

# Energy Efficient In-Network Data Processing in Sensor Networks

Prakash G L, Thejaswini M, S H Manjula, K R Venugopal, L M Patnaik

**Abstract**—The Sensor Network consists of densely deployed sensor nodes. Energy optimization is one of the most important aspects of sensor application design. Data acquisition and aggregation techniques for processing data in-network should be energy efficient. Due to the cross-layer design, resource-limited and noisy nature of Wireless Sensor Networks (WSNs), it is challenging to study the performance of these systems in a realistic setting. In this paper, we propose optimizing queries by aggregation of data and data redundancy to reduce energy consumption without requiring all sensed data and directed diffusion communication paradigm to achieve power savings, robust communication and processing data in-network. To estimate the per-node power consumption POWER-Tossim mica2 energy model is used, which provides scalable and accurate results. The performance analysis shows that the proposed methods overcomes the existing methods in the aspects of energy consumption in wireless sensor networks.

**Keywords**—Data Aggregation, Directed Diffusion, Partial Aggregation, Packet Merging, Query Plan.

## I. INTRODUCTION

RECENT advances in science and technology have led to the production of cost effective chip consisting of number of transistors. Processing capacity of chip is exponentially growing every year. These advances have led to the production of cheaper and smaller mechanical structure, battery-powered, sensing, processing and computing wireless sensors. As the technology is advancing the size of the sensors are available in smaller size. These sensors sense the field and forces in environment where they are deployed.

Figure 1 illustrates the generic architecture of a sensor node. It is composed of a power unit, processing unit, sensing unit and communication unit. The processing unit is responsible to collect, process signals captured from sensors and transmit them to the network. The processing unit is used to compute and process the data locally. Sensors are devices that produce a measurable response to a change in a physical condition like temperature and pressure. The wireless communication channel provides a medium to transfer signals from sensors to external world or to a computer network and helps to establish and maintain wireless sensor network which is usually ad-hoc. Advances in Micro Electro Mechanical System (MEMS) technology and its associated interfaces, signal processing and Radio Frequency (RF) circuitry have enabled the development of wireless sensor nodes.

Prakash G L, Thejaswini M, S H Manjula and K R Venugopal are with the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore 560 001, e-mail: glprakash@yahoo.com.

L M Patnaik is a Vice Chancellor, Defence Institute of Advanced Technology (Deemed University), Pune, India.

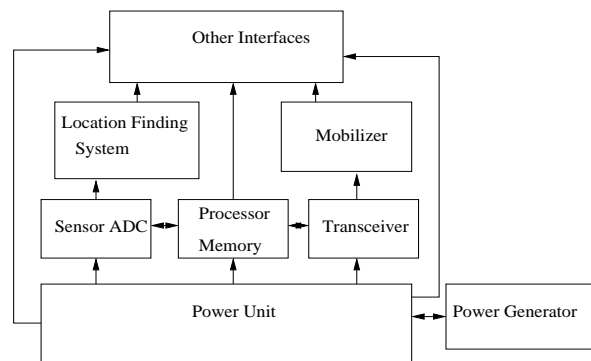


Fig. 1: Sensor Node Hardware

Each of the sensor is a separate data source that consists of node-id, location, time stamp, sensor type and the value of the reading. Sensor data might contain noise; it is not often possible to obtain more accurate results; but it is possible to obtain accurate results by fusing data from several sensors. Aggregation of raw sensor data is thus more useful in sensor applications than individual sensor readings. For example, when monitoring the pressure of a fluid flow in an industry, one possible query is to measure the average value of all sensor readings in that region, and report whenever it is higher than some predefined threshold.

Applications of sensors can be viewed as environmental, habitat monitoring, agriculture, intelligent systems, medical field, disaster management and object tracking. While designing sensor networks, resource constraints such as power consumptions, communication bandwidth, computing power, memory size and uncertainty in sensor readings are important parameters. Energy optimization is a paramount issue in wireless sensor networks as the sensor nodes are expected to have a few years of lifetime. The collection of information from sensors must be carefully managed with limited power and radio bandwidth [1].

Aggregation in sensor network is a very fast-evolving field. Two sensors geographically to each other to produce similar values. Similarly, a single sensor which is continuously monitoring a physical variable typically produces a stream of values which are correlated in time. Aggregation algorithms which exploit such correlations can significantly cut down

the amount of processing and communication. Acquiring and processing such event data can itself be a challenge.

*Motivation* : Each sensor in a network takes time-stamped measurements of physical phenomena. A sensor network database consists of sensor data. Maintenance of the database generate several new challenges. The challenges are: (1) The high energy cost of communication encourages in-network processing during query execution. (2) Limited storage on nodes and high communication costs imply that older data has to be discarded. The database system can try to maintain more high-level statistical summaries of the detailed information. (3) Additional operators have to be added to the query language to specify durations and sampling rates for the data to be acquired.

In-network data processing techniques improve the energy efficiency, which is a typical measure of performance of sensor networks. Sending all raw data to the sink node consumes more energy. Onboard processor of a sensor node carry out computation locally, reducing the power consumption of radio communication. In this paper, to evaluate the energy efficiency of in-network data processing approaches and data dissemination method called directed diffusion are proposed.

*Contribution* : To propose a solution to evaluate queries, query optimization for specific types of queries and the data routing approaches such as multihop ad-hoc distance vector routing and the directed diffusion method for data dissemination and processing in sensor networks. In a simulation study the performance of in-network data processing approach and the performance of different query plans are compared. In-network data processing techniques improve the energy efficiency; a typical parameter measure of performance in sensor networks.

*Organization* : The organization of the rest of the paper is as follows. Section II gives related work, Problem formulation and Detailed system design is presented in Section III and Section IV respectively; Detailed algorithm is developed Section V; Analysis of in-network data processing approaches and comparison of simulation results are given in section VI; Finally, Section VII contains conclusions.

