The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Hopping Sensor Relocation in Rugged Terrains

Yuanteng Pei, Fernando J. Cintrón, Matt W. Mutka, Jianguo Zhao, and Ning Xi

Abstract— **Hopping sensors are a type of low cost mobile sensors that are small in size, have limited capability and imprecise movement. However, their unique method of movement makes them suitable for rugged terrains. Sensors may fail when deployed in a rugged terrain or in an obstacle-abundant environment. Therefore, redundant sensors may be identified and relocated to the sensor holes.**

This paper addresses the problem of relocating such capability-constrained sensors in an obstructive environment. We propose an enhanced Quorum-Grid solution with Binary Splitting Message Forwarding (BSMF), which is decentralized and can detect both existing and newly appearing obstructions in the supplier and consumer cells matching process. Furthermore, a grid-based movement model is introduced for the hopping sensors. Simulation shows that our scheme significantly reduces the communication overhead and achieves relatively constant total energy consumption with varying amount of obstructions.

I. Introduction

Sensor networks have increased in interest to serve a broad range of applications with their ability to monitor large scale real-world phenomena. Hopping sensors are a class of mobile sensors whose mobility design is more adaptable for rugged terrains or difficult areas where wheeled mobility cannot perform well (Figure 1). In addition, their diminishing cost enables these inexpensive sensors to be deployed in large numbers. The applications of hopping sensor networks working in such areas include weather sensing, environment monitoring, disaster management, and battlefield surveillance [1]. Along with the benefits of hopping sensor networks there are several challenges such as imprecise movement, limited power and occasional failure owing to problems regarding the node, link and global maintenance and communication [2].

In addition to the inadequacies of hopping sensors, rugged terrains may also adversely affect their behaviors. Rugged terrains are the areas that are largely inaccessible to wheeled vehicles, where hopping sensors may operate by trading-off movement accuracy. We model the rugged terrains with two characteristics. The first one is their unevenness with diverse slope patterns. Due to such diversity and arbitrary nature of the terrains we did not model them with slopes in the relocation process. However, we find that hopping sensors are more energy efficient when moving on certain slope ground by comparing their energy consumption with the wheeled ones.

The second feature is the existence of great obstructions in the field that can significantly influence the sensor relocation. To illustrate, water, huge rocks, and other obstructions are inaccessible areas to deploy sensors. It is critical to design a movement scheme that can wisely avoid or circumnavigate these obstructions to reach the expected position. Given an initial sensor deployment, the proposed solution is to match and relocate a redundant sensor to the needed position in an obstructed environment. Furthermore, it must also consider the limited number of hops per sensor and the incapability of sensors landing at a precise location.

The rest of the paper is organized as follows. Section II briefly covers related work. In Section III, the motivation for using hopping sensors is presented. A description of our system model is in Section IV. The matching scheme is in Section V. Section VI defines the Grid Based Movement Model. Section VII provides simulations and performance analysis. Conclusion and future work are presented in Section VIII.

II. Related Work

Sensor relocation problems are extensively studied in [3]–[8]. When a sensor node depletes its energy or fails, sensor coverage holes evolve. Areas with redundant sensors can supply sensors to be relocated to these holes to maintain coverage.

Grid-based solutions for sensor relocation have been widely proposed [4], [7]–[9]. Wang, et al. [7] introduced two kinds of nodes, the grid head and its members, in their model and adopted a Grid-Quorum approach. Their scheme

Fig. 1. Hopping sensors we designed deployed near a construction area

This research is supported in part by NSF grants NO. CNS-0551464 and CNS-0721441, U.S. Army Contract No. W56HZV-04-2-001, and U.S. Army Research Office Contract No. W911NF-07-D-001.

Yuanteng Pei, Fernando J. Cintrón, and Matt W. Mutka are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA {peiyuant,cintrong,mutka}@cse.msu.edu

Jianguo Zhao and Ning Xi are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA {zhaojia1,xin}@egr.msu.edu

makes use of some notations. A *quorum* is mentioned in [10] as an element in a subset of a given set. In [11], a quorum system is defined as a collection of sets such that the intersection of any two sets is always non-empty. Grid cells with redundant sensors are identified as *suppliers*. They send an *advertisement message* to form a *supply quorum*. *Consumers* are grid cells in need of sensors, which send a *request message* to form a *consumer quorum*. However, with the presence of obstructions, supply and consumer quorum are not guaranteed to intersect on the grid for the matching between the supplier and consumer cells.

