Energy Efficient and Seamless Data Collection with Mobile Sinks in Massive Sensor Networks

Taisoo Park, Daeyoung Kim, Seonghun Jang, Seong-eun Yoo, and Yohhan Lee
Real Time Embedded Systems Laboratory, School of Engineering, Information and Communications University
Daejeon, South Korea
taisoo_park@tmax.co.kr, {kimd, exigent, seyoo, yohhanlee}@ icu.ac.kr

Abstract— **Wireless Sensor Networks (WSNs) enable the surveillance and reconnaissance of a particular area with low cost and less manpower. However, the biggest problem against the commercialization of the WSN is the limited lifetime of the battery-operated sensor node. Taking this problem into account, a mobile sink is deployed as a robot, vehicle or portable device to only activate the sensor nodes that are interesting to the sink and leaves other nodes deactivated for. This can considerably extend the lifetime of the sensor nodes compared to existing power management algorithm of using all nodes. However, in this environment, the mobility of the sink raises new issues of energy efficiency and connectivity in communications. To solve these issues, we propose a DRMOS (Dynamic Routing protocol for Mobile Sink) method that includes a designated wake-up-zone to make sensor nodes prepare for an incoming sink. The shape of the wake-up-zone is dynamically changing to reflect the past moving patterns of the sinks. Moreover, we present the extensive simulation results and recommend parameters for practical use of DRMOS from the simulation analysis.**

I. INTRODUCTION

Wireless sensor network technology provides monitoring environment, surveillance of property, and collecting data of massive field at a low cost. Since, the sensor nodes are connected to each other in a wireless ad-hoc manner and powered by batteries, the issue of power management arises. Many research have been conducted to devise an algorithm for extending the lifetime of sensor nodes, however, in real applications [1], the indeterministic lifetime of sensor node is still an ongoing issue. Therefore, by employing mobile sink into a robot, vehicle, or portable device, we can activate reduced number of sensor nodes which are in the region of our interest rather than activating all the sensor nodes in the field. This approach can drastically maximize the lifetime of sensor nodes.

For the expedition of extremely severe area such as Antarctica or a jungle area where it is difficult to redeploy the sensor nodes or change the batteries, efficient management of the lifetime of the sensor nodes is critical. In addition, collecting data nearby the mobile sink is a matter of the highest priority. For example, assume that a person lost his/her way in the New York City. In this case, if he/she can be assisted by the network of sensors, the most critical thing for him/her is figuring out the information of the road where he/she is on at the moment rather than having all the information of the road in United States. Therefore, for those kinds of applications, the mobile sink does not need to activate all the sensor nodes in the field but only need to selectively activate the sensor nodes to reduce unnecessary power consumption.

Two challenging issues in using mobile sinks are: seamless data collection and energy conservation. Since the location of the sink keeps changing, data reports from the sensor nodes can be lost because an existing path can become invalid when the sink moves. To remedy that situation, communication paths, or routing trees should be exhaustively rebuilt. However, frequent updates of the routing tree cause large energy consumption and significant delays for restarting communication. Since the sinks are collecting data only from the nodes inside of a certain range, a protocol should have policies for sleeping and waking up the sensor nodes to save energy.

The research projects [2], [4], [6], [7] have been performed to handle the mobility of the sink in the sensor networks. Their ideas can be divided into three categories: stop-and-update, prediction, and active-multicast.

The first approach is the stop-and-update approach, which requires a sink to stop to update a routing tree. In virtual sink rotation protocol [2], while a sink is moving, a temporary virtual sink is picked among

---

This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2009-(C1090-0902-0047))

This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MOST) (No. R0A-2007-000-10038-0)

sensor nodes to store data reports from other sensors. When the real sink stops and is reconnected to the network, the virtual sink forwards data reports from the sensor nodes. So, data reports are temporarily stored in a virtual sink while a real sink is moving. This approach is problematic because it causes critical delays for real time applications such as surveillance and reconnaissance sensor network.

