Minimum-Energy Data Dissemination in Coordination-based Sensor Networks

Hung Le Xuan, Dae Hong Seo, Sungyoung Lee, and Young-Koo Lee
Department of Computer Engineering, Kyung Hee University
{lxhung, comafire}@oslab.khu.ac.kr, { sylee , yklee }@khu.ac.kr

Abstract

Recent years, many efficient data dissemination protocols for mobile sinks in large scale sensor networks are currently under developed by researchers. In this paper we propose CODE, a COordination-based data Dissemination protocol for wireless sEnsor networks. CODE relies on grid structure and GAF protocol to achieve better energy consumption by establishing an efficient data dissemination path and turning off unnecessary nodes. Our simulation results show that CODE achieves more energy efficient and longer networks life time compared with other approaches while still handling efficient data delivery to mobile sinks [1]

1. Introduction

A sensor network is randomly deployed by hundreds or thousands of unattended and untethered sensor nodes in an area of interest. These networking sensors collaborate among themselves to collect, process, analyze and disseminate data. Limitations of sensors in terms of memory, energy, and computation capacities give rise to many research issues in the wireless sensor networks. In this paper, we propose A Coordination-based Data Dissemination protocol (CODE), addresses mobile sinks. We are motivated by the fact that handling mobile sinks is challenge of large-scale sensor network research. Though many researches have been published to provide efficient data dissemination protocols to mobile [2,4,5,8,9], they have proposed how to minimize energy consumed for network communication, without considering idle energy consumption. However, energy consumed for nodes while idling can not be ignored [10,11]. Y.Xu *et al* [10] and M.Stemm *et al* [11] suggested that energy optimizations must turn off the radio to reduce number of packets transmitted and to conserve energy. In CODE, we take into account of energy for both communication and idle. CODE is based on grid structure and coordination protocol GAF [1] to provide an energy efficient data dissemination path to mobile sinks for coordinating sensor networks.

[1] This research work has been partially supported by Korea Ministry of Information and Communication ITRC Program joint with SunMoon University

The rest of this paper is organized as follow. Section 2 mentions about related work. In section 3, we introduce our proposed protocol. We analyze the communication overhead of CODE in section 4. The performance evaluation is presented in section 5. Finally, we conclude the paper in section 6.

2. Related Work

Many sensor network protocols have been developed in recent years. [2,5,6,7,8,9]. One of the earliest work, SPIN [5], addresses efficient dissemination of an individual sensor's observation to all the sensors in the network. SPIN uses meta-data negotiations to eliminate the transmission of redundant data. Directed Diffusion [2] is similar in that it takes the data-centric naming approach to enable in-network data aggregation. In Directed Diffusion, all nodes are application-aware. This enables diffusion to achieve energy saving by selecting empirically good paths and by caching and processing data in-network. GRAB [9] targets at robust data delivery in an extremely large sensor network made of highly unreliable nodes. It uses a forwarding mesh instead of a single path, where the mesh's width can be adjusted on the fly for each data packet. GEAR [6] uses energy aware neighbor selection to route a packet towards the target region. It uses Recursive Geographic Forwarding or Restricted Flooding algorithm to disseminate the packet inside the destination regions.

While such previous work only addresses the issue of delivering data to stationary sinks, other work such as TTDD [4], SAFE [8] and SEAD [7] target at efficient data dissemination to mobile sinks. TTDD exploits local flooding within a local cell of a grid which sources build proactively. Each source disseminates data along the nodes on the grid line to the sink. However, it does not optimize the path from the source to the sinks. When a source communicated with a sink, the restriction of grid structure may multiply the length of a strait-line path by $\sqrt{2}$. Also, TTDD frequently renews the entire path to the sinks. It therefore increases energy consumption and the connection loss ratio. SAFE uses flooding that is geographically limited to forward the query to nodes along the direction of the source. SAFE uses geographically limited flooding to find the gate connecting itself to the tree. Considering the large number of nodes in a sensor networks, the network-wide flooding may introduce considerable traffic. Another data dissemination protocol, SEAD, considers the distance and the packet traffic rate among nodes to create near-optimal dissemination trees. SEAD strikes a balance between end-to-end delay and power consumption that favors power savings over delay minimization. SEAD is therefore only useful for applications with less strict delay requirements.