Similarly, [3] deals with the matching and relocation for hopping sensors. However, the issue that potential obstructions hamper the relocation process also remains unsolved. In [12], the problem of clustering sensors under obstructive and inhospitable conditions is addressed. However, matching and relocation are not specifically tackled. In this paper, we propose a simple and distributed Grid-Quorum based solution to effectively cope with obstructive environment like rugged terrains with hopping sensors.

III. Motivation

The motivation of using hopping sensors over wheeled ones in rugged terrains is twofold. The first one is that hopping sensors can access many difficult areas whereas wheeled sensors cannot. Second, for certain types of slopes, which frequently exist in rugged terrains, hopping sensors are more energy efficient. In this section, we focus on the second motivation to compare the energy consumption for two cases: flat plane and slope. To make the comparison reasonable, only the energy used to drive the sensor is considered.

We first conduct an analysis on a flat plane. Under ideal condition, the hopping sensor's trajectory is a projectile motion. In [3], wind influence is modeled as a predictable and stable parameter, which is a strong assumption due to its dynamic nature. Here we assume a negligibly weak wind setting and no sliding occurs during initial state of hoping. Given the sensor's mass m, the initial velocity v, the takeoff angle α, and the distance traveled by the sensor d, the energy per hop will be:

$$E_h = \frac{1}{2}mv^2 = \frac{mgd}{2\sin 2\alpha} \quad (1)$$

Fig. 2. Hopping sensors we designed in jumping action

For wheeled sensors, we suppose they have the same weight and need to traverse the same distance. Let the rolling friction coefficient for the flat plane be μ and the sensor run at a constant speed. Then energy for traversing such a distance is:

$$E_w = \mu mgd \quad (2)$$

To compare:

$$\frac{E_w}{E_h} = 2\mu \sin 2\alpha \quad (3)$$

The rolling friction coefficient between wheels and land is generally less than 0.1 [13], therefore $E_w/E_h < 0.2$, which means $E_h > E_w$. Therefore, for flat plane, the wheeled sensor is more energy efficient. This is because part of the energy for hopping sensors is converted to the potential energy for hopping height.

Now suppose both sensors need to traverse the same distance along a slope with an inclination angle β. For hopping sensors, with the same assumption in the flat plane, the jumping distance along the slope can be found from the intersection of the projectile trajectory and the slope. Establish a coordinate frame as shown in Fig. 3, where O is the starting point. Then the x coordinate for the intersection point is:

$$x = d(1 - \cot\alpha \tan\beta) \quad (4)$$

Thus $d' = x/\cos\beta$. The energy for hopping sensors to travel d' is the same as traveling d on the flat plane.

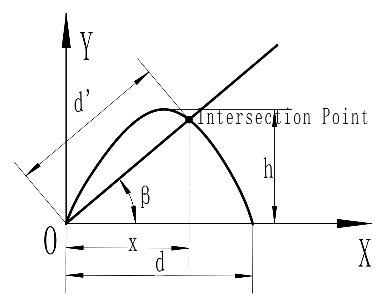


Fig. 3. Coordinate frame for slope plane

For wheeled sensors, using the same assumption for flat plane, the energy consumption of traversing d' along the slope is:

$$E_w = d(\mu mg + mg\tan\beta)(1 - \cot\alpha\tan\beta) \quad (5)$$

To compare:

$$\frac{E_w}{E_h} = 2\sin 2\alpha(\mu + \tan\beta)(1 - \cot\alpha\tan\beta) \quad (6)$$

Typically, the rolling friction coefficient for car tire on concrete is between [0.01, 0.015] [13]. Let $\mu = 0.0125$. Given $\alpha = 75°$, the takeoff angle of the hopping sensor we have developed [14] (Figure 2), we derive the diagram in Fig. 4 from equation (6) to show the change of above energy ratio with respect to slope angle $0° \leq \beta \leq 75°$.