A prediction based approach does not require a sink to stop because it has algorithms to predict the future location of the sink. Therefore, nodes in the network automatically wake up and sleep by predicting where a sink might be located. In the research [3], a Kalman filter was used with a link RSSI to try to predict movement of a sink in mobile network. However, these prediction mechanisms using the Kalman filter require too heavy computation and storage resources for sensor nodes. In addition, since tools such as GPS and other localization engines can provide frequent measurement updates of location, additional prediction processes between each update of measurement are not meaningful

In the active multicast based approach, a sink or a head of sensor nodes actively notifies where a moving entity will go. As a result, sinks can prepare their sensor nodes to be ready. In [4], they proposed a spatiotemporal multicast protocol that distributes a message to mobile delivery zones. A delivery zone is computed by a sink or a group leader of sensor nodes and is multicasted in order to wake up nodes. The delivery zone moves at a certain distance ahead of the mobile entity. Nodes in the delivery zone rebroadcast the messages to the forwarding zone and wait for the mobile entity to arrive. However, since this research did not consider multi hop data transmission, actual routing protocol in the network is excluded. This protocol is far estranged from real application of ad hoc sensor networks in the field because one hop communication has limitation in communication range. Moreover, they assumed that entities are moving at a fixed velocity. It is highly unlikely that a moving entity would advance with a fixed velocity and unchanging direction.

We propose our protocol, DRMOS with consideration for those aforementioned shortcomings. Our protocol uses dynamic updates of the routing tree without requiring a sink to stop. In addition, it can accurately reflect the past movement pattern of a sink without using expensive prediction algorithms. Moreover, our protocol is a multi-hop based routing protocol which can handle movement that varies in direction and velocity.

DRMOS is a set of protocols that includes algorithms for handling sink mobility by reusing a routing tree and differentiating states of the sensor node. In addition, for the seamless data collection, we employ the concept of a wake-up-zone to make sensor nodes ready for the coming sink. The shape of the wake-up-zone is dynamically changing to reflect past moving patterns of the sinks. We manipulate the actual wake-up-range to reflect the past moving pattern of the mobile sink. With this idea, we present simulation analysis to find the optimal frequency of routing tree updates and the optimal ratio of wake-up-zone range to actual data collection range. Therefore, we can have empirical analysis when we apply our protocols on real application of mobile sink based sensor network.

The remainder of this paper is organized as follows. Overview of our protocol is covered in Section II. Detailed energy efficient mobile sink based routing protocol is explained in Section III. Simulation parameters and environmental variables are presented in Section IV. The analysis of simulation results are discussed in Section V. Finally, section VI concludes with a summary of this paper.

II. DYNAMIC ROUTING PROTOCOL FOR MOBILE SINK

Several terms, and basic mathematics are employed to deal with problems of the mobile sink based sensor networks. We define three different states (black, gray, white), and three different zones (root-zone, collection-zone, and wake-up-zone) to simplify the explanation of the protocol.

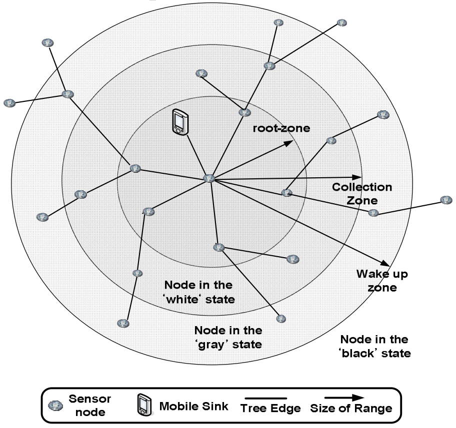

Figure 1. Node state and each range

Initially all the nodes are deactivated and powered off. On receiving a wake-up-signal from the sink, the nodes change their state as 'black'. Nodes in the 'black' state have only their RF module turned on and wait idly to receive a signal from the sink. In contrast, nodes in the 'gray' state have their RF module and CPU turned on, and then connect itself to the network. Gray nodes, however, do not actively collect data. 'White' nodes are the only nodes that actively collect data as well as report that data to the sink.

A 'root-zone' is the region which is inside the root's RF range. The root is the closest node to the sink. A 'collection-zone' is the area from which a sink wants to collect data. It is circular shaped and its radius is determined by the user. Nodes in the 'collection-zone' change their status to 'white' and report their sensory data to a sink through its parent. A 'wake-up-zone' is the largest region, and its shape and size can be changed by the sink. If nodes are determined to be in this zone, they go into the 'gray' state. A 'wake-up-zone' is for waking up nodes to be prepared to send sensory data. Since it takes time to be connected to the network, nodes should be woken up and prepared before the sink comes near.

