Energy Efficient Data Management in Wireless Sensor Networks

Kavitha Kayiram
Dept. of CSE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, Telangana, India
kavitha.bits@gmail.com

Rahul Surana
Senior Member Technical
DE Shaw India
Hyderabad, Telangana, India
rahasurana@gmail.com

Dr. R.Gururaj
Dept. of CSIS
BITS-Pilani, Hyderabad Campus
Hyderabad, Telangana, India
gururaj@hyderabad.bits-pilani.ac.in

***Abstract*—In a Wireless Sensor Network (WSN), sensor nodes are small inexpensive sensing and computing devices collaborating to form a network. However, they work with limited power and memory in hard-to-reach locations and extreme weather conditions. In general, a large number of tiny sensor nodes are usually deployed randomly to monitor one or more physical phenomena like- temperature, humidity etc. Due to limited power resources sensing, storage and retrieval of data become critical issues in wireless sensor data management. In this paper we propose a novel data management scheme that facilitates- (i) power efficient sensing, (ii) optimal utilization of storage space and (iii) energy and time efficient query processing in a WSN.Our proposed data sensing and storage schemes are based on sleep-scheduling and collaborative sensing techniques respectively. Next, our query processing scheme involves a *Lookup* technique that exploits the collaborative sensing which is a result of sleep-scheduling of nodes in a WSN. Our simulation results have proved the validity of our proposal.**

***Keywords*—Wireless Sensor Network, data management, collaborative sensing, query processing, sleep-scheduling.**

I. Introduction

A Wireless Sensor Network (WSN) [1] consists of multiple sensor nodes working collaboratively to form a network. Typical applications of WSN include Air-Pollutant Monitoring, Traffic Monitoring, Habitat Monitoring, Green House Monitoring, Waste Water Monitoring, Earth-Quake Detection System, Land-slide Detection system, Industrial applications, Scientific Simulation Systems etc. In a WSN, one of the nodes is designated as the *Base-Station* (BS) that connects the network with the outside world. Each sensor node is powered by a battery and predominantly consists of a processor, sensors (to sense one or more physical phenomena) and a radio communication module. The sensor node makes use of radio communication module for transmitting data to the BS (possibly via other intermediate nodes). In most of the cases the BS has abundant resources like power, memory and computing abilities. The radio communication module is a dominant consumer of energy. In general, replacement or recharge of batteries is almost impossible as their deployment is in remote hard-to-reach locations, and sometimes in extreme weather conditions. Thus, it is desirable to reduce the consumption of energy in order to enhance the lifetime of the network. Hence, efficient management of battery power is desirable in the design of any WSN.

A. Query Processing Model

In a WSN, sensor nodes are widely spread across the area of interest. Each sensor node has a fixed transmission range. Each sensor node can transmit data to its neighboring nodes within this range. The sensor nodes sense data continuously and store in their local memory unit and/or transmit the same to the BS. The queries from external world arrive at the BS. After processing a query, the BS is responsible for pumping the result back to the requester. Generally, in a WSN, all nodes store sensed data locally, and send the same to the BS at some specified time intervals. This is because the storage space available at nodes is limited. When a query arrives at the BS, and if all the data is available at BS cache, then the query can be answered at BS itself, without consulting the nodes. On the other hand if the data is not yet received at the BS, then the BS will send the query to the concerned nodes to extract the data. Such queries are known as ad-hoc queries. In such cases, nodes will execute the query and send the results to the BS which carries out the process of compilation of final results. In another scenario, it is possible that some amount of data is available at the BS and the remaining is at network nodes. To answer such queries, BS has to extract data from local memory of the self and a set of network nodes. To answer these types of queries BS has to perform data extraction using earlier two approaches, and compile final results.

B. Techniques for Energy efficiency

It is desirable to incorporate energy efficient schemes in various operations of the WSN like- sensing, storage, data transmission, computations etc. Some of the important techniques that are adopted for achieving energy efficiency are- Sleep-scheduling [2], Data aggregation [3], Data compression [4], and Load-balancing [5].

