

Delay Efficient Distributed Data Aggregation Algorithm in Wireless Sensor Networks

Prakashgoud Patil

B.V..B. College of Engineering and Technology
Hubli – 580031, Karnataka, India

Umakant Kulkarni

S.D.M. College of Engineering and Technology
Dharwad – 580002 Karnataka, India

ABSTRACT

In wireless sensor networks data aggregation is a very important one and at the same time the aggregation should be energy efficient and a lesser amount of delay. To solve the problem we propose Delay Efficient Distributed Data Aggregation (DEDA) Scheduling Algorithm for Wireless Sensor Network (WSN) to handle the delay and energy tradeoff in the process of aggregation using the timeout concept. In this paper, first we are building the aggregated tree. After building the aggregation tree the Distributed Data Aggregation Scheduling algorithm to achieve the optimized energy efficiency and delay aware data aggregation. Decision Making Unit (DMU) has been used to handle energy and delay trade off. The DMU will control the tradeoff between energy and delay by generating the timeout values. The main advantage of this approach is to achieve the ideal energy consumption by limiting a number of redundant and unnecessary responses from the sensor nodes.

Keywords

Wireless Sensor Network, Data Aggregation, Dynamic Timeout and Energy Efficiency.

1. WIRELESS SENSOR NETWORK

Sensor networks contain a set of sensor nodes and each node is equipped with one or more sensors, storage, processing resources and communication subsystems. The sensors in a node observe facts such as temperature, light, sound, seismic, and acceleration events, while the processing and other components analyze the raw data and formulate answers to specific user requests [1]. A sensor network is composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it. Recent advances in technology have covered the design and implementation of new generations of sensor network nodes.

A wireless sensor network is a collection of autonomous nodes organized into a cooperative network. Each node consists of processing capability, may contain multiple and different types of memory (program, data and flash memories), have a RF transceiver (usually with a single omnidirectional antenna), have a power source, and accommodate various sensors and actuators. The nodes communicate wirelessly and often self-organize after being deployed in an ad hoc fashion. Systems of 100s or even 1,000 of nodes are anticipated. Such systems can revolutionize the way we live and work. [2].

This technology provides the unlimited potential to the numerous application areas like environmental, medical, military, transportation, entertainment, crisis management, homeland defense, and smart spaces. Currently, wireless sensor networks are beginning to be deployed at an accelerated pace. It is not unreasonable to expect that in the next 10-15 years that the world will be covered with wireless sensor networks with access to them via the Internet. This can

be considered as the Internet is becoming a physical network. [3]

The key factor that makes wireless sensor networks so unique and promising both in terms of research and economic potentials, is their ability to be deployed in very large scales without the complex pre-planning, architectural engineering, and physical barriers that wired systems have faced in the past. For wireless sensor networks, the systems are wireless, have scarce power, are real-time, utilize sensors and actuators as interfaces, have dynamically changing sets of resources, aggregate behavior is important and location is critical. Many wireless sensor networks also utilize minimal capacity devices which places a further strain on the ability to use past solutions. [4].

1.1 Data aggregation in WSN

The process of gathering information and presenting in a summary form is called as data aggregation. In data aggregation, the data is gathered or collected from multiple sensors to eliminate the redundant transmission and provide fused information to the base station. Data aggregation usually involves the combination of data from multiple sensors at intermediate nodes and transmission of the aggregated data to the base station (sink). Most of existing data aggregation algorithms are, however, not feasible for WSNs owing to their size and complexity. [5]

Data aggregation tries to collect the most critical data from the sensors and make it accessible to the sink in an energy efficient manner with minimum data latency. Data latency is important in many applications such as environmental monitoring where the freshness of data is also an important factor. It is critical to develop energy efficient data aggregation algorithms so that network lifetime is enhanced. The factors such as network architecture, the data aggregation mechanism and the underlying routing protocol are determining the energy efficiency of a sensor network. In this paper, we describe the influence of these factors on the energy efficiency and delay of the network in the context of data aggregation. [6]

We are using the aggregated result of sensor data at the sink nodes in making the important decisions because WSNs are not always reliable, it cannot be expected that all nodes reply to all requests. To do this we must know the sensor information such as Node Identifications, IDs and the communication cost of transmitting IDs of all contributed sensor nodes along with the aggregated data to be minimized.

