

On Setting up Energy-Efficient Paths with Transmitter Power Control in Wireless Sensor Networks

Yangfan Zhou and Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China
Email: {yfzhou, lyu}@cse.cuhk.edu.hk

Jiangchuan Liu
School of Computing Science
Simon Fraser University
Vancouver, Canada
Email: jcliu@cs.sfu.ca

Abstract—Energy-efficiency is an important design consideration of communication schemes for wireless sensor networks (WSNs). In this paper, we investigate the problem of energy-minimized sensor-to-sink communications with adaptive transmitter power settings. We devise a novel network- and application-aware model for this problem, and present a Broadcast-On-Update (BOU) solution. However, BOU suffers from the high overhead due to explosive broadcasting in path setup. We then show a waiting scheme, BOU-WA, that effectively mitigates the broadcast explosion. In BOU-WA, the waiting time before each broadcast is proportional to the probability that a node could find a more energy-efficient path to the sink. We provide an efficient approximation algorithm to calculate this probability. The performance of BOU-WA is evaluated under diverse network configurations, and the results demonstrate its superiority in conserving energy.

I. INTRODUCTION

A wireless sensor network (WSN) consists of a number of in-situ battery-powered sensor nodes, which communicate through multi-hop paths. A WSN can be employed to collect the data about physical phenomena of interest [1][2][3]. There is a large collection of potential applications, including environmental monitoring, object tracking, surveillance, *etc.* [1][4]. Although WSNs have diverse task-specific requirements, many of them rely on a sensor-to-sink communication scheme to transfer the information that is collected by the sensors to a sink node.

In general, the battery in a sensor node is limited and not rechargeable [5]. Since wireless communications consume most of the energy in typical WSN applications [6], an energy-efficient data communication scheme is greatly desired. To be energy-efficient, the data communication scheme to convey the desired information on the event of interest through the established sensor-to-sink paths should cost as low energy as possible.

One important approach to save communication energy consumption is to perform transmitter power control (which is also called topology control). Obviously, setting the wireless transmitter power of each sensor node in different levels will result in different network topologies, as the neighboring nodes

that a node can directly reach are determined by the node's transmitter power setting.

A topology control scheme enables each node to set its power level to a minimum value under the constraint that the packet sent by this node could just reach its intended neighboring node. The energy consumption of data communication can thus be reduced. Transmitter power control is an important technique to save the energy consumptions of sensor nodes and prolong the lifetime of a network.

The prerequisite of transmitter power setting scheme is that each sensor node can set its own wireless transmitter power level. This is true in typical sensor node implementations. For example, the Berkeley Mica Mote [7] provides such program interfaces.

The notion of transmitter power control (topology control) has been extensively studied in wireless mobile ad hoc networks (MANETs). [8] is a good bibliography in the field. Although much work is done (*e.g.*, [9][10][11][12][13][14]) on transmitter power control in MANETs and WSNs, researchers mostly focus on network connectivity analysis, network lifetime (a network is alive if it is 'somehow' connected) analysis. They usually propose transmitter power setting schemes for energy-efficient communications between an arbitrary node pair and for energy-efficient broadcasting and multicasting.

Our objective of transmitter power setting is to achieve energy-efficient sensor-to-sink data communications. We model the transmitter power setting problem based on this consideration and the network and application features of WSNs. With this analysis and modeling work, we investigate implementation issues and analyze the proposed schemes to solve the transmitter power setting problem.

The contributions of our work are twofold. First, transmitter power setting problem is studied in this paper for achieving energy-efficient sensor-to-sink data communications. We do not emphasize to construct an energy-efficient communication path between an arbitrary node pair. This is the main consideration of the transmitter power setting problem in MANETs, as the MANET traffic is mainly unicasting peer-to-peer traffic. Instead, we aim at finding an energy-efficient communication

path between an arbitrary node and a given node, *i.e.*, the sink. This can greatly simplify the complexity of the problem. We show that this problem is tractable .

Second, we tailor the solution of the problem to adapt to the features of WSNs. We investigate the implementation issues for setting up the energy-efficient paths for sensor-to-sink traffic. Although high node density and large network scale of WSNs are major challenges for algorithms that set up each node's transmitter power level, we provide a low-overhead algorithm to address the transmitter power setting problem.

The rest of the paper is organized as follows. In Section II, we model our transmitter power setting problem according to the network and application features of WSNs. Section III investigates the implementation issues and analyzes the algorithm that solve the transmitter power setting problem. In Section IV, we present our simulation results. Section V discusses the related work of this paper. Section VI provides conclusion remarks and our future directions.

II. TRANSMITTER POWER SETTING FOR ENERGY-EFFICIENT SENSOR-TO-SINK DATA COMMUNICATIONS

A. Network, communication, and energy consumption models

The wireless signal fading models investigated in the literature [15] give the condition that packets transmitted from node u can be successfully received by the destination node v if the transmitter power setting of node u satisfies the following inequation:

$$Pr(u) \geq c \cdot (D(u, v))^n \quad (1)$$

Here c is a constant whose value is related to the system parameters such as the wavelength of the wireless signal, the antenna gains, and the threshold that a signal can be successfully detected in the destination node. n is the signal fading factor whose value is typically in the interval (2, 5) in an application environment. $D(u, v)$ is defined as the physical (Euclidian) distance between node u and node v ,

$$D(u, v) = \|X(v) - X(u)\|, \quad (2)$$

where $X(\cdot)$ denotes the physical location of a node.