## II. RELATED WORK

The evolution of sensor networks, challenges and opportunities is presented in [2]. A Survey of number of data processing methods, communication architectures and the features influencing the sensor network design have been described in [3]. Intanagonwiwat et al., [4] has proposed a directed diffusion paradigm and a Robust scalable communication to achieve energy savings by selecting the empirically good paths and by caching and processing data in-network. The disadvantage is that data is processed individually in the network and it consumes high energy.

Polastre et al., [5] has proposed data collection models in which data is stored in traditional databases and can be queried using standard techniques. Such a data collection model is easy to deploy but short lived when high data rate sensors are used, since the data communication requirements overwhelm the available energy resources.

The TinyDB and Cougar Operating System proposed in [6], [7] are equipped with query processing engine in which a user injects (in an extended SQL) a query at the sink node. Upon receiving the query, the sink node collects data from all nodes participating in the query. Based on the collected data, the sink node generates a single query plan that defines the sequence of data to be collected from its sensors which consumes high energy and it reduces the lifetime of a network.

The COUGAR [8] and query processing [6], [9] focuses on executing queries over sensor and stored data. Sensors are represented as a new data type, with special functions to extract sensor data when requested. COUGAR addresses scalability (increasing numbers of sensors) by introducing a virtual table where each row represents a specific sensor. The COUGAR system inspired many ideas in the early design phases of Nile, specifically, the stream data type and the table representation of streams. Seshadri et al. [10] presented the sequential model and implementation for sequence databases. A sequence is defined as a set with a mapping function to an ordered domain. Sequence databases is included in the extension of SQL:1999, which supports the notion of window queries over static data streams.

The challenge of maximizing the data collection from energy-limited store and extracts WSNs is examined in [11]. Tian He et al., [12] explained on the trade-off between energy awareness and surveillance performance by adaptively adjusting the sensitivity of the systems in WSNs. Mathew et al., [13] propose bootstrapping as a possible phase for energy saving in which the entities of the network are made aware of all or some of the other entities in the network. It aims at saving energy by reducing the number of collisions and turning off radio. The disadvantage is that it requires nodes to be highly synchronized.

Sabbini et al., [14] presented a new dissemination protocol for data collection in WSN. It uses location information to reduce redundant transmissions, thereby saving energy. Virtual grid formation is used to achieve location aided flooding. This reduces the redundant transmissions of same packet by a node resulting in energy saving. TOSSIM [15] provides a scalable simulation environment for sensor networks based on TinyOS [16]. Unlike machine level simulators, TOSSIM compiles a TinyOS application into a native executable that runs on the simulation host. This design allows TOSSIM to be extremely scalable, supporting thousands of simulated nodes. Deriving the simulation from the same code that runs on real hardware greatly simplifies the development process. TOSSIM supports several realistic radio-propagation models and has been validated against real

deployments for several applications.

Kalpakis et al., [17] have formulated the maximum-lifetime data-gathering problem has a linear programming formulation by taking data aggregation in to consideration and presented a polynomial-time algorithm to solve the problem. although this optimization framework yields satisfactory performance it makes the simplistic assumption of perfect data correlation, where intermediate sensor node can aggregate any number of incoming packets into a single packet. A perfect data correlation can also be found in [18], which analyzes the performance of data-centric routing schemes with in-network aggregation.

### III. PROBLEM DEFINITION

Given a WSN of size  $N$ , where  $(n_i, n_j)$  are connected if both the nodes  $i$  and  $j$  and the network model is connected graph  $G(N, E)$  where the node  $n_i$  and  $n_j$  are connected iff they are able to communicate and transmit data among themselves, the objectives are

- To improve a data processing method to reduces the data size.
- To improve a communication model to lower the number of transmissions.
- To reduce energy by sending the data to be transferred to the basestation.

#### A. Assumptions

- 1) A query issued in an environment typically specifies sensing types(photo, light, temperature, location, acceleration, magnitude), source node, set of predicates and sample period.
- 2) Every node holds a symmetric connectivity list of its neighbours.
- 3) Every node maintains a black list of neighbours of insufficient connectivity. All packets from or to a black node are dropped.
- 4) Every node holds an interest cache and a data list.
- 5) All nodes have similar capability and equal significance.
- 6) Each of the node is battery operated and fixed residual energy level.

#### B. Example

Consider the following example, where an average reading is computed over a network of six nodes arranged in a three-level routing tree in Figure 2. In the server based approach, where the aggregation occurs at an external server, each sensor sends its data directly to the server. This requires a total of sixteen message transmissions. Alternatively, each sensor may compute a partial state record, consisting of (sum, count), based on its data and that of its children, if there are any. This requires a total of only six message transmissions to server.

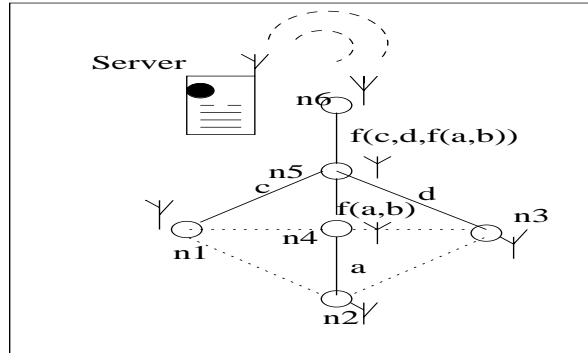


Fig. 2: In-network Aggregation at Nodes

TABLE I: NOTATIONS

Symbols	Definition
$N$	Number of nodes in the network
$E$	Number of Edges
$x, y$	Location of the node
$C1, C2$	Constants
$T_{energy}$	Total Energy
$A_{energy}$	Average Energy
$E_{ctrans}$	Cost for the Transmission
$E_{tx}$	Transmitter Energy
$s$	Packet size
$d$	Average distance between any two nodes
$E_{amp}$	Amplifier Energy
$E_{cpu}$	CPU Energy
$E_{adc}$	ADC Energy

In-network aggregation and query processing typically involve query propagation and data aggregation. To push query to every node in a network, an efficient routing structure have to be established. Transmitting all raw data to the sink nodes consumes more energy than pushing computation into the network. It requires different optimizing techniques for in-network data processing in sensor networks.

