

Proxy-Based Sensor Deployment for Mobile Sensor Networks

Guiling Wang, Guohong Cao, and Tom La Porta
Department of Computer Science & Engineering
The Pennsylvania State University
University Park, PA 16802
Email: {guiwang,gcao,tlp}@cse.psu.edu

Abstract

To provide satisfactory coverage is very important in many sensor network applications such as military surveillance. In order to obtain the required coverage in harsh environments, mobile sensors are helpful since they can move to cover the area not reachable by static sensors. Previous work on mobile sensor deployment is based on a round by round process, where sensors move iteratively until the maximum coverage is reached. Although these solutions can deploy mobile sensors in a distributed way, the mobile sensors may move in a zig-zag way and waste a lot of energy compared to moving directly to the final location. To address this problem, we propose a proxy-based sensor deployment protocol. Instead of moving iteratively, sensors calculate their target locations based on a distributed iterative algorithm, move logically, and exchange new logical locations with their new logical neighbors. Actual movement only occurs when sensors determine their final locations. Simulation results show that the proposed protocol can significantly reduce the energy consumption compared to previous work, while maintaining similar coverage.

1 Introduction

Recent advances in digital electronics, microprocessor, micro-electro-mechanics, and wireless communication have enabled the deployment of large scale sensor networks where thousands of small sensors are distributed over a vast field to obtain fine-grained, high-precision sensing data. Due to many attractive characteristics of sensor nodes such as small size and low cost, sensor networks [6, 9, 12, 14, 2] become adopted to many military and civil applications, from military surveillance to smart home [15], from formidable remote environment monitoring, to in-plant robotic control and guidance, from data collection on other planets to guarding the agricultural field. Due to the limited sensing range of the sensor nodes, deploying

sensors appropriately to reach an adequate coverage level is critical for the successful completion of the issued sensing tasks [4, 16].

Sensor deployment has received considerable attention recently. Many of these works [4, 5, 11, 10] assume that the environment is sufficiently known and under control, and sensors can be deployed manually. However, when the environment is unknown or hostile, such as remote harsh fields, disaster areas and toxic urban regions, sensor deployment cannot be performed manually. To scatter sensors by aircraft is one possible solution. However, using this technique, the actual landing position cannot be controlled due to the existence of wind and obstacles such as trees and buildings. Consequently, the coverage may be inferior to the application requirements no matter how many sensors are dropped. In these scenarios, it is necessary to make use of mobile sensors, which can move to the right places to provide the required coverage.

There have been some research efforts on deploying mobile sensors, but they are based on centralized approaches. For example, Zou and Chakrabarty [19] proposed to let a cluster head or central server collect the initial location of mobile sensors, calculate their target locations to increase the coverage and notify them to move. However, in many sensor deployment environments such as disaster recoveries and battle fields, a central server may not be available and it is hard to organize sensors into clusters due to network partitions. Further, the centralized approach suffers from the problem of single point failure.

Recently, several distributed sensor deployment algorithms [7, 8, 19, 18], assuming all sensors are mobile, have been proposed. However, to equip each sensor with a motor will increase the network cost and may not be necessary under certain initial distribution, such as random initial deployment. In these cases, a mix of static and mobile sensors may be able to provide the satisfactory coverage while keeping the network cost low. In our previous work [17], we designed a bidding protocol for deploying a mix of static and mobile sensors. In this protocol, mobile sensors move itera-

tively from small coverage holes (areas not covered by any sensors) to larger ones to increase the sensor coverage. The algorithm terminates when the sensor movement cannot increase the sensor coverage. Although the bidding protocol can be used to achieve high coverage, sensors may move in a zig-zag way and waste lots of energy compared to moving directly to the *final location*, which is not known initially and sensors have to find out through the round by round process.

In this paper, we design a proxy-based sensor deployment protocol to distributedly identify the target locations of mobile sensors, and then move there directly. To achieve this goal, we propose the idea of *logical movement*, where sensors logically move from small holes to large holes by exchanging messages, and assigning proxies. Mobile sensors move directly to their final locations only after they find those locations. Simulation results show that the proposed protocol can significantly reduce the energy consumption, while maintaining similar coverage.

The rest of the paper is organized as follows. In the next section, we give an overview of the basic bidding protocol. The proxy-based protocol will be presented in section 3. Section 4 presents performance evaluations, and Section 5 concludes the paper.

2 The Bidding Protocol

As far as we know, the basic bidding protocol [17] is the only previous work addressing the problem of deploying a mix of mobile and static sensors. In the basic bidding protocol, a mobile sensors is treated as the server to heal the coverage hole and its service has a *base price* which is related to the size of any new hole generated by its movement. Static sensors detect coverage holes locally by utilizing the Voronoi diagram [3]. They estimate their sizes as *bids*, and *bid* the closest mobile sensors with a *base price* lower than their *bids*. Mobile sensors choose the highest bids, move to heal the coverage holes detected by the bidder, and update their base prices to be the bid. At the new locations, mobile sensors receive bidding messages from static sensors which lost in the previous bidding or from static sensors which did not find suitable mobile sensors to bid. Mobile sensors move again if new biddings are accepted. In this way, mobile sensors move iteratively from small coverage holes to larger ones, and stop when they arrive at the largest coverage holes detected by the static sensors.

