© 2018 IJSRST | Volume 4 | Issue 5 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Energy Efficiency of Cluster Node Using Trust Method

Manoharan L[1], Prasaad G[1], Selva Prabhu J[1], Praveen D[2]

[1]M.E. Assistant Professor, Department of Electronics and Communication Engineering, JEPPIAAR SRR Engineering College, Old Mamallapuram road, I.T. Highway, Padur, Chennai, Tamil Nadu, India

[2]Department of Electronics and Communication Engineering, JEPPIAAR SRR Engineering College, Old Mamallapuram road, I.T. Highway, Padur, Chennai, Tamil Nadu, India

ABSTRACT

Wireless sensor network (WSN) brings a new paradigm of real-time embedded systems with limited computation, communication, memory, and energy resources that are being used for huge range of applications where the traditional infrastructure-based network is mostly infeasible. The sensor nodes are densely deployed in a hostile environment to monitor, detect, and analyze the physical phenomenon and consume considerable amount of energy while transmitting the information. It is impractical and sometimes impossible to replace the battery and to maintain longer network life time. So, there is a limitation on the lifetime of the battery power and energy conservation is a challenging issue. Implement Ad Hoc On-Demand multicast Distance Vector (AOMDV) Routing Protocol for multipath communication.

Keywords: Energy conservation, Sensor Nodes, (AOMDV) Routing Protocol, Wireless sensor network (WSN).

I. INTRODUCTION

In most WSN applications nowadays the entire network must have the ability to operate unattended in harsh environments in which pure human access and monitoring cannot be easily scheduled or efficiently managed or it is not feasible at all. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance, today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring. Naturally, grouping sensor nodes into clustershas been widely adopted by the research community to satisfy the scalability objective and generally achieve high energy efficiency and prolong network lifetime in large-scale WSN environments. The corresponding hierarchical routing and data gathering protocols imply cluster-based organization of the sensor nodes in order that data fusion and aggregation are possible, thus leading to significant energy savings. In the hierarchical network structure each cluster has a leader, which is also

II. WSN & CLUSTER OF NODES

Numerous techniques have been proposed to improve the energy consumption rate, such as clustering, efficient routing, and data aggregation. WPANs are used to convey information over short distances (about 30 feet) among a private, intimate group of participant devices. WLAN, a connection made through a WPAN involves little or no infrastructure or direct connectivity to the world outside the link. The environment plays a key role in determining the size of the network, the deployment scheme, and the network topology. The size of the network varies with the monitored environment. For indoor environments, fewer nodes are required to form a network in a limited space whereas outdoor environments may require more nodes to cover a larger area. Wireless sensor networks contain hundreds or thousands of these sensor nodes, and these sensors have the ability to communicate either among each other or directly to an external base station.

III. IMPLEMENTATION OF AODV

In the existing work, a node is chosen as a cluster head on certain parameters and the chosen node will act as the cluster head for a certain period of time. It is assumed that sensor nodes send the data after detecting an interesting event. CH collects these data, aggregates it and send to the base station. To save some energy, CHs can send the data to BS to utilize the bandwidth efficiently. The parameters to choose a node are based on remaining battery power, mobility, centrality and chance. In WSNs, the sensor nodes can fail either due to harsh environmental conditions or due to node hardware failure. The faulty nodes degrade the performance (detection accuracy) of the sensor network. The detection accuracy is limited by the amount of noise associated with the measurement and the reliability of sensor nodes. The sensor nodes are usually low-end inexpensive devices and sometimes exhibit unreliable behavior. In existing presented a novel method, DBA, which can effectively identify faulty nodes. The chance value to be calculated considering three input parameters such as remaining battery power, mobility and centrality by using Fuzzy rules. It will generate misleading results. The implementation of AODV method also decreases the energy efficiency of the cluster head.

