Data Aggregation, Query Processing and Routing in Sensor Networks.

Kaushal Mittal
Roll No.- 04329024
MTech IT, KReSIT
IIT Powai, Mumbai.
Email: kaushal@it.iitb.ac.in

Anshu Veda
Roll No.- 04329022
MTech IT, KReSIT
IIT Powai, Mumbai.
Email: anshu@it.iitb.ac.in

Bhupendra Kumar Meena
Roll No.- 04322008
MTech, Reliability Engg.
IIT Powai, Mumbai.
Email: bhupen@ee.iitb.ac.in

***Abstract*— Wireless sensor networks consists of battery powered, small computing devices called sensor nodes. They are used for a wide range of application ranging from millitary applications to many scientific, research and statistical applications. They are constrained with energy, memory and computing power and hence efficient techniques are required for data aggregation, query processing and routing in sensor networks. We present a breadth wise coverage of the variety of protocols developed for aggregation and routing for query processing in sensor networks.**

I. Introduction

Sensor nodes are small, inexpensive, battery-powered wireless devices with small amount of memory, processing power, and communication capabilities. Each can host a variety of sensors. For eg. MOTE [9], developed at UC Berkeley has 4Mhz Atmel microprocessor with 512 bytes RAM, 8 KB code space and 917 MHZ RFM running at 10kbps and 32 kb EEPROM. It hosts sensors like temperature, light, magnetic field, sound and power.

Wireless Sensor network is a collection of sensor nodes deployed in an adhoc fashion. Being battery powered and deployed in remote areas they have limited energy resource and hence limited. lifetime. Other constraints include, limited memory, processing power, bandwith, QoS support and large latencies. The accuracy is location dependent. For eg a temperature sensor near AC will not be able to report the correct temperature of the environment.

Due to these limitations data aggregation is an important consideration for sensor networks. The idea is to combined the data coming from different sources and enroute it further, after eliminating redundancy, minimising number of transmissions and thus saving energy. This being different from traditional address centric approaches, shifts the focus to data centric approaches.

They can be used for a wide variety of monitoring and research applications, inventory maintenance, health care , military, object recognition and tracking, research and study of biological and environmental phenomenon. All the applications depend on the ability to extract data from the network.

Sensor networks generate a large amount of data. For extracting information, we need to collect and query the data from sensor networks. The primary focus is on aggregates-summarized data. A Query can be a request for information or orders to collect more data. In section 2, we discuss the query processing support required for aggregate queries. In-network aggregation /processing protocols have been proposed for efficiently utilising the bandwidth, saving energy and increasing the lifetime of the sensor nodes. These are discussed in section 3. Such schemes require support from network layer routing protcols. We present a brief overview of routing protocols in section 4.

II. Query processing

In earlier approaches sensor nodes had an embedded program to compute the data and sent it to the sink via the underlying network protocol. The drawback of this approach is that the user cannot change the behavior of the system on the fly. This requires an interface that separates the query processing form the underlying networking protocols and details.

A. 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 databse 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.

A query plan decides the amount of computation to be pushed down the network and specify the role and responsibility of each sensor node. This is represented by the flow blocks. Creatig flow blocks use resources. They need to propogate the flow blocks to related sensor nodes. So the flowblocks must be resued. [9] 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.

B. Queries and Aggregates

The probable queries for the sensor networks can be categorised 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 continous queires that returns values periodically at specified time intervals. Eg. " SELECT AVG (temperature) FROM sensor where node = z"

The query interface proposed by [5] and [20] is similar to SQL, supporting the SQL clauses like SELECT, FROM, WHERE, GROUP BY, HAVING and aggregate clauses like MAX, AVG, MIN, COUNT and SUM. The difference is they support continous monitoring queries by adding the clauses like DURATION and EVERY [5] representing the lifetime of the query and the rate of answering the query.

[20] TAG uses the EPOCH clause to specify the amount of time to wait before acquiring and disseminating next sample. During this duration the node may sleep. This reduces the power consumption and increases the life time. The queries like SUM, AVG, COUNT are sensitive to duplicate data whereas the MAX, MIN queires are not. This kind of heuristic information is helpful in optimising the aggregation and routing protocols to provide energy efficient aggregation.