3. Proposed Protocol

In CODE, we rely on the assumptions that all sensor nodes are stationary and aware of their residual energy and geographical location. Once a stimulus appears, the sensors surrounding it collectively process signal and one of them becomes a source to generate data report. Sink and source are not supposed to know any *a-priori* knowledge of potential position of each other. CODE has three major phases: *data announcement*, *query transfer* and *data dissemination*.

3.1. Grid Formation

We assume that we have partitioned the network plane in virtual *MxN* grids. Each grid ID which has a typed [CX.CY] is assigned based on the coordinate (x, y) as:

$$CX = \left\lfloor \frac{x}{r} \right\rfloor, \ CY = \left\lfloor \frac{y}{r} \right\rfloor \tag{1}$$

where r is the grid size.

On the other hand, each node is supposed to maintain a neighboring table by using the simple HELLO protocol at the beginning of network life.

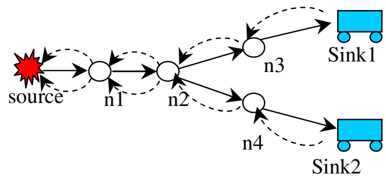
3.2. CODE Algorithms

a) Data Announcement

When a stimulus is detected, a source propagates a *data-announcement* message to all coordinators using simply flooding mechanism. Every coordinator stores a few piece of information for the data dissemination path discovery, including the information of the stimulus and the source location. In this approach, the source location does not mean the precise location of the source, but its grid ID. Since the coordinator role might be changed every time, the grid ID is the best solution for nodes to know the target it should relay the query to. To avoid keeping data-announcement message at each coordinator indefinitely, a source includes a *timeout* parameter in *data-announcement* message. If this timeout expires and a coordinator does not receive any further *data-announcement* message, it clears the information of the stimulus and the target's location to release the cache.

b) Query Transfer

Every node is supposed to maintain a *Query INformation Table* (QINT) in its cache. Each entry is identified by a tuple of (*query, sink, uplink*) (*sink* is the node which originally sends the *query*; *uplink* is the last hop from which the node receives the query). We define that two entries in QINT are identical if all their corresponding elements are identical. For example in Fig.1, node n1 and node n2 receive a query from sink1 and sink2, therefore they maintain tables *QINT n1* and *QINT n2*.



QINT n1

query	sink	uplink
A	sink1	n2
A	sink2	n2

QINT n2

query	sink	uplink
A	sink1	n3
A	sink2	n4

Figure 1. Query transfer and *QINT* table

Find_Next_Grid(NODE, packet* p){

If (NODE is Source)

NODE.**send_data();**

Else{

$\Delta_{CX} = \texttt{p->src\_addr.CX - NODE.CX};$

$\Delta_{CY} = \texttt{p->src\_addr.CY - NODE.CY};$

$\delta_{CX} = (\Delta_{CX} == 0)\,?\,0 : \dfrac{\Delta_{CX}}{|\Delta_{CX}|};$

$\delta_{CY} = (\Delta_{CY} == 0)\,?\,0 : \dfrac{\Delta_{CY}}{|\Delta_{CY}|};$

$NextGrid.CX = NODE.CX + \delta_{CX};$

$NextGrid.CY = NODE.CY + \delta_{CY};$

If (lookup_neighbor_table(*NextGrid***)** = = TRUE)

return *NextGrid*;