IV. ENERGY EFFICIENCY USING TRUST METHOD

First node dies in AODV very fast whereas it survives almost double time in the proposed model. Proposed protocol AOMDV is more stable than the AODV protocol. Presents about end to end delay. End to end delay is defined as the maximum time taken by the packets to travel. Here we also implement AOMDV protocol for multipath communication purpose in this protocol we can call as pro-active protocol and main advantage of this route discovery process always better performance in wireless sensor network. To overcome the wireless sensor networks communication path failure recovery using check point recovery algorithm node blocking. Data trust is evaluated based on the data sensed and collected from multiple devices; node trust is assessed in two dimensions, i.e., functional trust and recommendation trust, which indicate how likely a node can fulfill its functionality and how trustworthy the recommendations from a node for other nodes will be, respectively. The effectiveness and efficiency of the proposed Path trace scheme is validate failure through extensive experiments and find out the malicious node and the same has been eliminated so that we are increasing the performance high. Sensor monitoring all node information second by second and also all sensors will communicate each other neighbor sensor, and giving correct information. Here sensor can able to update the energy level of neighbor nodes and also with the help of sensor we can replace the low energy node into high energy node with the help of checkpoint recovery algorithm. So we can able to maximize the data transaction speed and data loss. However, a failure of an actor may cause the network to partition into disjoint blocks and would, thus, violate such a connectivity goal. Transmit aggregated data to the data sink. Reducing number of nodes taking part in transmission. Useful energy consumption. Scalability for large number of nodes. Reduces communication overhead for both single and multi-hop.

3.1 Working Methodology

The selection of CH nodes in the sensor networks can provide the following three benefits: 1. The average energy utilization for transmitting the data from the sensor node to the BS will be much lesser than the energy utilized for homogenous networks. 2. Improving reliability of data forwarding - It is generally known that the links tend to be low reliability. Each hop significantly minimizes the packet delivery rate. In heterogeneous nodes, there will be lesser hops between the nodes and the BS. Hence, the heterogeneous sensor networks can achieve a much better packet delivery ratio than the homogenous networks. 3. Decreasing latency for data transmission - Computational heterogeneity can minimize the latency in immediate nodes. The heterogeneity among the links can minimize the waiting time in the forwarding queue. Choosing lesser hops among the nodes to BS will reduce the forwarding latency. Data aggregation based on data ensemble: A cluster of nodes is replaced with a single node without changing the underlying joint deployment of the network. During node aggregation, the data ensemble process also takes place.

3.2 Cluster Formation

1. Each Node Broadcast (Head message) Contains Current Energy, ID of Node and Trust Level. 2. Based on the received Head message, each node determines Zone. Cluster Head for this round (random selection with obstacle). 3. Received strength is positive gets node's ID + Current Energy + header. 4. Calculate the Distance Based on their node's ID. 5. Calculated the distance based on nearby hop with High Energy. 6. Elects Cluster head remaining Nodes act like Cluster Member and having ID and low Energy to sense data and send to Cluster Head. 7. TDMA Schedule prevents collision among data messages.

3.3 Election of Cluster Head

Tc = W1 *D +W2 *Ef

Where,

D = Sum of the distances of all the CH nodes with each other (Proximity factor).

Ef = Energy required in a transmission to Source from this CH - Current energy of this CH node (Energy factor).

W1, W2 = Weight factors that determine the transmission grading.

3.4 Significance of Weights W1 And W2

W1=W2=1

For any two cluster-head nodes CH1 and CH2 let (D, Ef) pairs be (1, -2) and (2, -4) respectively,

Let, Tc = W1*D + W2*Ef

CH1: 1*1 + 1 *(-2) = -1

CH2: 1*2 + 1 *(-4) = -2 (Elected)

In the above case CH2 gets elected as the super cluster head due to its lower Tc value. Note that sign is taken into consideration.

Now, let us consider

W1= 10, W2= 1

CH1: 10*1 + 1* (-2) = 8 (Elected)

CH2: 10*2 + 1* (-4) = 16

Now, CH1 gets elected due to the W1 consideration. Since the sum of inter CHdistances is greater in CH2 than in CH1 therefore to minimize the energy dissipated in the network we elect CH1 as the SCH.

Again let,

W1=1 and W2=10

CH1: 1*1 + 10*(-2) = -18

CH2: 1 *2 + 10*(-4) = -38 (Elected)

Now, CH2 gets elected as the SCH due to W2 consideration. Since the energy required in the transmission is less in CH2 than in CH1.

Cluster-heads can be chosen stochastically (randomly based) on this algorithm:

If n <T(n), then that node becomes a cluster-head.

