A Coordination-based Data Dissemination Protocol for Wireless Sensor Networks

#Hung Le Xuan [1], Sungyoung Lee [1]
[1]*Departement of Computer Engineering, Kyung Hee University*
Giheung-Eup, Yongin-Si, Gyeonggi-Do, 449-701, Korea, { lxhung, sylee}@oslab.khu.ac.kr

Abstract

In recent years, many research in fields of data dissemination protocol to mobile sinks for wireless sensor networks have been published. Such protocols inspire us to design CODE, a ***CO****ordination-based data* ***D****issemination protocol for wireless s****E****nsor networks. CODE considers energy efficiency and network lifetime, especially to sensor networks with high node density. CODE is based on grid structure to establish an efficient data dissemination path between sources and mobiles sinks. Our simulation result shows that CODE consumes less energy and has a longer network lifetime compared with other approaches.*

1. INTRODUCTION

Recent advances in electronic and communication technology have enabled large scale sensor networks with hundreds or thousands of unattended sensors. These distributed sensing systems enable remote monitoring and event detection in a large region or an inhospitable area. Data sources are usually located where environment activities of interest take place. Mobile users use hand-held devices such as PDAs to gather sensing information from environment.

Handling such mobile sinks brings new challenges to large-scale sensor network research. In recent years, many research have been published to provide efficient data dissemination protocol to mobile sinks [3,4,5,8,9,12]. Most of these efforts focus on energy efficiency. In sensor networks, the energy consumption of each node is dominated by the cost of communication, rather than computation. Other research [13,14,15] have shown that we can not ignore energy consumed when nodes fall into *idling* mode. For example, M.Stemm *et al* [14] and Y.Xu *et al* [15] show that energy consumption for *idle:receive:send* ratios are *1:1.05:1.4*, respectively. Consequently, they suggest that energy optimizations must turn off the radio. Doing this not only simply reduces packet transmission but also conserves energy both in overhearing due to data transfer, and in idle state energy dissipation when no traffic exists, specially in sensor networks with high node density. Turning off unnecessary nodes and maintaining a data dissemination path to mobile sinks via the others should be designed in efficient manner.

Considering such issues, we propose CODE, a Coordination-based Data Dissemination protocol. In CODE, not all sensor nodes have to participate in network processing but fall into sleep mode based on GAF protocol [11]. Also, CODE is based on grid structure to establish a fast and precisely direct data dissemination path from sources to mobile sinks without flooding and additional path setup phase.

The rest of this paper is organized as follow. Section 2 describes the system model. In section 3, we introduce our proposed protocol. We present the performance evaluation in section 4. Section 5 mentions about related work. Finally, we conclude the paper in section 6.

2. SYSTEM MODEL

We consider a large scale sensor network with thousands of sensor nodes scattered randomly. Each node acts as either a source to sense information from the environment or a router to forward data through the sensor field to the interest users. Many users moving within the region equipped with mobile devices such as PDAs. They query via sensor nodes for data about the current status or summary of the recent activities of the target. For example in a forest, a user's query might be "Is there any dangerous animal in 200m around me" or "Let me know if a fire is detected".

In our model, we rely on the assumptions that all sensor nodes are stationary. Each sensor is aware of its residual energy and location. Location awareness can be based on either GPS [23] or other techniques [16,24]. In addition, to make redundant nodes stay in the sleeping mode, CODE uses modified GAF protocol [11], i.e. we adapt GAF approach to our model with following modification in order to provide necessary services for CODE layer:

- Before a coordinator is replaced, it handles all its routing information to the new coordinator.
- Each coordinator is replaced by another only if its residual energy is less than a predetermined *Threshold*. Doing this conserves energy for handling coordinator's information.
- Grid IDs are indexed according to CODE rules (section 3.1) in order to provide the best path routing.

- Services of GAF to support mobile nodes are removed to conserve energy and reduce collision since all nodes are stationary in our scenario.

Fig.1 depicts our general model where the routing algorithm is implemented above the modified GAF protocol. In this paper, we only focus on CODE routing algorithm. Details of GAF algorithm can be referred in [11].

