Quick Convergence Mobility Control Schemes in Wireless Sensor Networks

Xiao Chen
Dept. of Comp. Sci.
Texas State Univ.
San Marcos, TX 78666
xc10@txstate.edu

Zhen Jiang
Dept. of Comp. Sci.
West Chester Univ.
West Chester, PA 19383
zjiang@wcupa.edu

Jie Wu
Dept. of Comp. Sci. and Eng.
Florida Atlantic Univ.
Boca Raton, FL 33431
jie@cse.fau.edu

Abstract

In the near future, wireless sensor networks (WSN) performing sensing and communication tasks will be widely deployed as technology rapidly advances. Communication is one of the essential functionalities of these mobile networks while power and computation resources in each sensor are limited. Recently, attention has been drawn to using mobility control to minimize energy consumption in wireless sensor networks. In this paper, we are going to provide quick convergence mobility control schemes to achieve optimal configuration in a single data flow. The key idea of our schemes is to use the optimal location information of each relay node as a guide for node movement while maintaining the connectivity of relay nodes along the data flow. Experimental results show that our schemes can speed up the convergence process to nearly the optimal and reduce the cost of it almost to the minimum, compared with the best results known to the date.

1 Introduction

With advancements in technology, wireless sensor networks (WSNs) performing sensing and communication tasks will be widely deployed in the near future because they greatly extend our ability to monitor and control the physical environment and improve the accuracy of our information gathering [5], [7], [10], [15], [19]. Sensor nodes can be deployed in inhospitable physical environments such as battlefields, remote geographic regions, and toxic urban locations. One specific example can be a group of mobile robotic insects sensing dangerous areas or enemy targets and sending back as much information as possible [2].

Before the sensor nodes are deployed, they are initially powered. In many situations, once they are dispersed into an environment, they cannot get recharged very often. Thus, the power in sensors is the scarcest resource. Therefore, communication mechanisms must be power efficient and simple. As capability of mobility becomes more readily available to WSNs [17], there are several recent studies on using mobility as a control mechanism to minimize energy consumption [6], [9], [11], [20], [21]. [6] presents a simple mobility control scheme using only one-hop neighborhood information in which the connections between neighbors will never be broken. [9] improves such a method by solving the oscillation problem. However, it may take many rounds of movement for nodes to reach their optimal locations as shown by an example in the later section covering preliminary works.

In this paper, we provide new schemes that move the relay nodes much more quickly to their optimal positions without oscillation while the connectivity is maintained, so that the energy consumption of WSNs can be reduced. We put forward two schemes: MCC and MCF. MCC speeds up the convergence process by avoiding the overreaction of a node to the movement of its neighbors, while MCF reduces the convergence time by moving the nodes as close to their optimal positions as possible. The key idea of our schemes is to use the optimal location information of each relay node as a guide for mobility control. In later experimental results, we will show how much faster this information can allow the process to converge. Finding the optimal location of each intermediate node is an easy task. It is not much of an overhead to the communication process because it can be carried out along with the routing algorithm by just adding a counter and the location information of the source and the destination. In summary, there are four major features of our schemes. The first is that, compared to existing mobility control schemes, our methods converge much faster and reach nearly optimal results. The second is that our schemes will not break the connections between a node and its two neighbors. The third is that our schemes reduce the node movement almost to the minimum. The fourth feature is that our schemes use only one-hop neighborhood information; no global information is needed.

In this paper, we assume that a path from the source s to the destination d has already been discovered using some routing protocol. Only one-hop neighborhood information is used here. Such information can be discovered by GPS or some GPS-free positioning algorithms such as the one in [3]. To make this paper easy to read, we use global location in discussion.

The rest of the paper is organized as follows: Section 2 introduces some related work, Section 3 contains the preliminary information, Section 4 presents our two quick convergence schemes, Section 5 shows the experimental results, and Section 6 concludes the paper.

2 Related Work

There is a large amount of literature on power-efficient topology control and routing; for examples see [13] and [18]. Recently there have been several studies on using mobility as a control primitive to minimize power/energy consumption in such mobile networks [6], [9], [11], [20], [21].