The algorithm is designed so that each node becomes a cluster-head at least once.

$$T(n) = \frac{P}{1-P*(r \bmod P^{-1})} \forall n \quad G \qquad T(n) = 0 \quad n \quad G$$

Where n is a random number between 0 and 1

P is the cluster-head probability and

G is the set of nodes that weren't

cluster-heads the previous rounds

This version has a deterministic threshold algorithm, which takes into account the amount of energy in the node.

$$T(n)_{new} = \frac{P}{1-P*(r \bmod P^{-1})} \frac{E_{n\_current}}{E_{n\_max}}$$

Where $E_{n\_current}$ is the current amount of the energy and

E_{n_max} Is the initial amount of energy.

T(n)$_{new}$ =

$$\frac{P}{1-P*(r \bmod P^{-1})}\left[\frac{E_{n\_current}}{E_{n\_max}} + (r_s \, div \, P^{-1})\left(1 - \frac{E_{n\_current}}{E_{n\_max}}\right)\right]$$

V. NETWORK SIMULATION (NS2)

In 1996-97, work on ns version 2 (ns-2) was initiated based on a refactoring by Steve Mc Canne. The core of ns-2 is also written in C++, but the C++ simulation objects are linked to shadow objects in OTCL and variables can be linked between both language realms. Simulation scripts are written in the OTCL language, an extension of the TCL scripting language. This structure permits simulations to be written and modified in an interpreted environment without having to resort to recompiling the simulator each time a structural change is made. In the timeframe in which ns-2 was introduced (mid-1990s), this provided both a significant convenience in avoiding many time-consuming recompilations, and also allowing potentially easier scripting syntax for describing simulations. ns-2 has a companion animation object known as the Network Animator, nam-1, originally written by Mark Handley, used for visualization of the simulation output and for (limited) graphical configuration of simulation scenarios.

Network simulation software enables us to predict behavior of a large-scale and complex network system such as Internet at low cost under different configurations of interest and over long period. Many network simulators, such as NS2, Openet, Qualnet, etc., are widely available. We'll use NS2 for this project. NS2 is a discrete event simulator written in C++, with an OTCL interpreter shell as the user interface that allows the input model files (TCL scripts) to be executed. Most network elements in NS2 simulator are developed as classes, in object-oriented fashion. The simulator supports a class hierarchy in C++, and a very similar class hierarchy in OTCL. The root of this class hierarchy is the TCL object in OTCL.

Users create new simulator objects through the OTCL interpreter, and then these objects are mirrored by corresponding objects in the class hierarchy in C++. NS2 provides substantial support for simulation of TCP, routing algorithms, queuing algorithms, and multicast protocols over wired and wireless (local and satellite) networks, etc. It is freely distributed, and all source code is available. Developing new networking protocols and creating simulation scripts are complex tasks, which require understanding of the NS2 class hierarchy, C++, and TCL programming. However, in this project, design and run simulations in TCL scripts using the simulator objects without changing NS2 core components such as class hierarchy, event schedulers, and other network building blocks. put all the results in the files with given names.

VI. IMPLEMENTATION AND WORK FLOW

5.1 Network Construction

Define a very simple topology with two nodes that are connected by a link. A new node object is created with the command '$ns node' the simulator object will connect the nodes with a duplex link with the bandwidth, a delay of and a Drop Tail queue. Next is to send some data from node to another node by creating an agent object that sends data from node to other agent object that receives the data on node. A CBR traffic generator is attached to set packet size, time interval. create a Null agent which acts as traffic sink and attach it to node. Tell the CBR agent when to send data and when to stop sending. the simulator object should give time interval for simulation to execute the 'finish' procedure.