Our work is based on this model: If node u knows the locations of itself and its one-hop destination node v , the optimal transmitter power setting for node u to send a packet to node v is computed as:

$$Pr(u) = c \cdot (D(u, v))^n = c \cdot \|X(v) - X(u)\|^n \quad (3)$$

We assume that each sensor node can know its approximate physical location with which we can calculate the transmitter power setting. The approximate location information is achievable if each sensor node carries a GPS receiver or if some localization algorithms are employed (*e.g.*, [16]).

We model the network as a graph. Let $G(V, E)$ be the graph constructed by the sensor nodes in a d -dimensional space

where V is the set of vertices which are the sensor nodes¹ and E is the set of edges that are the wireless links connected by the pairs of the sensor nodes which can communicate with each other at the maximum power setting Pr_{max} . Denote s ($s \in V$) as the sink node which is the final destination of the sensor-to-sink traffic.

Let $P(V) = \{Pr(u): \text{for each } u \in V\}$ be the transmitter power setting scheme for the sensor-to-sink communications. $P(V)$ should assure that each node can send packets (possibly, in a multi-hop manner) to the sink s .

Denote edge set E' as the set of the wireless links under the transmitter power setting scheme $P(V)$. Obviously, each $Pr'(u)$ in $P(V)$ is not larger than Pr_{max} . Therefore, the graph $G'(V, E')$ is the subgraph of $G(V, E)$.

Note that $G'(V, E')$ is a directed graph. With the transmitter power setting scheme $P(V)$, node u can send packets to node v if $Pr(u)$ satisfies Equation (1) and thus $\vec{e}'(u, v)$ is formed. But $Pr(v)$ may not necessarily satisfy similar requirements and thus $\vec{e}'(v, u)$ may not be formed. This consideration is because the power level of a node's downstream neighbor is not necessarily larger than that of the node to let the downstream neighbor respond acknowledgement (ACK) packets to the node, as hop-by-hop packet ACK mechanism (*i.e.*, the packet ACK mechanism in MAC layer) is usually not employed for energy saving. Note that if a hop-by-hop packet ACK mechanism has to be employed, we can simply adjust a node's transmitter power to different levels: One for sending sensor-to-sink packets to its downstream neighbor; the other for sending ACK packets to the upstream neighbor. The technique is trivial and we do not discuss it in the rest of this paper.

Let $\vec{\ell}(u_1, s)$ (where u_1 is the source node and s is the destination node) be the path in $G'(V, E')$ along u_1, u_2, \dots, u_i, s ($u_1, u_2, \dots, u_i \in V$). Consider the sensor-to-sink traffic path $\vec{\ell}$. We assume all sensor-to-sink packets are of the same size. The energy consumption of a sensor-to-sink packet delivery along this path is modeled as:

$$\sum_{n=1}^i (\gamma Pr(u_n)) + \sum_{n=2}^i (\gamma Rr(u_n) + Ps(u_n)) + \gamma Rr(s) + Ps(s), \quad (4)$$

where γ is a constant related to the packet size, $Rr(\cdot)$ denotes the receiver power of a node, and $Ps(\cdot)$ denotes the energy consumption to process this packet. We assume that the energy consumed to receive and process a packet of each node is the same. Equation (4) can then be written as:

$$\gamma \sum_{n=1}^i (Pr(u_n) + \beta) \quad (5)$$

where β is a constant related to the power consumed to receive and process a packet.

We define Equation (5) the *path cost* of the path $\vec{\ell}$, denoted by $\omega(\vec{\ell})$. $\omega(\vec{\ell})$ reflects the energy consumption of

¹The terms 'vertex' and 'node' both refer to a sensor device. In the rest of this paper, they are used interchangeably.

the communication along the path $\vec{\ell}$. We define the *node cost* of a node u ($u \in V$) as the minimum value of the path costs of the known possible paths from node u to the sink s . We denote $\eta(u)$ as the node cost of node u . $\eta(u)$ reflects the known minimum energy required to transfer a packet from node u to the sink.

B. Transmitter power setting problem for energy-efficient sensor-to-sink data communications

Since WSNs are employed to sense and convey the phenomenal data of interest, sensor-to-sink traffic dominates the traffic of the networks. In typical WSN applications, usually traffic sources are a set of sensor nodes responsible for reporting the data of some nearby phenomena of interest and the traffic destination is a given sink.

If no data fusion/aggregation approaches are employed, the sensor-to-sink traffic of the network is simply many-to-one traffic. The transmitter power setting scheme for the network, in this case, should aim to minimize the energy consumption of the communication between an arbitrary node and a given sink.

If some data fusion/aggregation approaches are employed, without loss of generality, the transmitter power setting scheme for the network should still aim to minimize the energy consumption of the communication between an arbitrary node and the given sink. The reasons are as follows.

First, in most application scenarios, we could simply consider the data fusion/aggregation center as the single data source that is reporting data packet to the sink in the transmitter power setting problem. Usually, the data fusion/aggregation center should be a sensor node located near the set of the source sensor nodes sensing and reporting the data on the physical phenomenon of interest. We can simply leave the consideration of how to report the data to the data fusion/aggregation center. Also, as data fusion/aggregation center consumes more energy than the other source sensor nodes, in practical applications, to avoid quickly draining the data fusion/aggregation center node, data fusion/aggregation center node should be selected in a rotational basis. (Details on how to select a data fusion/aggregation center and how to report data to the center are beyond the scope of this paper.) This means that in a long term point of view, each sensor node could be voted as the center node. Therefore, minimizing the energy consumption of the communication between an arbitrary node to a given sink is desired in this case.

Second, if we cannot consider the data fusion/aggregation center as the single data source, then for a given set of the source nodes, the optimal transmitter power setting problem (i.e., how to minimize the total energy consumptions of the sensor-to-sink communications) is very hard to solve. It is a minimum Steiner tree problem [17] which is NP-Hard. An approximate algorithm is to minimize the energy consumption of the communication between each source node to the sink. The packets are fused/aggregated only at the nodes in which the paths from the source sensor nodes to the sink intersect [18]. In this case, the transmitter power setting scheme should

still minimize the energy consumption of the communication between an arbitrary node to a given sink.