In [6], the authors prove that in a single active flow between a source and a destination pair, if the energy cost function is a non-decreasing convex function, the optimal positions of the relay nodes must lie entirely on the line between the source and destination, and that the relay nodes must be evenly spaced along the line. Based on this, despite the randomness of the initial deployment, if the nodes can move towards their optimal locations under mobility control, the energy consumption can be minimized. In their paper, the authors put forward synchronous and asynchronous mobility control algorithms to reach optimality based on the averaging algorithm [8], [14]. That is, each node's optimal location is the average of its left and right neighbors' locations. In this paper, the *left* and *right* neighbors of a node refer to the left and right neighbors on the line between the source and the destination. Thus, a node moves along with the movement of its two neighbors. The algorithms are simple; they only require one-hop local information of the node's left and right neighbors, and are distributed, which is suitable for a mobile environment. Also, the authors prove that the movement of a node in this way will not break the connections between the node and its neighbors. However, there is an implementation problem that the authors do not mention in the paper. That is, due to round-off errors [1], the nodes may oscillate around their optimal locations and never stop. Thus, their energy will be depleted very quickly before they can perform useful tasks.

This problem is pointed out by [9] and fixed by a threshold which is set to 0.0001. In this way, if the difference between the node's current location and the next location is no greater than the threshold, the node can stop moving. However, as we can see from the experimental results, the convergence process is still very slow. It can take many rounds of movement for the nodes to reach their optimal locations. The more movement there is, the faster the power in each node will be depleted. Thus, slow convergence can be a negative factor to justify the effectiveness of the mobility control primitive in power-efficient communication.

Actually, if the location of the source $L(s)$, the location of the destination $L(d)$, and the label i of an intermediate relay node u_i are known, the optimal location $L'(u_i)$ of u_i can be calculated as $L'(u_i) = L(s) + i \times \frac{L(d)-L(s)}{n}$, assuming that $i \in [0, n]$, u_0 is source s and u_n is destination d [6]. Thus, the nodes can move to their optimal locations directly, without oscillation, in one round. However, this method can break the connections between a node u_i and its neighbors. When neighbors are disconnected, the data sent is lost and has to be resent from the source after timeout. The neighbors have to reconnect through sending each other Hello messages. As indicated in [4, 12, 16], all of these will decrease the communication efficiency.

3 Preliminary

We assume that all the sensor nodes have the same transmission range. If two sensor nodes are within each other's transmission range, they can communicate directly and they are called *neighbors*. Otherwise they have to rely on intermediate nodes to relay messages for them. We define a WSN as a graph $G = (V, E)$, where V is the set of all sensor nodes and E is the set of all edges between pairs of sensor nodes. If two sensor nodes can communicate directly, there is an edge between them in G. The location of each node u is (x_u, y_u), simply denoted as $L(u)$. $|L(u) - L(v)|$ is the physical distance between two nodes u and v. $L'(u)$ denotes the target location of u in its movement.

We assume that a path from the source s to the destination d has already been discovered using a routing protocol, e.g., a greedy routing protocol or one of the ad hoc routing protocols. We also assume that both s and d are not moving during the process. Otherwise, the path is always broken and a new routing path needs to be established. We label the nodes from the source to the destination as $0, 1, \cdots, n$. Node u_0 is the source, node u_n is the destination, and nodes $u_1, \cdots, u_{n-1}$ are intermediate relay nodes. For each node $u_i, 1 \leq i \leq n-1$, node u_{i-1} is its left neighbor and node u_{i+1} is its right neighbor. Neighbors can share information by exchanging short messages. To simplify the discussion, we describe the schemes in a synchronous, round-based system. All the schemes presented in the paper can be extended to an asynchronous system. However, to make our schemes clear, we do not pursue the relaxation.

In a single flow of communication between a source s and a destination d, the optimal configuration of relay nodes is established in [6] as follows. First, assume that the energy cost function is a non-decreasing convex function. Then, the optimal positions of the relay nodes must lie entirely

Algorithm MCD[6]: Mobility control at each relay node u_i.