The deployment is a round-by-round process. Each round is composed of three phases: service advertisement, bidding, and serving. Mobile sensors broadcast their base prices and locations in the service advertisement phase. Static sensors detect coverage holes and send bidding messages to the closest mobile sensors in the bidding phase. In the serving phase, mobile sensors choose the highest bid,

move to heal the hole and update the base price.

One more thing to clarify is the *multiple healing*. One coverage hole may be detected by several static sensors, and these static sensors may bid different mobile sensors separately. If they all win the bidding, those mobile sensors will move to heal the same hole. This situation is called *multiple healing*. In the basic bidding protocol, we propose to let mobile sensors self-detect the occurrence of *multiple healing*. Once this happens, all involved mobile sensors except the winner reduce their base prices to zero and advertise them in the new round. In the new round, these mobile sensors will be bid to heal other holes.

2.1 Problems of the Bidding Protocol

Although the basic bidding protocol can achieve a high coverage, there is still room for improvement in terms of energy efficiency and load balance. First of all, moving iteratively in a zig-zag path consumes much more energy than directly moving to the final location. Secondly, multiple healing wastes energy since only the winner will remain and others have to move again.

Finally, some sensors may move much longer than others. As shown in Figure 1, in the first round, holes A, B, C bid mobile sensor s_b , and hole D bids s_a . Hole A and hole D win due to their large size, and these two sensors move. Then hole C bids for sensor s_a and hole B does not bid for one round since it does not know the existence of s_a due to the limited advertisement radius. After s_a moves to hole C, hole B knows it and bids it. Then s_a has to move the third time to reach its final location and it moves a much longer distance than s_b in total. From the figure we can see, the mobile sensors closest to the large holes are likely to move a short distance since after they move to heal those large holes, they are not likely to move again with a high base price. However, other sensors may have to move a long zig-zag way from small holes to larger and larger holes, which are located far away from them. If some balancing scheme exists, e.g., moving s_a to hole A and s_b to hole B, these two sensors will have more balanced load.

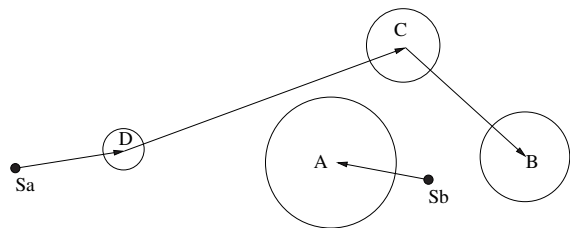


Figure 1.

Our proxy-based deployment protocol is motivated by the concern of energy efficiency. It overcomes the weakness of the basic bidding protocol by conducting one-time

movement, multiple healing avoidance and load balancing, while achieves almost the same coverage. We present the proxy-based protocol in the next section.

3 Proxy-based Sensor Deployment Protocol

3.1 Basic Idea: Logical Movement and Proxy Sensor

As mentioned in the previous section, instead of iterative movement, mobile sensors should identify the largest coverage holes first and then conduct one-time movement. However, in a distributed manner, the global largest holes cannot be obtained beforehand. This is why in the basic bidding protocol, mobile sensors identify the largest holes through iterative movement. They move to larger and larger holes, with an increasing base price and stop only when no bidding message is received, which means no static sensor can detect a larger hole and propose a higher bid than their base prices.

To identify the largest coverage holes, but not through iterative movement, we propose the idea of *logical* movement, which means mobile sensors logically move from small holes to large holes iteratively, and only move after knowing that no larger holes can be detected. This *logical* movement can also solve the problem of load balance by exchanging the logical positions among mobile sensors. If the distance of a mobile sensor to its logical position is too long, it can exchange its logical position with another sensor before really move so that no sensor is penalized to move too long. In addition, the *logical* movement can naturally solve the problem of *duplicate healing*, since they do not really move until the last step.

The difficulty in realizing the logical movement is how to obtain the position-related information without really being placed in the logical position. According to be bidding protocol, mobile sensors advertise their base prices within certain advertisement radius, collect the bidding messages from the static sensors, and then move to the new location (the largest hole in the advertisement area), and advertise again. One possible solution is to use network-wide broadcast. However, this method will greatly increase the traffic overhead. To obtain the location-related information with a low message complexity, we propose that each mobile sensor chooses a static sensor closest to its logical position as its proxy. Proxy sensors are responsible for advertising the base prices of mobile sensors and choosing the largest holes for them. Proxy sensors can also negotiate with the proxy of other mobile sensors about exchanging the target locations for load balance. New proxies will be selected as mobile sensors logically move. Finally, if proxy sensors do not receive any bidding messages, they will notify their delegated mobile sensors to move.