Eliminating redundancy, minimizing the number of transmissions and thus saving energy are the main idea and main usage of aggregating the data. This model shifts the focus from the traditional address-centric approaches for networking (finding short routes between pairs of addressable end-nodes) to a more data-centric approach (finding routes

from multiple sources to a single destination that allows in-network consolidation of redundant data). [7]

1.2 Energy and Delay Trade Off in WSN

Saving energy is a very critical issue in wireless sensor networks since sensor nodes are typically powered by batteries with a limited capacity. Since the radio is the main cause of power consumption in a sensor node, transmission/reception of data should be limited as much as possible. In the process of communication some duplicate or unnecessary data may be received by sensor nodes multiple times. For an example, let us assume that the user is interested to find the max value of the sensors readings. For this query, there is only one needed value and the other values are the same or smaller than it are redundant. Therefore, such redundant messages would increase the average energy consumption of network. [8]

The main idea in the data gathering scheme is each sensor should have expended the same amount of energy in each data gathering round and it is the energy efficient if it maximized the functionality of the network. This idea is captured by the network lifetime which quantifies the energy efficiency of the network. In data aggregation algorithms the main important metrics are network lifetime, data accuracy, and latency. The definitions of these measures are highly dependent on the desired application. The design of efficient data aggregation algorithms is an inherently challenging task. There has been intense research in the recent past on data aggregation in WSNs.

There are lot of other factors for aggregation delay For example.- Resolving collisions in MAC layer could incur a large amount of energy consumption and a large delay during aggregation. We have to make the tradeoff between the energy and delay. We need the low delay and energy efficient algorithm for the data gathering and aggregation in wireless sensor network.

In this work, we propose delay aware and energy optimized distributed data aggregation algorithm using dynamic timeout concept for wireless sensor networks.

2. LITERATURE REVIEW

Xiaohua Xu et al., [10] have proposed efficient distributed algorithm that produces a collision-free schedule for data aggregation in WSNs. This paper focuses on data aggregation scheduling problem to minimize the delay (or latency). In this paper, authors study the problem of distributed aggregation scheduling in WSNs and propose a distributed scheduling method with an upper bound on delay. This is a nearly constant approximate algorithm, which significantly reduces the aggregation delay.

Changhee Joo et al., [11] have formulate the scheduling problem that minimizes the delay sum of data in wireless sensor networks, and qualify the gain of in-network aggregation techniques for tree networks. They formulate a delay minimization problem in the presence of wireless interference, and quantify the gain of the in-network aggregation for reducing the total delay in wireless sensor networks. They provide a system wide lower bound on the delay performance of the optimal joint data aggregation and scheduling policy. The specific bounds depend on the underlying network topology and the associated weight assignment. Under certain network settings, it is shown that the bound can be achieved. They evaluate our analysis using binary trees with different weight assignments.

Mohammad Hossein Anisi et al., [12] have proposed an in-network data aggregation approach. This approach achieves ideal energy consumption by limiting a number of redundant and unnecessary responses from the sensor nodes. They could limit a number of redundant and unnecessary responses from the sensor nodes. Also this scheme makes use of the available path redundancy in the network to deliver a correct aggregate result to the data sink. It can increase the chance of receiving data packets at the destination and cause to more accurate results.

Xundan Shi et al.,[13] have proposed Enhanced Greedy Aggregation (EGA) algorithm for Directed Diffusion Protocol (DDP) to encourage multiple-path sharing. It encourages multiple paths sharing and early path sharing. It also improves energy efficiency for the case with multiple sources and sinks. This method does not introduce extra energy cost for diffusing interests and exploratory events. It trades off delay for energy efficiency.

Soonmok Kwon et al., [14] have proposed novel timeout control scheme which dynamically changes timeout according to the amount of currently accumulated data in a node. First they perform an in-depth analysis of the data aggregation process and validate the strong relationship between the amount of currently accumulated data in a node and the data aggregation timeout to achieve energy efficiency with little delay overhead. Their scheme is based on local information without involving control messages so that it is not only robust but also light-weight.

The approach which is proposed in [14] is a novel timeout control scheme. They designed a trade-off mechanism which dynamically varies timeout according to the amount of currently accumulated data elements in a node. Their solution is a network layer protocol which is composed of the functional unit and the decision making unit.

The following drawbacks are observed in this approach:

1. This scheme is not ideal energy consumption because it cannot limit the number of unnecessary responses from the sensor nodes.
2. This method is not a collision free because each sensor node does not have the capability of detecting whether a collision occurs or not.

We propose Delay Efficient Distributed Data Aggregation Algorithm for Wireless Sensor Network.

3. OVERVIEW

In this paper, first we use in-network data aggregation algorithm in which the aggregating tree is built. The node has Primary, Backup and Side edges. If node is dropped due to any error, the node data can be recovered using primary and, backup and side edges and these three methods are useful in error recovery. After building the aggregation tree, we apply the Distributed Data Aggregation Scheduling algorithm. Every node transmits its data to the neighboring nodes with the lowest level. Data are aggregated from children to their parents, and finally, to the root node of the tree. A time slot is allocated for a node to transmit, whenever the node has collected the aggregated data from all its children nodes in the data aggregation tree T. The interleaving of these two stages will reduce the delay greatly since it increases the number of simultaneous transmissions.