1. Exchange $L(u_i)$ with u_{i-1} and u_{i+1}.
2. Receive $L(u_{i-1})$ and $L(u_{i+1})$. Set $L'(u_i) = \frac{L(u_{i-1})+L(u_{i+1})}{2}$.
3. Set damping factor g a random value $\in (0, 1]$, move toward $L(u_i) + g \times (L'(u_i) - L(u_i))$.

round	s_x	$node1_x$	$node2_x$	$node3_x$	$node4_x$	d_x
0	92.11134	86.99914	80.11193	74.99975	69.11155	63.99937
1	92.11134	86.11163	80.99944	74.61174	69.49956	63.99937
2	92.11134	86.55539	80.36169	75.24949	69.30556	63.99937
3	92.11134	86.23651	80.90244	74.83362	69.62444	63.99937
..	..	..	..	..	..	..
53	92.11134	86.48894	80.86656	75.24416	69.62177	63.99937
54	92.11134	86.48895	80.86655	74.24417	69.62176	63.99937
55	92.11134	86.48894	80.86656	74.24416	69.62177	63.99937
56	92.11134	86.48895	80.86655	74.24417	69.62176	63.99937
57	92.11134	86.48894	80.86656	74.24416	69.62177	63.99937
58	92.11134	86.48895	80.86655	74.24417	69.62176	63.99937
..	..	..	..	..	..	..

Table 1. The oscillation of nodes to reach optimal locations

on the line between s and d. Furthermore, the relay nodes must be evenly spaced along the line. A uniform distributed algorithm that allows the relay nodes to move to their optimal positions is introduced in [6] (see Algorithm MCD). The key ingredient of this algorithm is the simple average calculation. Note that although a relay node computes the average of its two neighbors, the node only moves toward this point, instead of reaching it in one round. In other words, the movement is damped using a damping factor g in the algorithm. The damping process is used to avoid overreaction of each node. The authors of [6] also prove that the connection between communicating neighbors using MCD is never broken.

However, MCD has an implementation problem; it may make each node oscillate around its optimal location endlessly due to round-off errors (see Table 1 for an example of oscillation). There are six nodes in the table, including the source and the destination. The transmission range of each node is 10. Suppose they are placed in a line and the y coordinate of each node is the same. Therefore, in the table, only the x coordinate of each node is shown. Round 0 displays the initial location of each node. Starting from round 1, each relay node uses MCD (suppose g is set to 1) to calculate its optimal location. From the table, we can see that the process to reach the optimal location of each node converges gradually. However, starting from round 53, the nodes oscillate around their optimal locations and never stop. This kind of oscillation is caused by the round-off errors in computers.

Algorithm MCM[9]: Mobility control with minimum total moving distance.

1. The source node s sends $L(s)$ and its label 0 to u_1. When each relay node u_i receives $L(s)$ and the label $i-1$, it will pass $L(s)$ and its own label i to the succeeding node along the path. Such a propagation will end at d.
2. Once $L(s)$ is received at the destination node d, d sends a message carrying $L(d)$ back to s along the path.
3. At each relay node u_i, once both $L(s)$ and $L(d)$ are received, set $L'(u_i) = L(s) + i \times \frac{L(d)-L(s)}{n}$ and move u_i to $L'(u_i)$.

It wastes computation resources and will deplete the energy in nodes very quickly. Besides, the convergence process is very slow, too; it takes about 53 rounds for the nodes to get close to their optimal locations.

To prevent oscillation, [9] sets a threshold $MDPR = 0.0001$ so that if the difference between node u_i's target position $L'(u_i)$ and the current position $L(u_i)$ is less than MDPR, the node does not have to move any more. Thus, each node can stop earlier. This algorithm is called $MC1$.

Obviously, connections between neighbors along the path are not lost using MC1 [9]. However, this algorithm still suffers from the slow convergence problem.

After the locations $L(s)$ and $L(d)$ in the absolute coordinate system are collected at a relay node u_i, its optimal position can be determined by $L'(u_i) = L(s) + i \times \frac{L(d)-L(s)}{n}$ and this node can move to the optimal position directly without oscillation [6]. An algorithm for this is written in detail in [9] and is presented in Algorithm MCM here.

Algorithm MCM has a very nice property: the total moving distance of nodes in MCM is minimum.