III. Data Aggregation

To support queries over sensor networks, the distributed data over sensor nodes need to be processed. This poses up an implicit requirement for aggregating the data. Optimal Data Aggregation is an NP-HARD problem [1]. There can be two approaches for data aggregation.

1) Centralised approach
2) In-Network Aggregation

A. Centralised 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 sends the data packets to a leader, which is a powerful node. The leader aggregates the data which can be queried. The underlying routing protocols needs minimal changes. Wireless protocol like AODV can be used. 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. The major drawback is the sensor networks are energy constrained and hence the approach is costly as requires a lot of messages to be exchanged for each query.

B. In-Network Aggregation

This is a data centric approach where the intermediate nodes can look at the content and perform aggregation on multiple packets it recieves. The tranmitting and recieving cost is greater than the computing cost. This has been the motivation for in-network aggregation. It implies the shifting of a part of the computation from clients to the sensor nodes - aggregating the results or filtering the irrelevant data records with an aim of reducing the message transfer and efficient use of bandwidth, thereby increasing the life time of the system. There can be two variations for In-Network Aggregation:

1) *Packet Merging*
   Sending individual packets incurs the extra cost of packet headers each time. A better approach would be to aggreagate the packets and send a single large packet. This will reduce the cost associated with packet headers. Also the nodes may transmitt the data periodically and hence can get time to go into power safe mode.
2) *Partial Aggregation*
   For aggregate queries the intermediate nodes can calculate the partial results that can be used to calculate the final results. This saves energy considerably. The limitation of such aggregation is the underlying networking protocol needs to provide support for synchronisation. Also the synchronisation increases the response time of the queries. Spanning trees are considered the best routing structures for energy efficient aggregation.

C. Cougar Approach

Cougar[9] have abstracted the data generated by the sensor networks as an append only relational table. In this table each attribute represents either an information about the sensor node or data generated by it.

Cougar[9] approach provides partial aggregation at child nodes for queries. Each node maintains a *waiting list* of child nodes sending packet to it. The node expects the same set of child nodes to send the packets again in the next hop. But the network being energy constrained and mobile, the child node may die out due to power failure or it might move to other location. Even the node itself may move. In any of the above mentioned conditions, the Node may have to wait indefinitely for a packet from the child node. To avoid such failures, the timer is used, after timeout the parent node assumes child to be inactive. The query is intialised at the root node. It sends an initialisation broadcast message, containing the hop count used to determine the depth of a node in the spanning route tree generated. Eventually, the nodes may die. This requires repair of routes. They propose local repair strategy that finds a new route in its neighbourhood.

The approach provides modification in AODV for intercepting the packets for aggregation at network layer because it does not generate duplicate packets.

D. TAG Approach

TAG [9] proposes the tag approach for sensor networks with Mote [20] nodes. It assumes that if a node A listens node B then node B also listens node A. This may not be true always. The spanning tree needs to be created for energy efficient routing of messages. The root node starts the broadcast by sending a message with level 0 and its sensorId. All nodes hearing the message increase the level field attach their id and broadcast it again. They select the source of the message as their parent.The process continues down the tree.

computing a query involves two phases - propagating the query and aggregating the result. P transmits the query packet . If their is child then it will transmit it again , and P will hear it and come to know that it has a child. other wise after time *epoch*, it assumes it is a leaf and sends the data related to the query.It assumes that the childs will report within time *epoch*.

For continous monitoring queries, pipelining approach, a modification of the above aggregation scheme, can be used. The query is initiated at the root. The immediate child nodes when recieves the packet send their current aggregated values to the root. It also brodcast the message to the child nodes. so after first hop root gets aggregate from immediate childs. Duuring next hop the childs of the root node will get the packets from their child and will send the aggregated values to the root. So the root will get the aggregated values upto level 2. This process goes on and after n hops the root gets the aggregates from the n^th level. The first complete aggregate will take a large number of messages to be generated. but a after that each new aggregate from the network will be generate in *n* messages where *n* is the depth of the tree.

snooping can also be used to build the tree. The client query is given to the root of the network. Root sends its own value.The packet sent by the root also reaches its child. When the child hears the root sending the value it assumes it should also send data and send its values to the root. The child of these nodes follow the same and the process goes on. This saves a lot of messages used during the propagation phase. Multiple parents can also improve the precision of the aggregated data. For min max queries , the snooping is a good approach. It hears the neighbouring nodes sending data. If its own value is a better result then it hears, only then it sends the data. Another approach can be finding the min or max value upto *k* level and then asking the nodes to send the value less than or greater than the value observed.