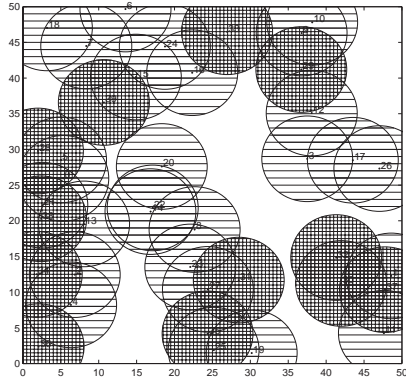
3.2 Protocol Overview

In the proxy-based protocol, mobile sensors act as hole-healing server, whose service has certain base price determined by the size of the new hole to be generated by their leaving. Static sensors detect coverage hole locally, use hole size as the bid and bid mobile sensors. Mobile sensors accept the highest bid, logically move to that hole and delegate the winning bidder as their proxy. The winning bidders must be the static sensors closest to their logical positions because static sensors detect coverage holes in their local area, the largest hole they find out must be closest to them than other static sensors. Therefore, the static sensor that wins the bidding of a mobile sensor must be the sensor closest to the logical position of that mobile sensor.

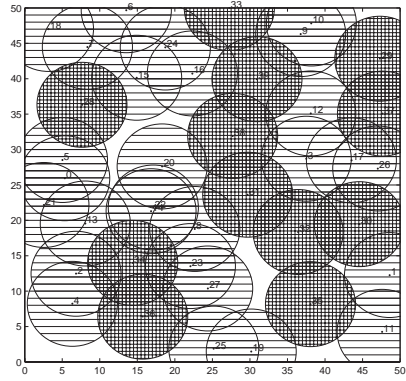
Proxy sensors advertise the service of their delegated mobile sensors, collect bidding messages and choose the highest bid. Then they delegate the bidder as the new proxy. Proxy sensors also examine the possible moving distance of their mobile sensors and do *hole exchange* for them if needed. When no bidding message is received for a number of rounds, denoted as δ , and no hole exchange is needed, proxy sensors conclude that their delegated mobile sensors already obtain the final position, and notify them to move.

Specifically, our protocol runs round by round until mobile sensors obtain their final locations and move there directly. Each round consists of four phases: service advertisement, bidding, logical movement, and hole-exchange. (1) In the service advertisement phase, proxy sensors advertise the logical locations, physical locations and base prices for their delegated mobile sensors. In the first round, a mobile sensor does not have a proxy and advertises its physical location and base price by itself. (2) In the bidding phase, static sensors detect coverage holes, estimate the hole size, choose a mobile sensor to bid and send *bidding* messages to its proxy. At section 3.4, we will describe how to choose mobile sensors in detail. (3) In the logical movement phase, proxy sensors (or mobile sensors without a proxy) choose the highest bid and send a *delegate* message to the bidder. The bidder becomes the new proxy. The base price of mobile sensors is updated by their new proxies. Also, proxy sensors need to check whether hole-exchange is needed. If yes, they choose the mobile sensor suitable for exchange and sends out *exchange* request to the proxy of that mobile sensor. (4) In the hole-exchange phase, proxy sensors check the received requests, choose one with the highest priority and return the confirm message to the requester. Then the mobile sensors delegated by these two proxy sensors exchange the hole to heal. The detail about hole-exchange will be described at section 3.3.

Before getting into the details, we use an example to show how our proxy-based deployment protocol works. Figure 2 (a) is the initial distribution of 40 sensors on a $50m$



(a) Initial deployment

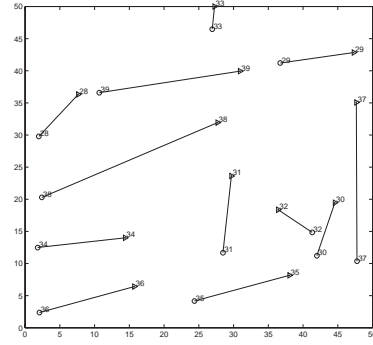


(b) Result of deployment

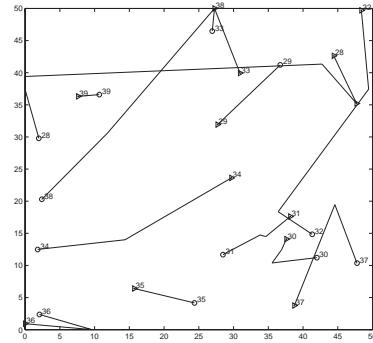
Figure 2. An example

by $50m$ target field, among which, 30% are mobile sensors. The sensing range is $6m$. The striped shadow is the sensing area of static sensors and the gridded shadow is that of mobile sensors. Figure 2 (b) shows the deployment results of our protocol from this initial distribution and Figure 3 (a) shows the moving trace of the mobile sensors. In this example, the mobile sensors move $13.65m$ on average. Sensor 38 moves the longest, $27.85m$. For comparison, Figure 3 (b) shows how mobile sensor moves in the basic bidding protocol under the same initial distribution, where mobile sensors move $23.77m$ on average. Sensor 28 has the longest moving distance. It moves 5 times, and $68.68m$ in total. From this example, we can see our proxy-based protocol is much more energy-efficient and load-balanced.

In the next several sections, we present some details, including *hole-exchange*, the setting of δ , how static sensors choose mobile sensors to bid, how to avoid the *multiple healing* problem, etc. After that, a formal description of our protocol is presented.