According to the Sleep-scheduling technique [6], each sensor node senses data and goes to sleep mode when there is no sensing or transmission activity. During sleep mode, as the node is not expected to perform sensing and communication activities, it would be operating in power saving mode with minimal power consumption towards other essential activities like running the clock. This ensures optimal power utilization at each of the sensor nodes.

When a tree topology is formed for data transmission, while sending data to the BS, each node has to send its own sensed data, and forward the data of its descendents. Thus, the intermediate nodes are always taxed with extra burden of query dissemination [7], and results forwarding as well. Due to this reason, the intermediate nodes closer to the root (BS), and/or having considerably large number of descendants die sooner than the nodes nearer to leaf level, and/or with lesser number of descendants.

In this paper, we propose an efficient data management approach in WSN with graph topology, to increase the power efficiency. Our approach is based on the following three techniques.

1. A novel sleep-scheduling scheme to conserve energy in the network.
2. A collaborative data acquisition scheme to ensure the availability of critical data in case of a disaster.
3. A lookup based data access to increase the efficiency of query processing.

We outline the latest work done so-far in the area of data management in WSN, in Section II. Our proposed data management scheme is described in detail with suitable illustrations in Section III. Then we evaluate the performance of the proposed approach in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

In this paper, as we propose a new data management approach based on sleep-scheduling, collaborative data acquisition schemeand a lookup based query processing, now we give a brief account of the research work done so-far in the fore mentioned areas. Sleep-scheduling of sensor nodes has been a very significant technique to utilize the power at a node in an optimal way. As mentioned earlier according to this approach, a node will go to sleep mode for some time thus consuming minimal power.

In S-MAC protocol [6], each node follows a periodic sleep and listen schedule to avoid idle listening and to reduce the energy wastage. Before going to sleep a node turns off its radio and sets a timer to wakeup itself later. After sleep-time the node wakes up and listens whether any communication is addressed to itself. To ensure collision free communication between nodes, each node uses *Ready to Send* (RTS) and *Clear to Send* (CTS) packets. The node whichever first sends the RTS packet wins the medium and the receiver will reply with a CTS packet. These participating nodes do not follow their sleep schedule until they finish transmission. The neighboring nodes in the network form a virtual cluster and also share their sleep schedule. To synchronize neighboring nodes with their sleep and listen cycles a SYNC packet is used. The SYNC packet contains the address of the sender and the time of its next sleep. The destination node will set its timer accordingly. Even though the source wants to send to a specific node, it can only broadcast and all the nodes in the vicinity listen. This leads to over hearing problem which is wastage of energy. To avoid overhearing, a node can go to sleep when it finds a RTS/ CTS packet for other node. There is a duration field in each transmitted packet, with this information other nodes know how long to keep silent.

The T-MAC protocol in [8] attempts to improve the performance of the S-MAC protocol. Instead of having a fixed sleep schedule like in S-MAC, it has variable sleep schedules. The idea is similar to that of a screen saver just like the screen switches itself to sleep mode when no activation event has occurred for a certain time period. The activation event can be receiving data, timer expiry and so on. In T-MAC, the node in sleep mode periodically wakes up to communicate with its neighboring nodes. One complete cycle of listen and sleep period is called a frame. Once the communication is over the node goes into sleep mode again until the next frame begins. During the sleep mode the radio is turned off so there is no reception for incoming messages. They may be lost. Hence, to avoid such situation, messages are queued at the neighboring nodes. T-MAC also makes use of RTS and CTS packets for collision free communication in the same way as in S-MAC. Every node transmits and listens as long as it is in active mode. Upon no activity, the node goes into sleep mode. All the messages are buffered during the sleep period. T-MAC moves all these messages to a burst at the beginning of next active period and transmits.

As an improvement to the S-MAC, a new protocol known as Pattern-MAC (P-MAC) [9] was introduced. According to this, each node decides its own sleep schedule based on the data transfer needs of self and neighbors. Hence, the sleeping pattern is asynchronous. Each node gets its neighbor information before hand through patterns. Now, the node can put itself into a long sleep for several time frames when there is no traffic in the network with the help of these patterns. Thus, a sleep wake-up pattern is a string of bits where bit 1 indicates that the node wants to stay awake in that slot while bit 0 indicates slot for node sleep. This way pattern exchange is made between neighbors ensuring optimal energy usage in the network. The energy of the network is conserved in P-MAC as a node is continuously in sleep mode and wakes up only when needed. Due to this P-MAC results in higher data throughput and better savings in energy.

