Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-11, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Energy Efficient Scalable and Reliable Wireless Sensor Routing

Keshav Sinha[1], Annu Priya[2], Partha Paul[3]
[1, 2, 3] Computer Science and Engineering, Birla Institute of Technology, Mesra, India

***Abstract**: The area of wireless sensor network (WSN) being fascinating among the community of researchers. The diverse area of WSN in defense, civilian and medical will enrich the popularity of WSN routing, but there are some dark side also while writing it only works for computationally and resource constrained micro-sensors. Above all it pressure the delay line-up in conventional routing protocol that developed for unrestricted ad-hoc wireless network. Routing protocol being planned for request WSN for energy efficient & durability of the network. We confirm a breeze, less time taking, energy efficient routing protocol having one-level data aggregation that assures upgraded durability of the network. The affirmed protocol being set side by side the other ad-hoc & sensor routing protocols. It's been perceived that confirmed protocol exceeds them in throughput, latency, the average energy consumption and average network endurance. The proposed protocol worth absolute time and node energy as the benchmark for routing, this cinch reliability and congestion avoidance.*

1. Introduction

A wireless sensor network is extended in various applications for a longer period of time. In the absence of maintenance of nodes & restoration of their energy sources. Each sensor [3] node in the network exhaust power, not exclusively in sensing data, but in processing data & broadcasting this processed information for more of the routing. Hence, a routing protocol framed for specified network [6] should be idolizing such that power utilization at every stage of the protocol's functionality minimum. Also, the network endurance to be kept greatest by appropriately making use of each of the services of the sensor [4] node for the sake of routing. A routing [7] protocol for WSN needs to be splendidly straightforward, having limited computational complexity, efficient [1], competent enough in power utilization, shall have upgraded endurance & having least latency for data transmission from the node to sink. The routing protocol that is to be presented in this work is in compliance with the aforementioned characteristics of WSN. Bandyo padhyay et.al [2], introduced the routing protocols framed for WSN be classified stationed on path selection. Placed on network architecture, it is being further classified as protocol recommended. The data-centric sensed being identified alongside attributes varying with specific applications. It's in cognizance with MCF it is assuming base station (BS) to be fixed and data needs to be requisite to be transferred to BS only. Not being enlisted in a diffusion of data as being accompanied by direct diffusion.

The organization of the paper is as follows: Section 2 describes the Proposed Protocol followed by an Introduction in Section 1. Section 3 deals with Data simulation finally in Section 4 conclusion is drawn.

2. Proposed Protocol

In this section LEO based routing [10] is proposed where every node to the base station (BS) is not recommended by all the nodes in the network. Each node and ad-hoc [5] network the information as respect to its neighbors only, thereby decreasing the memory demand of each node. The node contains the two types of information on the neighbors is saved in the neighbor table for each and every node. (i) The absolute time required for a packet to reach the BS from that node, and (ii) The remaining node energy. This protocol is comprehensive and has not been devised to optimally work in any specific application. The application specifies to bind the time and route temperature data. When the temperature increases beyond an outset, then it reloaded the sensors [9] is posting. Unlike the popular existing protocols such as DD, SPIN, LEECH, TEEN, GAF and GEAR, this protocol, defend the intensity of energy overhead involved for diffusion in diffusion-based routing protocols formation of cluster heads in hierarchic routing protocols, and communication overheads in the geographical [8] instruction placed routing protocols. In sum, the vital countenance of this protocol is its simplicity, less computational ramification, highly depressed routing overheads and the need for the nodes to broadcast only once to create the neighbor table. There are several steps for protocol routing is discussed as follows:

Step 1: Packet header

The packet header of LEO is presented in Table 1. The fields contain the packets, timetoreach_bstn and node_energy are the vital parameters to intend the routing paths. This is because the time taken from the neighboring node to the BS and the node's energy is the yardstick for routing in our protocol.

Table 1 Packet Format

Pkt_src	A node which originated this packet
Pkt_dst	Destination node
Fwd_set	A node which lies b/w the source and the sink
Nbr_set	Neighboring node
Pkt_seq_num	Sequence number of the packet
Pkt_start_time	Packet start time to each node
Timetoreach_bstn	Time to reach the base station to the node
Node_energy	Energy of the node
Nodetrans_id	Node Packet transmission ID
α , β	Turning factor

Step 2: Functionality and Algorithm of LEO

In this protocol each node is sensing the transmitting and receiving capabilities. It is able to compute simple aggregation functions using a neighbor table and a forwarding table. Sensors are being stationed (in a grid or random structure). Here random format is being used & to this random format, random function is being used. The neighbor table at each node stores only three data (i) the identifier of the neighbor node from which it has received the packet (pkt_src), (ii) the absolute time taken for a packet to reach the BS from the neighbor node (timetoreach_bstn) and the (iii) residual energy at the neighbor node (node_energy). The forwarding table of each node stores the information of that node which lies between the sender and the BS. The initialization phase is invoked by the BS and is done immediately after the nodes are deployed. This process initializes the neighbor table, which is otherwise empty. During the initialization phase, each node is made to broadcast only once. This ensures less routing overheads but taking care of the availability of the required information at each node for routing. Initialization is done only once and after the initialization process, all the nodes are ready for routing packets to the BS.

