Journal of Network and Information Security
6 (2), December 2018, 05-09
http://www.publishingindia.com/jnis

Scheduling Sensor Nodes for Enhancing Energy Savings in a Wireless Sensor Network

Shashank Srigiri[1*], Rishabh Ranjan[2] and Itu Snigdh[3]
[1]Department of CSE, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India.
Email: shashank10456@gmail.com
[2]Department of CSE, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India.
Email: rishabh.ranjan01567@gmail.com
[3]Assistant Professor, Department of CSE, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India. Email: itusnigdh@bitmesra.ac.in
*Corresponding Author

Abstract: Energy and delay are critical issues to any WSN. Among the many solutions till date, intelligent data forwarding has been extensively researched. This grounds us to consider a query routing tree to analyze the impact of schedule-based data transmissions to the sink. On the assumption of a hierarchical topology, our algorithm analyses and compares the route tree structure generated based on a pre-schedule to the usually adopted minimum spanning tree and Dijkstra algorithm-based routing trees. Our approach also analyses the impact of considering residual energy of the nodes. We apply both residual energies as well a pre-determined schedule to transmit data and observe that the energy conservation of the nodes is increased manifold.

Keywords: Dijkstra algorithm, Energy conservation, Minimum spanning tree, Query routing tree, Wireless sensor networks.

I. INTRODUCTION

A WSN essentially comprises of sensor nodes and a communicating interface. The sensor nodes are autonomous enough to decide which node to communicate to, given their energy level constraints and assigned sensing range. However, they are not intelligent enough to decide a schedule to forward data so as to minimize the network congestion or optimize the channel utilization. The associated intelligence in these contexts are added by means of scheduling algorithms or efficient MAC protocols [1], [2]. These approaches are usually either to reduce the energy expenses or to reduce the incurring delay in the transmission of data packets. As sensor nodes are responsible for forwarding messages and data to one another in the network, scheduling of sensor nodes for their effective utilization becomes critical for the overall network health.

These constraints however are different for a query driven sensor network. In a query driven WSN network application, communication is initiated by the sink, and the data needs to travel from a source S to a destination D which is usually unidirectional or converge cast. This is so because every typical query based WSN has a coordinator or a sink to store the collected data and forward it for further processing and hence they control the flow of data in the network. The efficiency of such WSNs depends on the coordinators' potential to collaborate as well as the extent to which the decided route is robust against the typical WSN constraints like battery power, network congestion, connectivity issues and link unavailability.

The initial phase of such sensor networks is its establishment if we assume that it would be static in nature once deployed. Post deployment, a query based WSN either arranges itself in the form of a graph or a hierarchical structure. The data and route queries in a query driven WSN are forwarded between the base station and the location where the target phenomenon has occurred or is observed as cited by Georgios *et al.* (2010) [9]. As data packets would be initiated only on the event of query initiated at the sink, it would result in lesser collision or contest for the channel thereby reducing the end to end delay. The first approach for realizing a Q-WSN is based on a single hop approach where each sensor node directly communicates with the base station or the sink.

We know that radio signals require a lot of power. Unlike messages running through wires, they decay in an accelerated manner. Also, as sound and radio decay according to the inverse square law [6], on doubling the distance we require four times the amount of power. The major limitation of this approach is severe energy depletion of the farther nodes and this eventually causes profound limitation in the lifetime of the network.

This shortcoming of the direct communication approach is overcome by multi hop packet transmissions over short communication ranges. This approach saves energy considerably and reduces the communication interference among the nodes competing for channel access. In addition to these approaches, scheduling has also been widely adopted to reduce delays and energy consumption. MAC scheduling algorithms have been applied to reduce data packet average waiting time, energy conservation and improvement of average successful transmission rate [1]. TDMA based scheduling has been proposed for multi hop networks characterized by many independent point to point flows in the network. Both node scheduling and node-based scheduling present methods to avoid conflicts while allocating slots in data transmission [3].