For group queries each partial aggregate record is assigned a group id based on the group expresiion. On hearing packets from child nodes, if the group number is same as its own group number, it combines the value, and if belong to other group stores it directly for forwarding. In case of *having* clause a filter is applied to the group aggregates created at each node. Groups can be large and hence main memory may fall short. To avoid this pre preaggregation technique is used.

The TAG offers a lot of advantages- saves energy, minimises the number of messages transfer, use of epoch allows nodes to sleep during idle time thereby saving energy.

E. TiNA

TiNA [7] provides further optimisations over COUGAR[10] and TAG[9]. It exploits temporal coherency tolerances, which further reduces power consumption by 60% extending the lifetime by 300%. The approach is to send the data only when there is a significant change in the data value. The concept of epoch as in TAG[9] is also used here for synchronising the reciepts of the packets from the child nodes and sending the aggregate. In TiNA[7], along with the WHERE clause a *tct* condition is given. A data value can be ignored if the variation from the previous value is withing the range specified by the *tct* condition. WHERE clause filters the data that do not meet the condition and *tct* filters the data whose value is in the tolerance range specified.

This requires higher memory requirements at each node because they need to store the intermediate results of the child nodes, partial aggregates. This is required because next time when they send the data, they first compares the view stored with the information recieved . If they are not in the range specified by tolerance, only then data is forwarded. The advantage is the significant reduction in number of messages over COUGAR[10] and TAG[9].

F. DQEB - Dynamic query-tree Energy Balancing Protocol.

Most of the above protocols assumes the construction of the static trees. They assume that the energy will remain the same throughout the time in operation. But the practical fact is - though initially the nodes may have same energy but with time the energy of the non leaf nodes decreases as compared to the leaf nodes. The non leaf nodes are often engaged in transmitting and recieving more data as compared to leaf nodes. So more energy is spent at non leaf nodes as compared to the leaf nodes.

The DQEB [6] approach deals with this by proposing an energy balanced approach of dynamically modifying the tree structure based on the energy left at nodes. In this approach the authors assumes that the nodes are organised into clusters with cluster heads. Each node is assigned a weight which goes on increasing with decreasing lifetime or energy.

As the energy decreases, it is wiser to move the node down the tree i.e. turn it in to a leaf node so that the tree do not get disconnected and involve a costly phase of repair. The energy cost depends on the number of leaf and non leaf nodes and the energy remaining at the node. The optimal solution is that the weights of the non leaf nodes should be minimum.

whenever the weight of a non leaf node goes down a threshold, the coordinator node asks all its child for alternating parents,Then, using a greedy approach it selects the alternating parents for all its childs and itself becomes a leaf node. The issues like the alternating parent may be an offspring for the current coordinator, or sibling of the child node. So the set of alternating parents consists of only those nodes that are not the offspring of the coordinator and not the sibling of the node. Since the nodes with less energy has become the leaf node it will live a little longer as now it only has to send its data. This increases the life time. Also in case, if the node die out due to power failure, the extra cost incurred in building the tree will not be required, since the tree is still connected.

All the aggregation schemes in turn require the support from the network layesr. The major modification required in the existing protocols is the need to intercept the intermediate packets. In the next section we discuss the routing protocols.