(a) In the proxy-based protocol



(b) In the basic bidding protocol

Figure 3. Comparison of the moving trace

3.3 Coverage Hole Exchange

There are two goals for *Coverage hole exchanging*. One is to reduce the average moving distance, and the other is to balance the load. Figure 4 is an example of the first case. After the exchange, the moving distance of both sensor s_a and sensor s_b are reduced. Figure 5 is an example of the second case. To heal the coverage hole, sensor s_a has to move a long distance before exchange. Thus, s_a and s_b exchange coverage hole to heal, even though s_b has to move a longer distance after the exchange.

To guarantee that no mobile sensor is penalized, we specify a maximum moving distance, d_{max} . The proxy sensor must examine whether its delegated mobile sensor has to move more than d_{max} . If yes, it checks the received service advertisement list and finds a mobile sensor which can exchange coverage hole with its delegatee so that the moving distance of both mobile sensors is less than d_{max} after the exchange. Then it sends an *exchange* message to the proxy of that mobile sensor specifying how long its delegated mobile sensor has to move without the exchange and wait for the confirmation.

To reduce the overall moving distance, the proxy sensor also calculates whether the overall moving distance can be reduced by exchanging coverage hole, such as the scenario in Figure 4. If yes, the proxy will send *exchange* message to

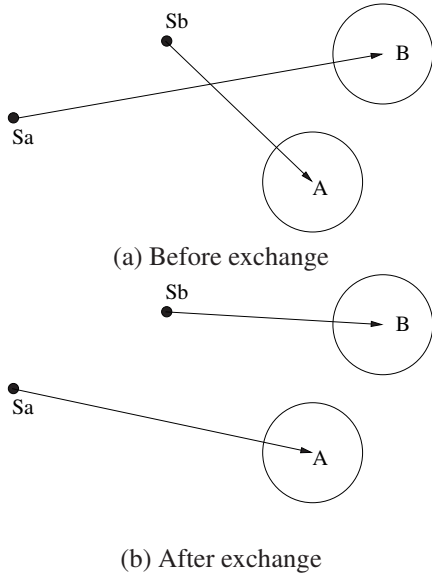


Figure 4. Hole exchange to reduce the average moving distance

the proxy of that mobile sensor. In the *exchange* message, the reduced moving distance after the exchange is specified.

The *exchange* requests have different priority. Generally, the *exchange* for load balance has a higher priority than the *exchange* for reducing the average moving distance since we must guarantee that no mobile sensor moves too long. Among the *exchange* for balancing load, the mobile sensor which may be penalized most without exchanging has the highest priority. Among the *exchange* for reducing average moving distance, the exchange with largest improvement has the highest priority. After receiving a list of *exchange* requests, the proxy accepts a request based on the priority and sends a confirmation message to the proxy whose request has been accepted.

The parameter δ mentioned in the previous section needs special management in hole exchange. δ specifies the number of rounds to wait for a proxy to notify its mobile sensor to move when no bidding message or exchange request is received. This parameter should be differentiated for different mobile sensors. For example, in Figure 1, δ for s_b should be larger than s_a and at least larger than two. Otherwise, when s_a moves logically to hole B after the third round, and its proxy finds it will move too long, s_b already moves to hole A. If they exchange hole at that time, s_b has to move another time. Therefore, we set a higher δ for the mobile sensors with a high base price and a short distance to their logical positions.

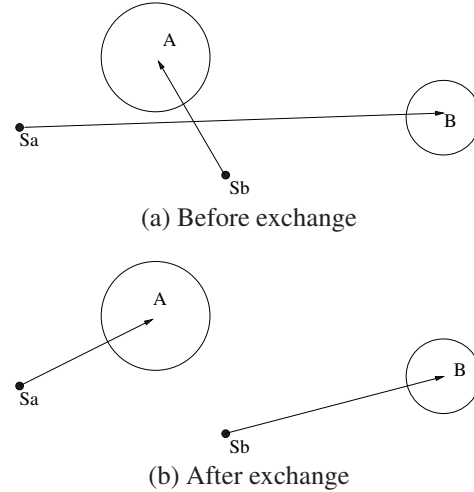


Figure 5. Hole exchange to balance the load

3.4 Distance-based vs. Price-based

After the service advertisement, each sensor has a list of mobile sensors with their locations, logical locations and base prices. Sensors having detected some coverage holes need to determine which mobile sensor to bid. In the basic bidding protocol, sensors bid the closest mobile sensors. We call this criteria as the *distance-based* approach. The distance-based approach has certain problems shown in Figure 6. Hole C bids s_j since it is the closest sensor and s_j moves to heal hole C . s_j 's move generates a new hole B and s_i will be bid to heal it. Apparently, sensor s_i can move directly to hole C , with a similar moving distance to hole B , while s_j can stay fixed, which is much more efficient. This is the motivation that we propose to let sensors choose the mobile sensor with the lowest price to bid (called *price-based* approach). Then hole C will bid sensor s_i since it has a lower base price, and sensor s_j can stay. The advantage of price-based approach is to reduce the number of movements. Each time, the sensor with the lowest price moves to the largest hole and it does not move again since no higher bid can be given.