Our review of the existing work related to data storage schemes for sensor networks reveal that some intelligent data storage schemes have been introduced so as to minimize the storage burden on individual nodes. According to this, nodes collaborate to decide about the data to be stored with minimal redundancy. If proper access mechanism is devised, this can significantly improve the query processing efficiency. In general, user queries the WSN for data or events like “List all the events whose temperature lies between 50 and 60”.As such adhoc queries are rarely found, the events are stored across the network at individual sensor nodes and are not stored at the BS. In Data-Centric-Storage (DCS) scheme [10], each event is stored in a sensor node, which is the owner of the event. If a query is intended for an event, it is directed to that sensor node. As the sensed events are stored in different nodes, there is non-uniform usage of storage at each of the nodes across the network. Thus, this scheme led to storage hot-spots. This results in irregular storage load distribution across the network. Certain nodes storing sensor events are assigned heavy load all time while other nodes are under-loaded. In Zone sharing (ZS) [11], load of a hot-spot node is distributed among its neighbors. This divides the hot-spot events among multiple sensor nodes leading to energy saving in the network. ZS is a distributed scheme for the decomposition of storage hotspots. ZS presents the solution in the context of the distributed index for multi- dimensional data (DIM) scheme [12]. The benefit of DIM structure is that it stores data with similar attributes nearby and enables energy-efficient query resolution. When a hotspot arises in a set of sensor nodes, those nodes falling on the border of the hotspot will transfer some loads of their zones to some of their less-loaded neighbors. It will send information to notify its neighbors and increase the cost of communication.

A. Limitations of the existing schemes:

Now, we brief on the limitations of the existing schemes mentioned in the preceding discussion. The P-MAC adaptively decides its sleeping cycles based on the self and the neighbors' sensing and transmission activities. Even during critical event it behaves indiscriminately. There is no provision for critical event detection and sensing highly sensitive data by more than one node. The responsibility of detecting critical events pertaining to highly sensitive data is assigned to a single node. And if due to some catastrophic if that single node ceases to work then the event cannot be detected. Thus compromising on the objective of disaster management. The T-MAC protocol has a problem with asymmetric sleeping patterns, where nodes may go to sleep while their neighbors are still in communion with them exchanging messages. The T-MAC doesn't facilitate load sharing and may result in less efficient query processing.

According to the ZS, though a node can distribute its load to its neighbors, it will incur extra overhead due to reconstruction of the DIM structure, which is a result of Zone sharing. As each node stores multiple events, the cost of communicating with neighbors for collecting one event data is relatively high.

B. Proposal

In this work we propose an efficient data management scheme for WSNs based on – (a) a novel sleep-scheduling, (b) collaborative sensing and (c) energy efficient query processing with a *Lookup* mechanism.

In this section, we propose sleep-scheduling technique in which at least two nodes are always active and sensing a phenomenon in a region which is critical for monitoring disasters. In case of a disaster and during that time if one of the nodes dies, we are able to get a notification since the other node is still active and sensing. Hence this approach is best suited for disaster monitoring WSN applications. The earlier sleep-scheduling techniques did not have this feature. Further, in our work, we introduce a new collaborative sensing scheme, according to which we consider that a set of nodes in a region form a group and sense the data in a collaborative way by adopting sleep-scheduling. Due to this data may be stored in more than one node. When a query is posed, instead of searching the memory space of involving nodes, the exact set of nodes that contain the data will be figured out with the help of a *Lookup* structure stored at each node. This reduces the query transmission and processing overheads.

Thus, our proposed data management scheme uses novel techniques in sleep-scheduling, collaborative sensing and query processing. Our experimental works conducted on custom built simulators, have proved the efficacy of our proposed approaches. In the next section, we give a detailed description of our proposed techniques for data management.

Thus, to address the drawbacks in the fore-mentioned schemes, we propose an efficient sleep-scheduling and storage scheme for WSN in Section III. We elaborate on the efficiency aspects of our approach with necessary graphs in Section IV.