The Decision Making Unit (DMU) has been used to handle energy and delay tradeoff in the aggregation. The DMU will

control the tradeoff the energy and delay by generating the timeout values. The time out values are used in each node to achieve the goal.

3.1 System Architecture

The tree given in Figure 1 indicates the system architecture of WSN used in the proposed work. The tree contains the gateway node and parent node. Each node has one parent node and gateway node. The gateway node is connected to all the parent nodes. Each node in the tree has the one parent node and one backup node.

The parent node will broadcast the messages to the all other nodes in network and build the aggregated tree in the network. All the other nodes will act as child nodes and the node which is in higher order that is selected as parent node. The message will reach to the child node through the parent node. When the node in the aggregated tree receives the message, then the node will make the sender of the packet as parent node.

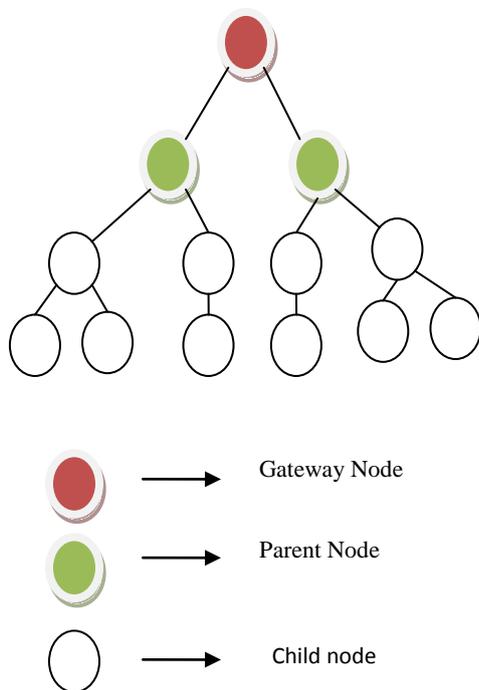


Figure 1 : System Architecture

3.2 Data Aggregation Tree

In this method, first the aggregation tree has been built. During the process of building the tree, the first gateway node or parent node will broadcast the messages to the other nodes. When the other nodes receive the message, the node will check whether the message is received for the first time. If this is received for the first time then, it will accept that node as its father / parent. Like this all the nodes will receive the messages and selects the parent nodes and thus a routing tree will be formed.

If the node gets the same message second time form some other node then the sender of that message is selected as its backup parent. This is a very important step in this algorithm. If the node gets the same message for the third time then the node will ignore that message.

Consider the Figure 2; initially there are no parent nodes in the network. In the process of building the tree each node will select its parent node and backup node. The node which is

parent to some nodes at the same time it may act as backup node to the other nodes. For building the aggregation tree we have formulated an algorithm which is as given below (Algorithm 1)

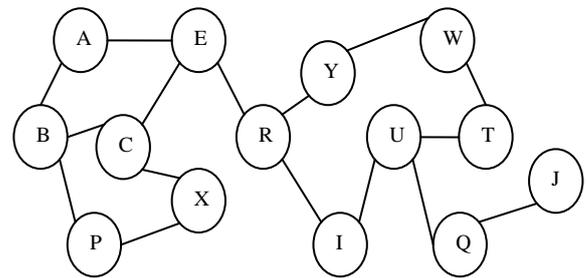


Figure 2: Before Aggregation

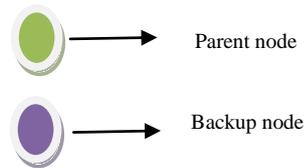
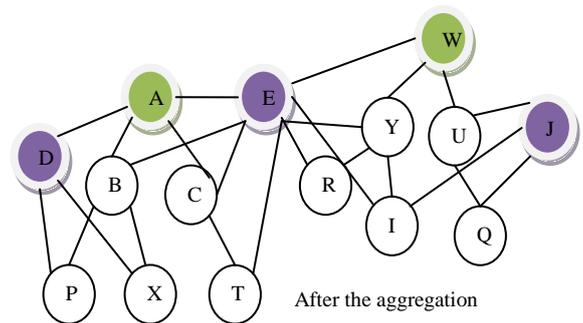


Figure 3: After Aggregation

1. Start
2. Define Counter =0: // Each node have the counter and initial value of that counter is zero
3. Parent node send the messages to all the other nodes
4. If (node.counter = 0)
 - 5. Accept the sender of that message as parent node
 - 6. Node.counter = 1;
 - 7. }
7. Else if (node.counter = 1)
 - 8. Add the sender of that message as its backup parent
 - Else
 - 9. Ignore all the messages
10. End

Algorithm 1: For Building the Tree

3.3 Error Recovery in Data Aggregation

Every node in the network has one primary parent and one backup parent. Every node in the tree connected to its parent and backup parent through edges as shown in the graph. The edges in the aggregation tree divided into three types:

primary, backup and side edges. The primary parent and backup parent are associated with the side edges. The node can be primary node and the same time it can be backup node.

