A New Approach of Data Aggregation in Wireless Sensor Networks

[1] **D. Bhargav Reddy,** [2] **Bitu Kumar,** [3] **B.Giridhar,** [4] **Mr Chandan Kumar Giri**

[1, 2] B.Tech Student

[3, 4] Asst.prof in CSE dept

Abstract - In this paper we discuss algorithms that allow the concealed data aggregation (CDA) in wireless sensor networks. We describe and evaluate three algorithms that were reported to suit to the WSN scenario. As result of the evaluation, where we emphasize the awareness to potential attack scenarios, we present a brief overview of strengths and weaknesses of the algorithms. Since no algorithm provides all desirable goals, we propose two approaches to cope with the problems. The first is the successive combination of two algorithms. It increases security, while the additional efforts can be minimized by carefully selected parameters. For the second approach we face specific weaknesses and engineer mechanisms that solve the particular issues. With the considered homomorphic message authentication code and a discussion of the id-issue we exemplary evaluate the two biggest issues of the very promising CMT algorithm.

Keywords **- Wireless Sensor Network.**

1. Motivation

Reducing the total required energy in a wireless sensor network is an outstanding goal. Beside the power required for the computation on the nodes, the power needed for sending and receiving the data packets in the network is a significant factor. Sending one bit requires the same amount of energy as executing 50 to 150 instructions on sensor nodes [11]. Thus, omitting as much network traffic as possible is a substantial task in the area of designing WSN applications. A well known approach, which is the basis for the following investigations, is the in-network aggregation (INA). In a WSN sensed values should be transmitted to a sink. In many scenarios the sink does not need the exact values for all sensors but a derivative such as sum, average, or deviation. The idea of the INA is to aggregate the data required for the determination of the derivatives as close to the source as possible instead of transmitting all sensed values through the entire network. See figure 1.

A serious issue connected with the INA is the security of the data. Considered that the data is transmitted encrypted, there is the problem that all aggregation nodes, i.e. the sensor nodes that perform the actual aggregation in the network, must have access to the decrypted values. Beside the lack of end-to-end (ETE) security, such a hop-by-hop (HBH) encryption as it is for example part of TinySec [5] has the drawback that the data must be decrypted and re-encrypted on every aggregation node. An approach that promises the combination of ETEsecurity and INA is the concealed data aggregation (CDA).

2. Background

CDA is an improved version of the INA, which in contrast to the classic HBH ensure the ETE-privacy, i.e. the encrypted Fig. 1: Principle of in-network aggregation values do not need to be decrypted for the aggregation. Instead, the aggregation is performed with encrypted values an only the sink can decrypt the result. Indeed, such an approach requires sophisticated cryptographic algorithms and properties, we will dwell on later.

Considered that such a secure INA exists, it has significant benefits compared to HBH and classic ETE encryption.

1) Network traffic: One major benefit of CDA is its efficiency of both computation effort and network traffic. Since the data is aggregated in the network, the network efficiency is better than ETE without aggregation. In [2] network configurations are described that reduce the network traffic by 85% due to CDA. In order to improve the network efficiency the packet size must be considered. Large encrypted packets could negate the positive network effect.

2) Computation effort: Compared to the HBH-aggregation, the computation effort can be assumed as smaller, because there is no need for decryption and encryption on the aggregation nodes. Indeed, this is only true if the cryptographic algorithms that allow the concealed aggregation do not require too many additional computations.

3) Security: Another benefit is the improved security in comparison to the HBH-aggregation. Since the values are not decrypted on every aggregation node, there are less points where an adversary could catch the unencrypted values. Up to this point we described the math behind the CDA as kind of magic that allows calculation with encrypted values. Now the mathematical background of this magic should be explained. The fundamental basis for CDA are cryptographic methods that provide the privacy homomorphism (PH) property. An encryption algorithm E() is homomorphic, if for given E(x) and E(y) one can obtain E(x_y) without decrypting x,y for some operation _. The concept was introduced by Rivest et. Al [12] in 1978. The two most common variations of PHs are the additive PH and the multiplicative PH. The latter provides the property $E(x \times y) = E(x)\ \ E(y)$. Well known examples of multiplicative PHs are RSA and the discrete logarithm ElGamal. But since the multiplicative aggregation does not have apparent applications in the field of INA on WSNs, we restrict our search to an efficient PH to the additive PHs with the property E(x + y) = E(x) _ E(y).