### IV. SYSTEM DESIGN

#### A. Network Architecture

A sensor network is modeled as a connected graph  $G(N, E)$ , where sensor nodes are represented as the set of vertices  $N$  and wireless links as the set of edges  $E$ .

Consider a scenario where several sensors that are deployed in a remote region have completed their sensing task and have some locally computed data. They are interested in collecting the required data possible from all these sensors at a sink node then to end user. Given some energy constraints in each of these sensors. Figure 3, shows a sample scenario with six source nodes, one sink node(node 0). Each node is labelled with its  $(x, y)$  coordinates, its available data and energy. The

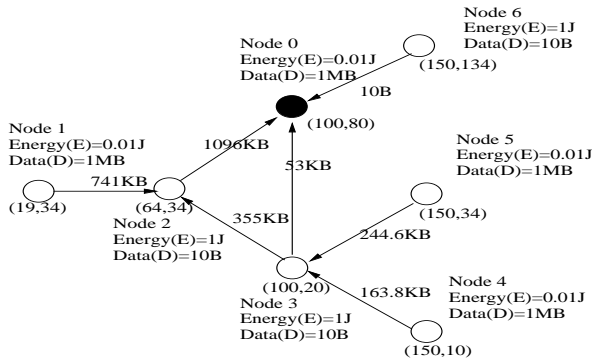


Fig. 3: Sensor Node Deployment

goal is to extract the data to the sink node. The arrows indicate the direction of data sent.

### B. Query Model

The order in which a node samples its sensors conventionally referred to as a query plan, This can be a crucial factor affecting the energy consumed by the sensor network. Such orderings for the nodes involved in a query are an essential part of query plan [19]. The data collected by the sink node can be used to determine energy-efficient query plans for the nodes participating in the query. It is important to note that the cost of determining the optimal query for a node depends on the complexity of the query. While for simple queries, a node may itself be able to derive the optimal query plan by spending a small amount of energy or memory, for complex queries, it might be desirable to delegate this task to the energy or memory rich sink node. Figure 4, shows the query for the sink node, which contains an AVG operator to compute the average value over all sensor readings and SELECT operator that checks if the result is above threshold.

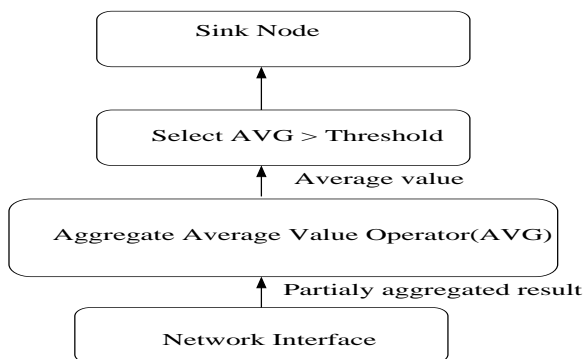


Fig. 4: Query plan at the Sink Node

The goal of the In-network query workload design is to reveal the performance characteristics of in-network query

processing techniques. The query workload as follows.

1) *In-Network Data acquisition query workload:* In this section, the different types of query plans are presented. In the workload, Q1-Q4 are data acquisition queries.

#### Q1: Single Sensory Attribute Projection

```
SELECT node_id, photo
FROM sensors
```

In this query plan the workload is just to select sensor photo readings.

#### Q2: Projection of Multiple Sensory Attributes

```
SELECT nod_id, photo, temperature, acceleration,
magnitue
FROM sensors
```

In this query multiple attributes such as photo, temperature, acceleration and magnitude readings in x and y directions are selected.

#### Q3: Single Sensory Attribute Projection and Selection

```
SELECT node_id, photo
FROM sensors
WHERE light ≥ C
```

Q3 studies the performance of selection queries on a sensory attribute. In comparison with Q1, this query adds a WHERE clause with a selection predicate on the projected light sensory attribute. In each epoch (sample interval) of the query, only those nodes whose recent photo readings satisfy the predicate will send out their data towards the sink even though all nodes in the network acquire their own light readings. The set of nodes that satisfy the predicate may vary from epoch to epoch depending on the data. The parameter  $C$  in the predicate is a user-specified constant value. It can be changed to achieve different selectivities of the predicate.

#### Q4: Conjunctive Selection on Multiple Sensory Attributes

```
SELECT node_id, photo, temperature
FROM sensors
WHERE photo ≥ C1
AND temperature ≥ C2
```

The query condition of Q4 is the conjunction of multiple selection predicates on sensory attributes. This query is used to investigate the predicate ordering issue in query evaluation. The number of predicates involved in the selection condition can be increased as necessary.  $C1$  and  $C2$  are two user specified constant values. Instead of sending all the raw reading query plan can be optimized by sending only readings which qualifies the criteria. Here the query condition is checked locally at the sensor nodes. The packets are transmitted only if the conditions are true.

2) *Aggregation Query Workload:* In this section, present the four SQL queries in the current version of query workload.

In the workload, Q5-Q7 are aggregation queries. All queries in the workload are continuous queries.

*Q5: Duplicate-Insensitive Simple Aggregation*

```
SELECT MAX(photo)
FROM sensors
```

Q5 tests the performance of the aggregation schemes for duplicate-insensitive aggregates. All nodes in the network participate in the aggregation process.