The packet loss may happen at any node primary node, backup node and side edge. The lost packets are recovered from the child node by using the side edges or backup parent. If the child node detects any packet loss, then, it will send the packet to the backup node. The parent node will associate with primary node and send to the child node using the side edges

If the loss occurs in the side edges then the child node sends the packet to the associate primary node or backup node. The primary node will resend the packet to the child node through another side node.

3.4 Example of Error Recovery in Data Aggregation Tree

We will recover the node by using the primary edge, backup edge and side edge. Consider the diagram given in Figure 4. In the diagram, each node has one parent node and one backup node. The node which is a backup node for one node and that backup node may be the parent node for another node.

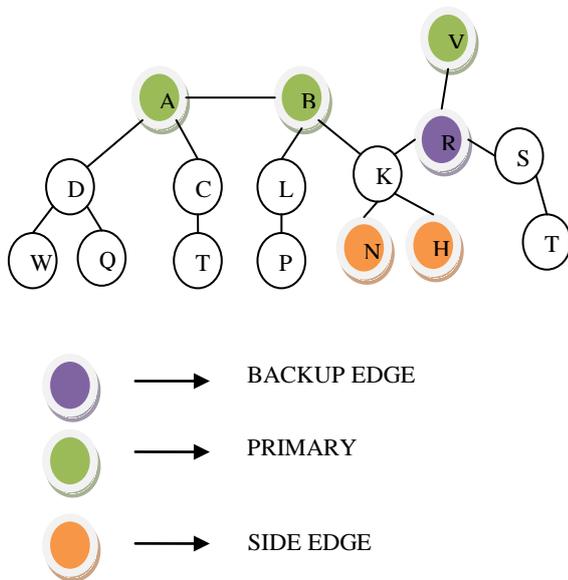


Figure 4: Error Recovery in Data Aggregation

Let's take the node K. The node K has the primary edge B, Backup edge R and Side edge N and H. The node K is used in the routing. Assume node K will be dropped due to any error. Now we have to recover the node K using the primary edge and backup edge. When the child node notices the node K was disappear then it will send the packet to the backup edge and side edges for recover that node. The backup edge and side edges have the information of the node and it will recover the node.

3.5 Distributed Data Aggregation Scheduling Algorithm

In Distributed Data Aggregation scheduling algorithm, all links are scheduled with TDMA in the tree such that the delay of aggregating all data to the sink node is approximately minimized. This algorithm will build the TDMA schedule of

the node based on the aggregate tree which was built as per the description provide in the section 3.1.

The time slot is allocated to leaf nodes to transmit their data without any conflict. First we will check is it leaf node or not. If it is leaf node then we will check for that leaf node has child nodes or not.

1. **Side node SN[i]:** these are the leaf nodes of the tree and it is to indicate whether the selected the node is leaf node in aggregated tree or not.
2. **Set of nodes S[i]:** It is the set of nodes such that each $j \in S[i]$. N_i and N_j are the two leaf nodes of the aggregated tree and both are unable to the send the data at a time due to the interference of parents. In this paper, each node in $S[i]$ is within a small constant number of hops of i .
3. **Ready to Send Set (RSS[i]):** It is the set of nodes that are received the data from all its children nodes and ready to send the parent. These are set of nodes which they collides with i .
4. **Transmit time slot (TTS[i]):** This is the time slot that is assigned to node N_i to send the data to its parent.
5. **Number of Childs (NC[i]):** This is the total number of the child nodes that node N_i has in the aggregated node.

We will use the above terms in the following algorithm

1. Start
2. Define $T =$ Data aggregated tree
3. $TTS[i] = 0;$
4. $Status = FALSE;$
5. Node N_i get the values of $NC[i]$ and $SN[i]$ based on the T
6. Initializes the set $S[i]$ based on the tree T
7. We will create the $RSS[i]$ using the given equation
8. $RSS[i] = CS[i] \cap j$ (j is an leaf node in T)
9. Node N_i select an random integer r_i to check the interference nodes
10. If $((r_i < r_j) \& ((r_i = r_j))$
{
11. If $(NC[i] = 0)$
{
12. Send message $READY(i, r_i)$ to all nodes in $S[i]$
13. If $((r_i, i) < (r_j, j))$ for each $j \in RSS[i]$ then
14. Send message $FINISH(i) \& TTS[i]$ to all nodes in $S[i]$
15. $Status = TRUE;$
}
16. Else
17. Go to the leaf node of that node
}
18. If $(N_i$ received a message $FINISH(j) \& TTS[i])$
{
19. Delete j from $RSS[i]$
20. $TTS[i] = \max (TST[i], TST[j]+1);$