3. Problem Definition

Data aggregation protocols aims at eliminating redundant data transmission and thus improve the lifetime of energy constrained wireless sensor network. In wireless sensor network, data transmission took place in multi-hop fashion where each node forwards its data to the neighbor node which is nearer to sink. Since closely placed nodes may sense same data, above approach cannot be considered as energy efficient. An improvement over the above approach would be clustering where each node sends data to cluster-head (CH) and then cluster-head perform aggregation on the received raw data and then send it to sink. Performing aggregation function over cluster-head still causes significant energy wastage. In case of homogeneous sensor network cluster-head will soon die out and again re-clustering has to be done which again cause energy consumption.

4. Data Aggregation: An Overview

Data aggregation is a process of aggregating the sensor data using aggregation approaches. The general data aggregation algorithm works as shown in the below figure. The algorithm uses the sensor data from the sensor node and then aggregates the data by using some aggregation algorithms such as centralized approach, LEACH(low energy adaptive clustering hierarchy),TAG(Tiny Aggregation) etc. This aggregated data is transfer to the sink node by selecting the efficient path.

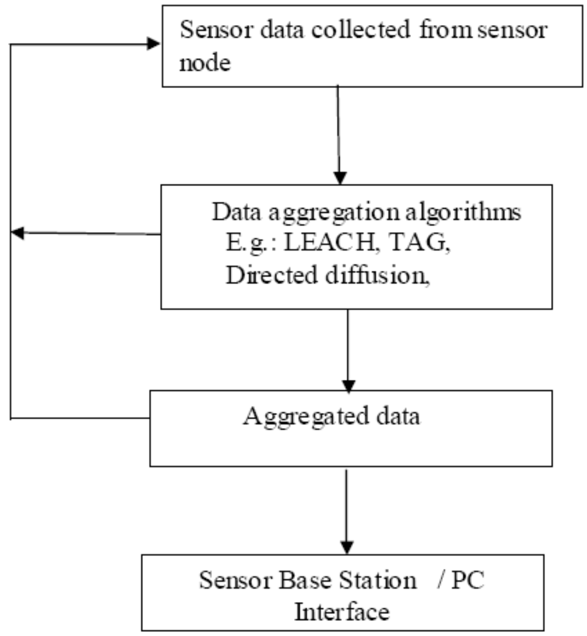


Fig 1: General architecture of the data aggregation algorithm

There are many types of aggregation techniques are present some of them are listed below.

Centralized Approach: This is an address centric approach where each node sends data to a central node via the shortest possible route using a multihop wireless protocol. The sensor nodes simply send the data packets to a leader, which is the powerful node. The leader aggregates the data which can be queried.

Each intermediate node has to send the data packets addressed to leader from the child nodes. So a large number of messages have to be transmitted for a query in the best case equal to the sum of external path lengths for each node.

In-Network Aggregation[7]: In-network aggregation is the global process of gathering and routing information through a multi-hop network, processing data at intermediate nodes with the objective of reducing resource consumption (in particular energy), thereby increasing network lifetime. There are two approaches for in-network aggregation: with size reduction and without size reduction. In-network aggregation with size
reduction refers to the process of combining & compressing the data packets received by a node from its neighbors in order to reduce the packet length to be transmitted or forwarded towards sink. In-network aggregation without size reduction refers to the process merging data packets received from different neighbors in to a single data packet but without processing the value of data.

Tree-Based Approach[8]: In the tree-based approach perform aggregation by constructing an aggregation tree, which could be a minimum spanning tree, rooted at sink and source nodes are considered as leaves. Each node has a parent node to forward its data. Flow of data starts from leaves nodes up to the sink and therein the aggregation done by parent nodes.

Cluster-Based Approach[6]: In cluster-based approach, whole network is divided in to several clusters. Each cluster has a cluster-head which is selected among cluster members. Clusterheads do the role of aggregator which aggregate data received from cluster members locally and then transmit the result to sink.

5. Query Processing

5.1.Query Models

COUGAR approach [10] proposes a query layer to support aggregate queries. With the interface provided, the clients can issue queries without knowing how the results are generated, processed and returned by the sensor network to them. The query layer processes declarative queries and generate a cost effective query plan. They follow a database approach to design a query interface for sensor networks. The view of cost is different for sensor networks. The major factor under consideration is the communication cost, involving the cost of routing the queries and aggregating data over the sensor networks. TAG also proposes a query model for supporting aggregate queries.

TAG and COUGAR are tightly coupled with the underlying aggregation schemes. [11] Proposes a Query Agent that provides application independent query interface and an API support to map the user specified queries to lower level semantics corresponding to underlying routing and aggregating protocols.