*Q6: Duplicate-Sensitive Simple Aggregation*

```
SELECT SUM(photo)
FROM sensors
```

Q6 tests the performance of the aggregation schemes for duplicate-sensitive aggregates. The duplicate-sensitivity of the aggregate requires extra effort in multi-path routing in order to ensure the correctness of query results.

*Q7: Aggregation with Sensory Attribute Selection*

```
SELECT AVG(photo)
FROM sensors
WHERE photo ≥ C
```

In comparison with Q5 and Q6, Q7 adds a selection predicate on the aggregation attribute. The predicate selects a subset of the nodes in the network to participate in the aggregation and this subset may change over epochs of the query depending on the data.

Here the data is processed locally. Here photo sensor readings are periodically sampled and compute the average of recent raw samples. To route the computed average values to the sink node we can use multihop protocol. A packet is forwarded by internal nodes along the route until the packet reaches its destination. Sensor nodes are limited by the transmission power of the wireless radio. In addition its limited communication channel and frequent topology changes make the sensor networks quite unstable. Routing protocols are required to overcome these limitations [20], [21].

### C. In-Network Data Processing

Data stored in sensor networks can be viewed as local, external and data centric. In local storage, data is stored on nodes locally; to retrieve data a query floods the network. In external storage, data is sent to sink node without waiting for the query. In data centric storage all communication is for named data.

*1) Broadcasting Query Message:* This is the simplest scheme. Sink node broadcast query message(BQ). Each source sensor node sends a data packet consisting of a record towards the sink. Computation will only happen at the sink after all the records have been received. This may consume more power to communicate with far nodes and computation at sink node.

*2) Processing Data Locally:* Instead of sending all the data to the sink node, send the locally processed data to the sink which will optimize the power consumption and communication radio energy, *e.g.*, instead of sending all the raw temperature readings, we send partially aggregated(PA) data such as average of every seven readings from intermediate node and send it to the sink for further processing.

*3) Packet Merging:* In Packet Merging(PM), instead of sending each sensor readings separately in a packet we can merge several records into large packet, consisting of many readings. Packet merging is the only way to reduce the number of bytes transmitted. This will save power consumption of source node and reduces the computation cost of sink node.

### D. Communication Paradigm for Sensor Networks

*1) Traditional Ad-hoc On-Demand Distance Vector Routing:* The Ad-hoc On-Demand Distance Vector Routing(AODV) stack has slightly different requirements than a Traditional Ad-hoc On-Demand Distance Vector Routing(TAODV) algorithm. It is a reactive algorithm, so it builds routes on demand when desired by source nodes. A source node desiring a route to the destination generates and broadcasts a route request (*RREQ*) message across the network. When the *RREQ* arrives at the destination or an intermediate node with the path to the destination, a route reply (*RREP*) message is generated and propagated along the reverse path. The nodes propagating the *RREP* back to the source add a route entry for the destination. *RREP* messages are only generated by the destination. No messages are generated to keep routes active because routes never expire. Route errors are generated when a data message can no longer be sent over the path. Using TAODV also reduces power consumption by routing data using multihop method.

*2) Directed Diffusion:* Applications of directed diffusion(DD) involve various types of sensors and sensor data and customizable in-network aggregation and processing (Filtering). Directed diffusion is a data centric in that all communication is named data. Here the sink node sends out interest, which is a task description to all sensors. Both data requests and data responses are composed of data attributes that describe the data. Each piece of the subscription/publish (an attribute) is described via a *key-operator-value* triplet. Key indicates the semantics of the attribute (latitude, frequency, etc.). Keys are simply constants (integers) that are either defined in the network routing header or in the application header. Allocation of new key numbers will be done with an external procedure to be determined. Operator describes how the attribute will match when two attributes are compared. Value has some type and contents. Some values also have a length (if its not implicit from the type). Each node stores interest in its cache, which contains a timestamp field and several gradient fields.

As the interest is propagated throughout the sensor network, the gradients from the source back to the sink are set up. When the source has data for the interest, the source sends the data along the interest gradient path. The interest and data propagation and aggregation are determined locally. The sink must refresh and refine the interest when it starts to receive data from the source. Directed diffusion is implemented using oneway pull assuming every node holds a symmetric connectivity list of its neighbours. A node maintains a Black List of neighbours of insufficient connectivity. All packets from or to a Black node are dropped. Every node holds an interest cache and a data list.

### E. Energy Model

To process a query, each source node samples its sensed data and checks if resulting readings satisfy the relevant predicates. To estimate the power consumption of per-node energy consumption Mica2 energy mode is used.

The total energy consumed  $T_{energy}$  is the sum of energy consumed by  $RADIO(E_{radio})$ ,  $CPU(E_{cpu})$ ,  $LEDs(E_{leds})$ ,  $ADC(E_{adc})$ ,  $MEMORY(E_{memory})$  and  $VOLTAGE(E_{voltage})$ .

The values of  $E_{leds}$  and  $E_{adc}$  are insignificant, then Equation becomes

$$T_{energy} = E_{radio} + E_{cpu} + E_{memory} + E_{voltage} \quad (1)$$

The average energy consumption  $A_{energy}$  of a node is given by sum of total energy  $T_{energy}$  by number of nodes  $N$

$$A_{energy} = \sum T_{energy}/N \quad (2)$$

The cost for transmitting data  $E_{ctrans}$  in terms of packet size  $s$ , the distance between the sender and receiver  $d$  can be formulated .

$$E_{ctrans} = s * E_{tx} + s * E_{amp} * d^2 \quad (3)$$

where  $E_{tx}$  is the cost for using the transmitter (i.e., the bit cost for the transmitter electronics) and  $E_{amp}$  for the amplifier cost.