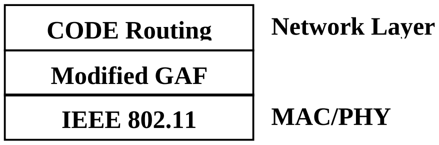

Fig. 1. CODE system model

3. PROPOSED PROTOCOL

In CODE, the sensor network field is divided into grids. Grids are indexed based on its geographical location (Fig.2). Each grid has one coordinator which acts as an intermediate node to cache and relay data. CODE has two major phases: *query transfer* phase and *data dissemination* phase. During *query transfer* phase, the data dissemination path is established based on grid IDs and target's location. A mobile sink selects a coordinator in the same grid to act as its *Agent*. Periodically, the sink checks its location. If the sink moves out of the grid, it first sends *cache-removal* message to clear out the previous data dissemination path and then re-sends a query to set up a new one. To establish the data dissemination path, CODE is based on grid IDs, instead of individual nodes (i.e. instead of knowing which next node a node should relay data to, CODE needs to know which next grid it should relay to).

A. Grid Indexing

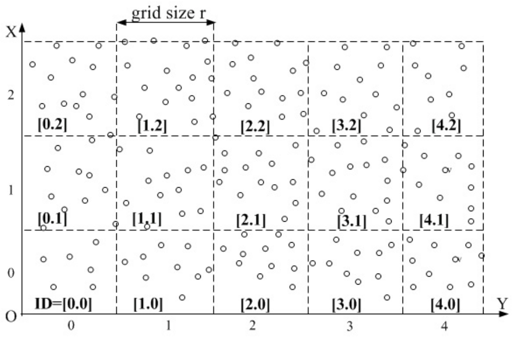

Fig. 2. Grid Indexing

We assume that we have partitioned the network plane in virtual *MxN* grids (for example in Fig.2 that is 5x3 grids). Each grid ID which has a typed [CX, CY] is assigned as follows: at the first row, from left to right, the grid IDs are [0.0], [1.0], [2.0], [3.0], and [4.0]. Likewise, at the second row, grid IDs are [0.1], [1.1], [2.1], [3.1], and [4.1], and so forth. To do this, based on the coordinate (x, y), each node computed itself CX and CY as follow:

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

where r is the grid size and $\lfloor k \rfloor$ is largest integer less than k.

B. Query Transfer and Data dissemination phases

Query Transfer phase

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). For example in Fig.3, node n1 and node n2 receive a query from sink1 and sink2, therefore it maintains a QINT as Fig.4.

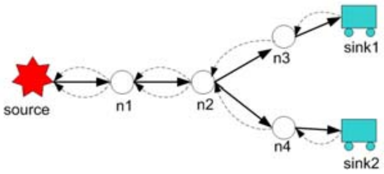

Fig. 3. Query transfer and data dissemination path setup

Node n1		
query	**sink**	**uplink**
A	sink1	n2
A	sink2	n2

Node n2		
query	**sink**	**uplink**
A	sink1	n3
A	sink2	n4

Fig. 4. Query information table maintained at nodes n1 and n2

Receiving a query from an uplink node, a node first checks if the query exists in its QINT. In our model, we define that two entries in QINT are identical if all their corresponding elements are identical. If so, the node simply discards the query. Otherwise, it caches the query in the QINT. Then, it computes the ID of next grid to forward the query. This algorithm is described in Fig.5. In this figure, *NODE* is the current node and *src_addr* contains the target's location. If NODE is in the target's region, it aggregates sensing information from surrounding nodes and sends along data dissemination path. Otherwise, it finds the next grid which closest to the target to relay the query. In case the next grid contains no node (so-called void grid), it try to find a round path. To do this, it first calculates the disparity δ_{CX}, δ_{CY} as:

$$\Delta_{CX} = p->src\_addr.CX - NODE.CX, \quad \delta_{CX} = \frac{\Delta_{CX}}{|\Delta_{CX}|}$$

$$\Delta_{CY} = p->src\_addr.CY - NODE.CY, \quad \delta_{CY} = \frac{\Delta_{CY}}{|\Delta_{CY}|}$$

Then, the next `GridID[NEXTHOP.CX, NEXTHOP.CY]` is

$$\mathtt{NEXTHOP.CX} = \mathtt{NODE.CX} + \delta_{CX}$$

$$\mathtt{NEXTHOP.CY} = \mathtt{NODE.CY} + \delta_{CY}$$

```
Find_NextHop(NODE, packet* p)
{
If (NODE is in Target's Location)
        NODE.send_data();
Else{
        $\Delta_{CX} = p->src\_addr.CX - NODE.CX$;
        $\Delta_{CY} = p->src\_addr.CY - NODE.CY$;
        $\delta_{CX} = (\Delta_{CX} == 0)\ ?\ 0 : \frac{\Delta_{CX}}{|\Delta_{CX}|}$;
        $\delta_{CY} = (\Delta_{CY} == 0)\ ?\ 0 : \frac{\Delta_{CY}}{|\Delta_{CY}|}$;
        NEXTHOP.CX = NODE.CX + $\delta_{CX}$;
        NEXTHOP.CY = NODE.CY + $\delta_{CY}$;
        If (lookup_neighbor(NEXTHOP)==TRUE)
              return NEXTHOP;
        Else
              find_round_path();
}
```