As we can see from the figure, when $30° < \beta < 72°$ (the curve between two vertical lines), $E_h < E_w$. The ratio

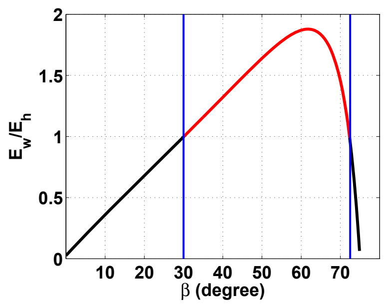


Fig. 4. Energy ratio for $0° \leq \beta \leq 75°$

decreases after about $62°$ because d' becomes increasingly smaller as the inclination angle is close to the takeoff angle, resulting in less energy consumed by wheeled sensor. The energy per hop for hopping sensors, however, remains the same. This explains why the ratio decreases. Nevertheless, most wheeled sensors can only move along slope with an angle less than $45°$ [15]. Hence we can neglect the declining part after approximately $62°$, and conclude the hopping sensor is more energy-efficient than the wheeled sensor for slope angle greater than $30°$ for takeoff angle $75°$. In fact, with different takeoff angles, a critical angle always exists when a hopping sensor consumes less energy than a wheeled one.

IV. System Model

Now that we have shown the hopping sensors are energy efficient on uneven ground, in this section we discuss the system model. A grid-based architecture is a natural solution in a network where nodes are relatively regularly deployed. Each grid cell is controlled by a gateway sensor which is capable of communicating with its peers in its immediate four neighbors (North, South, East, West). The gateway sensors can retrieve their absolute positions and have a larger sensing range than other sensor nodes. Sensor nodes within a grid cell register to the cell gateway and perform their required tasks.

The width (size) of a square grid cell is defined by: $W_c = R_g/k$, where R_g is the effective communication range of the gateway and k is the coverage factor. Assuming the gateway located at the center of the cell and given k=1.41 or k=1.5, 88.4% or 97.2% of 4-neighbor area can be covered respectively for effective inter-gateway communication. Due to space limitation we leave the analysis of the optimal grid cell size in future work.

When a sensor dies, a redundant sensor needs to be relocated to cover the sensing hole area. A redundant sensor can be easily identified on a given cell by the gateway. To maintain the sensor coverage when sensor holes appear, two major steps remain. In the first step, a match is needed between the supplier and consumer cell. Note that more than one supplier may exist. The consumer decides which supplier is selected. Then, a viable path from the supplier to the consumer needs to be computed for the movement. Second, the consumer triggers the relocation process by notifying the selected supplier, which in turn selects one of its redundant sensors and commands it to reallocate to the neighbor cell included in the path. We mainly focus on single sensor relocation. The problem of multiple sensor relocation can be easily solved by executing the scheme repeatedly.

The principal objective of this work is to provide an optimized matching path between a consumer and a supplier cell, involving the least amount of intermediate cells possible. A matching process is fundamentally important to the actual sensor movement and must remain workable in presence of obstructions. In our model, obstructions are represented as non-functional grid cells in which a gateway is hard to be placed, or if it exists, it fails to communicate with neighboring cells. These obstructed cells, with high probability, will not allow an intersection cell to match consumer and supplier cell. To overcome this problem, centralized or decentralized algorithms can be considered.

A centralized solution usually inherits the single-point-of-failure weakness, which is less fault-tolerant in hostile environment or a rugged terrain where adverse circumstances are frequent. In addition, it is a strong assumption that a single node has adequate energy and computation-capability to communicate with all the other sensor nodes and to find the optimized path for relocation. Since the existence of supplier and consumer is ad hoc and the relocation should be executed efficiently, the latency incurred by gathering supplier and consumer information becomes a drawback.

To be more fault-tolerant, less susceptible to the impact of mobility, and to reduce message overhead and latency in the process of collecting information [16], our algorithm should depend only on local information collected by each gateway to establish the matching path between the supplier and the consumer. To make it more practical, gateway communication is restricted to be with the four-neighboring gateways, and between gateway and managed sensors in the cell.

V. Matching Process

A. Message Forwarding Process

Previous Grid-Quorum relocation would fail if there are obstructions in the middle of the supplier or consumer quorum. We propose Binary Splitting Message Forwarding (BSMF) to cope with obstructions. The algorithm provides a matching path for a consumer and supplier cell, and it considers previously known and newly appearing obstructions.