## V. ALGORITHM

Data is exchanged when there are matching between subscriptions and publications. Algorithm for matching rules is given in Table II. Since diffusion is based on the core concept of subject-based routing, it is very important to make sure attributes in publications, subscriptions and filters match. For both Publish/Subscribe and Filters interfaces, matches are determined by one way match applying the following rules between the attributes associated with publish (P) and subscribe (S).

*Subscribe* : Each subscription causes Diffusion to send an interest message to the network. These interest messages are broadcast throughout the network. Figure 5(a), shows interest message broadcast. On arrival of an interest message

TABLE II: DATA MATCHING RULES

```

Data_Match( $S_a, P_a$ )
//  $S_a$  is a set of Subscribe Attribute
//  $P_a$  is a set of Publish Attribute
//  $S_a.op$  is a Subscribe Operator
//  $S_a.key$  is a Subscribe key
//  $P_a.value$  is a Publish Value
begin
  for every attribute  $S_a \in S$  and any
  operator  $S_a.op$ 
  begin
    for every attribute  $P_a \in P$ 
    begin
       $S_a.key = P_a.key$ 
       $P_a.value$  satisfies  $S_a.op$ 
      if (none exists)
        exit(no match)
      else
         $S$  matches  $P$ 
    end
  end
end
end

```

to a node, it is matched against the interest cache. Duplicate interests are dropped. An interest gradient is set in the interest cache based on first arriving interest which is shown by Figure 5(b). When an interest arrives to publishers with matching data, a simple hop-by-hop route is set up from the publisher to the subscriber.

*Publish* : A publisher sends a data message in reply to an interest or reinforcement. From a publisher point of view there is no difference between an interest and reinforcement. Periodically, a publisher compares its data list to its interest cache. Matching data is aggregated and send in a data message. Data messages are sent only through interest gradients of unique neighbours. On arrival of data message to a node, it is first matched against a data list. Duplicate data messages are dropped. Later, it is matched against the interest cache. Matching data message is forwarded down stream through interest gradients of unique neighbours. On a match the data gradient list is updated. Figure 5(c), shows the data delivery path of matching data.

*Filter* : On data message arrival it's first matched against all subscribed filters. On a match a copy of the interested data is forwarded to the filter. The data then is matched against other interests. The filter can decide if to drop a data message or forward it down stream modified or unmodified. Diffusion allows for aggregation of data, thus multiple attributes of the same kind can arrive at the same attributes array. The application layer is responsible to extract and verify multiple arriving data since as long as at least one match of data to an interest is attained, the data will be forwarded to the sink.

The temperature reading task can be described as an interest e.g.,

Attribute key temperature  
Operation equal

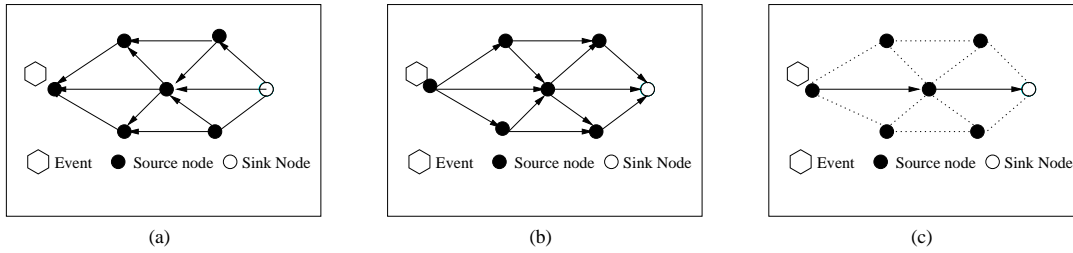


Fig. 5: A simplified schematic for directed diffusion

Attribute value 40  
Interval 20 ms  
Duration 10 seconds

The data sent in response to the above interest are also named using the similar scheme, *e.g.*,

Attribute key temperature  
Sensor\_id 2  
Attribute value 40  
Timestamp 01:23:42

## VI. PERFORMANCE EVALUATION

Simulation results performed on a test bed using TOSSIM simulator for TinyOS. Using PowerTOSSIM to estimate the total energy consumption of in-network data processing approaches. To estimate the power consumption of the mica2 sensor node mica2 energy mode is used. Table III, shows energy dissipation for mica2 mote .

TABLE III: THE ENERGY DISSIPATION OF OPERATIONS FOR MICA2

Operation	Energy Dissipation(mA)
CPU Active	8.93
CPU Idle	4.13
CPU ADC Noise Reduction	1.0
CPU Power Down	0.103
CPU Power Save	0.110
CPU Standby	0.216
CPU Initialization	3.2
Radio Default Power	15.00
EEPROM Read	6.24
EEPROM Write	18.40

Suppose a sensor is operating at 3 Volts and capable of transmitting data at a rate of 40 *Kbps* at 0.012 *Amp* transmit current draw. Hence, the energy cost of transmitting ( $TE_{trans}$ ) one bit in *Joules* is computed as  $TE_{trans} = 3 * 0.012 * (1/40,000) = 0.9\mu\text{Joules}$ .

### A. Simulation Setup

In this section, simulation studies are compare the performance of the packet broadcasting, packet merging with packet aggregation and the Directed Diffusion with TAODV methods with respect to its lifetime using TOSSIM simulator

in windows operating system. Different number of Sensors are randomly distributed in a query region over 100m x 100m area. The Simulation is run for 60 seconds, and each simulation run for different network size. The simulation parameters for query processing and directed diffusion are listed in Table IV and Table V respectively.