Node scheduling adopts sleep scheduling of the nodes where they turn on and off according to the algorithm while node-based scheduling deals with the scheduling of the levels in routing tree [2]. Nodes are also periodically scheduled in the form of organized disjoint sets [4] that remain active for a transmission period and remain in sleep mode in the remaining time. These mechanisms claim to increase the system lifetime by conserving energy of redundant nodes that can be used alternatively. Similar literatures mention adoption of scheduling algorithms on mobile nodes that act as data ferries to transport data to the sink. These adopt the process schedule algorithms like Earliest Deadline First (EDF), EDF with lookahead which parameters like delay and buffer capacity of nodes.

Our approach adopts a query routing tree which computes the best forwarding node on the basis of the pre-decided schedule, the last time of communication between forwarding nodes as well as the node residual energy of competitive nodes.

The article is organized as follows: Section 2 explains the query driven network scenario followed by a system model and assumptions. Section 3 illustrates the algorithm employed and presents the results with conclusion in Section 4.

II. System Model

Energy is a critical issue in every battery aided network [7], [8]. Our article focuses on conserving energy by time slicing the transmission period for each node. In addition, every node computes the best possible neighbour depending on the residual energy of its forwarding node. For this it maintains a routing table that contains the nearest neighbours' ID and their respective distance. For computing the query tree we use the Dijkstra's algorithm to find the shortest path among the given source and destination and compare it to a corresponding minimum spanning tree that is usually adopted in many literary works [11], [12], [13]. We assume the sink to serve as the destination for each iteration and every other node is the source.

Fig. 1 depicts a graph comprising of three different sensor nodes as S, H and T respectively connected to a sink; symbolizing the different characteristics they exhibit in terms of data collection, transmission power or connection strategy. It may be seen that all the S type of nodes connect only to a sensor of its type and so is the case with H and T. This is to enable only one type of the sensors to be active for processing one type of query. The remaining sensors can remain in their sleep mode. Unlike the other strategies of sleep and wake scheduling where nodes wake up and listen to the channel before forwarding data, our approach allows the sink or coordinator node to assign time slots for each type of node. To enable this strategy the sink first allocates a time-based schedule to each node lying within its one hop range. These first order nodes connect to the remaining nodes with the help of a spanning tree algorithm and allocates a time frame to transmit within its allocated schedule. The same procedure is followed till all the nodes are connected.

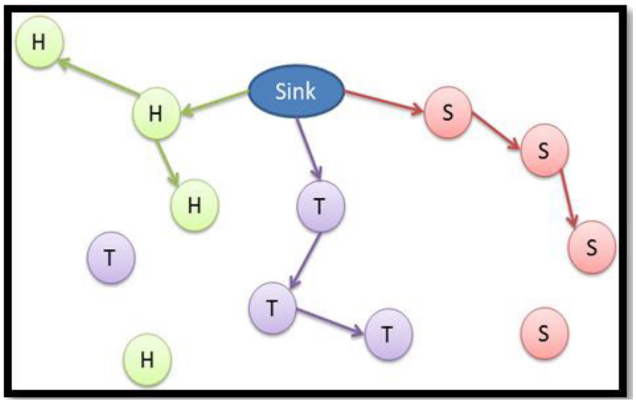


Fig. 1: Tree Based Backbone

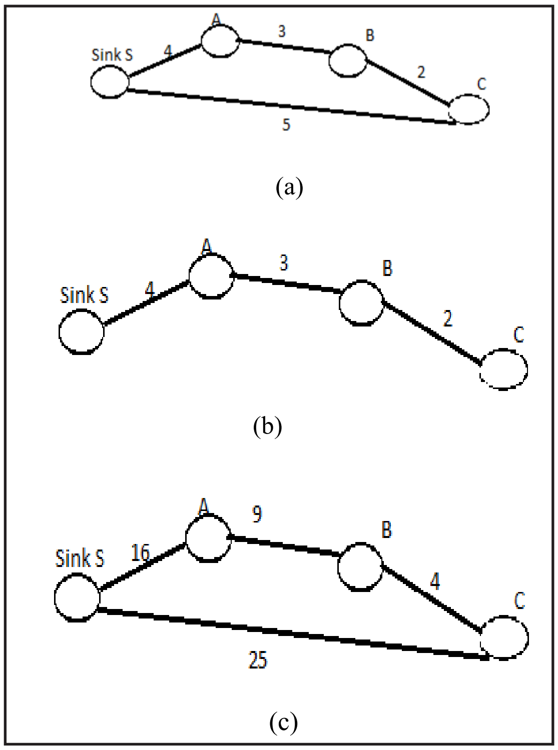
A. Proposed Strategy Aintaining the Integrity of the Specifications