When a gateway receives an advertisement or request message, it verifies whether it satisfies the demand or needs to forward the message. Before forwarding messages, gateways check the availability of the succeeding neighbor. Succeeding neighbors will vary depending on the message type. Advertisement messages are forwarded to neighbor grid cells in a grid row manner (East-West), while request messages are forwarded in a grid column manner (North-South). When a gateway in a grid cell receives both the advertisement and the request message, the cell becomes an intersection node and the gateway in it will be responsible to match the request to the advertisement. Without obstructions,

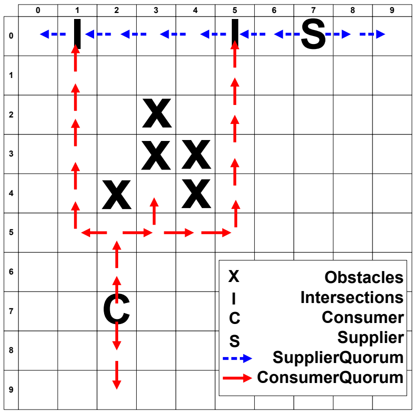


Fig. 5. Binary Splitting Message Forwarding

it is ensured by geographic relations that an intersection is made [7]. If a known or a new obstruction is detected in the next forwarding grid cell, this gateway will change the course by modifying message forwarding behavior in an effort to successfully forward the message. Given an obstruction, a request message may attempt to be forwarded to the East-West grid cell neighbors, while an advertisement message may be forwarded to the North-South neighbors. However, they will return to their default (initial) directions whenever it is possible. If both of the intended grid cell destinations are obstructed or visited before, the message will be forwarded in a direction opposite to the default until one available grid cell is found. Algorithm BSMF gives a step representation of this procedure.

Algorithm 1: Binary Splitting Message Forwarding (BSMF)

Input: Current cell's gateway G receives an incoming message with i) default forward direction D_{fw}; ii) message traversed path
Output: Destination neighbor cell(s) $\{NC_d\}$ of the message

Begin: Supplier and consumer cells have sent row advertisement or column request messages respectively
// Set opposite and two perpendicular directions of D_{fw}
$D_{opp}, D_{left}, D_{right} \leftarrow D_{fw}$
// NC denotes a "neighbor cell"
if *NC on D_{fw} is* available $\wedge$ unvisited **then**
  $\{NC_d\} \leftarrow NC$ *// On default whenever possible*
else
  $\{NC_{uv\_av}\} \leftarrow$ unvisited $\wedge$ available NC on D_{left}, D_{right}
  if $\{NC_{uv\_av}\} \neq \emptyset$ **then**
    $\{NC_d\} \leftarrow \{NC_{uv\_av}\}$
  else
    if *NC on D_{opp} is* available **then**
      $\{NC_d\} \leftarrow NC$ on D_{opp}
    else
      $\{NC_d\} \leftarrow NC$ from which the message was sent
foreach $i \in \{NC_d\} \wedge$ *i is out of border* **do** $\{NC_d\} \leftarrow \{NC_d\} \backslash i$
The message is forwarded to $\{NC_d\}$
if *both advertisement and request messages are found* **then**
  The advertisement of the closest supplier with the request are forwarded to $\{NC_d\}$ on the path to the consumer

Figure 5 illustrates an example of BSMF. A supplier S(7,0) sends the advertisement message in the first row (X,0) while the consumer C(2,7) initiates the request message in column (2,X). When the gateway at (2,5) receives the request message, it sends a message to check grid cell (2,4) availability. As (2,4) fails to reply, it indicates a new obstacle. Consequently the message is split to (2,5) West and East neighbors (1,5) and (3,5). The message continues to be forwarded in the column of (1,X), while in (3,4), the gateway finds the targeting grid cell (3,3) is an obstructed one. It attempts to ask its East-West neighbors to help forwarding but they do not respond. The gateway at (3,4) sends the message back to (3,5). The gateway at (3,5) will try to split the message but since it finds that (2,5) has been visited it forwards the message to (4,5) only. As (4,4) contains an obstacle, (4,5) will forward the message to (5,5) as well, which then is able to forward in an upward direction. Finally the intersections I(1,0) and I(5,0) are made. Both of them will forward the path information to the consumer. The consumer in (0,7) calculates the length of the two paths gathered as 15 and 13 and selects the path with length 13. It then triggers the movement process by notifying the supplier.