It supports different communication models - anycast, unicast, multicast and broadcast. Query agent will support a wide variety of routing and aggregation protocols selecting the best combination based on the type of the query.

5.2 Query Language in TinyDB

TinyDB's query language is based on SQL, and we will refer to it as TinySQL. Query Language in TinySQL supports selection, projection, determining sampling rate, group aggregation, user defined aggregation, event trigger, lifetime query, setting storing point and simple join [13].

5.3. Queries and Aggregates

The probable queries for the sensor networks can be categorized into:

1) Simple Queries
These are non aggregate queries.
Eg. "SELECT temperature FROM sensor WHERE node = z".
These are generally mapped into broadcast or point to point queries.

2) Complex Queries
They may contain sub queries.
Eg. "SELECT temperature FROM sensor WHERE room = (SELECT room WHERE floor = '3')"

3) Event Driven Queries
These are the continuous query that returns the values periodically at specified time intervals.
Eg: "SELECT AVG (temperature) FROM sensor where node = z"
The Grammar of TinySQL query language is as follows:
SELECT select-list
[FROM sensors]
WHERE predicate 294
[GROUP BY gb-list]
[TRIGGER ACTION command-name[(param)]]
[EPOCH DURATION time]

Where, select–list is the attribute list of the unlimited virtual relational table, which can include an aggregation function. Predicate is the query condition. gb–list is an attributes list. command–name is a trigger operation. Param is the parameters of trigger. Time is the value of time. TRIGGERACTION is the subordinate clause which defines the trigger. It determines the operations executed when WHERE clause is satisfied. EPOCH DURATION defines the query cycle. The meaning of the other clauses is the same as SQL. Following is an example of a TinyDB query.

SELECT nodeid, AVG(light), AVG(temp)
FROM sensors
WHERE AVG(light)=100
GROUP BY nodeid
EPOCH DURATION 5min
The meaning of the query is detecting nodeid per five minutes in which the average light is equal to 100 and eturning the nodeid and its average light and temperature.

Currently, the functions of TinyDB are very limited. Some functions supported by SQL are not supported by TinyDB.

6. Simulation and Experimental Analysis

Simulation Tools: We have plenty of simulation tools or simulators for simulating wireless networks. The simulators which are most popular are TOSSIM, NS-2, OPNET, OMNet++, J-Sim, GlomoSim, and Qualnet and so on. TOSSIM is a discrete event simulator for TinyOS (TinyOS is a popular sensor network operating system) sensor networks. Instead of compiling a TinyOS application for a mote, users can compile it into the TOSSIM [20] framework, which runs on a PC. This allows users to debug, test, and analyze algorithms in a controlled and repeatable environment. As TOSSIM runs n a PC, users can examine their TinyOS code using debuggers and other development tools. TOSSIM's primary goal is to provide a high fidelity simulation of TinyOS applications. For this reason, it focuses on simulating TinyOS and its execution, rather than simulating the real world.

Simulation run

This simulation is run for the following with aggregation and clustering Query-1.