The minimum spanning tree employs Prim's algorithm to construct the query tree and routes data as and when the sink invokes a request. A fixed schedule of data transmission for nodes is assigned to the child nodes. The Dijkstra's algorithm based query routing tree also uses the same schedule. The third algorithm uses a greedy based approach which uses the query route which has nodes with maximum residual energies. It just considers the shortest path as well as the energy of the nodes it forwards data to. It does not consider any schedule as in the above cases. For minimizing energy consumption every node must take the shortest path as the energy to transmit and receive majorly depends on the inter node distance. Literatures advise computing energy consumption in transmitting or receiving to be computed by the formula

$$E_{TX/RX} = kld^{\alpha}\text{; where, } \alpha=2, 4. \quad (1)$$

Therefore, we can assume that the cost of each edge is square of the distance between the nodes. Thus, energy consumption becomes proportional to the distance between the nodes. As energy expedited is proportional to square of the distance between nodes we adopt a methodology wherein the cost of each edge is square of the distance between the nodes with the sink as the source node. The advantage of Dijkstra's algorithm with squared edge weights over MST can be illustrated by the given example in Fig. 2 (a), (b) and (c).

Fig. 2 (a) depicts a graph with edges as the distances between the nodes A, B and C communicating to the sink S. If we assume node C to deliver a message to S, the path followed is depicted in Fig. 2 (b) from node C → B → A → S and the total energy consumed is 16+9+4 equals 29 units. While considering Fig. 2 (c), which considers Dijkstra with squared edge distances, the path would be node C → S and the total energy consumed would be 25 units.



(a) (b) (c)

Fig. 2: (a) Graph with Edge Weights (b) MST Based Edge Weights (c) Squared Dijkstra Based Edge Weights

Dijkstra with squared edges will always consume less than or equal to energy consumed using MST as an MST considers the best node from the current position while our approach with squared edges considers the best path to the sink.

Table I: Symbol table

Parameters	Symbol
Euclidean Distance between Nodes i and j	Di,j
Shortest Path 1st-Level Heading	Si,j
Node Remaining Energy	NRE
Sensing Range or Transmission Radius	R
Number of Nodes	N

B. Algorithm

- For each randomly generated nodes N={i, j,…n}.
- Construct graph G (V, E) such that Di,j <= R.
- Compute edge weight = cost of message delivery α (Di,j).
- Compute the shortest distance among all nodes using edge weight and Dijkstra's algorithm and sink as the root node 'S'.
- Remove redundant edges and store the minimum energy path for each node.
- Store node ID, edge weight with respect to all connected nodes.

C. Assumptions

We consider the network as a graph 'G' with 'V' vertices and 'E' edges. The vertices are the sensors and the edges are the communication links between the edges. The initial network topology depends on the following constraints:

- Nodes once deployed remain static.
- The existence of communication link depends on the constraint that ensures two nodes to connect only when their inter node distance is less than their sensing radius. Mathematically, if SR is the sensing radius and CR is the communication radius; then SR <= CR where CR = Di,j and Di,j = Euclidean distance between nodes i and j.
- Nodes adopt unicast communication model though they are connected to all nodes within their sensing radius to reduce redundant transmissions.
- Each node maintains a routing table about the connecting nodes and their respective distance among them.
- Every node computes the minimal energy constraint before forwarding message to its neighbour.