III. EFFICIENT DATA MANAGEMENT

Now, we describe our proposed techniques used in data management. First we elaborate on our novel collaborative sensing mechanism. Then we describe the proposed *Lookup* based query processing scheme. All the fore-mentioned techniques are synthesized carefully to derive more comprehensive data management approach for WSNs.

A. Proposed collaborative sensing scheme based on sleep-scheduling

The sensor nodes in a WSN can work collaboratively to- (i) avoid packet collision, (ii) minimize energy wastage in idle listening, and (iii) for load-balancing. Here, our focus is to achieve energy efficiency in sensing activity in the network. Duty-cycling based on sleep-scheduling is one prime concept used in this collaborative working of nodes.

According to this, a node can go to sleep mode for a specified time period, after assigning its sensing responsibilities to one of its neighbors.

Now, we elaborate on our proposed collaborative sensing model, which is best suited for WSNs deployed for monitoring disasters, where we are in need of critical data generated during the time of disaster. Every sensor has a fixed transmission range for communicating with other sensors. Two or more sensors which are able to communicate with each other, and deployed in the close proximity form a group. As the neighbor nodes (nodes in a group) are deployed in the same geographical location, they sense the same data and usually record similar readings for a given physical property. This sensing of the same data by more than one sensors leads to energy wastage. In our approach, each node of a group is paired up with another node of the same group. Once pairs are formed, two nodes of a pair will have same sleep-schedule. Nodes which don't find a pair will follow any one of its neighboring nodes. In a group, pairs will follow duty cycles such that during non-critical period at any point of time only one pair will be sensing. But, whenever the difference between two consecutive readings exceeds the defined threshold value, this indicates the possibility of a disaster and sensing activity becomes critical. Due to this, the monitoring system needs sensed data from all the nodes. To facilitate this, the active pair notifies all other sleeping pairs. After receiving a notification a sleeping pair wakes up and participates in the sensing of data. Thus, during non-critical period when a single pair is active in case of failure of one node, backup data is always available with the other node of the same pair. But, during the critical period data is available with all the pairs. As two neighbors are sensing, sleeping and transmitting in a collaborative manner, we can avoid data collision. This collaborative sensing model helps in minimizing the power consumption during non-critical periods and also facilitates high degree of reliability in the case of criticality. Hence, this model is more suitable for disaster monitoring and also can be used in normal applications.

B. Data Lookup at Sensor node

As the sensor nodes operate in collaborative mode as explained in the preceding discussion, the sensed data in a group is scattered across its group members. So for answering a query, data need to be extracted from more than one node. In our model, a query targeted at a group is received by one predefined node *N*. On receiving the query *N* has to execute the query on the data stored in its own memory and send the result to the BS. But, sometimes a node may not contain any data to answer the given query. If data is not found at a node, this indicates that it is also not there at other node of the pair, and then the query is passed on to the next pair. The same will happen in case a node contains partial results. If data is not available at a node, still a node will waste its resources by executing the query but with no results. This may lead to increased response time and wastage of computational resources at a node. To alleviate this, we introduce a special lookup structure to be maintained at each node. This lookup structure consists of date and time-stamp of the sensed data. This entry of date and time in the lookup confirms about the availability of sensed data in the sensor node memory. In general, a sensor node is queried for sensor readings recorded for a particular time interval. Upon receiving a query, the node first checks for the required timestamps in the lookup structure rather than searching the memory space for the required data. If the date and time are found to be matching node proceeds to search its own storage, else it will stop searching. Further this node doesn't direct the query to its paired node, as it also has the same data. Now, the query is transmitted to the next pair in the group. This way we can reduce the overhead of unnecessary execution of a query at nodes by maintaining a lookup table, which makes query execution efficient.

C. *Putting proposed techniques together*

Our proposed techniques, if put together as a data management approach, are best suited for WSN applications that monitor disasters, where access to sensor data is critical, and hence redundant sensing is essential for ensuring effectiveness. Our proposed collaborative sensing scheme which is based on sleep-scheduling, allows nodes to take a break from sensing activity and go to sleep, during non-critical periods. But on sensing some abnormality in readings, all nodes will wake-up and participate in sensing. This makes sure that when a disaster occurs, critical data can be had from multiple nodes. Even during non-critical periods, data is available at least two nodes in a given group. This approach minimizes the energy utilization in a network while assuring the availability of critical data. The described collaborative sensing scheme allows us to devise a lookup structure to minimize the query processing and response time overheads.