QUERY-1: SELECT AVG (light) FROM SENSORS GROUP BY NODEID % 2 SAMPLE PERIOD 2048

QUERY-2: SELECT MAX (temp), AVG (light) FROM SENSORS SAMPLE PERIOD 2048

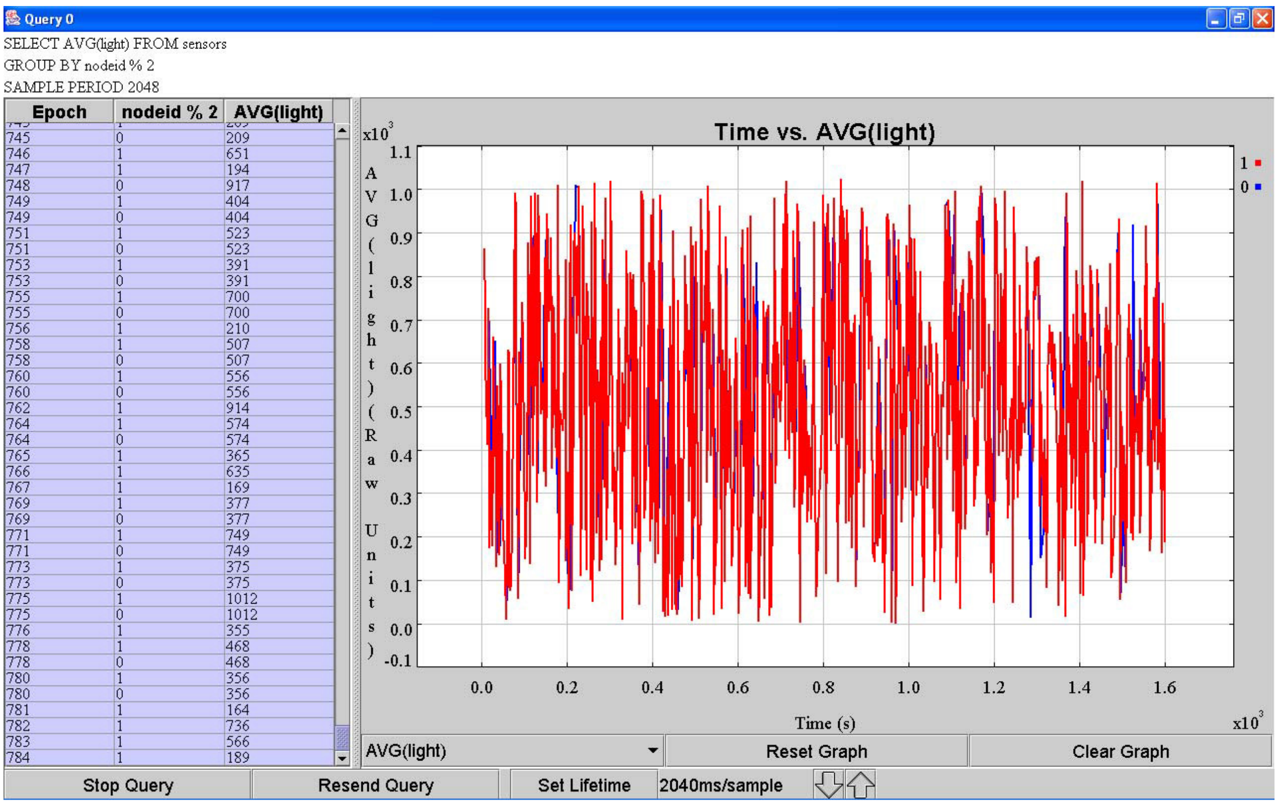
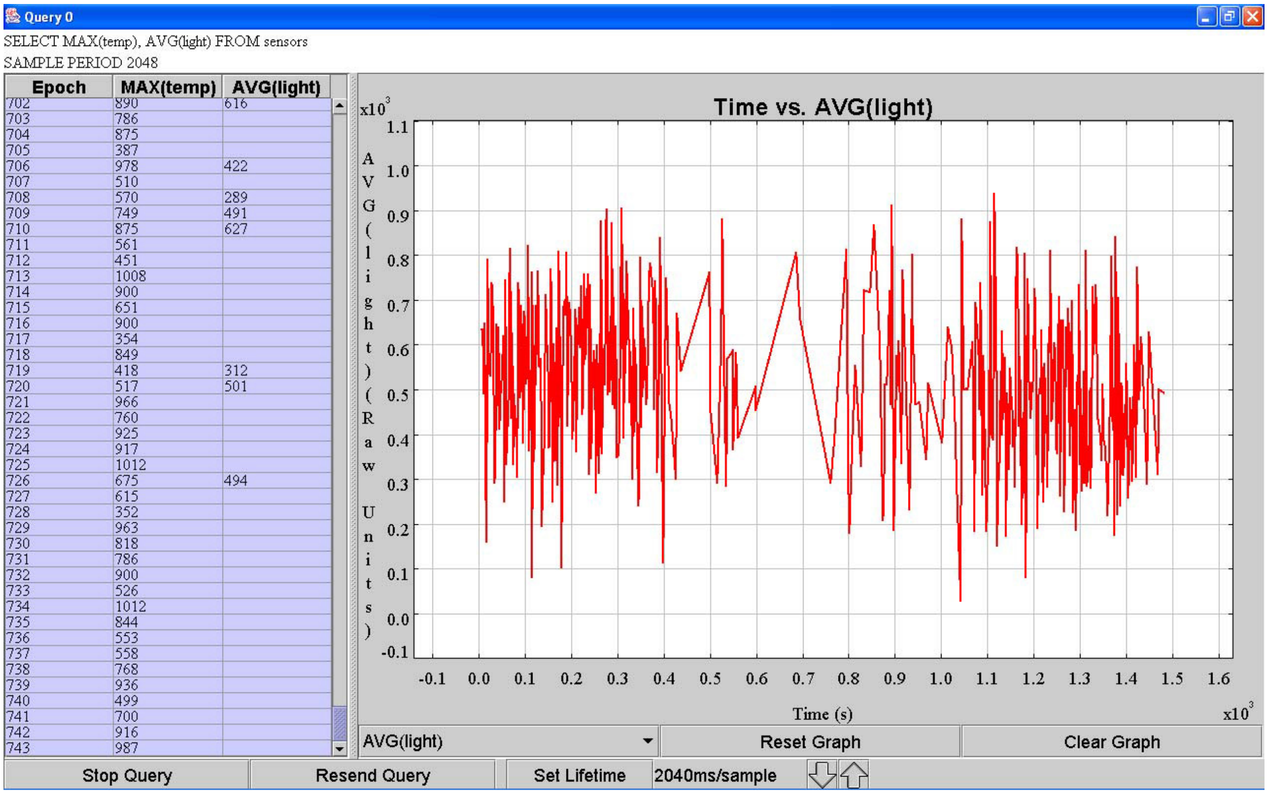