People may think the price-based criteria also has certain problem, like the situation shown in Fig. 7. With the price-based approach, s_j should heal hole B , while s_i should heal hole C since hole B has a higher bid and it wins the bidding for the cheaper sensor. With the distance-based approach, s_i heals hole B , while s_j should heal hole C since both hole B and hole C bid the closest sensor. It seems like the distance-based approach is better than the price-based approach in this kind of situation. However, our hole-exchange scheme can ask s_i and s_j to exchange hole and get the same result as the distance-based approach. Therefore, we believe the price-based approach is better than the distance-based

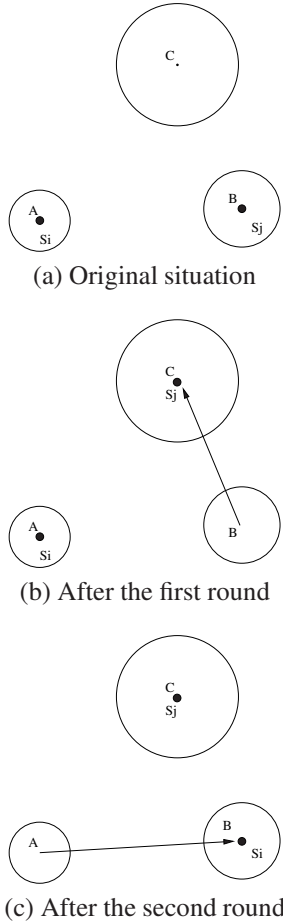


Figure 6. Problem of the distance-based criteria

approach in our proxy-based deployment protocol. Simulation results also verify this.

We mentioned in the previous section that the proxy should wait for δ rounds before sending notification to the mobile sensor if no bidding message is received. The setting of δ is different for distance-based and price-based approaches. In the price-based approach, sensors tend to bid the same mobile sensor since there is only one cheapest mobile sensor within a certain area, while in the distance-based approach, sensors tend to bid different mobile sensors since the sensor closest to one hole is likely different with the sensor closest to another hole. To wait for the same number of rounds, sensors can determine that no more bidding message will come in the distance-based approach, but they cannot in the price-based approach. Therefore, δ should be relatively smaller in the distance-based approach than that in the price-based approach.

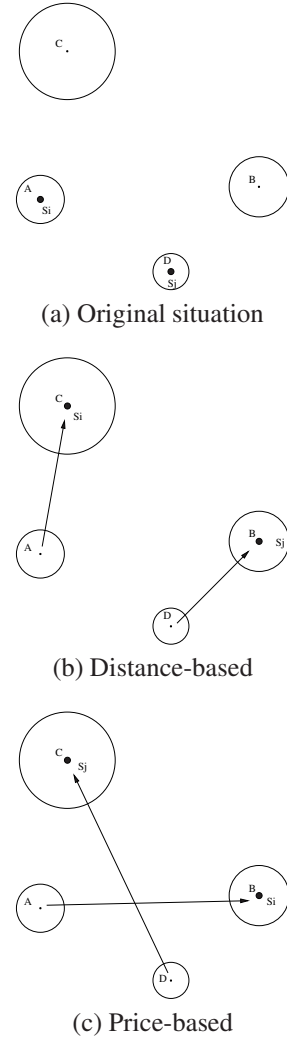


Figure 7. Problem of the price-based criteria

3.5 Multiple Healing Avoidance

Several mobile sensors may be bid to heal the same hole by different static sensors. This happens because of the limited service advertisement radius. Static sensors may have different knowledge about the mobile sensors. It is possible that several static sensors independently bid different mobile sensors for the same coverage hole since the cheapest mobile sensor or the closest mobile sensor in their views are different. If some of them succeed in bidding, multiple mobile sensors may move to heal the same hole, which is not necessary. In the basic bidding protocol, mobile sensors self-detect the occurrence of multiple healing after they already moved. If multiple healing happens, mobile sensors lower their base price to zero and move to other hole in the next round.

In our solution, proxy sensors check whether *multiple*

healing would happen before they notify their mobile sensors to move. To check the *multiple healing*, proxy sensors do the coverage hole detection again, assuming that their mobile sensors are not in their logical position (the hole position). Without multiple healing, there still exists a hole since only their mobile sensors are in the hole area. Otherwise, a multiple healing happens. Once proxy sensors detect multiple healing happens, they reduce the base price of their mobile sensors to be zero, and re-advertise the new base price in the subsequent rounds. In this way, mobile sensors avoid unnecessary movement.

To avoid that all proxies detect the same multiple-healing and reduce the base prices of their delegated mobile sensors to zero altogether, each proxy check whether the moving distance of its mobile sensor from its physical position to the hole is the shortest among those mobile sensors for the same hole. If not, it reduces the base price of its delegate to zero; otherwise, it waits for the leaving of other mobile sensors.

Having presented the idea of the proxy-based protocol, we show the formal algorithm in Fig 8. The hole detection part is not shown since it is the same as the basic bidding protocol.

3.6 Optimization on Reducing Message Complexity

Since communication also consumes energy, we should reduce the message complexity as much as possible. We propose optimizations to reduce the number of bidding messages and service advertising messages, which are the two major control messages in our protocol.