Based on the above discussion, to save energy consumption of the wireless communications, we do not have to consider how to minimize the sum of the energy consumed for the communication along a path between any arbitrary node pair. What we should consider, instead, is how to minimize the energy consumption of the communication between an arbitrary node to the sink. This consideration can greatly simplify the transmitter power setting problem. We model the transmitter power setting problem for energy-efficient sensor-to-sink data communications as follows.

Problem 1: Given graph $G(V, E)$ and a sink s ($s \in V$), compute $P(V)$ such that in the resulting subgraph $G'(V, E')$, there exists at least one path $\vec{\ell}(u, s)$ from each node u ($u \in V$) to the sink s and $\eta(u)$ is minimized. ■

Denote $cost(e)$ and $cost(\vec{e})$ as the cost functions of edge $e(u, v)$ and $\vec{e}(u, v)$ ($e \in E$, $e' \in E'$, and $u, v \in V$), respectively. $cost(e)$ and $cost(\vec{e})$ are defined as follows.

$$cost(e) = cost(\vec{e}) = \gamma(c \cdot (D(u, v))^n + \beta) \quad (6)$$

With the cost function, the shortest path from each node u ($u \in V$) to a given node s in the graph $G(V, E)$ can be found. The solution of Problem 1 is simply setting each node's transmitter power to the value with which it can just send packets to the downstream node along the shortest path.

With the information of the physical location of each node, $D(u, v)$ in Equation (6) can be calculated and thus $cost(e)$ can be obtained. Theoretically, the shortest paths can easily be found, for example, with the Dijkstra algorithm [19].

III. SETTING UP THE TRANSMITTER POWER LEVELS FOR SENSOR-TO-SINK TRAFFIC

Although a theoretical algorithm to set up the transmitter power levels for energy-efficient sensor-to-sink data communications is simply based on the modeling work in Section II, there are many practical implementation issues in WSNs that should be carefully considered.

Usually, the scale of WSNs is very large containing hundreds to thousands of sensor nodes. In order to obtain high reliability, the networks are usually with high density, i.e., the number of each node's neighbors is large. Moreover, sensor nodes are usually deployed in a non-deterministic manner, which means the location of each node is not known *a priori*. It is therefore not feasible for each sensor node to achieve a global picture of the network (i.e., graph $G(V, E)$ and the location $X(u)$ of each node u in set V) because exchanging the location information of every node is very expensive.

As each of the nodes does not have a global picture of the whole network, the shortest paths should be constructed with only localized information. We have to implement a solution to the problem in a completely distributed way, i.e., each node u should determine who its downstream neighbor in the shortest path is with only localized information. Here a node's localized information means the information that can be obtained by a node from its one-hop neighbors (i.e., its adjacent nodes in graph $G(V, E)$).

How to find out the downstream neighbor in the shortest path in an energy-efficient way (*i.e.*, exchange as small number of packets as possible) is a challenging implementation issue. We analyze and solve this problem in this section.

A. BOU: the basic algorithm

A direct way to set up the transmitter power level is broadcasting. Broadcasting is performed by setting the power level to the maximal value in order to reach all possible one-hop neighbors. We call the broadcast packets which carry the information to set up in-network nodes' transmitter power level the *configuration packets*. A configuration packet describes the location, the identity, and the node cost η of the node that sends the configuration packet.

The sink first broadcasts a configuration packet. The node cost of the sink is obviously set to zero. Upon receiving a configuration packet, an in-network node may update the node cost of itself and broadcast another configuration packet with the updated node cost.

Each node that receives a configuration packet computes its own wireless transmitter power setting with which it can reach the node where it receives the configuration packet according to Equation (3). Then the cost of the edge (the wireless link) from this node to the neighbor is calculated with Equation (6). The sum of the edge cost and the node cost of this neighbor is the path cost of the path from the node via this neighbor to the sink. If the node has not received any configuration packet before, this path cost is saved as its node cost. Otherwise, this path cost is compared with the current node cost. If the current node cost is smaller, the packet is simply dropped. Otherwise, the node cost is replaced with this path cost and this updated node cost is encapsulated in a configuration packet together with the node's location. The node then broadcasts the configuration packet.

It is straightforward to show that this process will finally converge and each node can know the location of its downstream neighbor through which the path to the sink is the shortest path. This process builds up a spanning tree rooted at the sink that initially sends out a configuration packet with node cost equal to zero. The path from each node to the sink in the spanning tree is the shortest path in graph $G(V, E)$ given the cost function of each edge e ($e \in E$) described in Equation (6). We call this approach *broadcast on update* (BOU) and formulate it in Algorithm 1.