When multiple suppliers exist, the consumer selects the supplier by the following rules. (i) An intersection cell informs the consumer of the advertisement from the closest suppliers. An intersection cell formed by immediate neighboring supplier does not continue to forward the advertisement in the same row. (ii) A consumer cell waits for a predetermined time after receiving the first supplier quorum to allow the arrival of other existing quorums. (iii) The level of sensor redundancy breaks the tie when multiple supplier paths are of equal distance. A supplier cell with more sensors is chosen.

B. Path Optimization

The matching process of the sensor supplier and consumer lays a foundation for the actual sensor movement. With the presence of obstructions, the path in the matching process may include some redundant cells that can be eliminated from the actual relocation process. Such an optimization can further reduce the movement time in the relocation process, which is defined by two additional rules below to check the cells in the matching process. Figure 6 also illustrates the idea.

- Rule 1 (Remove dead-end routes): The cell sending a

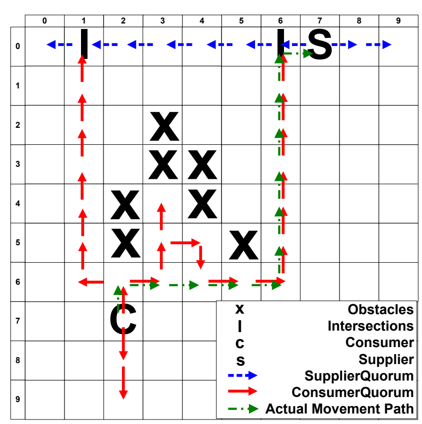


Fig. 6. Movement Path Optimization

backward message is removed. (e.g., In Fig.6, cell (3,4) is removed.)

- Rule 2 (Remove non-straight neighboring routes): The nodes between two adjacent neighboring path nodes are removed. (e.g., In Fig.6, cells (3,5), (4,5) are removed.)

VI. Sensor Migration

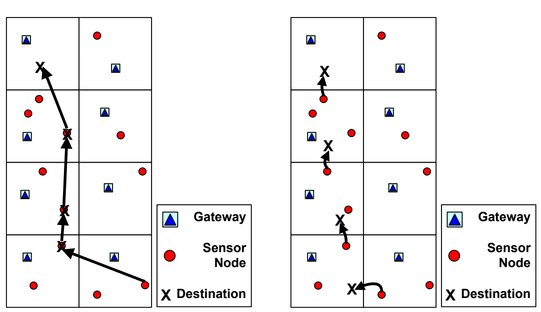
A. Grid Based Movement Model

Having obtained the matching path between the consumer and supplier cells, we need to determine how to move the sensor to the target location. There are typically two migration methods:

- Direct movement: transfer the sensor from supplier to consumer directly;
- Cascaded movement: using intermediate nodes as relaying ones.

Nevertheless, both methods are based on Precise Movement Model (PMM) [7], which uses a sensor to exactly replace another one. In [3], although the authors assume that the sensors can reach the destination if they hop into the target cluster, this is also PMM since the clusters are treated as points. Thus each sensor's target is also precise. Due to the inaccurate nature of hopping sensors, we propose a Grid Based Movement Model (GBMM), in which the sensor is only required to move to the target grid cell. Such a pattern fits hopping sensors better since it is not easy for them to be relocated to the exact position and the gateway can easily manage the sensors within its grid cell. Moreover, relaying sensors can move concurrently to save time.

Using GBMM, the most energy efficient way is to transfer the sensor closest to the border between its current cell and the neighboring target cell. The main idea of PMM and GBMM is shown in Fig. 7. In Fig. 7(a), the predecessor is to move to the successor's position to replace it until the last sensor arrives at the destination. Fig. 7(b) illustrates the GBMM. As long as a sensor enters its destination cell, its relocation is accomplished. When the path is the same, it is easy to observe that GBMM can consume less energy compared with PMM because of the decrease of the path length for each transfer.