III. ROUTING PROTOCOL DESCRIPTION

A. Initialization

After receiving wake-up-signal from the sink, all nodes have turned on their radio and set their state as 'black'. Once a sink activates its network, it first finds a root node that is closest to the sink. Then the sink initiates the DRMOS session by unicasting a TreeBuild message to the root. The TreeBuild messages are then broadcasted in the network. The node that received the TreeBuild message sets the source node of the TreeBuild message as the parent node. Then, it responds to the parent node with a Join message. Upon receiving the Join message, the parent node sends a JoinAccept message that includes the address of the child node. This is the three way handshakes to build a routing tree.

The TreeBuild message includes the designated radius of both the collection-zone and wake-up-zone, and the location of the sink. Upon receiving the TreeBuild message, the root broadcasts it to the network. At the initialization phase, both the collection-zone and wake-up-zone are circular shaped. After receiving the TreeBuild message, nodes should report their location to the sink. Then by comparing the distance from the sink and the radius designations for each zone, nodes can determine which zone they are located in.

If nodes are located inside the collection zone, they rebroadcast the TreeBuild message to other nodes to build a tree. Then they change their status to 'white' and their CPUs and sensors are activated to begin sensing and reporting to the sink. Nodes which are located inside the wake-up-zone and outside the collection-zone also rebroadcast the TreeBuild message and change their status to 'gray'. Their network paths are built, but the sensors do not start to sense yet. Nodes which are located outside the wake-up-zone do nothing but report their location to the sink. As a result of sending the TreeBuild message, the sink can change the status of nodes depending on where they are located and can receive sensory data from its collection range.

B. Uniform Velocity Movement

In this phase, the sink enters the field with uniform velocity. After broadcasting the TreeBuild messages in the initialization phase, the sink receives location information from sensor nodes. With that information, the sink builds a convex hull which becomes the parameter of the new wake-up-zone. The convex hull [4] of a set Q of points is the smallest convex polygon P for which each point in Q is either on the boundary of P or in its interior. With Graham's scan, it requires time complexity of O(n*log(n)) for a sink to build convex hull with 'n' nodes and requires time complexity of O(m) for a node to check whether a node itself is inside the convex hull or not, with 'm' nodes in the convex hull.

As the sink moves from place to place, the new wake-up-zone, will shift as much as the sink has moved. In the uniform velocity movement, the moving sink continues to issue the TreeBuild message, including the radius of collection-zone, location of the sink, and the location of nodes in the shifting convex hull. When the sink moves with uniform velocity, a circular shaped wake-up-zone is enough to handle the movement of the sink. However, if the velocity of the sink is changing, the prediction of the sink's movement is required to ensure that sensor nodes are awakened at the required times. Therefore, a change in shape of the wake-up-zone is required to handle changes to the sink's movement vector. The wake-up-zone is a practical representation and cannot be defined as a simple circle or ellipse. Therefore, the formulizing patterns associated with circles and ellipses are inapplicable. Instead, we use the convex hull to monitor the dimensions of the wake-up-zone.

C. Non-Uniform Velocity Movement

Intuitively, if the velocity of the sink keeps increasing in a certain direction, it is better to wake up nodes further ahead in the direction the sink is moving. Along the same lines, if a sink is changing direction from north to east, nodes to the east of the sink should be woken up and prepared because there is a larger possibility of the sink going east than west or south. Therefore, with DRMOS, the wake-up-zone is constantly manipulated, and nodes which have more possibility of being in the collection-zone are prepared in advance. Therefore, their network paths are built, whereas nodes which have less possibility of being in the collection-zone in the future can stay idle

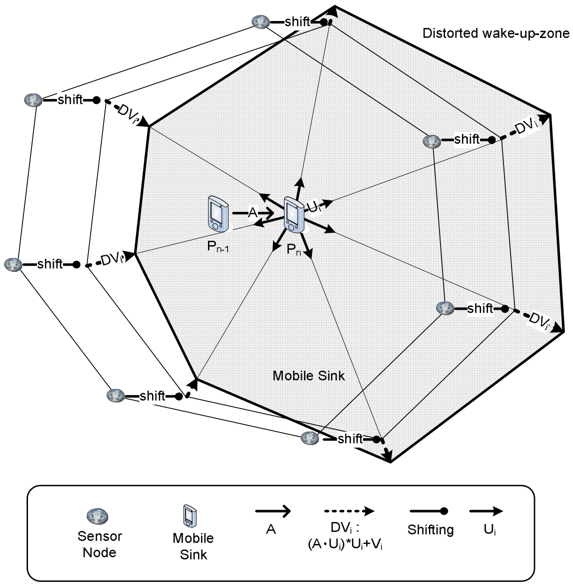

Figure 2. Distortion of the wake-up-zone when a mobile sink moves from ‘P_{n-1}’ to ‘P_n’ with the change of velocity as A