21. If j is a child of i then
22. $NC[i] = NC[i] - 1;$
}
23. If (N_i received a message $READY(j, r_j)$)
{
24. If j is in $CS[i]$ then
25. Add j to $RSS[i];$
}
26. End

Algorithm 2: For Distributed Data Aggregation Scheduling algorithm

In the algorithm, first we will get the node number (N_i) of the child's and leaf node from the aggregated tree. Then create the set of nodes using that aggregated tree. After this, create the ready to send set. The RSS will contain the nodes which are ready to transmit.

The random integer is taken to check interference is exist in the aggregated tree or not. If the interference is there then the delay will be increased so we have to create no interference in the nodes. If the single node is present then we will send the ready message to all the nodes. We will send finish message to all the nodes because the node does not have any other leaf nodes. If it contains any other leaf nodes then it will go to the leaf node and again the same process will run. When the node gets the ready message then a time slot will be allocated to node to transmit the packets. Initially the time to transmit slot is 0 and the status is false. When process starts the values will change based on the execution.

3.6 HANDLING DELAY AND ENERGY TRADEOFF

Delay and energy tradeoff is handled by using the decision making unit (DMU). By fine tuning the timeout the DMU controls the energy delay performance of the system. The DMU will generate timeout values with the parameters and that parameters are maintained at each node at the initialization of the network.

- C: Aggregation limit
- PA: Packet arrivals per unit time
- TF: Energy-Delay Tradeoff Factor which is controlled between zero and one.
- E: Energy trade off factor
- D: Delay trade off factor
- AD: Currently accumulated data elements

If the TF is high then delay is low. The TF value is between one to zero. The energy E is always less than the $-8*(D/C)$. If the energy is low then we will increase the value by using the following equation-1.

$$D = \frac{20}{PA} (1 - TF^2) \quad (1)$$

In the equation (1), we will calculate the energy factor. If the energy is low then we will increase the delay by the using equation. The delay factor is calculated from the following equation -2.

$$E = -1.8^{TF} * \frac{D}{C} \quad (2)$$

The delay factor is controlled using the equation (2). TF is the trade off factor between the energy and delay. This scheme will successfully balance the tradeoff the energy and delay. It achieves high energy efficient with little delay overhead and vice versa.

3.5 Total Workflow:

In this section the total work flow of our proposed work is summarized. At the beginning, the aggregation tree is built. The tree contains two parents one is primary and second one is secondary. If any errors occur then it will recovered using the recovery method.

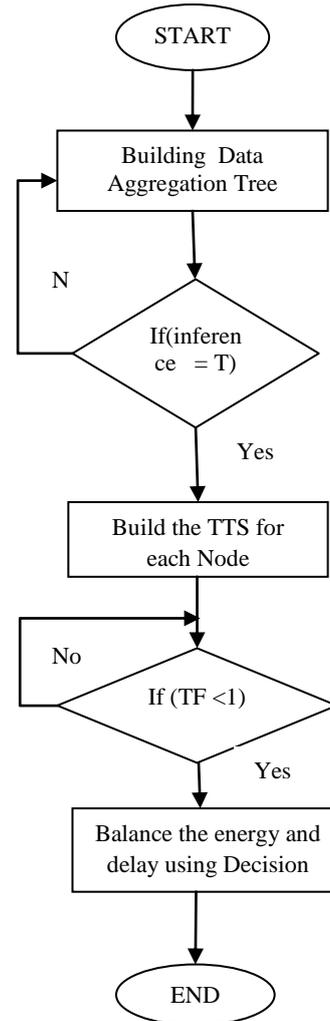


Figure 5: Flow chart for total work flow

After building the aggregated tree the distributed data aggregation scheduling algorithm is applied. In the distributed data aggregation scheduling algorithm the time slot is assigned to each and every node in the aggregated tree. The node will transmit their data during the assigned time slot.

The decision making unit (DMU) is used to handle energy and delay tradeoff. The DMU will control the tradeoff the energy and delay by generating the timeout values and these are the values that are used in each node.

Using the proposed approach, we achieved the ideal energy consumption by limiting a number of redundant and

unnecessary responses from the sensor nodes thus the energy and delay parameters are optimized for the best result.