Else

find_round_path(); }

Figure 2. Algorithm to find next grid ID

Receiving a query from an uplink node, a node first checks if the query exists in its QINT. If so, the node simply discards the query. Otherwise, it caches the query in the QINT. Then, based on target's location stored in each coordinator, it computes the ID of next grid to forward the query. This algorithm is described in Fig.2. Each node is supposed to maintain a *one-hop-neighbor table*. If a node can not find the next grid's coordinator in this table, it considers that grid as a void grid. For example in Fig.3, the sink1 sends query to the source *src* along the path [4.1], [3.2], [2.3], [1.3], [0.3]. However, with the sink2, the grid [3.0]'s coordinator can not find grid [2.1]'s neighbor (due to void grid) and grid [3.1]'s coordinator also can not find grid [2.2]'s neighbor (due to unreachable node) in its one-hop-neighbor table. Therefore, it finds the round path as [3.1], [3.2], [2.3], [1.3], [0.3]. A data dissemination path is discovered by maintaining a QINT at each intermediate node. A query from a sink is re-transmitted when the sink moves to another grid (section 3.3).

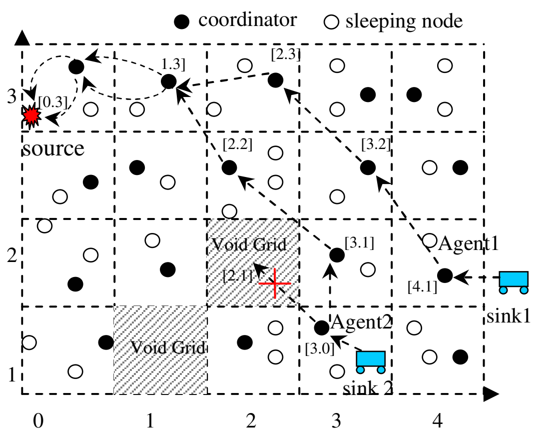


Figure 3. Multi-hop routing through coordinators

c) Data Dissemination

A source starts generating and transmits data to a sink as it receives a query. Receiving data from another node, a node on the dissemination path (including the source) first checks its QINT if the data matches to any query and to which uplinks it has to forward. If it finds that the data matches several queries but with the same uplink node, it forwards only one copy of data. Doing this reduces considerable amount of data transmitted throughout the sensor network. For example in Fig.1, node n1 receives the same query A of sink1 and sink2 from the same uplink node (n2). Therefore, when n1 receives data, it sends only one copy of data to n2. Node n2 also receives the same query A of sink 1 and sink 2 but from different uplink nodes (n3, n4). Thus, it must send two copies of data to n3 and n4. Likewise, the data is relayed finally to the sinks.

3.3. Handling Sink Mobility

CODE is designed for mobile sinks. In this section, we describe how a sink keeps continuously receiving updated data from a source while it moves around within the sensor field.

Periodically, a sink checks its current location to know which grid it is locating. The grid ID is computed by the formula (1). If it is still in the same grid of the last check, the sink does nothing. Otherwise, it first sends a *cache-removal* message to its old Agent. The *cache-removal* message contains the query's information, the sink's identification and the target's location. The old Agent is in charge of forwarding the message along the old dissemination path as depicted in Fig.4. Receiving a *cache-removal* message, a node checks its QINT and removes the matched query. When this message reaches the source, the whole dissemination path is cleared out, i.e. each intermediate node on the path no longer maintains that query in its cache. Consequently, the source stops sending data to the sink along this dissemination path. After old dissemination path is removed, the sink re-sends a query to the target location. A new dissemination path is established as described in section 3.2b. By doing this, the number of queries which is needed to be re-sent is reduced significantly compared with other approaches. Hence, collision and energy consumption is reduced. Also, the number of loss data packet is decreased. In case the sink moves into a void grid, it selects the closest coordinator to act as its Agent.