Among all the bidding messages for a mobile sensor, only one with the highest bid will be accepted and others are wasted. Also, several static sensors may detect the same coverage hole and send out multiple bidding messages for it. To remove these redundant bidding messages, sensors first broadcast their bids and the target mobile sensor. If a sensor finds that another sensor chooses the same mobile sensor with a higher bid or with the same bid but bigger *id*, it cancels its bidding message. Although this approach adds some local broadcasting messages, it reduces the chance of having multiple bidding messages which may propagate several hops to the mobile sensor (or proxy). Therefore, the total message complexity can be reduced.

The number of service advertising messages is related to the broadcast radius. If the radius is more than two hops, rebroadcast is necessary. If sensors rebroadcast right after they receive some service advertisement messages, there will be high message complexity. For optimization, we propose that a sensor should wait for some slots to rebroadcast after receiving the advertising messages. During the waiting period, if several advertising messages about different

Notations:

S_i : the service list received by sensor s_i .
 loc_i, loc'_i : physical position and logical position of s_i
 \mathcal{P}_i : the proxy of s_i

- (0) Upon entering Advertising phase:
 - set *timer* to be *advertise_interval*,
 - enter *Bidding phase* upon timeout
 - if** s_i is a mobile sensor without proxy **then**
 - broadcast $service(s_i, loc_i, loc_i, base_price_i, s_i)$
 - if** $s_i = \mathcal{P}_j$ **then**
 - broadcast $service(s_j, loc_j, loc'_j, base_price_j, s_i)$
- (1) Upon receiving $service(s_j, loc_j, loc'_j, base_price_j, \mathcal{P}_j)$:
 - add $(s_j, loc_j, loc'_j, base_price_j, \mathcal{P}_j)$ to S_i
- (2) Upon entering Bidding phase:
 - set *timer* to be *bidding_interval*
 - enter *Logical movement phase* upon timeout
 - calculate bid_i if hole exists
 - choose the closest/cheapest sensor s_j from S_i
 - where $base_price_j < bid_i$
 - send $bidding(s_i, loc'_j, bid_i)$ to \mathcal{P}_j
- (3) Upon receiving $bidding(s_j, loc'_k, bid_j)$:
 - record it if it has the highest bid
- (4) Upon entering Logical movement phase:
 - set *timer* to be *logical_interval*,
 - enter *Hole-exchange phase* upon timeout
 - if** record $bidding(s_k, loc'_i, bid_k)$ **then**
 - send $delegate(s_i, loc_i, loc'_i, bid_k)$ to s_k
 - else if** $s_i = \mathcal{P}_j$ and record $bidding(s_k, loc'_j, bid_k)$
 - send $delegate(s_j, loc_j, loc'_j, bid_k)$ to s_k
 - else if** $s_i = \mathcal{P}_j$
 - if** hole-exchange with s_m is needed
 - send $request(s_i, s_j, loc_j, priority)$ to \mathcal{P}_m
 - else if** it is time for s_j to move **then**
 - send $notice(loc'_j, base_price_j)$ to s_j
- (5) Upon receiving $delegate(s_j, loc_j, loc'_j, base_price_j)$:
 - $\mathcal{P}_j = s_i$; record $loc_j, loc'_j, base_price_j$.
- (6) Upon receiving $notice(loc'_i, base_price_i)$:
 - move to loc'_i and record $base_price_i$
- (7) Upon receiving $request(s_j, s_k, loc_k, priority)$:
 - record it if it has the highest priority
- (8) Upon entering Hole-exchange phase:
 - if** $s_i = \mathcal{P}_m$ and record $request(s_j, s_k, loc_k, priority)$
 - send $confirm(s_m, s_k, loc_m)$ to s_j
 - $\mathcal{P}_k = s_i$; $loc'_k = loc'_m$
 - $base_price_k = base_price_m$
 - record loc_k
- (9) Upon receiving $confirm(s_m, s_k, loc_m)$:
 - $\mathcal{P}_m = s_i$; $loc'_m = loc'_k$; $base_price_m = base_price_k$
 - record loc_m

Figure 8. The proxy-based protocol at sensor

s_i

mobile sensors are received, they should be packed into one single message and hence reduce the message complexity.

4 Performance Evaluations

As far as we know, the basic bidding protocol [17] is the only previous work addressing the deployment problem with a mix of mobile and static sensors. It has been shown to have good performance in coverage. Therefore, two main objectives of our performance evaluation include (1) whether the energy consumption can be much reduced and the load can be more balanced compared with the basic bidding protocol (2) whether the proxy-based protocol can achieve a similar coverage as the basic bidding protocol. In addition, we compare the distance-based approach and the price-based approach. In all the figures showing the evaluation results, we use 'Proxy-price' to represent the price-based approach and 'Proxy-distance' to represent the distance-based approach. We use 'Basic' to represent the basic bidding protocol.

We implemented the proxy-based deployment protocol in ns2 (version 2.1b9a). The total number of sensors is set to 40 and the percentage of mobile sensors is varied from 10% to 50%. The target field is a $50m * 50m$ flat space. The initial placement of sensors follows random distribution. 802.11 is the MAC layer protocol and DSDV is the routing protocol. The physical layer is modeled after the RF MOTE from Berkeley, with 916.5MHZ OOK 5kbps as the bandwidth and 20 meters as the transmission range. Sensing range is 6 meters.