It is noted that MCM does not create much overhead in the system because it can be combined with the routing process. When source s sends a message to d, it can also send its $L(s)$ and its label 0 along with the message. Each intermediate node will do the same thing until the message reaches d. Then d sends an acknowledgement plus its $L(d)$ back to s. When each relay node u_i has $L(s)$, $L(d)$ and its label, it can calculate its target position.

However, when nodes are moving towards their optimal positions using MCM, it is likely that the connections between some nodes will be lost. For example, suppose nine nodes $u_0, u_1, \cdots, u_8$ are aligned in a line. Node u_0 is the source and u_8 is the destination. The transmission range of each node is 10. Node u_i ($i \in [0..4]$) is at location i. Node u_5 is at location 14, node u_6 is at 23, node u_7 is at 32 and node u_8 is at location 41. According to Algorithm MCM, the optimal location of relay node u_i should be $L(u_i) = 0 + (41 - 0)/8 * i$. Therefore, the optimal location of node u_5 is 25.625. If node u_4 is still at its old location 4 when node u_5 moves toward its optimal location, the connection between them is lost.

Now the challenge lays in determining how to speed up the convergence process without loosing the connection between communicating neighbors. This is our topic in the next section.

4 Two Quick Convergence Schemes

In this section, we describe two mobility control schemes, MCC and MCF, to let nodes move to their optimal locations much more quickly. MCC speeds up the convergence process by avoiding the overreaction of a node to the movement of its neighbors, while MCF reduces the convergence time by moving the nodes as much closer to their optimal positions as possible. Both schemes use the information of the optimal locations of the relay nodes. This optimal location information is obtained by running Algorithm MCM before the convergence process starts. The oscillation problem is still solved using the threshold MDPR.

4.1 Scheme MCC

The first scheme *MCC* combines MCM with the Modified MCD (details shown in Algorithm MCC). This scheme still uses the average calculation in MCD. The difference is that in MCD, a node will move as the locations of its left and right neighbors change. However, in MCC, a relay node knows its optimal position by MCM, and if the distance between its new position (which is calculated as the average of its two neighbors' positions) and its optimal position is larger than the distance between its current position and its optimal position, it does not move. In this way, a node can avoid unnecessary movements. Therefore the time it takes to complete the convergence process can be reduced. Again, the MCM part of the algorithm can be combined with a routing algorithm to reduce overhead. The MCM is only called once if the locations of the source, the destination, and the number of relay nodes do not change for a period of time.

It can be proved that the connectivity is kept in MCC while the nodes are moving.

Theorem 1 Any connection between communicating neighbors is not lost in MCC.

Proof. Without loss of generality, in our proof, we need to cover cases where a node will move with the location changes of its two neighbors and cases where a node will not move if the new location is farther away from its optimal location than its current location. To cover both cases,

---

Algorithm MCC: MCM combined with Modified MCD.

Apply MCM to obtain the optimal location $OL(u_i)$ for each relay node u_i.

For each relay node u_i,

1. Exchange $L(u_i)$ with u_{i-1} and u_{i+1}.
2. Receive $L(u_{i-1})$ and $L(u_{i+1})$. Set $L'(u_i) = \frac{L(u_{i-1})+L(u_{i+1})}{2}$.
3. If $|L'(u_i) - OL(u_i)| > |L(u_i) - OL(u_i)|$ no movement; else
4. If $|L'(u_i) - L(u_i)| \geq MDPR$, move to $L'(u_i)$.

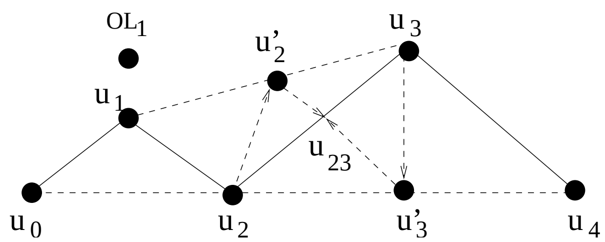

Figure 1. Illustration of Theorem 1

we come up with a network as shown in Figure 1. There are five relay nodes $0, 1, \cdots, 4$ at locations $u_0, u_1, \cdots, u_4$. Assume all the nodes have the same transmission range r. The solid lines between two nodes indicate that the two nodes can communicate with each other directly (they are neighbors), while the dashed lines are used to indicate the distance between the two nodes.