IV. Routing Protocols

Routiong is inherent to any network system. Sensor Networks are wireless networks. With centralised approach of aggregation, the exisiting routing protocols like AODV may work. The recent researches have significantly highlighted the importance of In-Network Aggregation in sensor networks. They require to intercept data at network layer, as opposed to the point to point routing protocol support in exisiting routing protocols. This motivates the requirement for development of routing protocols for sensor networks. We discuss the routing protocols for sensor networks under the following heads :

1) Conventional Routing Algorithms
2) Diffusion Routing Protocols
3) Location aware routing Protocols
4) Energy Conservation based Routing Protocols

A. Conventional Routing Algorithms

In conventional routing protocols aimed at making data available at all the nodes, as soon as an event is encountered.

1) Flooding and Gossiping[4] These focus on reducing the convergence time of data delivery. Whenever a node encounters an event, in flooding, it forwards it to all its neighbours. This incurs a a large number of message transfers. Flooding is therefore highly energy consuming approach. A modification suggested was gossiping, where the node probabilistically selects one neighbour to forward the data packet. The obvious problems with these protocols are Implosion, Overlapping Data and Resoure Blindness. To overcome these problems,[4] proposed SPIN protocols.
2) SPIN: Sensor Protocols for Information via Negotiation It basically follows the publish subscriber model. In SPIN, each node that gets data, advertises it to its neighbours using ADV packets. It uses the concept of Meta-data, which is sent with advertisement. The nodes use the meta data to check if they have the corresponding data item. When not found, they respond via REQ Packet which is answered by DATA packet. Each new DATA packet is advertised to neighbours. Each node also has a resource manager, which can be used for resource adaptation and optimizations.

   The main advantage of SPIN is it can be run in completely unconfigured n/w and when a change in topology occurs changes only have to travle one hop and the algorithm continues. Also it does not send any redundant DATA packet. The only overhead is of REQ and ADV packets, which authors consider to be of ignoreable sizes.

   However, with the emergence of reactive approaches like the ones discussed below, these schemes seem to have grown completely outdated, as request-driven approach makes more sense in case of mobile, ad-hoc networks and avoids unnecessary propagation of data throughout the network.

B. Diffusion Routing

These protocols gave a radical approach to routing. The major change was that instead of flooding multiple copies of each data item into the network, a reactive approach where destination initiated querying was proposed.

1) This method cited in[8] was initially designed for location tracking systems. It uses application specific naming schemes, to identify data items during initial interest phase. Interest phase involves transmission of interest messages by the sink nodes which are flooded through the network. Each node which recieves an interest message forms a gradient, towards the requesting nodes. Thus each node just stores neighbour's information. The sources send their data to their gradients in the data propagation phase, which propagates in the same manner. Once the sink starts recieving messages it reinforces some paths, by sensing cost effective paths. For this purpose it uses those paths which deliver data in shortest time.
2) An enhancement of the same is CLUDDA [3], where the authors propose to use clustering for more efficient use of resources and extend the query format to support dynamic aggregation. The idea of clustering is to prevent the flooding during interest propagation, which in this case is limited to the cluster heads. The regular nodes do not participate unless they can support some request. The revised format of interest allows the node to handle unfamiliar queries and formation of data aggregation. The cluster heads act as dynamic data aggregation points thereby ensuring aggregation as close to the source as possible. Noteably, these points are dynamic. Operation of node in unfamiliar environment, is a promising feaure supported by this scheme as well as layered data aggregation.

C. Location Aware Routing

These are tree based algorithms, based on central link as root node. The construction of routing tree is quite similar to the approach followed in TAG[9]. The root node sends a packet with level 0 and its sensorId. The childs who recieve the packet, sets their own level as one more then the level in the packet they recieve. for eg. if a node recieves a packet with level i, then it will set its own level as $i+1$. It sets the sender of the packet as its parent. And again broadcast the packets for further formation of the tree.

[13] suggest some modifications to routing tree construction thereby introducing the concept of group-aware network configuration. The group identity is incorporated in tree construction. These groups are the same as discussed in In-network aggregation. The basic aim is to support aggregation schemes like TAG[9], Cougar[10] and TiNA[7]. In the GaNC protocol, when the root node prepares a query message, with level it also appends the parent Group id. In case a node gets messages from two parents from same level, same groupid can be used as a preference. The major advantage seen by this scheme is minimization of number of message transmisissions, by *clustering* data from same groups, which accounts for lesser partial state maintenance.