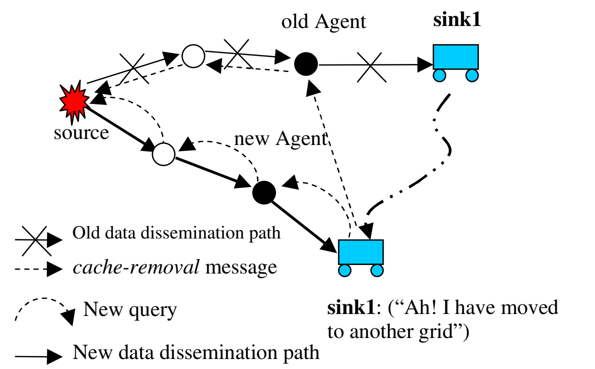


Figure 4. Handling sink mobility

4. Communication Overhead Analysis

We analyze the communication overhead of CODE to show the benefit of using grid structure and GAF-based approach. We also compare CODE with TTDD and the sink-oriented data dissemination approaches (henceforth called *SODD*) such as Directed Diffusion[2], GRAB[9]. Since query aggregation and data aggregation techniques are adopted in CODE, TTDD and SODD as well, we do not consider these aggregations when we compare communication overhead. We suppose a similar model and notations in [4].

Let's consider N nodes deployed uniformly in a sensor field of A square meters. Each cell has a size $R/\sqrt{5}$ as computed in [1], where R is nominal radio range of sensor nodes. This cell size guarantees connectivity of the whole sensor field. There are k sinks moving with max speed v, while receiving d data packets from a source in a time period of T. Each data packet is a unit size long and the others have a size l. To model sink mobility, we assume each sink traverses m cells ($m \le 1 + \frac{vT}{R/\sqrt{5}}$). Consequently, each sink has to send a *cache-removal* message and a *query* m times, and receives $\frac{d}{m}$ data packets between two consecutive location updates.

There are c^2 cells in the whole sensor field (where $c = \left\lceil \sqrt{A}/(R\sqrt{5}) \right\rceil$; $\lceil x \rceil$ is the smallest number larger than x). We analyze the communication overhead in the worst-case, i.e. the source and the sink are furthest away from each other.

For a query from a source to reach a sink, it traverses *c* cells throughout the sensor field, in other words, it traverses throughout *c* hops. Therefore, the overhead is 2.*c.l,* including sending a *cache-removal* message.

Similarly, the overhead to deliver *d/m* data packets from a source to a sink is *c(d/m)*. Since we have *k* mobile sinks, the total overhead to receive *d* data packets is:

$$k.m\left(2c.l + cd/m\right) = 2kmcl + kcd$$

Plus the overhead $c^2.l$ for *data-announcement* message to reach all coordinators using simply flooding and the overhead in updating the mission of the sensor network $c^2.l$, we have:

$$CO_{CODE} = 2kmcl + kcd + 2c^2 l$$

In CODE, there is no overhead for constructing the grid, since each node computes its grid ID based on its geographical location. As analyzed in [1], the overhead due to GAF *discovery* message is small. Though GAF periodically sends out discovery message if the node is in the discovery or active state, the frequency will be very low. Since the broadcast is limited in one hop around a node, such overhead will not affect the whole system energy dissipation too much. Therefore, the total overhead of CODE is:

$$CO_{CODE} = 2kmcl + kcd + 2c^2 l \qquad (2)$$

This equation shows that the communication overhead of CODE is not affected by the node density. This is major advantage of CODE when using grid structure and GAF-based approach.

For TTDD, as analyzed in [4], the total overhead is

$$CO_{TTDD} = Nl + \frac{4N}{\sqrt{n_1}} l + km_1 n_1 l + kc_1(m_1 l + d)\sqrt{2N} \quad (3)$$

And for the *SODD,* the communication overhead without considering aggregation as computed in [4] is:

$$CO_{SODD} = k\, m_1\, N\, l + kc_1 d\sqrt{N} \qquad (4)$$

Where m_1 is the number of cells that a mobile sink traverses ($m_1 \le 1 + vT/\alpha$, where α is the TTDD's cell size), n_1 is the number of nodes in each cell ($n_1 = N\alpha^2/A$), and $c_1\sqrt{N}$ is the average number of sensor nodes along the straight-line path from the source to the sink. ($0 < c \le \sqrt{2}$).