III. Results

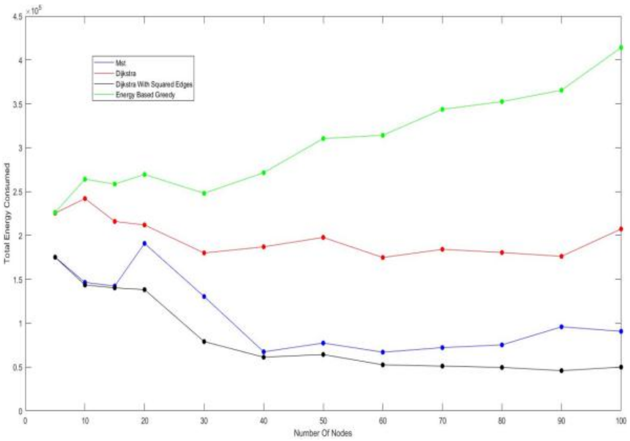
We employ four algorithms namely Minimum spanning tree (Prim's), Dijkstra's algorithm, Modified Dijkstra's algorithm and a greedy based energy optimal algorithm to analyze our scheduling strategy.

Table II: Simulation Parameters

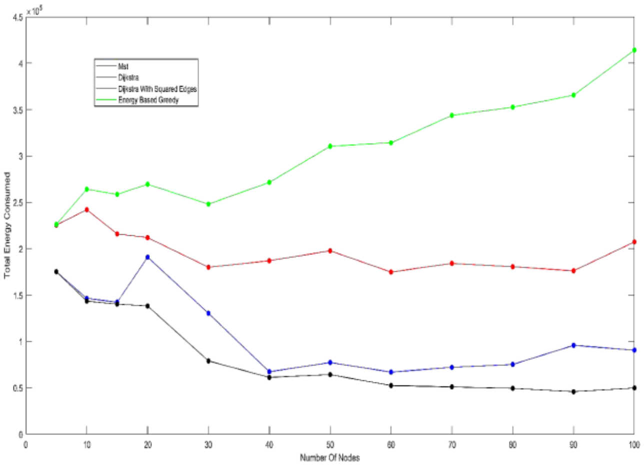
Parameters	Case 1	Case 2	Case 3
Number of Iterations	5..1000	5..1000	5..1000
Number of Nodes	Varies from 5 to 100	Varies from 5 to 100	Varies from 5 to 100
Initial Energy of Each Node	1+e7	1+e7	1+e7
Range of all the Nodes are Same	70	100	220
Area of the Region of Interest	100x100	200x200	250x250

Based on the above-mentioned controlling parameters graphs have been plotted for three scenarios case 1, 2 and 3 and 15 iterations respectively. We depict the overall energy consumption of the network which is the total energy consumed per node. From the graphs in Fig. 3 (a), (b) and (c) we observe that the energy consumption is more when either the number of nodes is less, for e.g. 5 nodes or when the range of nodes is less as compared to their area of deployment. This is so because the total consumed energy is dependent on the link distance and the node positions and when the nodes placed or the range the nodes are less, the randomly placed nodes either do not fall in the connection range and stay disconnected or they need to connect to nodes at a farther distance. In such cases, the impact of scheduling becomes limited. With increase in the number of nodes we can see the actual improvement of our adopted strategy.

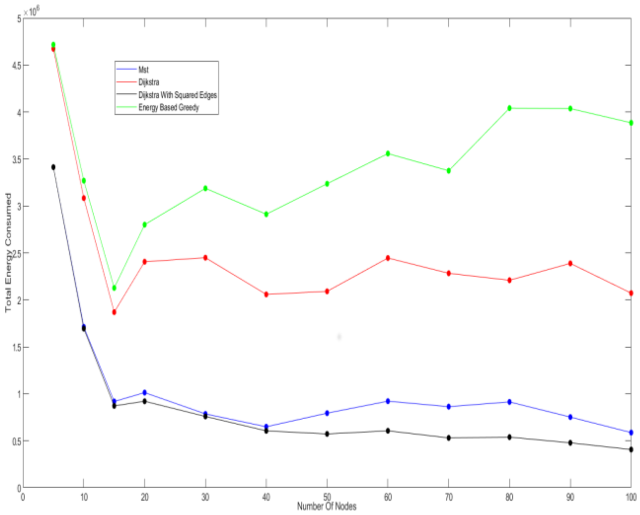
We observe that the minimum spanning tree for route construction ensures choosing the most optimal node to forward the data and is always the shortest path and hence the energy depletion is minimum. The second strategy that adopts the Dijkstra's algorithm depicts a balanced energy depletion with the increase in number of nodes. The energy-based algorithm performs the worst as it only considers the residual energy of the node while forwarding data and not the inter node distance. The best performance is shown by Dijkstra's algorithm with squared edge weights as depicted in Fig. 3 (a), (b) and (c).