(a) Precise Movement Model (b) Grid Based Movement Model

Fig. 7. PMM vs. GBMM

B. Energy Computation

Using the hopping model in [3] and given the average distance per hop for the sensor as r, the final landing point is subjected to a two dimensional normal distribution $\Delta\mathbf{r} \sim N(\mathbf{0}, \delta^2\mathbf{I})$, where $\mathbf{I}$ is the $2{\times}2$ identity matrix. Note that we assume the independence of the two dimensions because the final landing point is independent of the directions. Assuming the acceptable landing area is within 3δ (this can ensure most of the jumps land in the target area), the number of hops to traverse a distance l satisfies:

$$\lfloor \frac{l}{r+3\delta} \rfloor \leq N \leq \lceil \frac{l}{r-3\delta} \rceil \quad (7)$$

We use the floor on the left and ceil on the right to ensure that the robot can travel the distance l. Assume the number of cells along the path is $n+1$ including the starting and destination cell, then the number of cell crossing is n. Suppose the distance from the closest sensor in each cell i to the border is l_i, $(i = 1, \cdots, n)$ (this can be maintained or computed by the gateway), then total hops H for GBMM is between:

$$\sum_{i=1}^{n} \lfloor \frac{l_i}{r+3\delta} \rfloor \leq H \leq \sum_{i=1}^{n} \lceil \frac{l_i}{r-3\delta} \rceil \quad (8)$$

If the energy for each hop is E, then we can get the upper bound and lower bound for the energy as EH for migration. We can also estimate the maximum energy as:

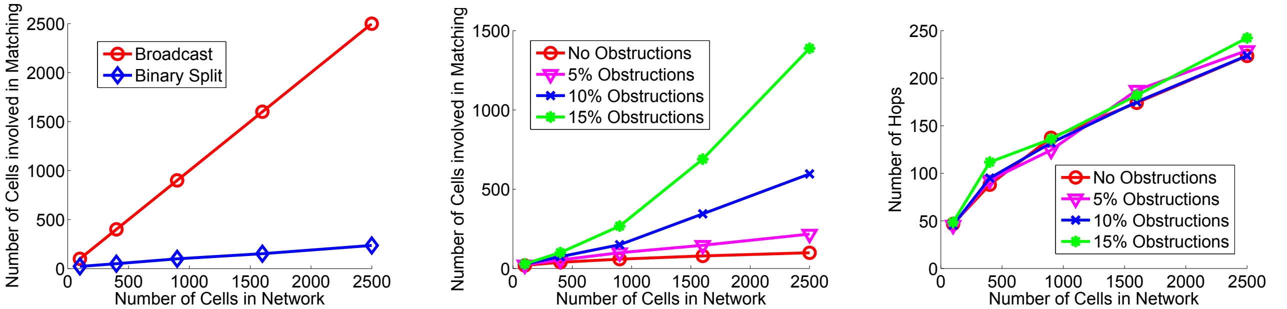
$$En \lceil \frac{W_c}{r-3\delta} \rceil \quad (9)$$

because l_i is always less than the width of grid cell W_c.

VII. Simulation

To validate the correctness and effectiveness of our algorithm and proposed models, we simulated the hopping sensor network with random positioned supplier and consumer cells. Scenarios were designed with grid sizes of $10*10$, $20*20$, $30*30$, $40*40$ and $50*50$ grid cells. Each grid cell, with 3-5 hopping sensors randomly distributed, represents an area of $60*60$ square meters, which allowed us to represent fields areas up to $9km^2$. Hopping sensors are assumed to have a hopping range with a landing precision radius varying from 2.1 to 3.9 meters. Each hop is modeled to consume a random time between 2 and 5 seconds. Obstructions are randomly distributed in the network and are simulated as non-functional grid cells. Simulations also compared varying number of obstructions with: 0%(no obstructions), 5%, 10%, 15% of obstruction ratio to the total number of grid cells. For each simulation, experiments are run 50 times to calculate average values.

The first two simulations are to evaluate the performance in the matching process by measuring network load (grid cells involved in packet forwarding). As shown in Figure 8(a), BSMF has a much fewer number of grid cells involved in the matching process with 5% obstruction ratio compared to the broadcast approach which guarantees a matching if it ever exists. The conventional quorum-based



(a) Broadcast vs. Binary Splitting Matching

(b) Number of Cells in Matching with Varying Obstructions

(c) Number of Hops in Movement with Varying Obstructions

Fig. 8. Simulation Results

matching fashion is not compared as it fails with presence of obstructions. In the extreme case, BSMF involves only 9.4% cells to participate compared with broadcast. In Figure 8(b), the performance of BSMF for varying grid sizes and amount of obstacles is tested and verified. On average, BSMF involves 80.2% less cells compared to the broadcast approach. Fewer amounts of involved cells indicate much lower total energy consumption, considerably less packet transmission and network traffic generated in the process.