TABLE IV: SIMULATION PARAMETERS FOR QUERY PLAN

Parameter Type	Test Value
Number of nodes	5,20,50,65,75,85,100
Sink node	Mote 0
Radio model	Lossy
Distance scaling factor	1.0 with empirical radius
Simulator hardcoded	4Mhz
Epoch Period	1000ms-10000ms
Aggregate operations	SUM,AVG,MAX
Sensor type	Photo sensor, Temperature sensor, Demo sensor, Accelerometer sensor, Magnetometer sensor

### B. Performance Analysis

From the simulation results, Figure 6, illustrates the performance analysis of a simple query(SQ) of sensing photo reading above some threshold value and increased workload

TABLE V: SIMULATION PARAMETERS FOR DIRECTED DIFFUSION

Parameter Type	Test Value
Number of nodes	5,20,50,65,75,85,100
Sink node	Mote 1
Radio model	Lossy
Distance scaling factor	4 with empirical radius
Maximum interest	10
Maximum gradients	2
Maximum gradients overrides	4
Maximum attributes	4
Maximum Data	25(data cache size)
Time to live	10
Timer period(msec)	125
Timer tics per second	1000 / Timer period(msec)
Interest sender period	5
Interest expire time(seconds)	15

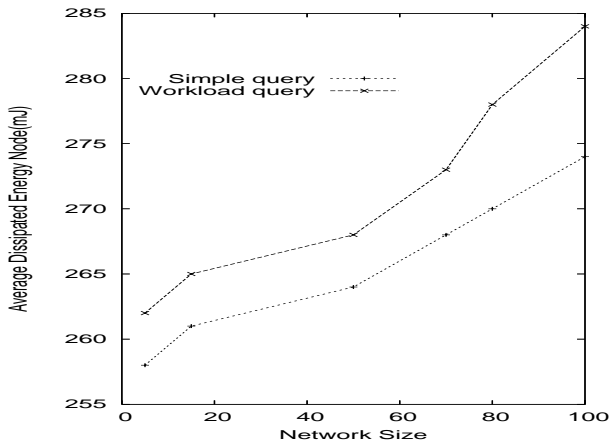


Fig. 6: Average Dissipated Energy versus Network Size for different query type

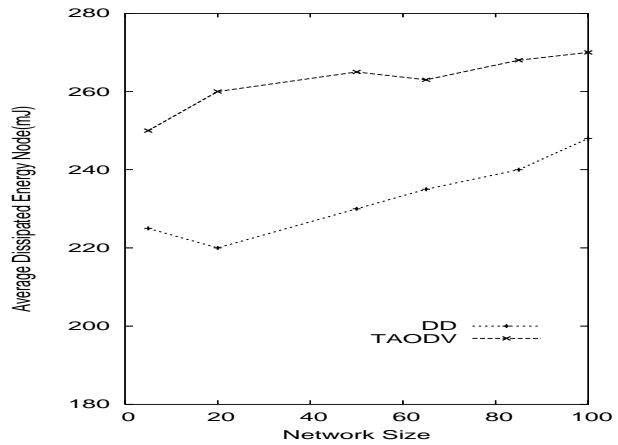


Fig. 8: Average Dissipated Energy for Multihop Ad-Hoc Routing versus Directed Diffusion

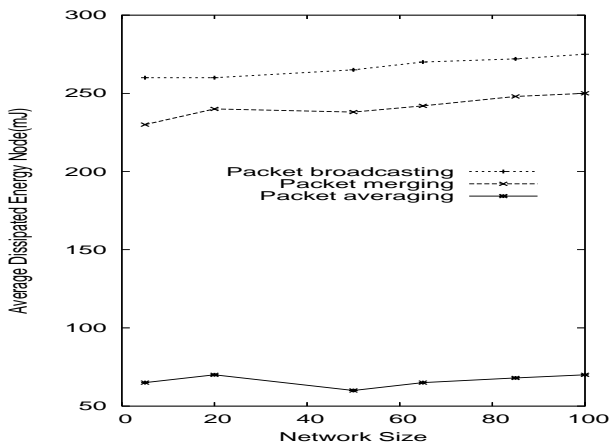


Fig. 7: Average Dissipated Energy for In-Networks Data Processing Techniques

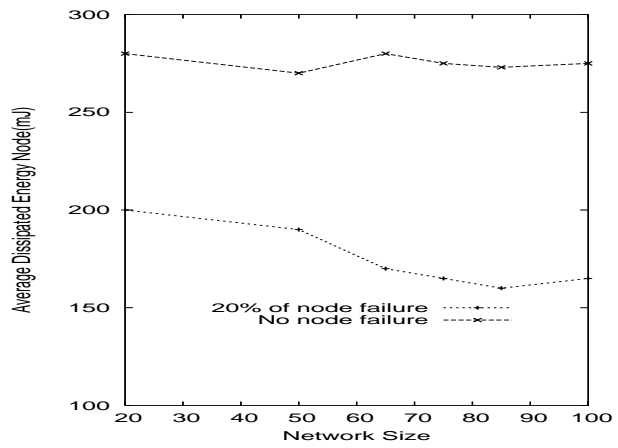


Fig. 9: Average Dissipated Energy for Node Failure on Directed Diffusion

query of detecting photo, temperature, accelerometer and magnetometer in  $x$  and  $y$  directions, and all readings above some threshold values which influences the performance metrics such as lifetime of the network. Energy consumption for sparse networks is increases linearly and for dense networks simple query increases faster than workload query.

Figure 7 illustrate the variation of average dissipated energy per node with different network size. This figure compares the energy dissipation of data processing techniques such as packet broadcasting messages, processing data locally that is partially aggregating values on local nodes, and packet merging. Without in-network data processing, each node has to send a data packet for each node whose route goes through  $n$  number of nodes, so energy consumption increases very fast. Packet broadcasting consists of all raw data, consumes more energy Packet merging consumes less energy than packet broadcasting as it consists of several sensor readings merged in a packet. Packet aggregation in in-network data processing method consumes less energy compared to other

methods, it reduces redundancy in sensor readings.