B. Packet implosion of BOU: the challenge

However, as mentioned before, a typical WSN is with high node density and with large number of nodes. In addition, the power level needed to communicate with a neighboring node is linearly related to the n th power of the physical distance to the neighboring node according to Equation (3). The BOU approach is surely not efficient. A major challenge encountered in this distributed implementation is that it might cause explosive broadcasting in the network. Let us take Figure 1 as an example. Note that to simplify our discussion, we adopt the ideal free space transmission model [15] (*i.e.*, $n = 2$ in Equation (6)), and we ignore the energy consumptions of

Algorithm 1 BOU: The basic algorithm

```

1: input  $X$ 
   /* $X$  is this node's physical location.*/
2:  $\eta \leftarrow +\infty$ 
   /* $\eta$  is the node cost of this node.*/
3:  $downstream\_neighbor \leftarrow NULL$ 
   /* $downstream\_neighbor$  is the neighboring node to which this node
   sends sensor-to-sink data packets.*/
4:  $Pr \leftarrow 0$ 
   /* $Pr$  is the power level setting with which the node sends sensor-to-sink
   data packets.*/
5: loop
6:   Wait until receiving a configuration packet
7:    $Y \leftarrow$  The location information obtained from the configuration packet
8:    $\eta_{neighbor} \leftarrow$  The node cost information obtained from the configura-
   tion packet
9:    $Pr' \leftarrow$  The power level calculated with Equation (3)
10:   $\omega \leftarrow \eta_{neighbor} + cost(e)$ 
   /* $cost(e)$  is the cost of the edge from this node to the neighbor
   calculated with Equation (6).*/
11:  if  $\omega < \eta$  then
12:     $\eta \leftarrow \omega$ 
13:     $Pr \leftarrow Pr'$ 
14:     $downstream\_neighbor \leftarrow$  the neighbor that has sent the config-
   uration packet
15:    Create a configuration packet with the node cost  $\eta$ , the location  $X$ 
   and the identity of this node
16:    Broadcast this configuration packet
17:  end if
18: end loop

```

receiving and processing a packet at the node inside area ϕ , where area ϕ is a circular area with diameter AC (AC is the segment from node A to node C).

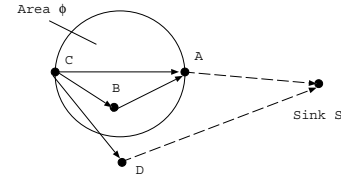


Fig. 1. A scenario of a network

Suppose node A in Figure 1 broadcasts a configuration packet. Node B and node C will approximately receive the packet at the same time. Normally, the processing time of the packet in node B and node C is almost the same. If node B and node C both find out that the paths along node A to the sink are the current shortest path to the sink, node B and node C will broadcast their configuration packets almost at the same time with their node costs.

In the next step, node C will receive the configuration packet from node B and notice that the path cost of the path through node B to the sink is lower than that of the path directly through A to the sink (because $\|AB\|^2 + \|BC\|^2 < \|AC\|^2$). Node C thus has to update its node cost computed with the node cost received from node B . Therefore, node C has to broadcast again a configuration packet.

If the network is with high node density and large scale, similar scenarios would cause severe problems. An in-network node might have to update its node cost for many times. Broadcasting has to be performed upon each update of the

node cost. This will cause explosive broadcasting of the network because the updated information is propagated in a tree-like manner to all upstream nodes of the nodes which have updated their node costs and broadcasted their configure packets.

Such kind of packet explosion should by all means be avoided to save the energy spent in configuring the transmitter power settings of the in-network nodes. Moreover, work in [20] enlightens us and strengthens our motivation to address this problem. In [20], the authors show that an *energy hole* around the sink is very likely to happen if the sink is fixed. We believe one easy way to avoid such energy hole is that we change the location of a sink frequently in the network area. In this case or in other application scenarios, when the location of the sink is not fixed during the network lifetime, the process to configure the optimal power setting of each node needs to be started each time the location of the sink changes. The efficiency of the implementation of the process therefore becomes a more critical issue. It is very desirable to address the aforementioned packet explosion problem.

One way to avoid the broadcast packet explosion problem is for each node to wait for a given period of time between the update of its node cost and the broadcasting of a configuration packet. An important research issue is therefore to determine this waiting time, which is investigated in the following subsection.

C. Determining the waiting time before broadcasting

The case that a node broadcasts a configure packet for more than one times happens only when the node needs to update its node cost after the first time it broadcasts a configuration packet. The reason of the update is that the node receives another configuration packet from a neighbor, which causes the change of its node cost. Therefore, if the waiting time before each in-network node broadcasts a configuration packet can be long enough, the node could have collected the configuration packet which indicates the actual shortest path from the node to the sink. It could avoid broadcasting for another time.

An intuitive solution is that each in-network node waits for the same period of time. But unfortunately, this idea does not work and the explosion situation still exists. For example, in the scenario described in Figure 1, node *B* and node *C* waits for the same time after they receive the packet from *A*. Node *C* will still receive the configuration packet from node *B* after it broadcasts the configuration packet with a node cost based on the node cost of node *A*. Then node *C* will update its node cost and it has to broadcast another configuration packet with the updated node cost.

We propose that the sophisticated waiting time should be proportional to the probability that a node will update the node cost in the future. Suppose that a node receives a configuration packet announcing a path $\vec{\ell}$ to the sink. It calculates the path cost $\omega(\vec{\ell})$ of this path. If $\omega(\vec{\ell})$ is smaller than the current node cost (if the node has not received any configuration packet before, the node cost is set to $+\infty$), the current node cost is updated to $\omega(\vec{\ell})$. The node should derive the

probability that there exists another path $\vec{\ell}'$ to the sink whose path cost $\omega(\vec{\ell}')$ is lower than $\omega(\vec{\ell})$, and waits for a period of time that is proportional to this probability. Now the problem left is how to calculate this probability.

Although a node cannot have a global picture of the network, if the node deployment scheme of the network is known, the probability distribution of a node's location can be modeled. For example, we can model this distribution as a uniform distribution if the sensor nodes are deployed randomly in a uniform way. Furthermore, the location $X(u)$ of each node u can be regarded as independent and identically distributed random variable if the deployment scheme of each in-network node is identical and independent of the others. In our following discussions, we regard $X(u)$ as independent and identically distributed random variable with probability density function $P_x(X)$.

The problem of computing the probability that a node will update the node cost in the future is formulated as follows.