Step 3: Algorithm Initialization Process

The absolute time taken by each packet to reach the BS from each node is calculated using the following algorithm. An initialization packet is broadcasted by the BS once the nodes are deployed. The initialization packet will have its timer switched on when it starts to form the BS and the field pkt_start_time is set to 0. The node energy is also set to the actual value. Once the neighboring nodes of the BS get the packet, the following algorithm is followed.

Algorithm

Step.1 Deploy sensor's (either in grid or random format). We use random format. For random format use random function.
Step.2 Once Nodes are deployed, store there (X, Y) coordinates in the array.
Step.3 Define a variable transmission range (let it be 20m).
Step.4 For each node N:
Calculate the Euclidean distance with all other nodes. Those which are less than trans-mission range, from the nbr_set (neighbor set) of the node N.
Step.5 Calculate forward set of the Node N:
All those nodes which are part of its Nbr_set and also lies between its location and the sink location constitutes its Fwd_set (Forward set).
Step.6 On the basis of two factors select the intermediate Node:
Factor 1. Remaining Energy: The node in the Fwd_set with maximum remaining energy must be selected.
Factor 2. Reliability: Node with minimum packet loss must be selected as:
α * (Residual energy /initial energy) + β * (1/Packet loss). Where α and β are tuning factors.

The node that has received the initialization packet calculating the time required for a packet to reach the BS from the information consigned by the node before. The existing node amends the field timetoreach_bstn, calculates the residual node energy after one transmission and amend the node_energy field and more of that it broadcasts the packet. It also updates its neighbor table with node ID, timetoreach_bstn and node_energy data being neighbored by packets that it has received. It also updates its forwarding table being accompanied by node ID, timetoreach_bstn and node energy data of those nodes which are closer to the BS. Calculate the Euclidean distance with all other nodes. Those which are less than transmission range, from the nbr_set (neighbor set) of the node N. Forward set reckoning of the set of node N: Each node whatsoever being in nbr_set set and is conjointly amid endemic for location and sink location complement its Fwd_set (For-ward set). Based on two components the intermediate one's among the node is stabbed. Component 1 - Remaining Energy: Node having superlative remaining energy value among the fwd_set is a requisite for impending to be tabled.

Component 2

Reliability: Node possessing minimal packet loss is a requisite to be tabbed among all other. This process is repeated until the entire network is covered. At the far end of aforementioned initiated by the process, each node will know all its neighbors in its radio frequency (broadcast) region, time to traverse through the BS from individual nodes and their residual energy. Each node can easily route the actual data packet backed by the data attainable in the adjoining table. There is no further requirement to spend time and re-sources in creating the route between the source and the BS.

3. Data Simulation

We used Matlab, R 2013a for simulating the network diversified by the dimensions. We calculated the performance of Modified-LEO on different node clusters. We compared the performance of Modified-LEO with the LEO algorithmic architecture. The comparison features include i) Throughput b) Latency and c) Packet Dropped. The network (Node clusters) has predefined Transmission Range, Node Energy, and Network coordinates for each Node {n1, n2…… nix_{-1}, n_x}. Whereas, (x=1, 2,…, ∞]. After initializing each node with node energies and transmission range. The simulator calculates the neighbor-Set (NS) of each node based on the Transmission Range (R). After calculating the NS for each node of the network, the Source-Node (S) and Destination-Node (D) are selected for packet (L) exchange. Depending on source node S and destination node D, the forward set is calculated starting from source node S, intermediate nodes, to destination node D. The forward set (FS) is calculated using the Euclidean Distance (Distance) formula as:

$$E\_distance\ (n_i,\ n_j) = \sqrt{((x_1 - {x_2}^2) + (y_1 - y_2)^2)} \quad (1)$$

Whereas,
i = Intermediate Node.
j = Destination Node D, and
x_i & y_j = Co-ordinates of intermediate node and D.

Based on the least Euclidean distance in the cluster, the Forward-Set (FS) is calculated, now source node (S), destination node (D), FS and NS are ready. We first simulated the network with 25 nodes and sent packets over the predefined network topology. We call the above function iteratively for sending n packets and then calculate the time elapsed for each transfer. Based on the total packets sent by the source node (S), packets delivered successfully (received by D), packets dropped, and time elapsed, we calculated the throughput and latency.

1. Throughput (T)

$$T = \frac{Packet\ Received}{BS\ total\ Packet\ Sent} \quad (2)$$

Whereas, BS is the base station.