For example, the sensor field *A=2000mx2000m*. The number of mobile sinks *k* is 4, moving with speed *v*=10. We suppose *m* and m_1 reach their maximum value, i.e. $m = 1 + \frac{vT}{R/\sqrt{5}}$ and $m_1 = 1 + \frac{vT}{\alpha}$, where the nominal radio range *R=250*, *T=200 seconds* and TTDD's cell size $\alpha = 200$. Suppose $c_1 = 1$, *l*=1 and *d*=100 data packets. We vary the number of nodes N from 0 to 1000 in order to show the predominance of CODE in node density.

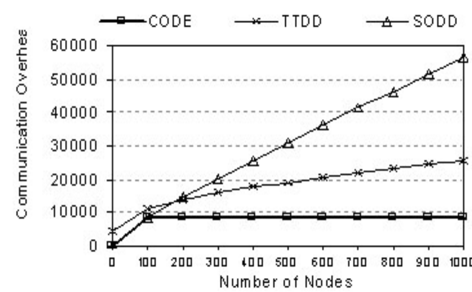
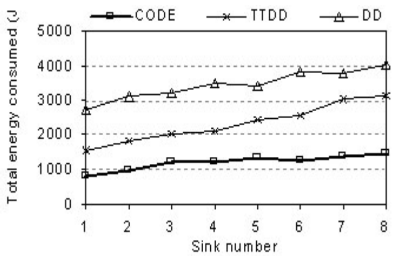


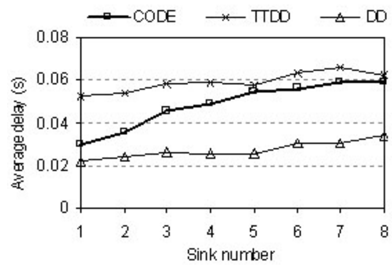
Figure 5. Communication Overhead analysis.

Fig.5 plots the communication overhead ratio of CODE compared with TTDD and SODD. In this figure, the communication overhead of CODE is much less than TTDD and SODD as the node density increases. The reason is evident that, in CODE, only the coordinators participate into sending and receiving packets. Therefore, the communication overhead only depends on the number of cell c^2 , instead of the number of nodes *N*. Whereas, in TTDD and SODD, most nodes participate in to communication process, thus the total communication overhead increases as the number of nodes increases.