The first part of MCC calculates the optimal locations of the relay nodes. In the figure, only the optimal location of node 1 is shown. Node 1 is the one that will not move because its new location which is at the midpoint of u_0 and u_2 will put it farther away from its optimal location. All others will move to their new locations, that is, node 2 will move to u_2' which is the midpoint of u_1 and u_3, and node 3 to u_3' which is at the middle of u_2 and u_4.

Now we want to prove that the connections of nodes in their new locations are not lost. That is, $|u_1 - u_2'|$ and $|u_2' - u_3'|$ are less or equal to the transmission range r.

First we prove that $|u_1 - u_2'| \leq r$ is true. Obviously in triangle $u_1u_2u_3$, either $|u_1 - u_2| \geq |u_1 - u_2'|$ is true or $|u_2 - u_3| \geq |u_2' - u_3|$ is true. Since $|u_1 - u_2'| = |u_2' - u_3|$, $|u_1 - u_2| \leq r$, and $|u_2 - u_3| \leq r$ are true, $|u_1 - u_2'| \leq r$ is also true.

Next we prove that $|u_2' - u_3'| \leq r$ is true. Denote the midpoint of u_2u_3 as u_{23}. $|u_2' - u_3'| \leq |u_2' - u_{23}| + |u_3' - u_{23}| = \frac{1}{2}(|u_1 - u_2| + |u_3 - u_4|) \leq \frac{1}{2}(r + r) = r$. So $|u_2' - u_3'| \leq r$ is true.

Therefore, the connectivity is not lost in Algorithm MCC.

□

4.2 Scheme MCF

The second scheme *MCF* also uses MCM to obtain the optimal location for each relay node. The idea is that the relay nodes should move toward their optimal locations as much as possible without breaking the connections with their left and right neighbors. In this way, for each node, there is no extra movement. The details of this algorithm are shown in Algorithm MCF.

In MCF, the target location $L'(u_i)$ can easily be calculated using a small program that solves mathematical equations. Theorem 2 shows that the connection between communicating neighbors is not lost in MCF.

Theorem 2 Connectivity is kept between communicating neighbors in MCF.

Proof. Without loss of generality, there are four relay nodes $0, 1, 2, 3$ at locations u_0, u_1, u_2, u_3 (see Figure 2). The transmission range of each node is r. The area covered by a node's transmission range is represented by a dashed circle in the figure. A solid line between two nodes indicates that they can communicate with each other directly while

---

Algorithm MCF: Move to optimal location as much as possible.

Apply MCM to obtain the optimal location $OL(u_i)$ for each relay node u_i.

For each relay node u_i,

1. Calculate target location $L'(u_i)$ which is the closest point to $OL(u_i)$ without breaking the connection with u_i's left and right neighbors u_{i-1} and u_{i+1}.

2. If $|L'(u_i) - L(u_i)| > MDPR$, move to $L'(u_i)$.

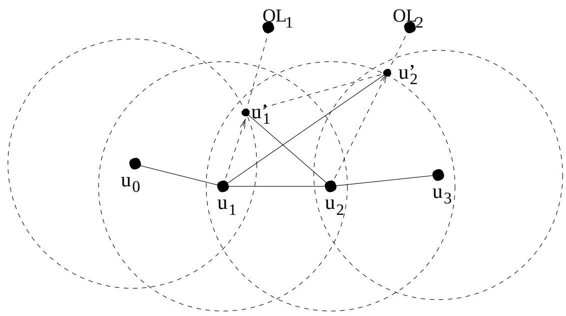

Figure 2. Illustration of Theorem 2

a dashed line indicates the distance between them. Node 1 has neighbors 0 and 2 and node 2 has neighbors 1 and 3. The optimal locations of nodes 1 and 2 are OL_1 and OL_2 respectively from MCM. Now node 1 and node 2 will move toward their optimal locations as much as possible without loosing the contact with their neighbors. The new locations of nodes 1 and 2 are u'_1 and u'_2 as shown in the figure.