Fig. 5. Pseudo-code of finding next hop algorithm

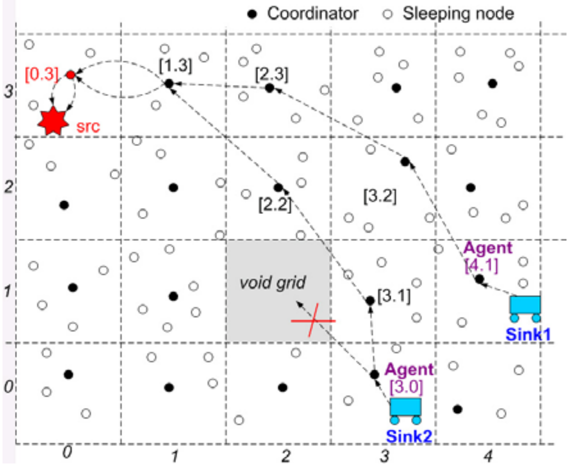

Fig. 6. Multi-hop routing through coordinators

For example in Fig.6, 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 coordinator in the grid [3.0] encounters a void grid [2.1], thus it finds a round path as [3.0], [3.1], [2.2], [1.3], [0.3]. A data dissemination path is discovered by maintaining a QINT at each intermediate node. A query from a sink is transmitted only once, and will be re-sent if the sink moves out of its grid (section 3.3).

Data Dissemination phase

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.4, 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.

C. Handling Sink Mobility

CODE is designed for mobile sinks. In this section, we describe how our approach maintains efficiently a data dissemination path as a sink moves within the sensor field.

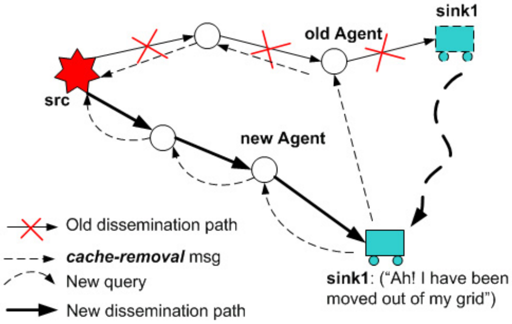

Fig. 7. Handling sink mobility

Every a duration of time τ_{grid}, 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 type, the sink address and the target's location. The old Agent is in charge to forwards the message along the old dissemination path as depicted in Fig.7. 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. 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 3.1. 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 chooses a closest coordinator to act as its Agent.

4. PERFORMANCE EVALUATION

A. Simulation Model

We developed a simulator based on SENSE simulator [22] to evaluate and compare CODE to other approaches such as TTDD [4]. To facilitate comparisons with TTDD, we use the same energy model used in ns2 [25] that requires about 0.66W, 0.359W and 0.035W for transmitting, receiving and idling respectively. The simulation uses MAC IEEE 802.11 DCF that SENSE implements. The nominal transmission range of each node is 250m [11].

Our goal in simulating CODE is to examine how well it actually conservers power, especially in dense sensor networks. In the simulation, we take into account of total energy consumed for not only transmitting, receiving but also idling. The sensor network consists of *400* sensor nodes, which are randomly deployed in a *2000mx2000m* field (i.e. one sensor node per *100mx100m* grid). *Two-ray ground* is used as the radio propagation model and an omni-directional antenna having unity gain in the simulation. 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 (i.e. the fastest human speed) according to *random way-point* model [21]. Two sources generate different packets at an average interval of *1* second. Initially, the sinks send a query to the sources. As a source receives a query, it starts generating and sends data to the sink along the data dissemination path. The simulation lasts for *200* seconds.

We use four metrics to evaluate the performance of CODE. The energy consumption is defined as the total energy network consumed. The success rate is the ratio of the number of successfully received packets at a sink and the total number of packet generated by a source, averaged over all source-sink pairs. The delay is defined as the average time between the time a source transmits a packet and the time a sink receives the packet, also averaged over all source-sink pairs. We define the network lifetime as the number of nodes alive over time.