One more consideration regarding distortion is that the wake-up-zone should not invade the collection-zone. Although the wake-up-zone keeps shrinking, distortion is limited by the collection zone.

Figure 2 represents distortion of the wake-up-zone when a mobile sink moves from ‘P_{n-1}’ to ‘P_n’. Also ‘S_n’ and ‘S_{n-1}’ represent the current velocity and the previous velocity, respectively.

In the distortion procedure, a sink first computes change to its velocity, or acceleration vector $\vec{A}$.

$$\vec{A} = \vec{S_n} - \overrightarrow{S_{n-1}} \quad . \qquad (1)$$

For the nodes in the convex hull, define $\vec{V_i}$ which starts from P_n and ends at the current position of node i, or N_i. Then define $\vec{U_i}$ by normalizing $\vec{V_i}$

$$\vec{V_i} = \overrightarrow{P_n N_i} \quad . \qquad (2)$$

$$\vec{U_i} = \frac{\vec{V_i}}{|\vec{V_i}|} \quad . \qquad (3)$$

Then, with the inner product of $\vec{U_i}$ and $\vec{A}$, the sink can give different weights on distortion vector $\overrightarrow{DV_i}$ depending on the phase between acceleration vector $\vec{A}$ and $\vec{V_i}$.

$$\overrightarrow{DV_i} = (\vec{U_i} \bullet \vec{A}) \cdot \vec{U_i} \quad . \qquad (4)$$

The $\overrightarrow{DV_i}$ is multiplied by a distortion coefficient ‘*k’* which determines the scale of $\overrightarrow{DV_i}$. Then the sink distorts the convex hull by adding k ($\overrightarrow{DV_i}$) to $\vec{V_i}$.

$$\vec{V_i} = \vec{V_i} + k(\overrightarrow{DV_i}) \quad . \qquad (5)$$

In the result, the end points of new $\vec{V_i}$ will form the new convex hull. As in Figure 3, after distortion, the region which $\vec{A}$ is heading will be expanded and the opposite region will contract.

Detailed procedures for distorting the convex hull are also summarized in Table 1.

TABLE I. DISTORTION ALGORITHM

```
Distortion( node[]  convexhull)
Input     node[]  convexhull
Output  node[]  newConvexhull
Set $P_n$ as current position of sink
Shift convex hull as much as $P_n - P_{n-1}$
Set a vector $\vec{S_n}$ as current velocity of sink
Set a vector $\vec{A}$ as $\vec{S_n}$ - $\overrightarrow{S_{n-1}}$
For $i$ = 1 to number of node in convex hull
    Define $\vec{V_i}$ which starts from $P_n$ and ends at node i
    Define $\vec{U_i}$ by normalizing $\vec{V_i}$
    Define $\overrightarrow{DV_i}$ by inner product of $\vec{U_i}$ and $\vec{A}$ to $\vec{U_i}$
    Add $\overrightarrow{DV_i}$ to $\vec{V_i}$
    Include $V_i$ in newConvexHull
End for
Set $P_{n-1}$ as $P_n$
Set $\overrightarrow{S_{n-1}}$ as $\vec{S_n}$
Return newConvexHull
```

Initially P_{n-1} and P_n are set as initial position of sink. Distortion is performed by a sink

D. Routing Tree Update

Issuing the TreeBuild message frequently guarantees seamless communication between a sink and node. However, it causes the problems of mass traffic in the network and exhaustive energy consumption. The reason for this is that building a tree requires a sink to broadcast the TreeBuild messages, and for every TreeBuild message, a Join and JoinAccept message follows to assign an address to the sensor nodes.