4. SIMULATION RESULTS

4.1 Simulation Model and Parameters

We use NS2 [15] to simulate our proposed DEDA protocol. In our simulation, the channel capacity of mobile hosts is set to 2 Mbps. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs has been used as the MAC layer protocol. It has the functionality to notify the network layer about link breakage.

In the simulation, the size of the sensor network is set to 750 meters x 750 meters. The simulation was carried out for 50 seconds. All nodes have the same transmission range of 250 meters. The mobile nodes move within the specified area of wireless sensor networks and the node speed is fixed as 5m/s. The simulated traffic is Constant Bit Rate (CBR). The simulation settings and parameters along with their values are summarized in Table 1.

Table 1. Simulation Parameters

No. of Nodes	10,20,30 and 40
Area Size	750 X 750
Mac	802.11
Routing Protocol	DEDA
Radio Range	250m
Simulation Time	50 sec
Traffic Source	CBR
Packet Size	512 bytes
Mobility Model	Random Way Point
Speed	5m/s
Rate	50,100,150,200 and 250Kb
Initial Energy	7.1J
Receiving Power	0.395
Sending Power	0.660
Idle Power	0.035

4.2 Performance Metrics

The performance evaluation of the proposed algorithm is measured using the following metrics.

Average Energy: It is the average energy consumption involved in the entire data transmission.

Average Packet Delivery Ratio: It is the ratio of the number of packets received successfully and the total number of packets transmitted.

Delay: It is the total amount of time taken by the packet to receive by the receiver.

Throughput: It is the total number of packets received by the receiver versus sent.

We compare our DEDA with the Distributed Data Aggregation protocol. The simulation results are presented in the next section

4.3 Results

A. Based on total number of nodes

In our first experiment we vary the number of nodes as 10,20,30 and 40.

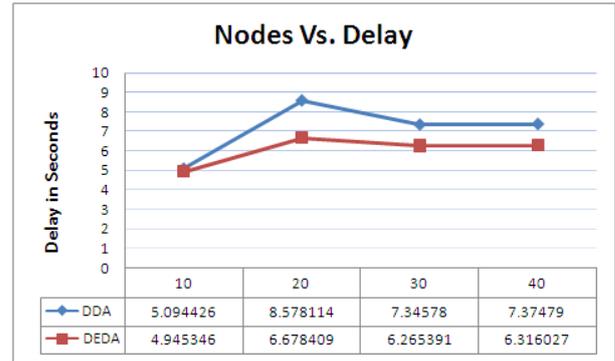


Figure 6: Nodes Vs Delay

It is observed from Figure 6 that the delay of proposed DEDA is less than the existing DDA protocol when the number of nodes in the network ranges from 10 to 40. From Figure 7, It is noticed that the delivery ratio of proposed DEDA is higher than the existing DDA protocol. The proposed algorithm gives better performance in terms of energy consumption. From Figure 8, it is noticed that, the energy consumption of our proposed DEDA is less than the existing DDA protocol. Figure 9 shows that the throughput of proposed DEDA is higher than the existing DDA protocol. Overall, the proposed algorithm performs better with respect to delay, Delivery ratio, energy and throughput.