B. Simulation result

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.8, Fig.9 and Fig.10 plot the energy consumption, delay and success rate for different numbers of sinks, respectively. In Fig.8, CODE shows much better energy consumption than TTDD. This is due to efficient process at each phase in CODE. Besides, the nodes which don't participate into forwarding data are turned off. Also, based on grid ID routing, CODE finds a shorter dissemination path than TTDD. In Fig.10, CODE demonstrates a comparable success rate with TTDD. Delay of CODE is shorter than TTDD and only slightly longer than TTDD when the number of sinks is 8 (Fig.9). This delay indicates that CODE transfers successfully most of the packet even to the high-speed sinks. This is because every intermediate node on dissemination path has to check its QINT before forwarding. And due to exchanging message for coordinator setting up, some packets are delayed in the node's queue for some time.

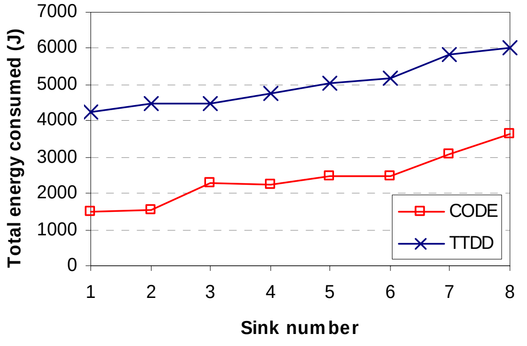

Fig. 8. Energy consumption for different numbers of sinks

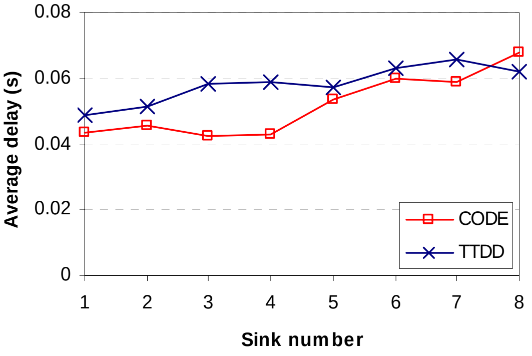

Fig. 9. Delay for different numbers of sinks

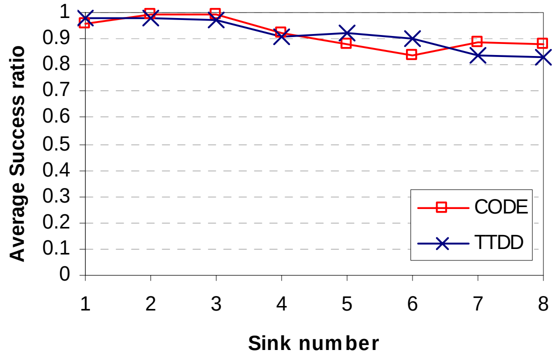

Fig. 10. Success rate for different numbers of sinks

Impact of Sink mobility

In order to examine the impact of sink mobility, we measure CODE for different sink speeds (0 to 30 m/sec). The network consists of 8 mobile sinks and 400 sensor nodes. Fig.11 demonstrates total energy consumed as the sink's speed changes. In both low and high speeds of the sinks, CODE shows the total energy consumed is about twice less than TTDD. Fig.12 shows the delay of CODE which is slightly higher than TTDD. The success ratio is comparable with TTDD as depicted in Fig.13. However the delay is somewhat longer than TTDD, due to the same reasons described above.