Fig 2:Result window for with aggregation and clustering

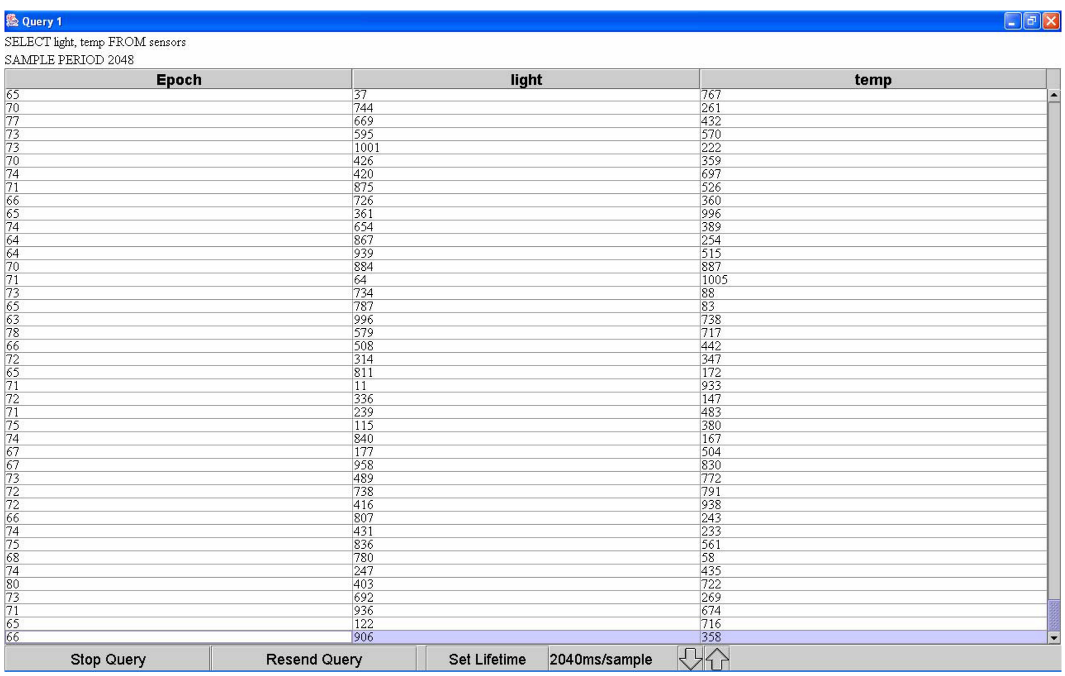


Epoch	MAX(temp)	AVG(light)
702	890	616
703	786	
704	875	
705	387	
706	978	422
707	510	
708	570	289
709	749	491
710	875	627
711	561	
712	451	
713	1008	
714	900	
715	651	
716	900	
717	354	
718	849	
719	418	312
720	517	501
721	966	
722	760	
723	925	
724	917	
725	1012	
726	675	494
727	615	
728	352	
729	963	
730	818	
731	786	
732	900	
733	526	
734	1012	
735	844	
736	553	
737	558	
738	768	
739	936	
740	499	
741	700	
742	916	
743	987	