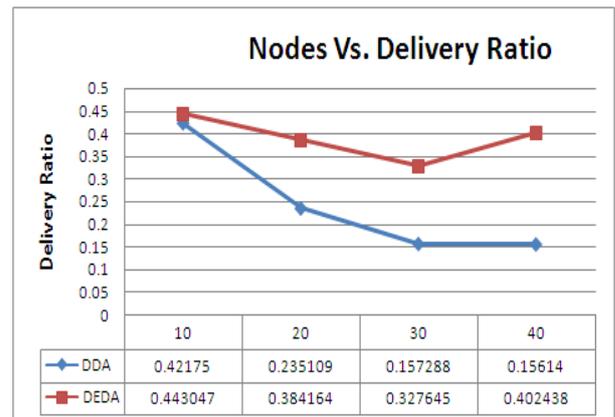


Figure 7 : Nodes Vs Delivery Ratio

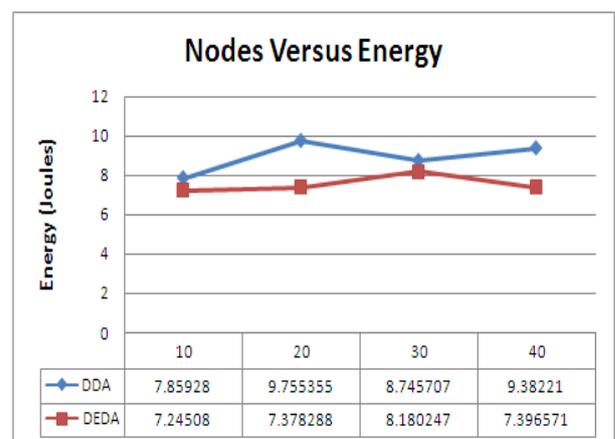


Figure 8 : Nodes Vs Energy

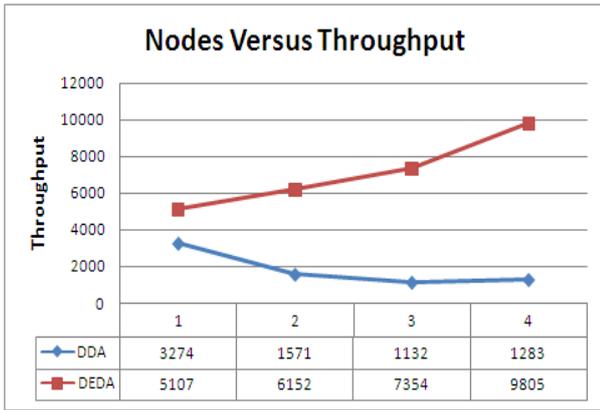


Figure 9: Nodes Vs Throughput

B. Based on Rate

In the second experiment, we vary the data rate from 50,100,150,200 and 250Kb. It is observed from Figure 10 that, the delay of our proposed DEDA is less as compare to the delay in DDA protocol. The delivery ratio of our proposed DEDA is higher than the existing DDA protocol which is as shown in the Figure 11. It is also observed from the figure 12 that , the energy consumption of our proposed DEDA is less than the existing DDA protocol. The throughput of proposed DEDA is higher than the existing DDA protocol which is shown by the figure 13.