D. Energy Conservation-Based Routing Schemes

Energy being the most important constraint, these protocols concentrate on minizing the energy spent in transmission. The basis could be minimizing the energy spent during one propogation, maximizing the lifetime by avoiding using depleted nodes. Most of these are reactive protocols too.

1) Energy-Aware Routing In this protocol[18] local flooding is done during the interest propagation phase to determine all routes to the destination. With interest message cost of route is propagated to each subsequent neighbour too. On recieving the interest message the nodes adds an energy metric to the cost. This energy metric is evaluated with a consideration of residual energy. Sink node assigns each path a probability inversely proportional to its cost. During data propagation the sink node selects a path based on this probability thereby preventing depletion of optimal paths.
2) Energy Efficient and MAC Routing This protocol[17] accounts for MAC Constraints in addition to minimizing the total energy consumed and maxmizing the network lifetime. The network is modelled as a DAG, with single sink as root. LPP approach is used to solve the optimization problem. The MAC constraints are based on the maximum rate supported by the wireless medium. Flow balance constraints ensure that total incoming traffic at each node is same as the outgoing traffic. Energy constraints ensure that none of the paths are broken. The flow balance The MAC contention, Flow balance and Energy consumption are formulated as the constraints of the LPP problem and Lifetime is maximized under these.

E. Other Approaches

1) Rumor Routing
   This algorithm[12] is an event-centric algorithm as approcah to the data-centric approaches explained above. Its intended for application where coordination system is not available for phenomenon of interest. It finds its useful in situations like those of emergency. The basic distinction from other schemes is that data here is ketyed on event and not geography or network topology.
   In [12] distinguish between, the query flooding based (like directed diffusion)and event flooding based approaches. They propose rumor to be compromise between the two. The algorithm employs a set of long-lived agents. Each node maintains an event list and neighbours' list. The agents travel through the network creating paths directed to the event. The algorithm is only useful when the number of queries and events is moderate.
2) Greedy Incremental Search
   In this routing protocol[16] the basic criteria is to incrementally establish the paths from source to the destination by the hueristic search phenomenon. In this approach the source node applies the heuristic search for finding the best neighbour node by analysing all the nodes which are lying near to the transmitting nodes.

V. Critical Analysis

The query interface must be separate from the sensor network. This frees theclient from the burden of knowing the netowrk topology , identification of the sensor nodes and the way the data is going to be aggregated. This will also help in developing a network with support for various aggregation schemes and routing protocol. Depending on the user queries, an efficient query plan will be generated that will decide, what aggregation schemes must be used, how the nodes mustbe organised and what routing protocol must be used. This can save the energy in the snsor nodes to a great extent.

After studying the various Data aggregation protocols, we feel that though the strategies like Cougar[10] and TAG[9] are performing In-Network aggregation thus, increasing the life time. The major drawback of these approaches is that they ignore the fact that the energy at non-leaf nodes differ from the energy of leaf nodes. At the time of a node failure, recovery involves a lot of communication cost, for rebuilding the tree. This also involves latencies. A better approach will be to have dynamic restructuring of the routing tree, as the non-leaf nodes reaches the end of their life time, in order to avoid, the breaking of the routing tree at inner levels. The break is costlier if it breaks near the root of the tree. Loss of data is severe. So, Dynamic aggregation protocols, that try to move the dying non-leaf nodes towards the lat level of the routing tree, making them leaf nodes, will be efficient.

Existing routing protocols for wireless networks will not be practical for sensor networks, due to the requirements of packet interception at intermediate nodes in network aggregation schemes. The convetional protocols for sensor networks cannot support data centric based aggregation schemes like TAG[9], COUGAR[10], TiNA[7] and DQEB[6]. Directed diffusion is the most widely used paradigm. More energy efficient protocols can be designed considering the MAC layer constraints and life time considerations. One comprehensive scheme based on this is [17].