Now we want to show that the communication between nodes 1 and 2 in their new locations is not lost, that is, $|u'_1 - u'_2| \le r$. From the figure, in shape $u_1 u_2 u'_2 u'_1$, either $|u'_1 - u'_2| \le |u'_1 - u_2|$ is true or $|u'_1 - u'_2| \le |u_1 - u'_2|$ is true. Since $|u_1 - u'_2| \le r$ and $|u'_1 - u_2| \le r$ are true, so $|u'_1 - u'_2| \le r$ is true. This means that the communication between nodes 1 and 2 in their new locations is still within the range r. $\square$

5 Experimental Results

In this section, we verify the improvement our new schemes offer on convergence speed and cost through experimental results. In a synchronous, round-based system, the speed of achieving stablization is measured by the number of rounds of node movement needed for convergence. The cost of mobility control schemes primarily comes from the energy consumed in node movement which is determined by the distance a node moves. In our experiments, the total distance of movement of all the nodes is used as a metric for the cost of mobility control schemes.

In the experiments, we compare the convergence speed and the cost of four algorithms: (1) MC1; (2) MCC; (3) MCF; and (4) MCM. For each algorithm, the number of rounds and the total distance of node movement are calculated. In our experiments, the number of nodes is set to 5, 10, 15 and 20, including the source and the destination. The transmission range is set to 20 and 40 [22]. The initial locations of the nodes are randomly generated. The damping factor in MC1 is set to 1 for the sake of convenience.

Figures 3 and 4 show the number of rounds of node movement for different algorithms when the transmission range is set to 20 and 40 respectively with the number of nodes varied. In the figures, MC1 has the most rounds of node movement, MCC has less, and MCF and MCM have the least. Due to the nature of MCM, we know that it only takes one round for the nodes to reach their optimal positions. From either figure, we can see that the line of MCF is almost overlapped with that of MCM. This shows that MCF can converge surprisingly fast. It almost reaches the optimal result of MCM.

Figures 5 and 6 show the total distance of node movement during the convergence process using different algorithms when the transmission range is 20 and 40 respectively with the number of nodes varied. The results in these two figures match those of the number of rounds of node movement. One very good result is that MCF is so close to

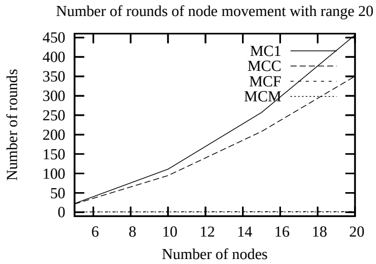

Figure 3. Number of rounds of node movement using range 20

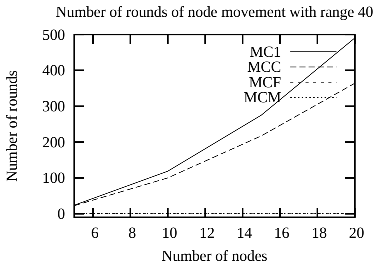

Figure 4. Number of rounds of node movement using range 40

MCM in terms of the total distance that their lines overlap in the figures. As we know, MCM achieves the minimum total movement. Therefore, the total movement using MCF is extremely close to the minimum.

In summary, these results show us how effective it is to embed the information of optimal locations of relay nodes into the schemes. Despite a very low overhead to gather this information at the beginning, the speed of the convergence process has been greatly increased and the cost has been greatly reduced - especially in the MCF scheme where it almost reaches the best results for both convergence speed and cost. In addition, unlike algorithm MCD, both MCC and MCF will not incur any node-overreaction. Therefore, there is no need to put the damping factor g into the algorithms, because it brings nothing but delay.