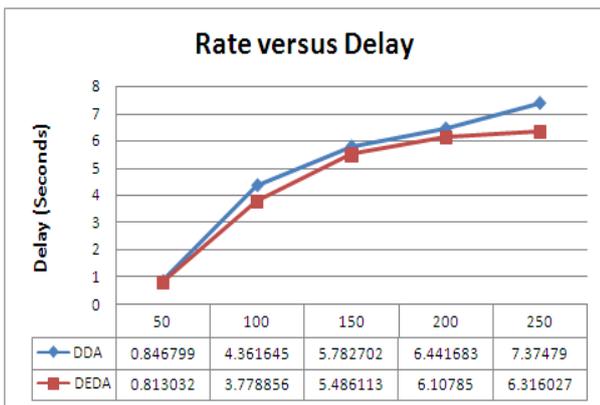


Figure 10: Rate Vs Delay

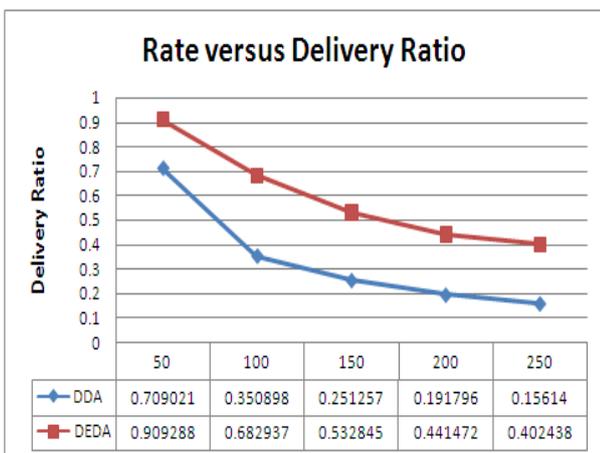


Figure 11: Rate Vs Delivery Ratio

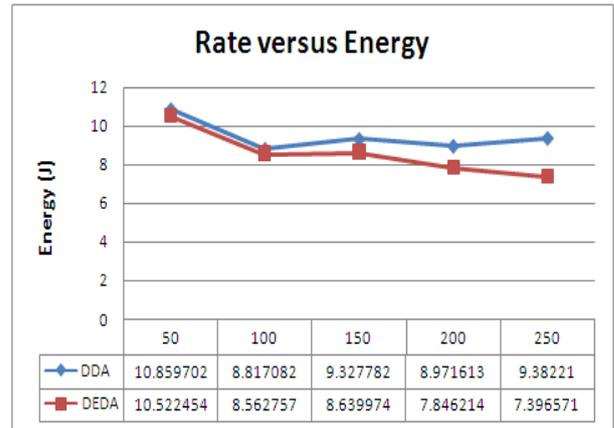


Figure 11: Rate Vs Delivery Ratio

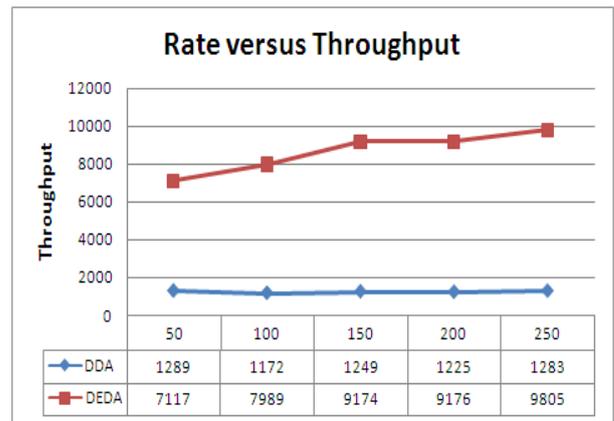


Figure 13: Rate Vs Throughput

5. CONCLUSION

In wireless sensor networks data aggregation is a very important one and at the same time the aggregation should be energy efficient and a lesser amount of delay. To solve the problem we proposed a Delay Efficient Distributed Data Aggregation (DEDA) Scheduling Algorithm for Wireless Sensor Network (WSN) to handle the delay and energy tradeoff in the process of aggregation using the timeout concept. The Decision Making Unit (DMU) has been used to handle energy and delay tradeoff in the aggregation. The DMU will control the tradeoff energy and delay by generating the timeout values. The performance of the proposed algorithm is measured using various QoS parameters such as Energy Efficiency, Delay, throughput under various conditions of sensor network by changing the total number of nodes and data rate. The proposed algorithm performs better than the existing distributed data aggregation algorithm.

6. REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", Elsevier, Page No: 393–422, 2002.
[2] John A. Stankovic, "Wireless Sensor Networks", University of Virginia, June 19, 2006.

- [3] F. L. LEWIS, “Wireless Sensor Networks”, Smart Environments: Technologies, Protocols, and Applications, New York, 2004.
- [4] Seapahn Megerian and Miodrag Potkonjak, “Wireless Sensor Networks”.
- [5] Cinzia Cappiello and Fabio A. Schreiber, “Experiments and analysis of quality and energy aware data aggregation approaches in WSNs”.
- [6] Ramesh Rajagopalan and Pramod K. Varshney, “Data aggregation techniques in sensor networks: A survey”, 2006.
- [7] Rabindra Bista and Jae-Woo Chang, “Energy-Efficient Data Aggregation for Wireless Sensor Networks”.
- [8] Ignacio Solis and Katia Obraczka, “In-Network Aggregation Trade-offs for Data Collection in Wireless Sensor Networks”, 2005.
- [9] Kemal Akkaya, Mohamed Younis and Moustafa Youssef, “Efficient Aggregation of Delay-Constrained Data in Wireless Sensor Networks”, *IEEE*, 2005.
- [10] Xiaohua Xu, Xiang-Yang Li, Xufei Mao, Shaojie Tang, and Shiguang Wang, “A Delay-Efficient Algorithm for Data Aggregation in Multihop Wireless Sensor Networks”, *IEEE*, 2011.
- [11] Changhee Joo, Jin-Ghoo Choi, and Ness B. Shroff, “Delay Performance of Scheduling with Data Aggregation in Wireless Sensor Networks”, *IEEE*, 2010.
- [12] Mohammad Hossein Anisi, Abdul Hanan Abdullah and Shukor Abd Razak, “Efficient Data Aggregation in Wireless Sensor Networks”, International Conference on Future Information Technology, 2011.
- [13] Xundan Shi and Alvin Lim, “An Energy-Efficient Data Aggregation Algorithm for Wireless Sensor Networks”, Science Academy Transactions on Computer and Communication Networks, 2011.
- [14] Soonmok Kwon, Jae Hoon Ko, Jeongkyu Kim and Cheeha Kim, “Dynamic timeout for data aggregation in wireless sensor networks”, Elsevier, 2010.
- [15] Network Simulator: <http://www.isi.edu/nsnam/ns>
- [16] Prakashgoud Patil, Umakant P Kulkarni, “Data Aggregation in Wireless Sensor Networks Using Fuzzy Aggregator”, International Conference on Soft-Computing and Engineering Applications – ICSEA2011, IRNet ISBN : 978-93-81583-01-2.
- [17] Prakashgoud Patil, Umakant P Kulkarni, “Some Issues in Clustering Algorithms for Wireless Sensor Networks”, IJCA Special Issue on “2nd National Conference-Computing, Communication and Sensor Network” CCSN, 2011.
- [18] Prakashgoud Patil, Umakant P Kulkarni, “Effect of Data Aggregation on Wireless Sensor Network Performance”, IEEE Conference - TRENDZ IN INFORMATION SCIENCES & COMPUTING – TISC 2011 ISBN 978-1-4222-9009-7