Fig 3:Result window for with aggregation and without clustering

QUERY-3: SELECT temp, light FROM SENSORS SAMPLE PERIOD 2048

2010 IEEE International Conference on Computational Intelligence and Computing Research



Epoch	light	temp
65	37	767
70	744	261
77	669	432
73	595	570
73	1001	222
70	426	359
74	420	697
71	875	526
66	726	360
65	361	996
74	654	389
64	867	254
64	939	515
70	884	887
71	64	1005
73	734	88
65	787	83
63	996	738
78	579	717
66	508	442
72	314	347
65	811	172
71	11	933
72	336	147
71	239	483
75	115	380
74	840	167
67	177	504
67	958	830
73	489	772
72	738	791
72	416	938
66	807	243
74	431	233
75	836	561
68	780	58
74	247	435
80	403	722
73	692	269
71	936	674
65	122	716
66	906	358

Fig 4 :Result window for with out aggregation and clustering

Simulation results and comparison With aggregation query

- SELECT MAX (temp), AVG (light) FROM SENSORS SAMPLE PERIOD 2048
  Without aggregation query
- SELECT light FROM sensors SAMPLE PERIOD 2048
  With aggregation and with clustering query
- SELECT AVG(light) FROM SENSORS
  GROUP BY NODEID % 2
  SAMPLE PERIOD 2048.

Table 1

	Without aggregation		With aggregation		Cluster based aggregation	
No of Nodes	10	20	10	20	10	20
No of message transmitted	895	687	87	72	266	122

7. Conclusion

In this work we have studied the two most important parts of data communication in sensor networks- query processing, data aggregation and realized how communication in sensor networks is different from other wireless networks. Wireless sensor networks are energy constrained network. Since most of the energy consumed for transmitting and receiving data, the process of data aggregation becomes an important issue and optimization is needed. Efficient data aggregations not only provide energy conservation but also remove redundancy data and hence provide useful data only.

The simulation result shows that when the data from source node is send to sink through neighbors nodes in a multihop fashion by reducing transmission and receiving power, the energy consumption is low as compared to that of sending data directly to sink that is aggregation reduces the data transmission then the without aggregation. We have showed how aggregate queries are efficiently executed in wireless sensor networks.

8. Future Scope

Future work will focuses on the using new different routing algorithms for routing the data from the source to the sink. Our approach should confront with the difficulties of topology construction, data routing, loss tolerance by including several optimization techniques that further decrease message costs and improve tolerance to failure and loss. In addition to implementing these techniques, we need to rethink some of these techniques to present more efficiency to network changes and external factors which could affect our approach such as node mobility, obstacles and other issues. In addition as future work, we could also extend our simulator to incorporate a 3D tree construction technique.

References

[1] S. Lindsey and C. Raghavendra, "PEGASIS: Powerefficient gathering in sensor information systems," in 2010 IEEE International Conference on Computational Intelligence and Computing Research Proceedings of IEEE AerospaceConference, vol. 3, Mar. 2002, pp. 1125–1130.

[2] M. Lee, and S. Lee, "Data Dissemination for Wireless Sensor Networks", in Proceedings of the 10 th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'07).

[3] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking", IEEE/ACM Transactions on Networking, Vol. 11, no. 1, Feb 2003.

[4] H. Cam, S. Ozdemir, P. Nair, and D.Muthuavina shiappan, "ESPDA: Energy-Efficient and Secure Pattern-based Data Aggregation for Wireless Sensor Networks", in Proceedings of IEEE Sensor- The Second IEEE Conference on Sensors, Toronto, Canada, Oct. 22-24, 2003, pp. 732-736.

[5] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", (MobiCom 2000) pp 56-67.

[6] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An Efficient Clustering-based Heuristic for Data Gathering and Aggregation in Sensor Networks", IEEE 2003.

[7] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey", IEEE Wireless communication 2007.

[8] M. Lee and V.W.S. Wong, "An Energy-aware Spanning Tree Algorithm for Data Aggregation in Wireless Sensor Networks," IEEE PacRrim 2005, Victoria, BC, Canada, Aug. 2005.

[9] M. Ding, X. Cheng, and G. Xue, "Aggregation tree construction in sensor networks," in Proc. of IEEE VTC'03, Vol. 4, Orlando, FL, Oct. 2003.

[10] "The Design Space of Wireless Sensor Networks" by Kay R¨omer and Friedemann Mattern

[11] http://en.wikipedia.org/wiki/Wireless_Sensor_Network

[12] Wendi B. Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, "Energy Efficient Communication Protocol for Wireless Microsensor Networks", (33rd

Hawaii International Conference on System Sciences, 2000).

[13] S. Nath et al., "Synopsis Diffusion for Robust Aggregation in Sensor Networks," ACM SenSys 2004, Baltimore, MD, Nov. 2004.