Table 2: Average Number of packets sent

Protocol	Total number of nodes in simulation			
	25	50	75	100
Modified-LEO	100	100	100	100
LEO	100	6082	5857	5536

Table 3: Packet Dropped during the simulation

Protocol	Total number of nodes in simulation			
	25	50	75	100
Modified-LEO	92	94	95	98
LEO	86	2379	2200	2238

Table 4: Packet during the simulation

Protocol	Total number of nodes in simulation			
	25	50	75	100
Modified-LEO	8	6	5	2
LEO	14	49	17	23

2. Latency (L_t): The latency is calculated is shown below

Table 5: Time taken by the packet to reach destination Node (D)

Protocol	Total number of nodes in simulation			
	25	50	75	100
Modified-LEO	0.247	0.00794	0.0196	0.025
LEO	0.001374	0.00552	0.0123	0.022

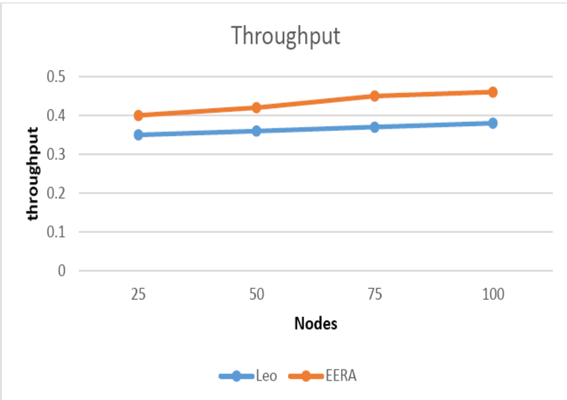

Figure 1: Throughput of networking using LEO Method

Reliability (R_t):

The reliability factor depends upon the node energy (E) left after each iteration and the type of node selected i.e. The more distant the node, greater will have its energy, because farther nodes are selected least in the starting of the simulation. The overall reliability of the network depends upon following factors:

$$Packet\_Loss\ (P) = \frac{Received\ packect - sent\ packet}{Received\ Packet} \quad (3)$$

Tuning factors α and β, where α=1-β, for ideal condition β=0. 5

$$R_t = \frac{(\alpha * Residual\ Energy)}{Intial\ Energy} + \frac{(\beta * 1)}{Packetloss\ (P)} \quad (4)$$

Here three conditions exist:

If, $\alpha = \beta = 0.5$

Then, it will be in ideal condition, in that case R_t will get halved.

If, $\alpha < \beta$

Then, the value of R_t will depend on residual energy and initial energy.

If, $\alpha > \beta$

Then, the value of R_t will depend on all three factors (initial energy, residual energy, packet loss

4. Conclusion

The novelty of this paper is the implementation of energy efficient, reliable routing algorithm (EERA), for packet delivery from source to destination. EERA works on four basic concepts throughput, reliability and latency. For higher network dimension the EERA is similar to Leo after nearest nodes (Nodes in feature set) get exhausted. There will be an increase in throughput for bigger networks if considerable node energy is supplied to each node in the future set of the network. The LEO based methods can also be implemented in the thrust area of networking for the future and the simulation results.

5. References

[1] S. Misra, P. Dias Thomasinous: "A simple least-time and energy-efficient routing protocol with one-level data aggregation for wireless sensor networks", The Journal of Systems and Software, 83, 852–860, 2010.

[2] Bandyo padhyay, S. Coyle: "An energy efficient hierarchical clustering algorithm for wireless sensor networks", In Proceedings of INFOCOM, vol. 3, pp. 1713–1723, 2003.

[3] Bellaachia, Abdelghani, Weerasinghe, Nilkamal: "Performance analysis of routing protocols in sensor networks", International Journal of Ad Hoc and Ubiquitous Computing, 3 (3), 167–173, 2008.

[4] Chandrasekar, R. Misra, S. Obaidat: "A probabilistic zonal approach for swarm-inspired wildfire detection using sensor networks", International Journal of Communication Systems, 21 (10), 1043-1047, 2008.

[5] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris: "SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks", Wireless Networks 8 (5), 481–494, 2002.

[6] M. Chu, H. Haussecker, F. Zhao: "Scalable information driven sensor querying and routing for ad hoc heterogeneous sensor networks", International Journal of High Performance Computing Applications 16 (3), 2002.

[7] Jose Nogueira, S. Marcos: "A proactive routing protocol for continuous data dissemination in wireless sensor networks", In Proceedings of 10th IEEE Symposium on Computers and Communications, 2005.

[8] J.H Ernest Chang: "Echo algorithms: depth parallel operations on general graphs", IEEE Transactions on Software Engineering 8 (4), 1982.

[9] M. Chu, H. Haussecker, F. Zhao: "Scalable information driven sensor querying and routing for ad hoc heterogeneous sensor networks", International Journal of High Performance Computing Applications 16 (3), 2002.

[10] S. K Dhurandher, S. Misra, V. M. S. Bansal, P. Singh, and V. Punia: "An energy-efficient on-demand routing protocol for wireless ad-hoc networks", International Journal of Communication Systems, 22(7), 789–817, 2009.