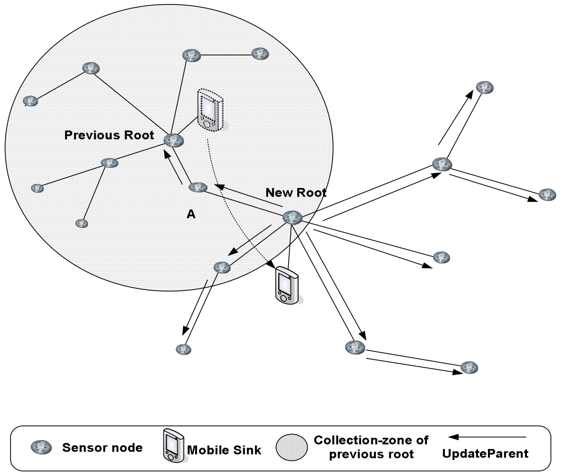

Figure 3. Avoid frequent rebroadcasting of TreeBuild packet when a newly discovered root is in the collection zone range which was determined by the previous TreeBuild packet.

To minimize the number of packet transmissions without degrading connectivity, DRMOS reuses an existing tree unless a sink moves out of the collection-zone that was built by the previous TreeBuild message. However, to reuse the previous routing tree, the parent-child relationship in that routing tree should be updated. For example, in Figure 3, node ‘A’ was reporting data to its previous root. After a sink moves to a different place and finds a new root, node ‘A’ should report data to the new root. Therefore its reporting path should be updated when a new root is determined by the sink. This can be done by sending an UpdateParent message to connected neighbors. Since it does not require broadcasting and responding, this process can reduce the packet transmissions for building the routing tree by one-third as well as reducing the time for constructing a routing tree and the battery consumption of nodes. Besides updating the parent-child relationship, the UpdateParent message also contains the dimensions of the collection-zone and a list of the nodes in the convex hull. This information is used to change the status of nodes.

E. Node Sleep and Wake up

Since a user is only interested in nodes inside the collection-zone, nodes which are located outside of wake-up-zone can go into ‘black’ state to save battery power. At the same time, nodes which newly come into the collection-zone will go into ‘white’ state and report sensory data to a sink. Even though a node used to be in the wake-up-zone or collection-zone, if a sink moves away and a node is no longer inside of the wake-up-zone, then that node will change its state to black. Then another a designated period time, have its power turned off. These dynamic joining and leaving processes are accomplished by message exchanging between a sink and nodes.

The joining process of a new node is done when the sink sends a TreeBuild or UpdateParent message. The joining process was covered in the previous sections. As shown in Figure. 4, the leaving process is done when the sink sends a Leave message. If the sink moves out of the RF range of a root, the root can not receive the ACK of its data from the sink. If there is no ACK for a certain period of time, the root sends a Leave message to nodes in the tree. Nodes which receive the Leave message from the root continue sending data to the root for a certain period of time then go to sleep. However within that period of time, if a node receives a TreeBuild or UpdateParent message from the sink, a node disables its timer and continues sending data. In addition, stored data in the previous root are forwarded to the sink through a new root.

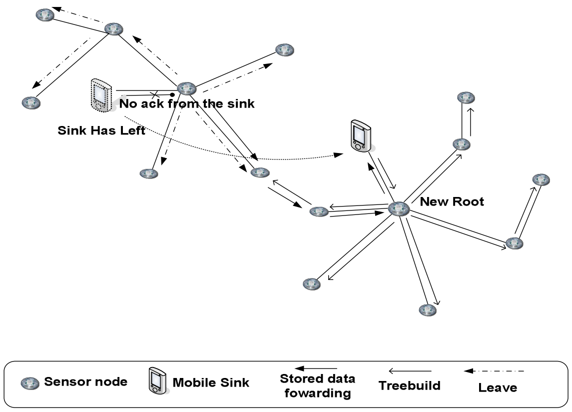

Figure 4. Sleeping procedure of nodes

IV. SIMULATION

To apply our protocol in real applications, we simulated several scenarios on the ns-2 network simulator. Initially all nodes were deployed in a grid-like pattern.

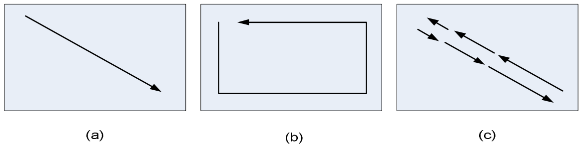
(a) (b) (c)

Figure 5. Three different models to represent the mobility of a sink: (a) Cruise; (b) Patrol; (c) Booster.