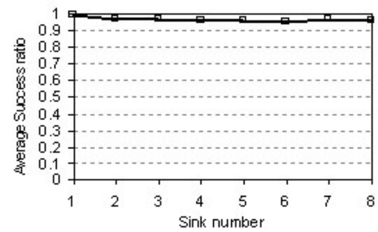
5. Performance Evaluation



(a) Energy consumption

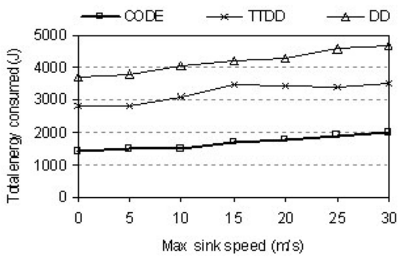


(b) Average delay

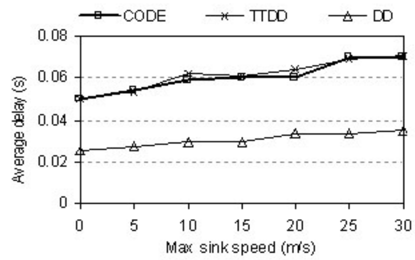


(c) Success ratio

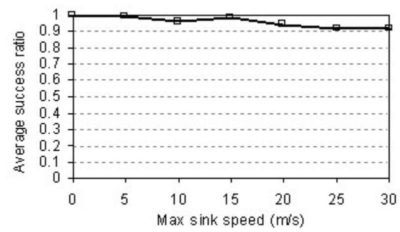
Figure 6. Impact of sink number



(a) Energy consumption



(b) Average delay



(c) Success ratio

Figure 7. Energy consumption for different sink speeds

5.1. Simulation Model

We simulated CODE on SENSE [3] and compared to other approaches DD [2] and TTDD [4]. Network comprises 400 nodes randomly deployed in a 2000mx2000m field. We use the same energy model used in ns2 that requires about 0.66W, 0.359W and 0.035W for transmitting, receiving and idling. The simulation uses MAC 802.11 DCF and nominal transmission range of each node is 250m. *Two-ray ground* is used as the radio propagation model. Each data packet has 64 bytes, query packet and the others are 36 bytes long. The default number of sinks is 8 moving with speed 10 m/sec according to random way-point model. Two sources generate different packets at an average interval of 1 second.

5.2. Simulation Result

a) Impact of Sink Number

We first study the impact of the sink number on CODE. In the default simulation, we set the number of sink varying from 1 to 8 with the max speed 10m/s and a 5-second pause time. Fig.6a shows total energy consumption of CODE. It demonstrates that CODE is more energy efficient than other protocols. This is because of two reasons. First, CODE uses QINT to efficiently aggregate query and data along data dissemination path. This path is established based on grid structure. Hence CODE can find a nearly straight route between a source and a sink. Second, CODE exploits GAF protocol, so that nodes in each grid negotiate among themselves to turn off its radio. Therefore, it reduces significantly energy consumption. In contrast, DD always propagates the new location of sinks throughout the sensor field in order for all sensor nodes to get the sink's location. In TTDD, the new multi-hop path between the sink and the grid is rebuilt. Also, data dissemination path of TTDD is along two sides of a right triangle. Fig.6b demonstrates the average end-to-end delay of CODE. As shown in this figure, the delay of CODE is shorter than TTDD and slightly longer than DD. In Fig.6c, it shows that the success rate of CODE is always above 90 percent. It means that CODE delivers most of data successfully to the multiple sinks.

b) Sink Mobility

In order to examine the impact of sink mobility, we measure CODE for different sink speeds (0 to 30 m/sec). In this experiment, the network consists of 8 mobile sinks and 400 sensor nodes. Fig.7a demonstrates total energy consumed as the sink speed changes. In both low and high speeds of the sinks, CODE shows the total energy consumed is better than other protocols, about twice less than TTDD and three times less than DD. The reason is that, aside from above explanations, the mobile sink contact with the coordinator to receive data while it is moving. Thus, the query only needs to resend as it moves to another grid. Fig.7b shows the delay of CODE which is comparable with TTDD and longer than DD. In Fig.7c, the success rate is also above 90 percent. These results show that CODE handles mobile sinks efficiently.

c) Impact of Node Density

To evaluate the impact of node density on CODE, we vary the number of nodes from 300 (1 node/cell on average) to 600 nodes (2 nodes/cell). Eight sinks move with speed 10m/sec as default. Fig.8 shows the energy consumption at different node densities. In this figure, CODE demonstrates better energy consumption than other protocols. As the number of nodes increase, the total energy consumption slightly increases. This is because of turning off node's radio most of the time. Therefore, energy is consumed mostly by the coordinators. While in TTDD and DD, nodes which don't participate in communication still consume energy in sleeping mode.