The third simulation evaluates the number of hops and time in the sensor movement process. Its objective is to measure the total amount of hops taken by sensors to perform the relocation movement after obtaining the relocation path by BSMF. As presented in Figure 8(c), the varying number of obstacles has little impact on the number of hops in our model, which demonstrates that the movement spends almost a constant amount of energy, regardless of the amount of obstacles in the movement process. The nearly constant number of hops is contrasted by the trend of network traffic in the second simulation. This result can be explained as the movement is only related to the quorum with the "optimized" intersection, while the matching process is to generate all the possible routes to link the consumer and supplier.

VIII. Conclusion and Future Work

We studied sensor relocation considering obstructions and proposed an enhanced Quorum-Grid relocation solution with an optimized BSMF algorithm. The proposed algorithm is decentralized and can detect new obstructions in the sensor supplier and consumer matching process. Furthermore, the grid-based movement model is developed and studied with energy computation, suitable for imprecise-movement hopping sensors. The relocation cost has been taken into account, by implementing a path optimization during matching process execution. Simulation with different grid sizes, random distribution of hopping sensors, and varying amount of obstructions shows that our scheme reduces the involved number of cells in the matching process by 80.2% on average compared to the broadcast approach and achieves relatively constant total energy consumption by evaluating total number of hops in the actual movement.

In future work, we will (i) model and analyze the unevenness and patterns of the rugged terrain; (ii) enhance the algorithm to cope with newly appearing obstructions during the sensor movement process; (iii) compute the optimal grid cell size for the grid structure.

References

[1] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks - Technology, Protocols and Applications*. John Wiley & Sons, 2007.

[2] M. Ringwald and K. Römer, "Deployment of sensor networks: Problems and passive inspection," in *5th Workshop on Intelligent Solutions in Embedded Systems. WISES 2007*, 2007.

[3] Z. Cen and M. W. Mutka , "Relocation of hopping sensors," in *Robotics and Automation. ICRA 2008. IEEE Int. Conf. on*, May 2008.

[4] S. Chellappan, X. Bai, B. Ma, and D. Xuan, "Sensor networks deployment using flip-based sensors," in *Mobile Adhoc and Sensor Systems. MASS 2005. IEEE Int. Conf. on*, 2005.

[5] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotics Systems. DARS 2002. Int. Symposium on*, 2002.

[6] G. Wang, G. Cao, and T.L. Porta, "Movement-assisted sensor deployment," *INFOCOM 2004. 23rd Conference of the IEEE Computer and Communications Societies*, vol. 4, 2004.

[7] G. Wang, G. Cao , T.L. Porta, and W. Zhang , "Sensor relocation in mobile sensor networks," *INFOCOM 2005. 24th Conference of the IEEE Computer and Communications Societies*, vol. 4, 2005.

[8] J. Wu and S. Yang, "Smart: a scan-based movement-assisted sensor deployment method in wireless sensor networks," *INFOCOM 2005. 24th Conference of the IEEE Computer and Communications Societies*, vol. 4, 2005.

[9] R. Akl and U. Sawant, "Grid-based coordinated routing in wireless sensor networks," *Consumer Communications and Networking Conference. CCNC 2007. IEEE*, 2007.

[10] G. Cao and M. Singhal, "A delay-optimal quorum-based mutual exclusion algorithm for distributed systems," *Transactions on Parallel and Distributed Systems. T.PDS*, vol. 12, no. 12, 2001.

[11] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks," *Mobile Networks and Applications*, vol. 10, no. 1-2, 2005.

[12] M. Younis, P. Munshi, G. Gupta, and S. Elsharkawy , "On efficient clustering of wireless sensor networks," *Dependability and Security in Sensor Networks and Systems. DSSNS 2006. IEEE.*

[13] T. E. Toolbox, "Friction and coefficients of friction," http://www.engineeringtoolbox.com/rolling-friction-resistance-d_1303.html.

[14] J.G. Zhao, R.G. Yang, N. Xi, B.T. Gao, and X. G. Fan, "Development of a miniature self-stabilization jumping robot," *Intelligent Robots And Systems, IROS 2009. IEEE/RSJ Int. Conf. on*, 2009.

[15] http://www.activrobots.com/.

[16] N. Li, J. Hou, and L. Sha, "Design and analysis of an mst-based topology control algorithm," *INFOCOM 2003. 22nd Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 3, 2003.