We then created three different models to represent the mobility of a sink, and named them as ‘Cruise’ (uniform velocity and unchanging direction), ‘Patrol’ (uniform velocity and changing direction, with the changes in direction forming a rectangular path), and 'Booster' (changing velocity and changing direction). We also varied the initial radius of the wake-up-zone, from the same size of the collection-zone, which means no wake-up-zone like [2], [3], to three times the initial size of the collection-zone. To test the performance of our protocol, we undertook the process both with distortion of the wake-up-zone and without distortion. Also, for update of a routing tree, we use three different update policies. The first policy is that a sink updates the routing tree whenever it goes out of the RF range of the root. The second policy is that the sink updates the routing tree only when it moves out of collection zone. Therefore if the sink is inside the collection-zone, only the relationship between a parent and child is updated. The third policy is that the sink updates the routing tree only when it moves out of wake-up-zone. If the sink is inside the wake-up-zone, only the relationship between a parent and child is updated. In Table 2, these parameters are summarized.

Under the simulation environments, we define three metrics that demonstrate the DRMOS protocol is an improved method for handling the mobility of a sink in a Sensor network.

The first metric is the number of data packets received at a sink. Using the same movement model as a basis for comparison, a larger number of data packets received by a sink means better performance of a protocol in terms of seamless data collection. The second metric is the amount of traffic. This is primarily affected by the number of broadcasts sent to update a routing tree. The third metric is the average energy consumption[1] of nodes. We computed the nodal energy consumption for sending and receiving messages, and repeated those computations for nodes in each of the three different states(white, gray, black). Therefore we can simulate how efficiently DRMOS lets nodes wake up and sleep according to the sink's movement.

TABLE II. PARAMETERS OF SIMULATION

Parameter	Value	Remark
Channel	WirelessChannel	
Propagation Model	TwoRayGround	
Phy	WirelessPhy	
Mac	802.11	
Dimension	5200 * 5200	Square meter
Simulation time	300	Second
Number of nodes	33 * 33	grid
Mobility model	Cruise, Patrol, Booster	Differentiating speed and direction of a moving sink
Initial wake-up-zone radius	1,1.5,2,2.5,3.0 times of collection-zone radius	Larger wake-up zone results in increase of energy consumption and decrease of message loss
Distortion Coefficient	0,3,6,9,12,15,18, 21,24,27, 30	Distortion is related to the handling acceleration of a moving sink

V. RESULT ANALYSIS

Simulations were conducted as described in the previous section. In this section the corresponding results are shown and analyzed.

A. Varying the size of the wake-up-zone

In the first scenario, we varied the initial radius of the wake-up-zone from the same size of the collection-zone to three times the initial size of the collection-zone. The same size of wake-up-zone and collection-zone means the sink does not wake up the sensor nodes beforehand. Since the sink should stop and rebroadcast packets to build a new routing tree, we can think of this case as stop-and-update approach[2], that was mentioned as related work. With the variation of the size of wake-up-zone, we computed the average energy consumption of the nodes and counted the number of packets received at the sink for the three different mobility models. In this first scenario, the distortion coefficient '*k*' was set as 1, and we did not use the UpdateParent packet, but only used the TreeBuild packet to update the a routing tree. In this scenario, with the distortion coefficient of 1 means no distortion, or no reflection of acceleration. Therefore, we can have an idea of the stop-and-update approach[2] and active multicast approach[4] with distortion coefficient 1.

As shown in Figure 6, the overall average energy consumption of all three models increases as the size of the wake-up-zone increases. The reason for this is that the larger wake-up-zone causes more nodes to have their CPUs and sensors turned on.

---

[1] For computing energy consumption of nodes, we quote data sheets of the CC2420 [8] and the atmega128 [9]. Energy consumption for a packet transmission and receiving as 17.4mA and 18.8mA, respectively, and for each state of black, gray, and white as 18.8mA, 26.8mA, and 46.8mA respectively.