VI. Conclusion

In this work we have studied the three most important parts of data communication in sensor networs- query processing, data aggregation and routing and realized how communication in sensor networks is different from other wireless networks. The implementation of a separate Query Agent seemed to be a promising approach for facilitatting the support for multiple applications via a single interface. The critical energy constraints can be overcomed by dynamic aggregation schemes where routing trees get optimally readjusted to save energy. Simply having efficient aggregation schemes is not sufficient. They must be backed by efficient routing protocols to preserve energy.

Acknowledgment

We would like to thank Prof. Parmesh Ramanathan for his valuable guidance and support in understanding the intricacies of the underlying concepts related to sensor networks and wireless networks.

References

[1] Krishanamachari, Estrin D., Wicker S., *The Impact of Data Aggregation in Wireless Sensor Networks.*, IEEE Confernece on Distributed Computing systems Workshop, 2002.

[2] Intanagonwiwat C., Estrin D., Govindan R., Heidemann J., *Impact of Network Density on Data Aggregation in Wireless Sensor Networks.*,2002.

[3] S. Chatterjea, P. Havinga, *A Dyanamic Data Aggregation Scheme for Wireless Sensor Networks.*, ProRISC 2003, Veldhoven, The Netherlands, 26-27 November 2003.

[4] Joanna Kulik, Wendi Rabiner, Hari Balakrishnan, *Negotiation based Protocols for Disseminating Information in Wireless Sensor Networks*, ACM Wireless Networks, 1999.

[5] J. Gehrke , Yong Yao , *Query Processing in Sensor Networks*, Pervasive Computing, IEEE , Volume: 3 , Issue: 1 , Jan. -March 2004 Pages:46 - 55.

[6] H. Yang, F. Ye,B. Sikdar, *A Dynamic Query-tree Energy Balancing Protocol for Sensor Networks*, Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE, Vol.3, Iss., 21-25 March 2004 Pages: 1715- 1720

[7] Jonathan Beaver, Mohamed A. Sharaf, Alexandros Labrinidis, Panos K. Chrysanthis, *Power-Aware In-Network Query Processing for Sensor Data*, MobiCom 2003.

[8] Intanagonwiwat C., Estrin D., Govindan R., Heidemann J., Fabio Silva, *Directed diffusion for Wireless Sensor Networking.*, Networking, IEEE/ACM Transactions on , Volume: 11 , Issue: 1 , Feb. 2003.

[9] S. Madden, M. Franklin, J. Hellerstein, *TAG: a Tiny AGgregation Service for Adhoc Sensor Networks*, OSDI December 2002.

[10] J. Gehrke , Yong Yao, *The Cougar Approach to In-Network Query Processing in Sensor Networks*,SIGMOD, 2002.

[11] Weisong Shi, Sivakumar Sellamuthu, Kewei Sha and Loren Schwiebert, *Query Agent: A General Query Processing Tool for Sensor Networks*, IEEE Conference on Parallel Computing Workshop, 2004.

[12] David Braginsky, Deborah Estrin, *Rumor Routing Algorithm for Sensor Networks.*, WSNA 02, September 2002, Atlanta, Georgia, USA.

[13] Jonathan Beaver, Mohamed A.Sharaf, Alexandros Labrinidis, Panos K. Chrysanthis: *Location-Aware Routing For Data Aggregation in Sensor Networks.*

[14] Thomas Hou, Yi Shi, Jianpin Pan, *A Lifetime-aware, Single-session Flow Routing Algorithm For Energy-constrained Wireless Sensor Networks.*, 2003.

[15] Thomas Hou, Yi Shi, Jianpin Pan, *Energy Efficient Routing With Guaranted Delivery in Wireless Sensor Networks.*, 2003.

[16] Solios I., Obraczka K., *The Impact of Timing in Data aggregation for sensor Networks.*, 2004.

[17] B.Hamdaoui, P. Ramanathan., *Energy Efficient and MAC-aware Routing for Data Aggregation in Sensor Networks*, 2004.

[18] Rahul C. Shah, Jan M. Rabaey, *Energy Aware Routing for Low Energy Adhoc Sensor Networks* IEEE, 2002.

[19] Samuel Madden, Robert Szewczyk, Michael J. Franklin and David Culler, *Supporting Aggregate Queries over Ad-Hoc Wireless Sensor Networks.*, IEEE, 2002.