In Figure 8, compares the directed diffusion (DD) with multihop Traditional Ad-hoc On-Demand Distance Vector Routing (TAODV) scheme for data dissemination in sensor networks. This figure shows that the average dissipated energy per node as the function of network size. Directed diffusion is scalable and robust data dissemination and processing approach consumes less energy than multihop ad-hoc distance vector routing.

From Figure 9, at any instant, In 10 to 20 percent of the nodes failures, Directed diffusion is able to maintain reasonable event delivery. The average dissipated energy actually improves in the presence of node failures. But it is also expected that directed diffusion would expend energy to find alternative paths. In addition, diffusion benefits significantly from in-network aggregation. Intermediate nodes suppress duplicate packet estimation. Figure 10 shows that diffusion expends nearly three times as much energy in



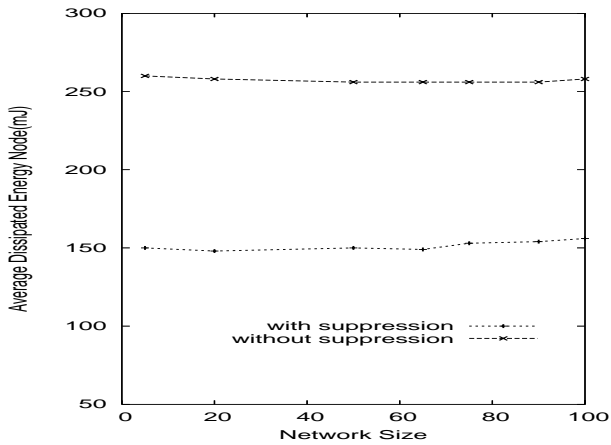


Fig. 10: Average Dissipated Energy for Duplicates Suppression

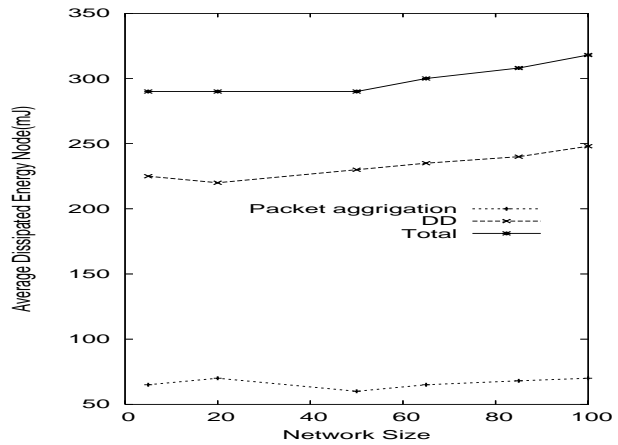


Fig. 12: Total Energy of Network using Packet merging and DD

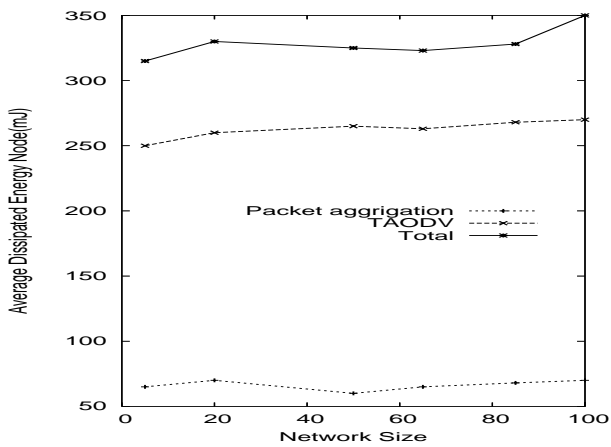


Fig. 11: Total Energy of Network using Packet merging and TAODV

TABLE VI: NETWORK LIFE TIME

Nodes	Average Energy(mJ)					
	Processing				Communication	
	SQ	BQ	PM	PA	TAODV	DD
5	212	260	230	65	280	225
15	265	260	240	70	260	220
50	268	265	238	60	265	230
70	273	270	242	65	263	235
80	278	272	248	68	268	240
100	284	275	250	70	270	248

smaller field, as when it can suppress duplicates. In larger sensor field, the ratio is 2.

Figure 11 and Figure 12 are the results, showing network lifetime; average dissipated energy for varying network densities; calculates the lifetime of network; compares TAODV with DD. It is observed that the life time of the network increases when packet averaging used for data processing and DD used for communication, which reduce the number of transmissions

The analysis of various in-network data processing and communication methods with respect to average dissipated energy as shown in the Table VI. In all methods as anticipated, the packet averaging for data processing and directed diffusion could significantly reduces energy consumption in sensor networks.

## VII. CONCLUSIONS

The key point of this paper is to stress the need for a simulation framework for data processing and communication algorithms in sensor networks from data generation to network simulation. The Energy optimization techniques are proposed such as in-network data processing methods such as, query optimization plans, processing data locally and packet merging, and communication paradigm directed diffusion. As compared to the existing data processing and communication methods, our approaches are more effective to minimize the total processing and transmission energy consumed by the network.

Future challenges include running queries from multiple users for long time over a sensor network, sharing the resources among the queries to balance and minimize overall resource usage.