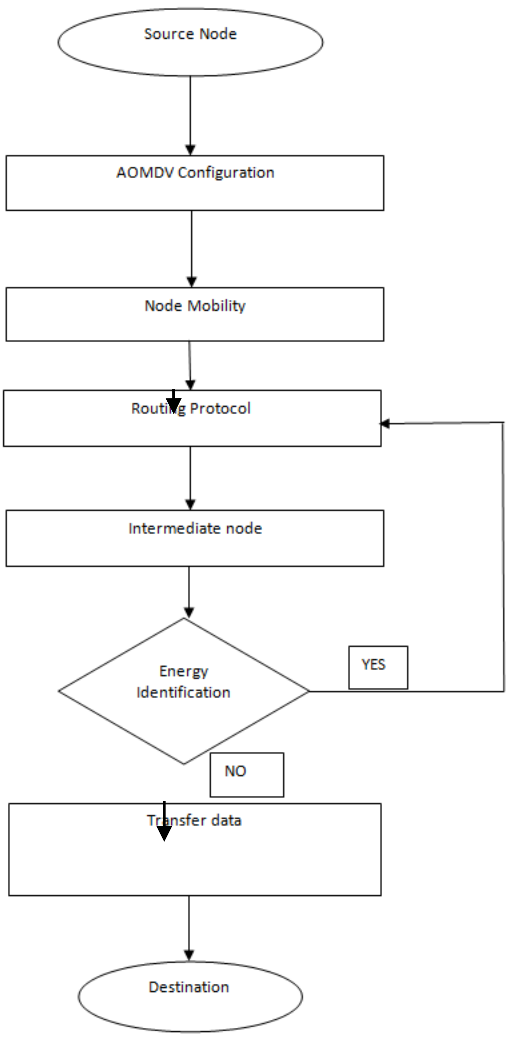

Figure 1. Flow Chart

5.2 Data Transfer Between Nodes

The Destination node that is present in the network. The cluster head is formed to save the energy of the nodes and also the life time of the network. It should be able to communicate with all the other nodes in the cluster and it should have the maximum energy when compared to all the other nodes. The sensor nodes periodically transmit there data to the corresponding nodes. The Nodes aggregate the data and transmit them to the Base or Destination either directlPy communication with other nodes. Because the nodes send all the time data to higher distances than the common nodes, they naturally spend energy at higher rates. But a problem arises when the Node got attack are not able to communicate.

5.3 Aomdv Protocol Implementation

An AOMDV referred as a pattern of normal system activity to identify active attempts. It is effective to detect new attacks. Because it doesn't maintain any database, but they continuously monitor traffic patterns or system activities. The deviations from this pattern may cause alarm to be triggered. To form node in a single network. The formation of the head is the same process as formed the cluster head in the previous module. All nodes should have both the properties .it should be able to communicate with all nodes and even with the other nodes.