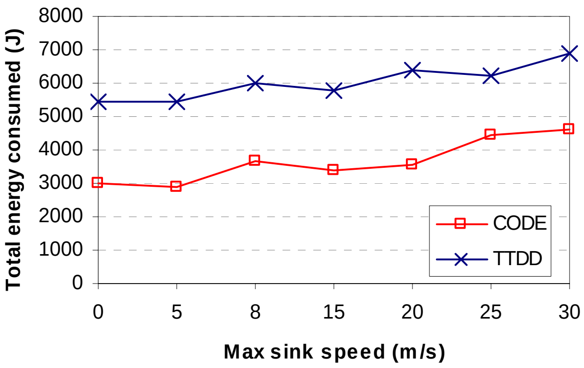

Fig. 11. Energy consumption for different sink speeds

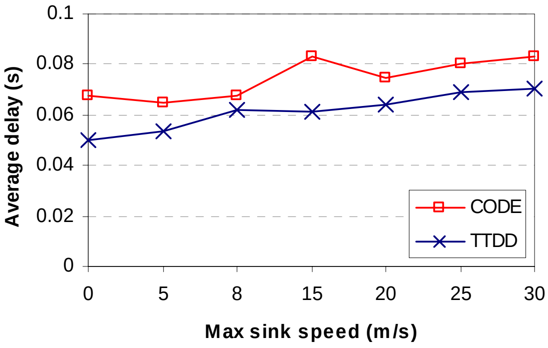

Fig. 12. Delay for different sink speeds

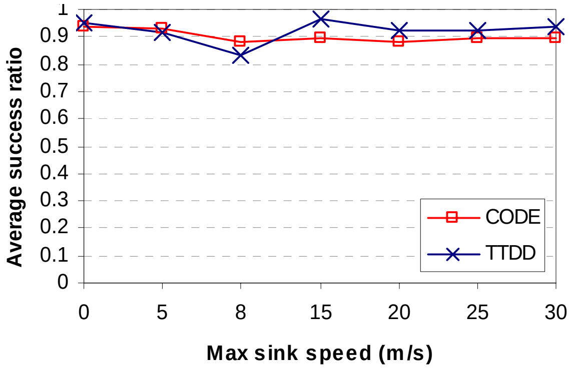

Fig. 13. Success rate for different sink speeds

Impact of Node density

To evaluate the impact of node density on CODE, we vary the number of nodes from 200 to 500 nodes. Eight sinks move with speed 10m/sec as default. Fig.14 shows the energy consumption at different node densities. In this figure, CODE consumes a little more energy than TTDD as the number of nodes is 200 to 250. The reason is that when node density is low, some grids may have no node inside (i.e. void grid). Therefore it has to find a round path to avoid these grids. However, when the number of nodes increases over 250, energy consumption of CODE is significantly less than TTDD. The first reason is that CODE turns off all the nodes which don’t participate into routing process. The second is that the number of void grids reduces as the number of nodes increases, so CODE can find a shorter dissemination path. In TTDD, nodes which don’t participate in routing process still consume energy for idling mode. Moreover, dissemination path of TTDD is along two sides of a right-angled triangle and TTDD uses local flooding. Fig.15 shows a shorter delay as the node density increases over 500. Fig.16 shows that CODE deliveries the most of data successfully. These results indicate that our main goal when designing this protocol is achieved, i.e. better energy efficiency, especially to dense sensor networks.

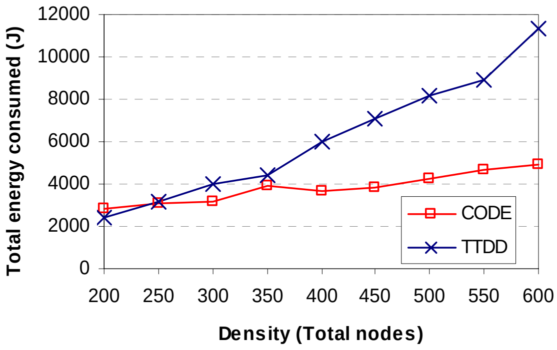

Fig. 14. Energy consumption for different node density

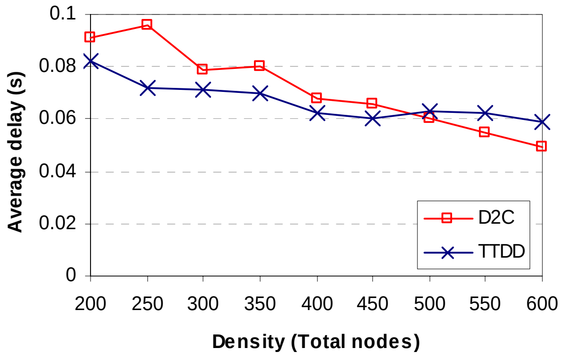

Fig. 15. Delay for different node density

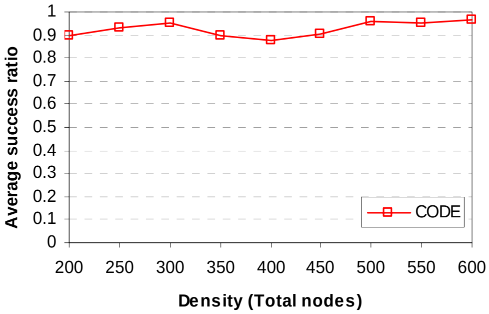

Fig. 16. Success rate for different node density

Network lifetime

In this experiment, the number of sinks is 8 moving with speed 10 m/sec. The number of sensor nodes is 400. A node is considered as a dead node if it has not enough energy to receive or transmit data. Fig.17 shows that number of nodes alive of CODE is about 19% higher than TTDD. 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.