Hence, the techniques proposed when applied to graph topology for routing are more effective and will increase the efficiency with respect to storage space, power utilization, and query processing.

IV. EVALUATION

In this section, we discuss various experiments and their results, conducted to measure the effectiveness of our proposed data management approach. To conduct experiments, we have implemented our proposed scheme on our custom-built simulator. Our simulator is meant for Windows platform and is console-based. This simulator allows us to define the geographical range of the network along with the number of nodes. We can also define the transmission range of each node. Since we plan to conduct experiments to assess the power consumption of our technique, we have also designed our simulator wherein the power allotted to each node can also be defined. The deployment of nodes using our simulator can be either random or predefined. For the simulation purpose we assume the connectivity to be constant.

The simulator was run for networks of sizes 50, 100, 150, 200 and 250 nodes. The input to the simulator is the location of nodes specified by their (x, y) co-ordinates. The radio communication range of each node is set as 3m. Each sensor node is initialized with 1J (Joule) of energy. The network is given some queries which are randomly generated at the BS. Each query is targeted to randomly selected nodes of the network.

To evaluate our proposed scheme, we have conducted a series of experiments on the sample data sequence obtained from Intel Berkeley Research Lab [13]. The sample data sequence comprises of sensor readings for physical attributes like temperature, humidity, voltage and pressure. For our experimentation we have taken readings for temperature alone. Each sensor node for every one minute will read one reading from the data file mentioned above and stores the same in the local table with appropriate time-stamp. This way each node senses (reads from the file) 60 times a minute. Thus we have readings recorded for 24 hours. Each sensor node reads data from different input files collected from [13]. In our experimentation our sample queries request for data pertaining to a time period in number of hours starting from 0 to 24. In our simulated network, we have a BS, where queries are generated continuously at regular intervals. To answer a query, data need to be retrieved from two sensors. (To make it simple, we have assumed each query requesting data from two sensor nodes meaning that the query has its results from two sensor nodes). So, we generate queries for two sensor nodes other than their pairs. The query targeted two sensor nodes are generated in a sequential order so as to make energy dissipation in WSN uniform. The BS disseminates query into the network. The node in a group near to the BS processes the query, with its lookup. If it doesn't find its complete results the node forwards it to the next node in the group. If the query is answered completely, it sends the results to the BS or else forwards the query to the next node.

All the experiments are conducted on the above described simulator and the environment. In the following discussion, we explain the experiments and discuss the analysis of results.

A. *Performance analysis*

Our implementation covered all our proposed techniques with respect to - 1.Sleep-scheduling, 2.Collaborative sensing, 3.Lookup based query processing. We have compared the performance of our approach with that of P-MAC [9] based on the following criteria.

1. The lifetime of the network.
2. The total load handled in a fixed time interval.
3. The total energy consumed in a fixed time interval.
4. The power efficiency in the network.

We have done our experimentation with a network having, 50, 100, 150, 200 and 250 nodes. The presented reading for each performance metric is the average of the five simulation runs conducted on above networks. We have computed: (i) the lifetime of the network, (ii) the number of queries processed, (iii) the total energy consumed and (iv) power efficiency in the network for each simulation. The average values of our results for each experiment are depicted in the graphs presented in this section.

We have implemented our scheme and P-MAC on our simulator. In both the cases the networks were initialized with same initial energy and were exposed to a stream of queries which are generated at uniform rate. We measured the lifetime of the network in each case, which is depicted in graph shown in Fig. 1.

The experimentation was done on networks with varying network size (in terms of number of nodes), as mentioned earlier. From the graph it is evident that our scheme results in longer life of the network when compared to P-MAC. We also observe that for the networks with size 150 and above our approach outperformed the P-MAC. As a result of increased lifetime of the network in our scheme it is very natural to see higher number of queries handled by the network during the lifetime. In case of P-MAC, each node transmits data leading to energy wastage thereby lessening the lifetime of the network.