Problem 2: Given

- A graph $G(V, E)$, a sink s ($s \in V$), and the cost function of an edge of the graph described in Equation (6);
- The probability density function $P_x(X)$ of the location of each node u ($u \in V$) where X is the possible physical location;
- The deterministic location x of a node m ($m \in V, m \neq s$) and the deterministic location y of the sink s ;
- The cost $\omega(\vec{\ell})$ of a path $\vec{\ell}$ from node m to the sink s ;

Compute the probability ρ that there exists a path $\vec{\ell}'$ from the node m to the sink s other than $\vec{\ell}$ such that the cost of $\vec{\ell}'$, $\omega(\vec{\ell}')$, satisfies $\omega(\vec{\ell}') < \omega(\vec{\ell})$. ■

With the solution of Problem 2, an improvement of BOU is to update line 16 of Algorithm 1 to Algorithm 2. We call this improved approach BOU-W (*wait before broadcast, on update*).

Algorithm 2 The broadcast scheme in BOU-W

- 1: If there is another configuration packet which is scheduled to be broadcasted, cancel it.
 - 2: Calculate the probability ρ that there exists another path to the sink of which the path cost is smaller than ω
 - 3: $T = \alpha \cdot \rho$
 - 4: Schedule that the configuration packet will be broadcasted in T seconds.
-

Here α is a constant whose value can be determined empirically.

Now we discuss the solution to Problem 2. Obviously, this problem is equivalent to computing the probability that the known path $\vec{\ell}$ is not the shortest path from the node m to the sink s . Therefore, in order to solve the problem, we should know the probability distribution of the node cost of a *given* node. However, it is very difficult to derive this probability distribution mathematically. But, we can perform Monte Carlo method to find this probability distribution. We discuss this approach in the following example.

Consider a network that contains 30 sensor nodes and a sink. The sensor nodes are deployed uniformly in a $30m \times 30m$ square and the sink is at the center of the square.

We fix the physical distance between one sensor node (denoted by node m) and the sink. This distance is denoted by d . Then we randomly generate the locations of the other 29 sensor nodes in a uniform way for h times and thus we get h graphs. For each graph, we perform the Dijkstra algorithm to find the shortest path from node m to the sink given the edge cost described in Equation (6) and record the node cost $\eta(m)$. Thus we get h results of $\eta(m)$. Let each number of series $N_i (i = 1, 2, \dots)$ be the number of results which is in interval $(0, i \cdot \tau]$, where τ is a constant. Obviously, if h is large enough and τ is small enough, N_i/h reflects the probability distribution of $\eta(m)$.

We gradually change the distance d with a step size equal to δ and perform the above process. In this way, we can get the probability distributions of $\eta(m)$ with different distances between node m and the sink. Figure 2 shows part of the results of the statistical data, in which δ is 3, h is 10^5 , and τ is 5. The probabilities is calculated with N_i/h .

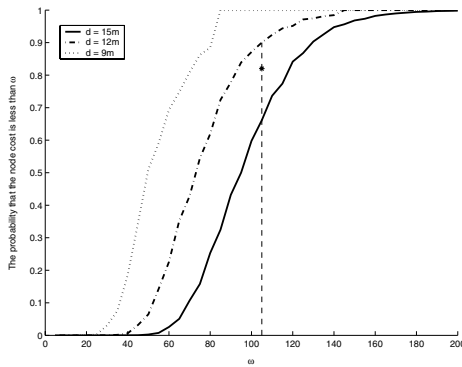


Fig. 2. The estimated probability distribution of node cost

With the statistical data achieved in the above approach, Problem 2 can be solved approximately. For example, when $\omega(\bar{\ell}) = 105$ and the distance $\|x - y\|$ is 13, the probability ρ is 0.82, which is approximately estimated with simple linear interpolation technique according to the data shown in Figure 2.

Note that the above approach to achieve the solution of Problem 2 requires only localized information. The cost $\omega(\bar{\ell})$ can be calculated as the sum of the node cost of the neighbor from which the node receives a configuration packet and the cost of edge between the node and the neighbor according to Equation (6). The location of the sink can be found in the configuration packet and the statistical data can be achieved with emulations before the sensor nodes are deployed and saved in the memory of each sensor node. The complexity to calculate the waiting time is negligible if we employ the statistical estimation approach discussed above to solve Problem 2.

D. BOU-WA: an approximate approach

In subsection III-C, we discuss how to determine the waiting time before a node broadcasts a configuration packet on the update of its node cost. However, the mathematical solution

of Problem 2 is not easy to be derived. Although Monte Carlo method helps to find approximate solutions, when the node deployment scheme cannot be well modeled, the statistical data cannot be achieved with emulations. Moreover, to obtain high accuracy, the above numerical solution of the Problem 2 requires that τ and δ are small. This means that huge volume of statistical data should be saved in a sensor node, which might not be practical due to the hardware constraint of the sensor node implementation [7]. In this subsection, we provide an approximate solution to determine the waiting time.

Let's still take Figure 1 as an example. For simplicity, we consider the space is 2-dimensional and we adopt the ideal free space transmission model [15] (*i.e.*, $n=2$ in Equation (6)). We ignore the energy consumed to receive and process a packet. Note that in actual application case, we can employ a more sophisticated model and without loss of generality, the approach proposed in our following discussions is still applicable.

We denote $\{C, A, \dots, S\}$ as the path from node C via node A and some other nodes to the sink S . If there exists a node B in the area ϕ , surely the path $\{C, B, A, \dots, S\}$ is shorter than the path $\{C, A, \dots, S\}$ because

$$(\|X(A) - X(C)\|)^2 > (\|X(A) - X(B)\|)^2 + (\|X(B) - X(C)\|)^2, \quad (7)$$

as node B is within the circular area with diameter AC

Note that there may exist another node D outside the area ϕ such that $\{C, D, \dots, S\}$ is shorter than the path $\{C, B, A, \dots, S\}$. But we can simply consider the probability that node B exists as the probability that a better path than the path $\{C, A, \dots, S\}$ exists as an approximation, although this probability is smaller than the actual probability that a better path than the path $\{C, A, \dots, S\}$ exists.