## REFERENCES

- [1] Weifa Liang and Yuzhen Lin, "Online Data Gathering for Maximizing Network Lifetime in Sensor Networks," in *IEEE Transactions on Mobile Computing*, vol. 6, pp. 2–11, January 2007.
- [2] Che-Yee Chong and Srikant P. Kumar, "Sensor Networks: Evolution, Opportunities and Challenges," in *Proceedings of the IEEE*, vol. 91, pp. 1247–1256, August 2003.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," *IEEE Communication Magazine*, vol. 40, pp. 102–114, August 2002.
- [4] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *IEEE/ACM Transactions on Networking*, vol. 1, pp. 56–67, February 2000.
- [5] J. Polastre, G. Bolriello, L. Girod, and R. Han, "Applications: Beyond Dumb Data collections(panel)," in *EmNets*, 2006.
- [6] Y. Yao and J. Gehrke, "Query Processing for Sensor Networks," in *First Biennial Conference on Innovative Data Systems Research, Asilomar, CA*, pp. 21–32, 2003.
- [7] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The Design of an Acquisitional Query Processor for Sensor Networks," in *ACM SIGMOD International Conference on Management of Data*, pp. 491–502, June 2003.
- [8] Yong Yao, Johannes Gahrke, "The Cougar Approach to In-Network Query Processing in Sensor Networks," in *SIGMOD Communication Magazine*, vol. 31, pp. 9–18, September 2002.
- [9] Johannes Gehrke and Samuel Madden, "Query processing in sensor networks," *IEEE Pervasiv computing Magazine*, vol. 40, pp. 46–55, JANUARY-MARCH 2004.
- [10] P. Seshadri, M. Livny, R. Ramakrishnan, "The Design and Implementation of a Sequence Database System," in *VLDB'96, Proceeding of 22nd International Conference on Very Large Data Base*, September 1996.
- [11] Narayanan Sadagopan and Bhaskar Krishnamachari, "Maximizing Data Extraction in Energy-Limited Sensor Networks," *IEEE INFOCOM*, September 2004.
- [12] Tian He, Sudha Krishnamurthy, John A. Stankovic, Tarek Abdelzaher, Liquian Luo, Radu Stoleru, Ting Yan and Lin Gu, "Energy-Efficient Surveillance System using Wireless Sensor Networks," in *ACM Transactions on Mobisys*, June 2004.
- [13] Rajesh Mathew, Mohamed Younis and Sameh M. Elsharkawy, "Energy-Efficient Bootstrapping for Wireless Sensor Networks," in *Innovations System Software Engineering*, 2005.
- [14] Harshavardhan Sabbineni and Krishnendu Chakrabarty, "Location-Aided Flooding: An Energy-Efficient Data Dissemination Protocol for Wireless Sensor Networks," in *IEEE Transactions on Computers*, vol. 54, pp. 36–46, January 2005.
- [15] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System Architecture Directions for Networked Sensors," in *Architectural Support for Programming Languages and Operating Systems*, pp. 93–104, 2000.
- [16] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2003.
- [17] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Efficient Algorithms for Maximum lifetime Data gathering and Aggregation in Wireless Sensor Networks," in *Comput. Netw. J.*, vol. 42, pp. 697–716, August 2003.
- [18] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," in *Univ. Southern California Comput. Eng.*, pp. 02–14, 2002.
- [19] Qiong Luo, Hejun Wu, Wenwei Xue, Bingsheng He, "Benchmarking In-Network Sensor Query Processing," in *Technical Report HKUST-CS05-09, Department of Computer Science, HKUST*, June 2005.
- [20] David Braginsky and Deborah Estrin, "Rumor Routing Algorithm for Sensor Networks," in *ACM International Conference on Mobile Computing*, pp. 22–31, September 2002.
- [21] Hock Guan Goh, Moh Lim Sim and Hong Tat Ewe, "Energy Efficient Routing for Wireless Sensor Networks with Grid Topology," in *IFIP International Federation for Information Processing*, pp. 834–843, 2006.



**Prakash G L** is an Assistant Professor with the Department of Computer Science and Engineering of Alpha College of Engineering, Bangalore, India. He received his B.E and M.E degrees in Computer Science and Engineering from Bangalore University, Bangalore. He is presently pursuing his Ph.D programme in the area of Wireless Sensor Networks in Bangalore University.



**Thejaswini M** is a Faculty with the Department of Computer Science and Engineering of Sri Jagadguru Mallikarjuna Murugharajendra Institute of Technology, Chitradurga, India. She received her B.E in Information Science and M.E degrees in Computer Science and Engineering from Visveswaraiiah Technological University and Bangalore University respectively. Her research interests include Wireless Sensor Networks.



**Manjula S H** is an Assistant Professor with the Department of Computer Science and Engineering of University Visveswaraya College of Engineering, Bangalore, India. She received her B.E and M.E degrees in Computer Science and Engineering from Bangalore University, Bangalore. She is presently pursuing her Ph.D programme in the area of Wireless Sensor Networks.



**K R Venugopal** is currently the Principal and Dean, Faculty of Engineering, University Visveswaraya College of Engineering, Bangalore University, Bangalore. He obtained his Bachelor of Engineering from University Visveswaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded Ph.D. in Economics from Bangalore University and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored 27 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++ etc. He has been serving as the Professor and Chairman, Department of Computer Science and Engineering, University Visveswaraya College of Engineering, Bangalore University, Bangalore. During his three decades of service at UVCE he has over 200 research papers to his credit. His research interests include computer networks, parallel and distributed systems, digital signal processing and data mining.



**L M Patnaik** is a Vice Chancellor, Defence Institute of Advanced Technology(Deemed University), Pune, India. During the past 35 years of his service at the Indian Institute of Science, Bangalore. He has over 500 research publications in refereed International Journals and Conference Proceedings. He is a Fellow of all the four leading Science and Engineering Academies in India; Fellow of the IEEE and the Academy of Science for the Developing World. He has received twenty national and international awards; notable among them is the IEEE Technical

Achievement Award for his significant contributions to high performance computing and soft computing. His areas of research interest have been parallel and distributed computing, mobile computing, CAD for VLSI circuits, soft computing, and computational neuroscience.