In the next several sections, the simulation results about coverage, moving distance, etc are shown.

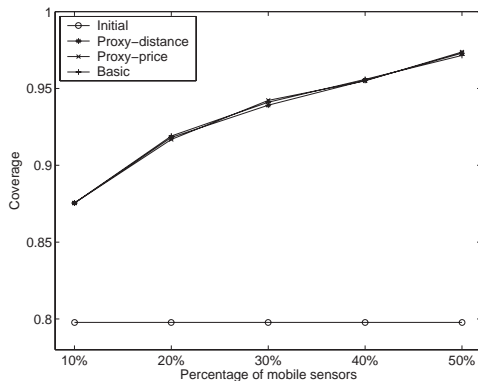


Figure 9. Coverage

4.1 Coverage

Figure 9 shows the coverage reached by our proxy-based protocol and basic bidding protocol under different mobile

percentage. We can see that the coverage obtained by running our protocol is pretty close to the basic bidding protocol.

4.2 Energy Consumption

Energy consumption in the deployment procedure comes from two sources, mechanical movement and electrical communication. We use the moving distance as the metric for the energy consumption in mechanical movement and the message complexity as the metric for the energy consumption in communication.

From Figure 10 and Figure 11, we can see that the proxy-based protocol has a much shorter moving distance (more than 40% difference), while the basic bidding protocol has a much lower message complexity. To consider both factors, we conclude that proxy-based protocol is much more energy efficient because the mechanical movement consumes much more energy than electrical communication. Calculated from the data of the mobile sensor platform [13], it costs a mobile sensor about $30J$ to move one meter, but it only costs the mobile sensor about $0.1J$ to send one message, which means to it costs 300 times the energy to move one meter than to send one message. When 10% mobile sensors are used, the proxy-based protocol saves 10-meter's movement per sensor, with the cost of less than 30 messages. Without any doubt, the proxy-based protocol is more energy efficient.

Generally, the moving distance decreases as the mobile percentage increases. The reason is as follows. The initial coverage holes may have different size, and the mobile sensor needs to heal the largest one, which may be faraway. With less number of mobile sensors, it may take them a long way to reach the largest holes. With more mobile sensors, the smaller holes are also to be healed, and reducing the average distance to the holes. However, in the basic protocol, the number of movement is not reduced monotonically with the increase of mobile percentage due to multiple healing. If several mobile sensors move to heal the same hole, all of them except the winner need to move again. With more mobile sensors, multiple healing happens more frequently, and the sensors move more as the mobile percentage increases.

Between the price-based approach and the distance-based approach, the former one has less movement. The reason is as follows. Each time, static sensor bids the cheapest mobile sensor, and a mobile sensor accepts the highest bid. Once a mobile sensor accepts a bid, it is not likely to be bid somewhere else, which means its logical position is stable before it finally moves. The proxy of the mobile sensor detects whether its distance to the logical position is too long at an early time and can find another mobile sensor to exchange the coverage hole. On the contrary, when static sensors choose the closest mobile sensor to bid, the situation

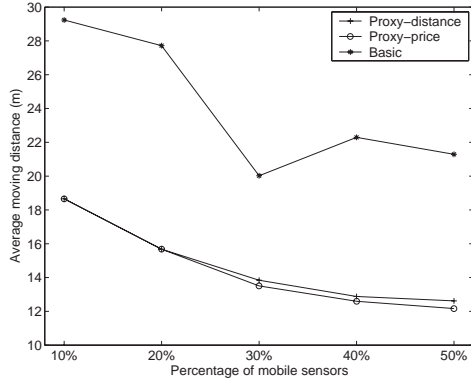


Figure 10. Moving distance

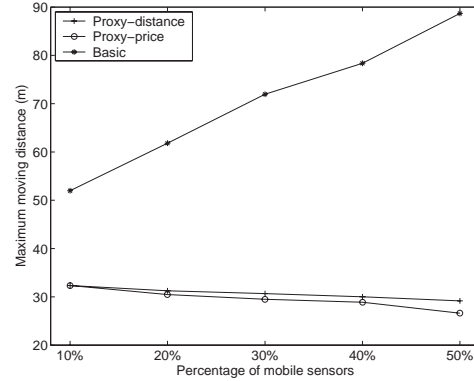


Figure 12. Maximum moving distance among the mobile sensors

shown in Figure 1 is likely to occur. The logical position of some mobile sensor is frequently changed. When the proxy sensors of these mobile sensors detect their possible moving distance is too long, they may not find a suitable sensor to exchange hole since they may move already. Also, in the situation shown in 6, the price-based approach also has less movement.