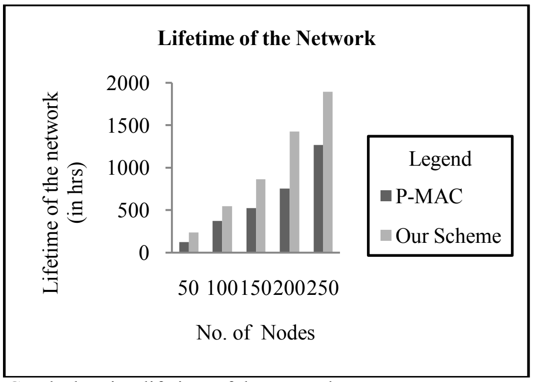

Fig. 1 Graph showing lifetime of the network

Next we conducted series of experiments to compute the load handled (total number of queries executed) by both approaches, in a specific time interval (15000 seconds). In our scheme the data is stored at the node level thereby facilitating fewer data transmissions as the data size is smaller than that of P-MAC. It is obvious that P-MAC spends more time in data transmission than our scheme, which is a direct indication that P-MAC gets less time to process queries. According to the graph shown in Fig. 2, we understand that the total number of queries handled by network adopting our scheme is higher than that of P-MAC.

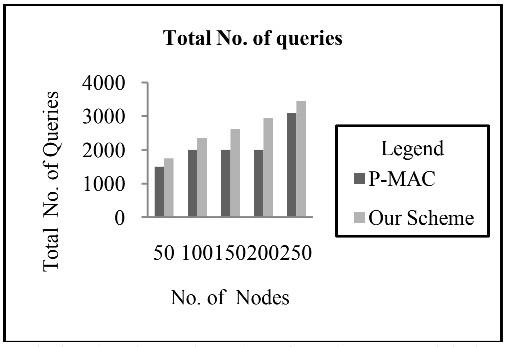

Fig. 2 Graph showing the total number of queries executed during the lifetime of the network

The graph in Fig. 3 shows the total energy consumption in fixed time duration of 15000 seconds. According to Fig. 3, we observe that our collaborative sensing and lookup based search strategy is highly energy efficient when compared to the P-MAC protocol.

The energy consumption in our scheme is lesser than P-MAC because the energy required for transmission through the secondary radio is 10 times less than the primary radio. In P-MAC the packets containing the data of the sleep-pattern (PETF – Pattern Exchange Time Frame) is sent through the primary radio to the neighboring nodes, thus resulting in higher power consumption as compared to our scheme where the data (a binary bit 0 to send a wake-up/accepted wake-up request and a binary bit 1 followed by the metadata (query)) is small resulting in less power consumption. And also P-MAC utilizes much of the node energy reserve in its communication activity, which is well addressed in our scheme. Hence, for the same workload the power consumption is more in P-MAC when compared to our scheme, which is proved in our simulation results presented in Fig. 3.

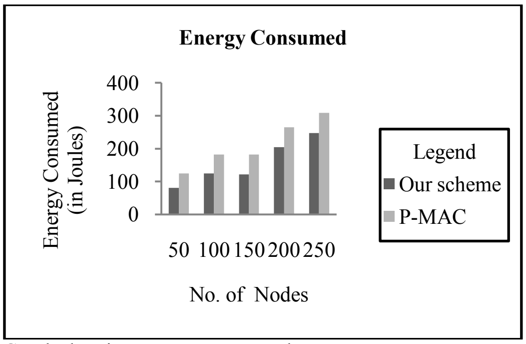

Fig. 3 Graph showing Energy consumed

The power efficiency is the number of queries executed per unit energy consumed as given in Equation 1. We could derive these metrics by running the simulations for 15000 seconds. As we see that the power efficiency is calculated as the total number of queries executed divided by the total energy consumed. Since the energy consumption is less and the total number of queries is high in our scheme when compared to that of P-MAC the value of the power-efficiency of our scheme is greater than that of P-MAC. As the P-MAC follows the sleep-pattern received from its neighbor, the sensing and sleep timings are followed according to that pattern irrespective of the present network condition. The P-MAC shows higher increase in the energy consumed, while energy consumed in our approach is moderate with respect to increase in the network size.

$$\text{Power efficiency} = \frac{\text{Total number of queries}}{\text{Total Energy Consumption}} \tag{1}$$

This means due to load balancing in our approach, the network behaves neutral. Hence, it is understood that the P-MAC shows lesser power efficiency when compared to our approach. This is represented in graph shown in Figure 4.