(a)



(b)



(c)

Fig. 3: (a) Case 1 (b) Case 2 (c) Case 3

IV. CONCLUSION

We analyzed the mentioned algorithms for different node coordinates, area of interest, different node ranges and different number of nodes. We observe that for schedule-based data transmissions considering squared distances as edge weights gives us better results than any algorithm employed to create a query routing tree. Thus, we conclude that rather than only considering the residual energy of the nodes if we adopt our approach, there is a significant energy saving of the overall network. Our work has considered only static schedules which has been put to further enhancement by adopting dynamic scheduling methods. The scheduling mechanism has been attached for reference.

REFERENCES

[1] Q. Ren, and Q. Liang, "An energy-efficient MAC protocol for wireless sensor networks," in *2005 Global Telecommunications Conference (GLOBECOM'05)*, vol. 1, IEEE, 2002.

[2] D. Tian, and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 32-41, ACM, September 2002.

[3] S. C. Ergen, and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Networks*, vol. 16, no. 4, pp. 985-997, 2010.

[4] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*

Proceedings IEEE, vol. 3, pp. 1976-1984, IEEE, March 2005.

[5] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International, pp. 296-305, IEEE, December 2004. M. Baldonado, C.-C. K. Chang, L. Gravano, and A. Paepcke, "The stanford digital library metadata architecture," *Int. J. Digit. Libr.*, vol. 1, pp. 108-121, 1997.

[6] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols and Applications*, John Wiley and Sons, 2007.

[7] J.-Z. Sun, "QoS aware query processing algorithm for wireless sensor networks," *Journal of Computers*, vol. 3, no. 11, pp. 32-41, November 2008.

[8] T. Arici, B. Gedik, Y. Altunbasak, and L. Liu, "PINCO: A pipelined in-network compression scheme for data collection in wireless sensor networks," *12th International Conference on Computer Communications and Networks*, pp. 539-544, 2003.

[9] G Chatzimilioudis, D. Zeinalipour-Yazti, and D. Gunopulos, "Minimum-hot-spot query trees for wireless sensor networks," *Proceedings of the Ninth ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 33-40, ACM, 2010.

[10] D. Messina, M. Ortolani, and G. L. Re, "A network protocol to enhance robustness in tree-based WSNs using data aggregation," *Internatonal Conference on Mobile Adhoc and Sensor Systems, (MASS 2007),* pp. 1-4, IEEE, 2007.

[11] J. Wang, and K. Jee-Hyong, 'ViTAMin: A virtual backbone tree algorithm for minimal energy consumption in wireless sensor network routing," *IEEE International Conference on Information Networking (ICOIN)*, pp. 144-149, Bali, 1-3 February 2012.

[12] S. Chakraborty, S. Nandi, and S. Karmakar, "A tree-based local repairing approach for increasing lifetime of query driven WSN," *14th International Conference on Computational Science and Engineering (CSE),* pp. 475-482, IEEE, 2011.

[13] H. Fariborzi, and M. Moghavvemi, "EAMTR: Energy aware multi-tree routing for wirelesss sensor networks," *IET Communications,* vol. 3, no. 5, pp. 733-739, 2009.

[14] M. W. Yingchi, L. Xiaofang, and L. Yi, "Workload based query routing tree algorithm in wireless sensor networks," *International Conference on Computational Intelligence and Software Engineering (CiSE),* pp. 1-4, December 2010.