If we determine the waiting time according to this approximate probability (*i.e.*, the probability that node B exists), then we are waiting a shorter period of time than the BOU-W scheme. Therefore, the risk that the node will update its routing information and broadcast again is larger. In our simulation study, we will show that this risk is manageable and the approximation works well.

We name the scheme that adopts the waiting time based on this approximation the BOU-WA (*BOU-W with approximation*) scheme. BOU-WA is an improvement of BOU by updating line 16 of the BOU mechanism described in Algorithm 1 to Algorithm 3.

Algorithm 3 The broadcast scheme in BOU-WA

- 1: If there is another configuration packet which is scheduled to be broadcasted, cancel it.
 - 2: Calculate the probability ρ' that there exists a node u such that the path cost of the path from this node to the sink, immediately via the node u and then immediately via the one that sends this node the configuration packet, is smaller than ω
 - 3: $T = \alpha \cdot \rho'$
 - 4: Schedule that the configuration packet will be broadcasted in T seconds.
-

Suppose the deployed node number is k and the deployment area is φ . Assume the nodes are deployed uniformly in area

φ . The probability ρ' that there exists at least one node in area ϕ is as follow.

$$\rho' = 1 - \left(1 - \frac{\phi}{\varphi}\right)^{(k-2)}, \quad (8)$$

The complexity to calculate the waiting time in this scheme is negligible as a node only has to solve ρ' in Equation (8). Also, this scheme requires no message exchange among sensor nodes. As k and φ are known before node deployment, they can be programmed into the node beforehand. ϕ can be calculated according to $X(A)$, $X(C)$ and the cost function described in Equation (6). In this example that adopts the ideal free space transmission model, it is as follows.

$$\phi = \pi \cdot \left(\frac{\|X(A) - X(C)\|}{2}\right)^2 \quad (9)$$

Our simulation work in Section IV will compare this approximate probability, *i.e.*, ρ' , and the probability that a better path exists, *i.e.*, ρ , which is obtained with the Monte Carlo method. How BOU-WA performs with different network scales will also be investigated.

IV. SIMULATION RESULTS

We program the BOU scheme and the BOU-WA scheme with NS-2 [21] and study the performance of these schemes with simulations.

TABLE I
THE SETTING OF THE SIMULATION NETWORK

| | |
|------------------------------|-------------------------------------|
| Area of sensor field | 100m*100m |
| MAC Protocol | IEEE 802.11 without CTS/RTS and ACK |
| Transmitter Power | 0.660W |
| Receiver Power | 0.395W |
| Wireless Communication Model | Free Space |
| Packet length | 36 bytes |

In our simulation work, we first investigate the improvement of BOU-WA with different values of α (α is used to calculate the waiting time described in Algorithm 3 in the BOU-WA scheme) comparing with the BOU scheme in terms of energy consumption overhead to set up the transmitter power level of each in-network sensor node. The converging times of these schemes are also compared. Different network scales (*i.e.*, different node numbers of a network) are adopted in the simulations to show the scalability of these schemes.

To study how the BOU-WA scheme could approach the BOU-W scheme, we also investigate the differences between the probability ρ' calculated with Equation (8) and the actual probability ρ estimated with the Monte Carlo method.

Detailed settings of the simulation network are shown in Table I.

A. The comparisons of BOU and BOU-WA

In the network described in Table I, we randomly deploy $k + 1$ nodes in a uniform manner. We randomly select a node as the sink node and the other k nodes as the in-network

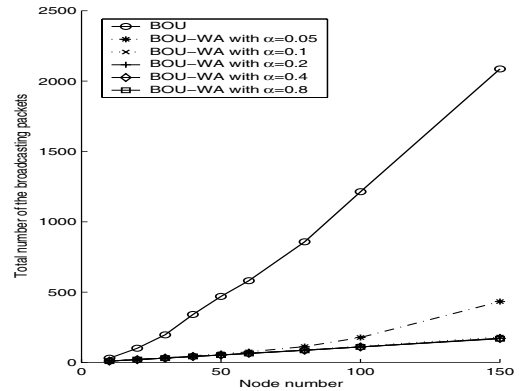


Fig. 3. Total number of broadcasts

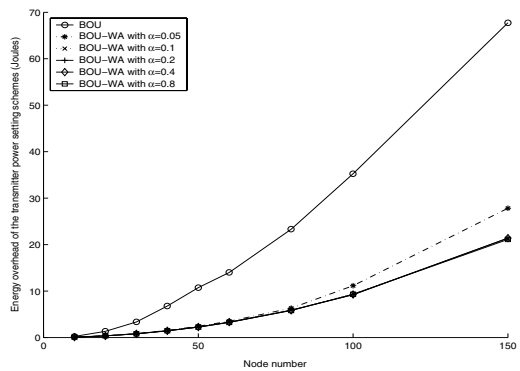


Fig. 4. The energy consumption overhead

sensor nodes. Let the sink node initiate the algorithms (*i.e.*, it broadcasts the first configuration packet with a node cost equal to zero). We set k as 10, 20, 30, 40, 50, 60, 80, 100 and 150. For each setting of k , we set the constant α as 0.05, 0.1, 0.2, 0.4, 0.8 (the waiting time before broadcasting is $\alpha \cdot \rho'$ seconds in the BOU-WA scheme). For each setting of k , we run the simulations of the BOU scheme for 1000 times and for each setting of k and α , we also run the simulations of the BOU-WA scheme for 1000 times. We average the results of all the simulation runs in each setting.