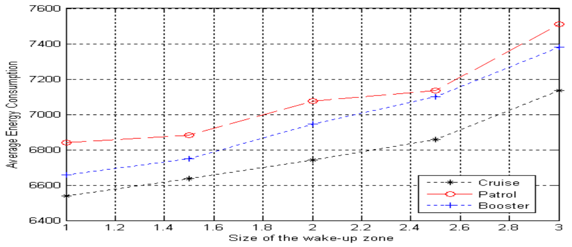

Figure 6. Average energy consumption of the node versus the size of the wake-up-zone compare to the size of the collection-zone.

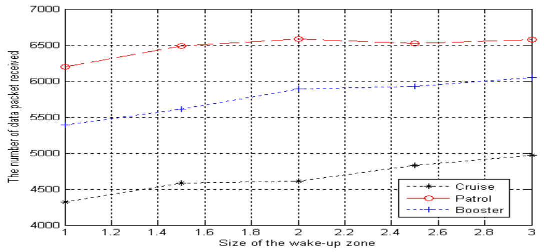

Figure 7. The number of packets received at the sink versus the size of the wake-up-zone compare to the size of the collection-zone.

Figure 7 represents the number of data report packets received at the sink versus the size of the wake-up-zone (again the size of the wake-up-zone is defined relative to the size of the initial collection-zone). In the same movement model, a larger number of data packets received indicates that the nodes were woken up timely enough to report their data to the sink. With a larger wake-up-zone, the sink can activate more nodes in advance. Thus, when the size of the wake-up-zone is increased, the number of data packets received at the sink should also increases. However, the number of packets does not continue to increase after the size of the wake-up-zone reaches 2.5 times the size of the initial collection-zone. This is because packet loss happens due to collisions from mass traffic. Therefore, for all three mobility models, we can conclude that the size of the wake-up-zone should be 2.5 times that of the collection-zone. This size provides the most efficient performance of the system with regard to energy consumption of the nodes, traffic, and seamless data collection.

B. Varying the distortion coefficient

In this scenario, we varied the distortion coefficient '*k*' in section III, from 3 to 30. The larger number of the distortion coefficient means large degree in reflection of the acceleration. We fix the size of the wake-up-zone as 2.5 times the size of the collection zone. Then, for each movement model, we simulated the number of data packets received.

As in Figure 8, in the 'cruise' model, because the movement is initially accelerated, a larger distortion coefficient leads to a higher number of data packets received. However, since the sink only accelerates for a short time, and then travels at a constant velocity, performance does not continue to improve beyond a distortion coefficient of 21, and in fact decreases slightly.

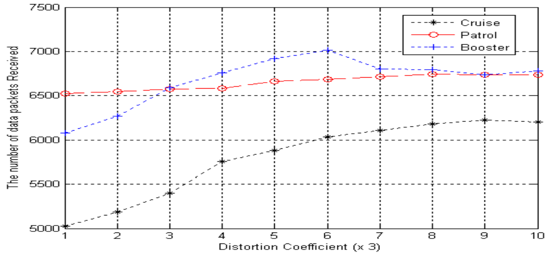

Figure 8. The number of data packets received at the sink versus the distortion coefficient. The size of the wake-up-zone is 2.5 times of the size of collection-zone.

In case of the 'patrol' model, performance is not significantly affected by the distortion coefficient because most of the time it moves inside the wake-up-zone. The number of data packets received is increased slightly until the distortion coefficient reaches 27, and remains constant after that point.

In case of the 'booster' model, velocity is constantly changing. As a result, performance is significantly increased by increasing distortion until the coefficient reaches 18. However, distortion beyond this point results in a small decrease in the number of data report packets because the sink changes its direction at the corner.

Therefore, for the 'cruise', 'patrol' and 'booster' movement models, we can conclude that the appropriate distortion coefficients are 24, 27 and 18, respectively.

C. The update policy of a routing tree

In this section we simulated three different policies for updating a routing tree. In the first policy, we only used the TreeBuild packet to update the routing tree when the sink moved out of the RF range of the root. In the second policy, if the status of a new root found by the sink was white, we utilized the UpdateParent packet (as opposed to the TreeBuild packet). For the third policy we use the UpdateParent packet only if the status of a new root was black. Therefore, the routing tree was rebuilt only when the sink moved out of the wake-up-zone that was built by the previous TreeBuild packet.