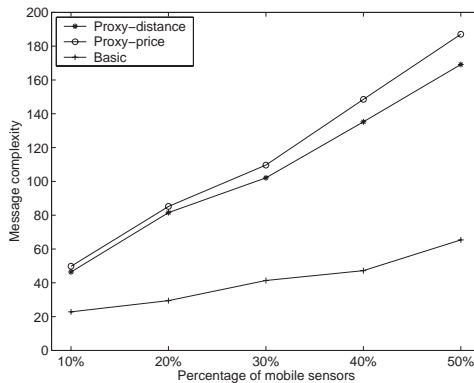


Figure 11. Message complexity

As for the message complexity, the basic bidding protocol performs better because they do not have messages for hole exchange. In addition, in the proxy-based protocol, mobile sensors only move after their final locations have been found and it needs more rounds to stop, while the basic protocol terminates earlier and consequently needs less message. Between the two criteria to choose mobile sensor to bid, the distance-based approach outperforms the price-based approach. This is because the price-based approach needs a relatively higher δ , and consequently needs more rounds to stop and more messages are transmitted. Consider both movement and message complexity, we prefer the price-based approach, since movement is the dominant factor in energy consumption.

4.3 Load Balance

To evaluate whether individual sensors are penalized to move a long distance, we show the maximum moving distance among the mobile sensors in Figure 12. Note that Figure 10 shows the average moving distance. From the figure, we can see the maximum moving distance in the proxy-based protocol is less than twice of the average moving distance, while it is several times more than the average moving distance in the basic bidding protocol. For example, when the mobile percentage is 50%, the maximum moving distance (88.69m) is about four times the average moving distance (21.29m). This result shows that our protocol is much more balanced. Also, the maximum moving distance in the basic bidding protocol is several times more than that in the proxy-based protocol.

Between the price-based approach and the distance-based approach, the price-based approach performs better. The reason has already been explained in the above. The logical position of some sensors are more likely to change in the distance-based approach. When the proxies of these mobile sensors detect their possible moving distance is too long, they may not find a suitable sensor to exchange hole since other sensors may move already. Then these sensors have to move a long way.

In the basic bidding protocol, the maximum moving distance increases with a higher mobile percentage. Multiple healing contributes this phenomenon. When several mobile sensors move to heal the same hole, all except the winner have to reduce the base price to zero and move again. With a higher mobile percentage, there will be more multiple healing.

In our proxy-based protocol, the maximum moving distance is decreased with a higher mobile percentage. This is because with more mobile sensors, the load of each mobile

sensor is lowered. In addition, the proxy sensors can detect the multiple healing, and mobile sensors need not move again for that.

5 Conclusion

In this paper, we addressed the problem of deploying a mix of mobile and static sensors. Based on a proxy-based protocol, mobile sensors can identify the final locations, and then move there directly. Simulation results show that the proxy-based protocol can significantly reduce the physical moving distance, while maintaining similar coverage, when compared to existing schemes. Since mobile sensors need to communicate with other sensors to achieve logical movement, the message complexity is increased. However, we believe this increase is well justified considering that physical movement consumes most of the power.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, pp. 102–114, August 2002.
- [2] F. Aurenhammer, "Voronoi diagrams — a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, pp. 345–406, 1991.
- [3] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K. k. Saluja, "Sensor Deployment Strategy for Target Detection," *First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.
- [4] S. Dhillon, K. Chakrabarty and S. Iyengar, "Sensor Placement for Grid Coverage under Imprecise Detections," *Proceedings of the International Conference on Information Fusion*, 2002.
- [5] W. R. Heinzelman, J. Kulik and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Network," *Fifth ACM/IEEE Annual International Conference on Mobile Computing and Networking (Mobicom'99)*, August 1999.
- [6] A. Howard, M. J. Mataric and G. S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," *Autonomous Robots, Special Issue on Intelligent Embedded Systems*, September 2002.
- [7] A. Howard, M. J. Mataric and G. S. Sukhatme, "Mobile Sensor Networks Deployment Using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem," *the 6th International Symposium on Distributed Autonomous Robotics Systems*, June 2002.
- [8] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication," *ACM SIGMOBILE, Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00)*, August 2000.
- [9] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks," *ACM SIGMOBILE, Seventh Annual International Conference on Mobile Computing and Networking (MobiCOM '01)*, 2001.
- [10] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Network," *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2001.
- [11] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, May 2000.
- [12] G. T. Sibley, M. H. Rahimi and G. S. Sukhatme, "Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks," *2002 IEEE International Conference on Robotics and Automation*, 2002.
- [13] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, "Protocols for Self-Organization of A Wireless Sensor Network," *IEEE Personal Communication*, vol. 7, no. 5, pp. 16–27, October 2000.
- [14] M. Srivastava, R. Muntz and M. Potkonjak, "Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments," *ACM SIGMOBILE, Seventh Annual International Conference on Mobile Computing and Networking (MobiCOM'01)*, 2001.
- [15] S. Tilak, N. B. Abu-Ghazaleh and W. Heinzelman, "Infrastructure Tradeoffs for Sensor Networks," *First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.
- [16] G. Wang, G. Cao and T. La Porta, "A Bidding Protocol for Deploying Mobile Sensors," *The 11th IEEE International Conference on Network Protocols (ICNP)*, November 2003.
- [17] G. Wang, G. Cao and T. La Porta, "Movement-Assisted Sensor Deployment," *The 23rd Conference of the IEEE Communications Society (INFOCOM)*, March 2004.
- [18] Y. Zou and K. Chakrabarty, "Energy-Aware Target Localization in Wireless Sensor Networks," *Proc. IEEE Int. Conf. Pervasive Computing and Communications*, 2003.