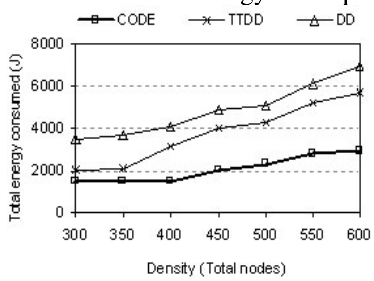


Figure 8. Energy consumption for different node density

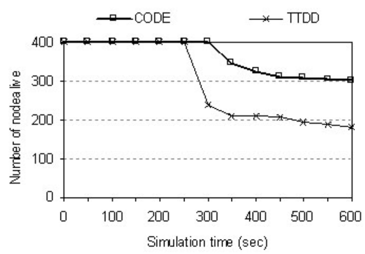


Figure 9. Number of node alive over time

d) Network Lifetime

In this experiment, the number of sensor nodes is 400. A node is considered as a dead node if its energy is not enough to send or receive a packet. Fig.9 shows that number of nodes alive of CODE is about 60 percent higher than TTDD at the time 600sec. This is because of two reasons. The first is that CODE focus on energy efficiency. The second is that rotating coordinators distributes energy consumption to other nodes, thus nodes will not quickly deplete its energy like other approaches. TTDD concentrates on dissemination nodes to deliver data, therefore such nodes will run out of energy quickly. We do believe that when the node density is higher, the lifetime of CODE will be prolonged much more than other approaches.

7. Conclusion

In this paper, we introduced CODE, a Coordination-based Data Dissemination mechanism for wireless sensor networks. This scheme is based on GAF protocol to conserve energy and prolong network lifetime. CODE employs grid structure and QINT to build up an efficient data dissemination path between sources and multiple mobile sinks. The simulation on SENSE shows that CODE is successful in meeting design goals of energy efficiency, network lifetime while delivering most of data successfully to mobile users. The simulation also shows that CODE is more energy efficient and has longer network lifetime than other approaches, especially to sensor networks with high node density.

References

[1] Y. Xu, J. Heidemannn, and D. Estrin. "Geography-informed energy conservation for ad hoc routing". In Proc. of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001), Rome, Italy, July 2001.

[2] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva. "Directed diffusion for wireless sensor networking" Networking, IEEE/ACM Transactions on Volume: 11 Issue: 1 , Feb. 2003 Page(s): 2 -16.

[3] Gang Chen et al "SENSE - Sensor Network Simulator and Emulator" http://www.cs.rpi.edu/~cheng3/sense/

[4] Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, Lixia Zhang. "Sensor Networks: A two-tier data dissemination model for large-scale wireless sensor networks" Proceedings of the Eighth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM 2002), Sept 2002, Atlanta, GA

[5] Joanna Kulik, Wendi Heinzelman, Hari Balakrishnan. "Negotiation-based protocols for disseminating information in wireless sensor networks" ACM Transaction on Volume 8 , Issue 2/3 March-May 2002.

[6] Yan Yu, Ramesh Govindan, Deborah Estrin. "Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks (2001)" UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, May 2001.

[7] Hyung Seok Kim, Tarek F. Abdelzaher, Wook Hyun Kwon "Dissemination: Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks" Proceedings of the first ACM international conference on Embedded networked sensor systems, November 2003

[8] Sooyeon Kim; Son, S.H.; Stankovic, J.A.; Shuoqi Li; Yanghee Choi; "SAFE: a data dissemination protocol for periodic updates in sensor networks" Distributed Computing Systems Workshops, 2003. Proceedings. 23rd IEEE International Conference on, 2003 Pages:228 – 234

[9] F. Ye, S. Lu, L Zhang. "GRAdient Broadcast: A Robust, Long-lived, Large Sensor Network" http://irl.cs.ucla.edu/papers/grab-tech-report.ps, 2001.

[10] Y.Xu, J.Hendemann, and D.Estrin. "Adaptive energy-conserving routing for multihop ad hoc networks". Technical Report TR-2000-527, USC/Information Sciences Institute, Oct. 2000. Available at ftp://ftp.isi.edu.isis-pubs/tr-527.pdf

[11] M. Stemm and R.H Katz. "Measuring and reducing energy consumption of network interfaces in hand-held devices". IEICE Transaction and communication, E80-B(8): 1125-1131, Aug. 1997