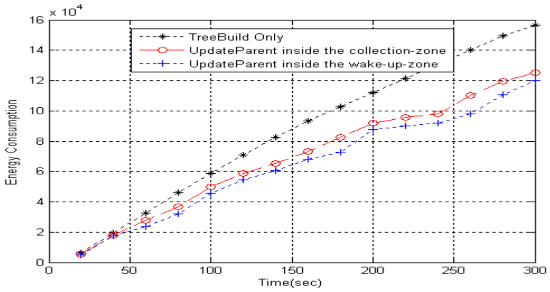

Figure 9. The energy consumption of nodes(mW), for three different policies of updating a routing tree.

Figure 9 and Figure 10 show the results of each policy in updating the routing tree. The traffic and the number of data packets from sensor nodes increase as simulation time passes. In the case of using only the TreeBuild packet, the traffic is significantly larger. However, the second policy shows great improvement of the traffic with minimal loss of data report packets. The third policy, although it results in the lowest nodal energy consumption, causes notable loss of data report packets when the acceleration of the sink is high. Therefore, in this scenario we can conclude that when the sink does not move out of the collection-zone, reusing the previous routing tree is preferable to rebuilding a whole tree.

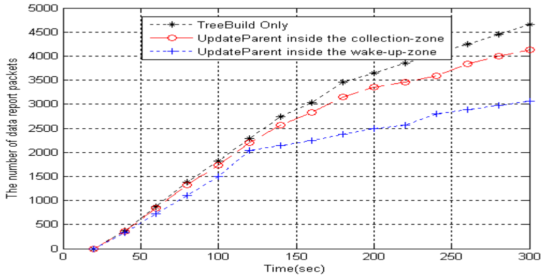

Figure 10. The number of data report packets received, for three different policies of updating a routing tree.

VI. CONCLUSION

When using mobile sink based sensor networks, the mobility of the sink brings out several issues that must be addressed. In DRMOS, we distorted the wake-up-zone to wake up sensor nodes in advance, and used the UpdateParent packet instead of the TreeBuild Packet to reduce unnecessary traffic from frequent updates of the routing tree. In the simulations, we proved that distortion of the wake-up-zone reduces the loss of data report packets from the sensor nodes. Moreover, use of the UpdateParent packet can significantly reduce energy consumption of the nodes. In addition, we determined the ideal size of the wake-up-zone and the distortion coefficients that provide optimal performance.

DRMOS is designed based on totally new schemes in order to handle mobile sink with dynamic and autonomous network configuration. Unlike other related works, the network model of DRMOS includes multi hop communications. In addition, in the movement model of the sink, by taking acceleration into account, degree of the simulation becomes complex. Although we can not exactly compare DRMOS with other related works due to those discrepancies in simulation model and network model, we can have idea on superiority of DRMOS by practicing synthetic simulations

REFERENCES

1. J. Sung, S. Ahn, T. Park, S. Jang, D. Yun, J. Kang, S. Yoo, P. Chong, D. Kim, “Wireless Sensor Networks for Cultural Property Protection”, IEEE 22nd International Conference on Advanced Information Networking and Applications (AINA) Workshops, 2008
2. Lynn Choi, Kwangseok Choi, Jungsun Kim, and Byungjoon Park, “Virtual Sink Rotation: Low-Energy Scalable Routing Protocol for Ubiquitous Sensor Networks”, *EUC Workshops 2005*, LNCS 3823, pp. 1128 – 1137
3. Pubudu N. Pathirana, Andrey V. Savkin, Sanjay Jha, “Mobility Modeling and Trajectory Prediction for cellular Networks with Mobile Base Station”, Proc. of *MobiHoc’03*, Annapolis, Maryland, USA
4. Qingfeng Huang, Chenyang Lu, and Gruia-Catalin Roman, "Spatiotemporal Multicast in Sensor Networks," First ACM Conference on Embedded Networked Sensor Systems (*SenSys'03*), Los Angeles, CA (2003)
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, The MIT Press; Second edition, pp. 947, (2001)
6. Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, Lixia Zhang, “A Two Tier Data Dissemination Model for Largescale Wireless Sensor Networks” Proc. of MOBICOM’02, Atlanta, Georgia, USA (2002)
7. H. Kim, Tarek F. Abdelzaher, W.Kwon “Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks”, *SenSys’03*, Los Angeles, California, USA (2003)
8. *Chipcon product from Texas Instrument,* Data sheet, 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, October, 2005
9. *Atmel Corporation*, Data sheet, 8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash, Atemga128, October, 2006