The total number of broadcasts in setting up the transmitter power levels and the energy consumption overhead of BOU and BOU-WA are shown in Figure 3 and Figure 4. It can be found that BOU-WA greatly improves the BOU scheme, especially when the number of nodes is large. Moreover, the greater α is, the better the BOU-WA scheme performs. But when α is large (*i.e.*, $\alpha = 0.2, 0.4$ or 0.8), different values of α do not have much different effects on the energy consumption overhead of BOU-WA.

The counter-effect of BOU-WA comparing to BOU is that BOU-WA might require larger converging time. The converging times of the BOU and BOU-WA are shown in Figure 5. These simulation results show that the greater α is, the longer the converging time of BOU-WA scheme is. Note that when α is less than 0.2, BOU-WA has a smaller converging time than BOU. This is because the number of broadcasts in BOU-

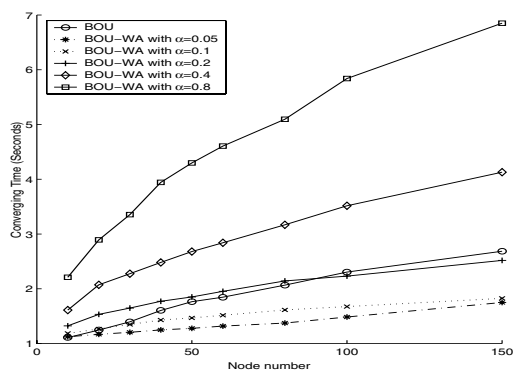


Fig. 5. The converging time

WA is much smaller than that that in BOU. As a result, the load of the wireless channel is lighter in case that BOU-WA is employed. Therefore, if a node wants to send a packet, it waits for less time until the channel is free in case that BOU-WA is employed.

These results show that BOU-WA, with a good parameter α , can perform much better than BOU.

B. The approximation of BOU-WA

To study the estimation error of ρ' in BOU-WA, we employ the Monte Carlo method to calculate ρ .

In the network described in Table I, we randomly deploy k nodes in a uniform manner and we place the sink at the corner of the square. We then randomly select two nodes in the network. One is a node that sends out a configuration packet, denoted by node s ; and the other is a node that receives the configuration packet, denoted by r . With the BOU-WA scheme, ρ' is calculated. We change the locations of the other nodes, except the sink, randomly for 10000 times and count the number of instances t in which node s is not the adjacent neighbor of node r along the shortest path from node r to the sink when the algorithm converges. $t/10000$ is regarded as the probability ρ that a better path exists than the path $\{r, s, \dots, sink\}$.

We perform the above process for 10000 times and the differences between ρ' and ρ are averaged. The average error can be regarded as the probability estimation error of the BOU-WA scheme.

We set k as 10, 20, 30, 40, 50, 80 and 100. We achieve the estimation error with the above method. The results are shown in Figure 6.

It can be seen that the estimation error is small. Moreover, the higher the node number is, the better accuracy the estimation achieves. It is worth to mention that finding a more accurate estimation is not necessary, because the average packet number that an in-network node should broadcast in the BOU-WA scheme is already close to the lower bound 1. Figure 7 shows the average broadcast number of a node when we set $\alpha = 0.8$ in the BOU-WA scheme. The lower bound is 1 because each node obviously has to broadcast at least once. This means that the room to further improve BOU-WA is already very small.

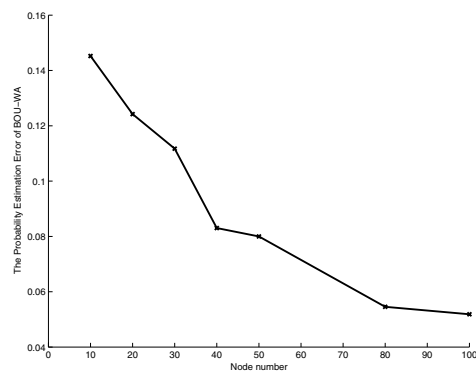


Fig. 6. The average estimation error of ρ'

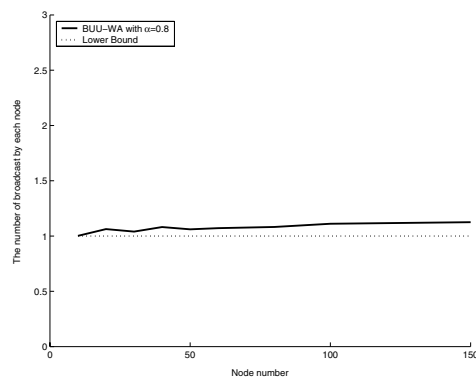


Fig. 7. The average number of broadcasts by each node

V. RELATED WORK

Research on many aspects of energy-efficient sensor-to-sink data communication has been conducted in the literature. In the work on data routing, directed diffusion [22] introduces the data-centric notion. It proposes that sensor-to-sink packets could be pre-processed at in-network nodes with data fusion and data aggregation techniques in order to reduce the total number of packets needed to convey the event information. Many other data communication schemes for WSNs have been proposed [23][24][25][26][27], all of which are striving to achieve energy efficiency, while maintaining other properties of the communication such as reliability or information fidelity. [28] is a good survey of these research issues.

In addition, traffic congestion will cause high packet loss rates, which result in low energy efficiency. To this end, mechanisms for detecting and even avoiding congestion have been studied [29][30]. In our recent work [31], we propose that the source reporting rates should be determined by the communication cost (which could be implemented to reflect the wireless communication conditions) and the importance of each source node (the metric that reflects how much information the source could provide on the event of interest) so that the communication scheme can effectively avoid congestion and provide reliable data transport.