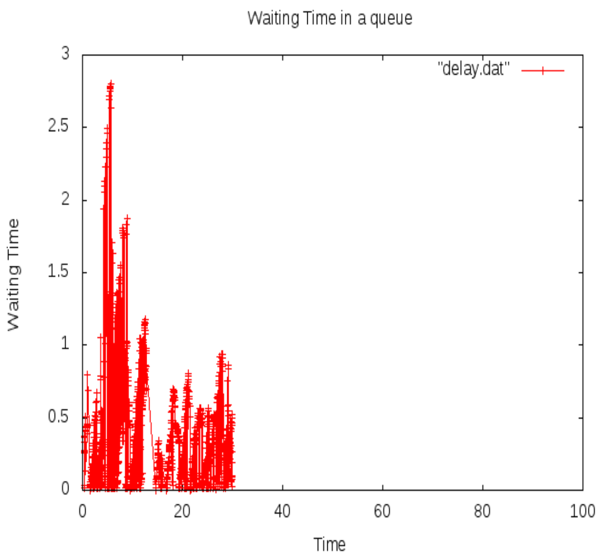

Figure 2. Average Delay

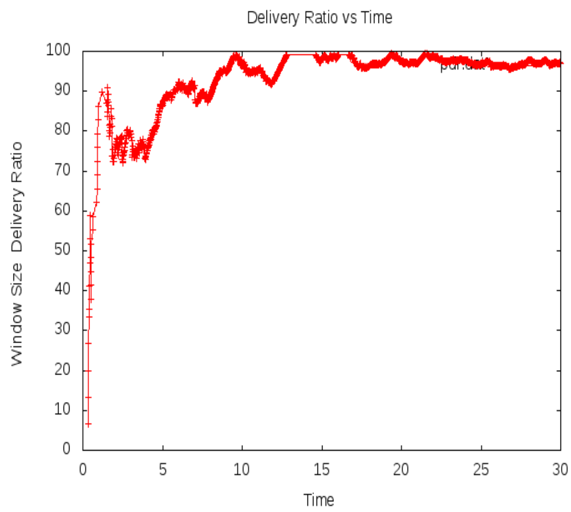

Figure 3. Packet Delivery Ratio

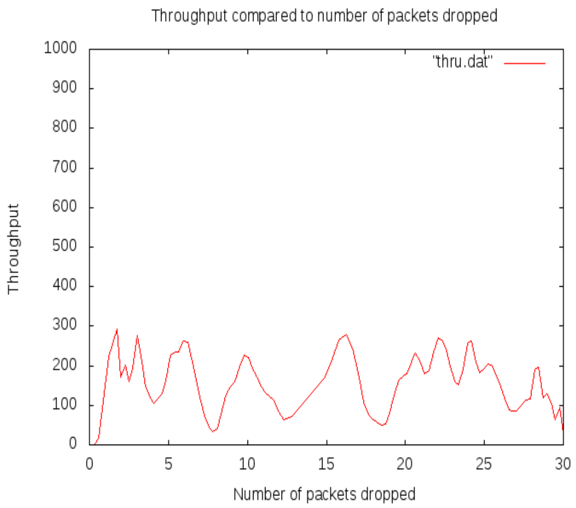

Figure 4. Throughput

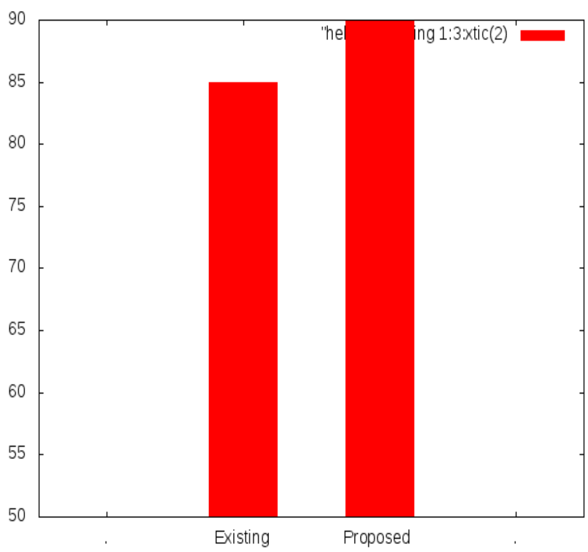

Figure 5. Energy Comparison

VII. CONCLUSION

This system evaluates feasibility and performance through both theoretical analysis and simulations. The results shows that different operational modes of can be configured to provide optimal performance in a variety of network configurations depending largely on the application of the sensor network. The results shows that this system performs better (in the worst case between 60-100 percent improvement in energy savings) than others while providing support for communication error handling, which was not the focus of earlier studies.

VIII. REFERENCES

[1]. Zhang, X., Zhang, Q., Peng, J., Zhao, Y., & Liu, W. (2017, July). A lifetime optimization mobile data gathering strategy with adaptive compressive sensing in WSN. In Control Conference (CCC), 2017 36th Chinese (pp. 8970-8975). IEEE.

[2]. Shen, J., Wang, A., Wang, C., Hung, P. C., & Lai, C. F. (2017). An Efficient Centroid-Based Routing Protocol for Energy Management in WSN-Assisted IoT., 5, 18469-18479. IEEE.

[3]. Feng, H., & Dong, J. (2017). Reliability analysis for WSN based on a modular k-out-of-n system. Journal of Systems Engineering and Electronics, 28(2), 407-412.

[4]. Jing, Y., Yuzhi, Z., Dan, D., Xiao, W., Ping, Y., Lingfang, C.,& Zetao, L. (2017, July). An early warning system of diseases and pests for blueberry based on WSN. In Control Conference (CCC), 2017 36th Chinese (pp. 8885-8889). IEEE.

[5]. Li, S., Fang, H., & Chen, J. (2017, July). Energy efficient multi-target clustering algorithm for WSN-based distributed consensus filter. In Control Conference (CCC), 2017 36th Chinese (pp. 8201-8206). IEEE.

[6]. Xu, J., Liu, W., Lang, F., Zhang, Y., & Wang, C. (2015). Distance measurement model based on RSSI in WSN. Wireless Sensor Network, 2(08), 606.

[7]. Chandra, S. (2013). Wireless network interface energy consumption. Multimedia Systems, 9(2), 185-201.