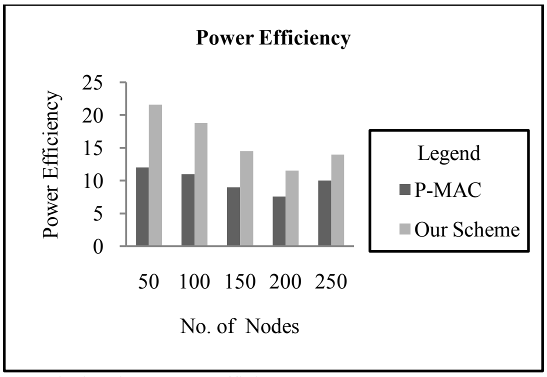

Fig. 4 Graph showing Power Efficiency

V. CONCLUSION

In this paper we have proposed an energy efficient data management scheme which is best suited for disaster monitoring Wireless Sensor Networks. Energy efficiency in our approach is achieved due to – (i) novel sleep-scheduling, (ii) collaborative sensing, (ii) efficient query processing using lookup structures. Our proposed approach ensures availability of critical data even during the disasters. Our experiments conducted on custom-built simulator have proved the efficacy of our approach in terms of power utilization in the network, and query response time. Though our proposed techniques put together, are best suited for disaster monitoring applications, can be used independently for achieving energy efficiency.

REFERENCES

[1] Raghavendra, Cauligi S., Krishna M. Sivalingam, and Taieb Znati, eds. Wireless sensor networks. Springer, 2004.

[2] Ye, Dayong. "A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks." IEEE transactions on cybernetics 48, no. 3, 2018: 979-992.

[3] Jiao, Xianlong, Wei Lou, Xinxi Feng, Xiaodong Wang, Libin Yang, and Guirong Chen. "Delay Efficient Data Aggregation Scheduling in Multi-channel Duty-Cycled WSNs." In 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 326-334. IEEE, 2018.

[4] Kimura, Naoto, and Shahram Latifi. "A survey on data compression in wireless sensor networks." Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on. Vol. 2. IEEE, 2005.

[5] Kavitha, Kayiram, Cheemakurthi Ravi Teja, and R. Gururaj. "Workload-Aware Tree Construction Algorithm for Wireless Sensor Networks." International Journal on Applications of Graph Theory in Wireless Ad Hoc Networks and Sensor Networks 4.1, 2012.

[6] Ye, Wei, John Heidemann, and Deborah Estrin. "An energy-efficient MAC protocol for wireless sensor networks." In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1567-1576. IEEE, 2002.

[7] Ye, Fan, et al. "A two-tier data dissemination model for large-scale wireless sensor networks." Proceedings of the 8th annual international conference on Mobile computing and networking. ACM, 2002.

[8] Van Dam, Tijs, and Koen Langendoen. "An adaptive energy-efficient MAC protocol for wireless sensor networks." Proceedings of the 1st international conference on Embedded networked sensor systems. ACM, 2003.

[9] Zheng, Tao, Sridhar Radhakrishnan, and Venkatesh Sarangan. "PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks." Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International. IEEE, 2005.

[10] Ahmed, Khandakar, and Mark A. Gregory. "Data-Centric Storage in Wireless Sensor networks." Wireless Sensor Networks: Current Status and Future Trends (2012): 33.

[11] Aly, Mohamed, et al. "Zone Sharing: A hot-spots decomposition scheme for data-centric storage in sensor networks." Proceedings of the 2nd international workshop on Data management for sensor networks. ACM, 2005.

[12] Li, Xin, et al. "Multi-dimensional range queries in sensor networks."Proceedings of the 1st international conference on Embedded networked sensor systems. ACM, 2003.

[13] Madden, Samuel. "Intel lab data." Web page, Intel (2004).