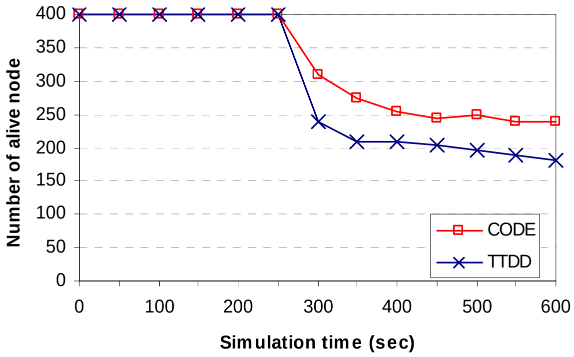

Fig. 17. Number of node alive over time

5. RELATED WORK

Many sensor network protocols have been developed in recent years. Previous work in fields of data dissemination [3,5,6,7,8,9,12] inspire us to develop an efficient approach for large scale sensor networks. One of 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 [3] and DRP [10] are similar in that they both take 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. DRP exploits application-supplied data descriptions to control network routing and resource allocation in such a way as to enhance energy efficiency and scalability. 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 neighbour selection to route a packet towards the target region. It also use 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 $\sqrt{2}$. Thus, this approach consumes more energy. 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 favours power savings over delay minimization. SEAD is therefore only useful for applications with less strict delay requirements.

CODE differs from such protocols in three fundamental ways. First, CODE exploits GAF protocol [11] to reduce energy consumption and data collision while the nodes make decision to fall into sleeping mode. Second, CODE establishes a better data dissemination path based on grid ID without flooding and additional phase. Third, the sinks do not need to periodically propagate its geographical location to the sources. In addition, like other approaches, CODE takes into account of query and data aggregation [1,2] to reduce the amount of data transmitted from multiple sensor nodes to sinks.

6. 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 extend network lifetime. CODE also adapts 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] B. Krishnamachari, D. Estrin, and S. Wicker. "*The impact of data aggregation in wireless sensor networks*". In Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, 2002.

[2] S. Maddes, R. Szewczyk, M. J. Franklin, and D. Culler. "*Supporting aggregate queries over ad-hoc wireless sensor network*". In IEEE Workshop on Mobile Computing Systems and Applcations, May 2002.

[3] 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.

[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 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 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] D. Co_n, D. V. Hook, S. McGarry, and S. Kolek. "*Declarative ad-hoc sensor networking. SPIE Integrated"* Command Environments, 2000.

[11] 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.

[12] Wensheng Zhang; Guohong Cao; La Porta, T. *"Data dissemination with ring-based index for wireless sensor networks"* Network Protocols, 2003. Proceedings. 11th IEEE International Conference on , 4-7 Nov. 2003 Pages:305 – 314

[13] G. J. Pottie and W. J. Kaiser. "*Embedding the internet: wireless integrated network sensors"*. Communications of the ACM, 43(5):51–58, May 2000.

[14] 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

[15] 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

[16] Nirupama Bulusu, John Heidemann, and Deborah Estrin. "*Gps-less low cost outdoor localization for very small devices*". IEEE Personal Communications Magazine, 7(5):28–34, October 2000.

[17] Wendi B. Heinzelman *et al*. "*An Application-Specific Protocol Architecture for Wireless Microsensor Networks"* IEEE transactions on wireless communications

[18] W. C. Y. Lee, *Mobile Cellular Telecommunications*. McGraw Hill, 1995.

[19] S. Singh and C.S. Raghavendra. "*Pamas: Power aware multi-access protocol with signalling for ad hoc networks*". ACM CCR, July 1998.

[20] O. Kasten. "*Energy consumption"*. ETH-Zurich, Swiss Federal Institute of Technology. Available at http://www.inf.ethz.ch/~kasten/research/bathtub/energy_consumption.html, 2001

[21] David B. Johnson and David A. Maltz. "*Dynamic Source Routing in Ad Hoc Wireless Networks*". In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

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

[23] US Naval Observatory (USNO) GPS Operations, http://tycho.usno.navy.mil/gps.html

[24] J.Albowitz, A.Chen, and L.Shang, "*Recursive Position Estimation in Sensor Networks*". ICNP'01, 2001.

[25] The Network Simulator ns-2 http://www.isi.edu/nsnam/ns/