The problem of broadcast storm has been extensively studied in multi-hop wireless networks. A series of solutions (*e.g.*,

[32]) have been proposed to mitigate the storm. However, their focus is on how to efficiently send a packet to every node in a network without much duplication. In our work, we reduce the number of broadcasts which is performed by each node to report its node cost to its one-hop neighbors for setting up energy-efficient paths.

VI. CONCLUSION REMARKS AND FUTURE WORK

This paper has examined the problem of transmitter power control for energy-efficient sensor-to-sink communications. We have modeled this problem based on the network and application features of WSNs. An intuitive implementation to solve this problem, namely BOU, has been presented. We have identified the broadcast explosion problem in BOU, and then improved BOU by allowing a waiting period before each broadcasting. We have shown that the waiting time should be proportional to the probability that a node would find a more energy-efficient path to the sink, and presented an efficient approximate algorithm to calculate the probability. Simulations have been designed to evaluate BOU and BOU-WA. The results have validated the effectiveness of BOU-WA; specifically, it can set up energy-efficient paths for sensor-to-sink traffic with low overhead in a reasonable converging time.

There are many possible future directions for this work. We are particular interested in integrating our algorithm with existing data fusion/aggregation schemes. We also interested in practical implementations, and we expect more issues can be identified in this process.

ACKNOWLEDGEMENT

The work described in this paper was substantially supported by two grants, RGC Project No. CUHK4205/04E and UGC Project No. AoE/E-01/99, of the Hong Kong Special Administrative Region, China. J. Liu's work is supported in part by a Canadian NSERC Discovery Grant 288325, an NSERC Research Tools and Instruments (RTI) Grant 613276, and an SFU President's Research Grant.

REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, April 2002.
- [2] D. Estrin, D. Culler, and K. Pister, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, January–March 2002.
- [3] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Communications of ACM*, vol. 43, no. 5, May 2000.
- [4] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. of the ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.
- [5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. of the 5th MobiCom*, Seattle, Washington, August 1999.
- [6] L. Doherty, B. Warneke, B. Boser, and K. Pister, "Energy and performance considerations for smart dust," *International Journal of Parallel and Distributed Systems and Networks*, vol. 4, no. 3, pp. 121–133, 2001.
- [7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proc. of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2002.
- [8] P. Santi, "A bibliography on topology control in ad hoc networks," Sep 2003, <http://www.imc.pi.cnr.it/~santi/papers/biblioTC.ps>.
- [9] H. Balakrishnan, R. Morris, K. Jamieson, and B. Chen, "Span: An energy efficient coordination algorithm for topology maintenance for in ad hoc wireless networks," in *Proc. of the 7th MobiCom*, Rome, Italy, July 2001.
- [10] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. of the 19th IEEE Infocom*, Tel Aviv, Israel, March 2002.
- [11] V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," *IEEE Journal of Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.
- [12] P. Santi, "The critical transmitting range for connectivity in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, May–June 2005.
- [13] P. Santi, D. Blough, and F. Vainstein, "A probabilistic analysis for the range assignment problem in ad hoc networks," in *Proc. of MobiHoc 2001*, Long Beach, October 2001, pp. 212–220.
- [14] R. Wattenhofer, P. Bahl, L. Li, and Y. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proc. of the 20th IEEE Infocom*, April 2001.
- [15] T. Rappaport, *Wireless Communications: Principles and Practices (2nd Edition)*. Upper Saddle River: Prentice Hall, 2002.
- [16] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE Personal Communication*, October 2000.
- [17] M. Goemans and D. Williamson, "A general approximation technique for constrained forest problems," *SIAM J. Comp.*, vol. 24, pp. 296–317, 1995.
- [18] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," in *Proc. of the 21st IEEE Infocom*, New York, 2002.
- [19] A. Gibbons, *Algorithmic Graph Theory*. Cambridge University Press, 1985.
- [20] J. Li and P. Mohapatra, "An analytical model for the energy hole problem in many-to-one sensor networks," in *Proc. of the IEEE Vehicular Technology Conference (To Appear)*, Dallas, Texas, Fall 2005.
- [21] K. Fall and K. Varadhan, *The ns manual*, Dec. 2003, <http://www.isi.edu/nsnam/ns>.
- [22] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. of the 6th MobiCom*, Boston, Massachusetts, August 2000.
- [23] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proc. of the First Workshop on Sensor Networks and Applications*, Atlanta, GA, October 2002.
- [24] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, August 2002.
- [25] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM Mobile Computing and Communications Review (MC2R)*, vol. 1, no. 2, pp. 295–298, January 2002.
- [26] C. Schurgers and M. Srivastava, "Energy efficient routing in wireless sensor networks," in *Proc. of the 2001 MILCOM*, vol. 1, 2001.
- [27] R. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proc. of the IEEE Wireless Communications and Networking Conference*, Orlando, FL, March 2002.
- [28] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Elsevier Ad Hoc Network Journal*, vol. 2, no. 3, pp. 325–349, 2005.
- [29] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz, "Esrt: Event to sink reliable transport in wireless sensor networks," in *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2003.
- [30] C. Wan, S. Eisenman, and A. Campbell, "Coda: Congestion detection and avoidance in sensor networks," in *Proc. of the the 1st ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, November 2003.
- [31] Y. Zhou, M. Lyu, J. Liu, and H. Wang, "Port: A price-oriented reliable transport protocol for wireless sensor networks," in *Proc. of the 16th International Symposium on Software Reliability Engineering (To Appear)*, Chicago, IL, November 2005.
- [32] Y. Tseng, S. Ni, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2/3, pp. 153–167, March 2002.