6 Conclusion

In this paper, two quick convergence mobility control schemes, MCC and MCF, have been put forward to improve communication in WSNs. MCC speeds-up the convergence process by avoiding node's overreaction to the movement of its neighbors, whereas MCF reduces the convergence time by moving the nodes as close to their optimal positions as possible. Both schemes have embedded the information of

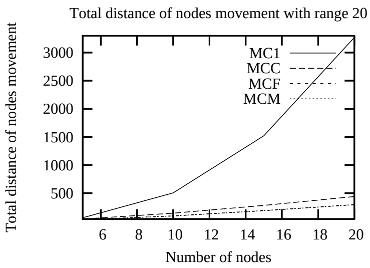

Figure 5. Total distance of node movement using range 20

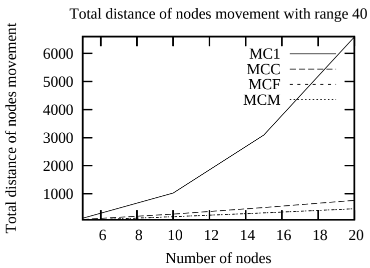

Figure 6. Total distance of node movement using range 40

the optimal locations of relay nodes into the mobility control. Compared to existing mobility control schemes, they can speed up the convergence process nearly to the optimal and reduce the cost to nearly the minimum in WSNs. This is especially true for MCF. These results provide strong evidence of support in justifying the effectiveness of using mobility control to reduce energy-consumption to improve communication efficiency in WSNs. In this paper, we have only discussed the communication of a single active flow between one source and one destination. Communication among multiple source-destination pairs was not addressed, but will be in our future work.

References

[1] http://mathworld.wolfram.com/RoundoffError.html.

[2] http://www.eecs.harvard.edu/ rjwood/media.html.

[3] S. Capkun, M. Hamdi, and J. Hubaux, "Gps-free positioning in mobile ad hoc networks," *Proc. of the 34th Annual Hawaii International Conference on System Sciences*, 2001, pp. 3481-3490.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverge control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, 2004, pp. 243-255.

[5] D. Estrin, D. Culler, K. Pister, and G. S. Sukhatme,"Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, 2002, pp. 59-69.

[6] D. Goldenberg, J. Lin, A. Morse, B. Rosen, and Y. Yang, "Towards mobility as a network control primitive," *Proc. of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc'04)*, May 2004, pp. 163-174.

[7] D. L. Hu, B. Chan, and J. W. M. Bush, "The hydrodynamics of water strider locomotion," *Nature*, vol. 424, no. 7, August 2003, pp. 663-666.

[8] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of autonomous mobile agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, 2003, pp. 988-1001.

[9] Z. Jiang, J. Wu, and R. Kline, "Mobility control for achieving optimal configuration in mobile networks," *Technical report*, Department of Computer Science, West Chester University, 2007.

[10] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Emerging Challenges: Mobile networking for 'Smart Dust'," *Journal of Communications and Networks*, vol. 2, no. 3, September 2000, pp. 186-196.

[11] L. Li and J. Halpern, "Minimum-energy mobile wireless networks revisited," *Proc. of IEEE Conference on Communications (ICC2001)*, vol. 1, June 2001, pp. 11-14.

[12] Y. Mao and M. Wu, "Coordinated sensor deployment for improving secure communications and sensing coverage," *Proc. of the* 3^{rd} *Workshop on Security of Ad Hoc and Sensor Networks*, 2005, pp. 117-128.

[13] C. Perkins, "Ad Hoc Networking," Addison-Wesley, 2000.

[14] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," *Proc. of The Ninth International Conference on Mobile Computing and Networking (Mobicom)*, September 2003, pp. 96-108.

[15] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, 1999, pp. 171-194.

[16] S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," *Proc. of IEEE International Conference on Robotics and Automation*, 2004, pp. 165-171.

[17] V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, 1999, pp. 1333-1344.

[18] E. M. Royer and C. K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, April 1999, pp. 46-55.

[19] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robomote: a tiny mobile robot platform for large-scale sensor networks," *Proc. of IEEE International Conference on Robotics and Automation*, 2002.

[20] W. Wang, V. Srinivasan, and K. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," *Proc. of the* 11^{th} *ACM international conference on Mobile Computing and Networking (MobiCom'05)*, 2005, pp. 270-283.

[21] Y. Wang, H. Wu, F. Li, and N. Tzeng, "Protocol design and optimization for delay/fault-tolerant mobile sensor," *Proc. of the* 27^{th} *International Conference on Distributed Computing Systems (ICDCS'07)*, 2007, pp. 7-7.

[22] J. Wu, M. Cardei, F. Dai, and S. Yang, "Extended Dominating Set and Its Applications in Ad Hoc Networks Using Cooperative Communication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, August 2006.