Performance Evaluation of improved Algorithm for Wireless Sensor Networks

Suresh Kumar Meena                                           Naveen Hemrajani
Assistant Professor, Deptt. of CSE,                                     Vice Principal,
Shri JJT University, Jhunjhunu, Rajasthan, India              SGVU, Jaipur, Rajasthan, India
suresh28skm@gmail.com                                           naven_h@yahoo.com

Abstract

Wireless sensor networks have recently come into prominence because they hold the potential to revolutionize many segments of our economy and life, from environmental monitoring and conservation, to manufacturing and business asset management, to automation in the transportation and health-care industries. Information collected by and transmitted on a sensor network describes conditions of physical environments for example, temperature, humidity, or vibration and requires advanced query interfaces and search engines to effectively support user-level functions.

Wireless sensor nodes can be deployed in various regions to collect the information but they have limitations like Limited power they can harvest, ability to cope with node failures (node failure problem), mobility of nodes, communication failures, heterogeneity of nodes, large scale of deployment and Load balancing problem. Node capacity is scalable, only limited by bandwidth of gateway node.

These are the existing problems in Wireless sensor nodes. The proposed algorithm in this paper handles load balancing, lifetime and mobility of nodes problems. The work has been simulated on OMNeT++ 4.0 simulator on Linux platform. And finally we have compared some results with an existing algorithm like Threshold sensitive Energy Efficient sensor Network algorithm and Low Energy Adaptive Clustering Hierarchy algorithm and have carried out comparisons/investigation on various parameters like average time for first node to die, network lifetime etc, the results are shown better in some cases for this proposed algorithm.

Keywords: Wireless sensor network, algorithm, bandwidth, mobility, Clustering.

1. Introduction

1.1 Wireless Sensor Network

A wireless sensor network consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. They are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control. [2] Wireless sensor nodes constitute part of a network, which is defined as “The networks consist of individual nodes that are able to interact with their environment by sensing or controlling physical parameters; these nodes have to collaborate in order to fulfilled their tasks as, usually, a single node is incapable of doing so; and they use wireless communication to enable this collaboration” [3].
1.2 System Architecture for the Sensor Nodes

A sensor network is composed of ten to thousand of the sensor nodes which are placed in the wide area. In the sensor network all the nodes are communicating with each other either directly or through the other nodes. TinyOS [4] is a component based operating system designed to run in resource constrained wireless devices.

1.3 Mobility in Wireless Sensor Nodes

In WSNs the participants are the sensor nodes, which according to the application are sinks, sources and intermediate sensor nodes. The sinks are identified as sensor nodes or external elements that interact with the network. Examples of sinks in the WSN are PDAs, Laptops or gateways to other networks. Mobile sensor networks are sensor networks in which nodes can move under their own control or under the control of the environment. Mobile networked systems combine the most advanced concepts in perception, communication, and control to create computational systems capable of interacting in meaningful ways with the physical environment, thus extending the individual capabilities of each network component and network user to encompass a much wider area and range of data. A key difference between a mobile sensor network and a static sensor network is how information is distributed over the network.

Wireless Sensor Networks (WSNs) are widely used network. These days, there are many researches running on in this field and many protocols have been designed for the Wireless Sensor Network. Most of these algorithms consider the Static Sensor Networks (SSNs). The Static Sensor Networks (SSNs) have various disadvantages such as first, less energy efficient, most of the gateway nodes loss their energy first means these nodes are die thus the whole network goes to die. Second, Static Sensor Networks (SSNs), the sensor nodes are static so its can not move to other places but in Mobile Sensor Networks the sensor node can move and reach the places where event is fired. Mostly the sensors are deployed randomly, as opposed to precisely, therefore there is often a requirement to move the sensor node for better sight or for close proximity.

1.4 Types of Mobility in WSNs

Considering the capabilities that wireless communication offers, the sensor nodes and the objects of interest can have mobility identified as:

a) Event mobility: Typical in applications with events generated for detection of objects, the one most commonly known is tracking. On these applications, the detection of the object(s) is by means of more than one sensor node.

b) Node mobility: Sensor nodes have the capability to adjust their position, offer better results for the task, or when the network is used to monitor a moving object. This kind of network requires a more flexible and self configurability behavior due to variables like the speed of the node, energy consumption for connectivity, requirements to maintain a good level of functionality and QoS of the WSN.

c) Sink mobility: It can be considering very different from the others, but can also be thought as very similar to the node mobility case. An example can be when the sink is an external element like a PDA or Laptop, whose function is to consult the information provided by the WSN.

1.5 Limitations and challenges in Wireless sensor Network

There are some limitations for WSN implementation. The sensor nodes have the various limitations such as low battery power, minimum computation capability, load balancing, network life time and mobility of nodes. The field of application of the WSN is now constrained by the capacity to instrument the nodes with sensors and by the required processing of the generated information. Some of the commonly applied sensors used are for measuring now, temperature, humidity, vibrations, pressure, brightness, mechanical stress, and proximity.

From the listed applications for Wireless Sensor Networks (WSNs) [5], it becomes clear that they require the interaction of elements or members that perform different tasks. The common participants in this WSN are:

- **Sources** that are in charge of generating the data of the task.
- **Intermediate nodes** that make additional processing or forwarding data, and
- **Sinks** where the information is received. In some applications, these sinks are part of the networks, whereas in others they are external elements that enquire information from the network.

Managing a wide range of application types in a WSN is hardly possible with a single conception and design of the wireless network. However, certain attribute identified are related to the characteristic requirements and the mechanisms of such systems. The realization of these characteristics with newer mechanisms is the major challenge foreseen to WSNs.
2. Review of Literature

Wireless Sensor Networks (WSN) design is influenced by many factors, which include fault tolerance, scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media, and power consumption [3].

2.1.1 Localization algorithm

We can divide our location algorithm into two parts based on the measuring system used: GPS based and GPS free localization algorithm.

A) GPS based localization Algorithms

In GPS based localization algorithm we require a GPS system because GPS system is the one way to find the position of the node. In this technique, few nodes commonly known as anchors, use GPS to determine their location using GPS system and, broadcast it position information in the network. This position information has cached by the other node in the network and with the help of this information other nodes (neighbour nodes) in the network calculate their own position without using GPS. Further, GPS based localization algorithm is divided into two parts, based on the range of the anchor nodes: Range-free and Range-Based. Monte Carlo localization algorithm [21], and color-theory-based dynamic localization algorithm [22] etc.

a) Sequential Monte Carlo localization Algorithm

The Monte Carlo localization algorithm is mainly used for the robot localization [34] which is based on its motion, perception, and possible pre-learned map of its environment. L. Hu and D. Evans [20] used this algorithm for mobile sensor node. This is the first range free localization algorithm for mobile sensor nodes. The localization of sensor nodes is more difficult than the localization of robot.

b) Color theory based Dynamic Localization algorithm

The colour theory based localization algorithm is proposed by the Shen-HaiSheet. al. [22], it comes under the category of Range-free localization algorithm. In this algorithm, a sensor node is represented by the set of RGB value. With the help of RGB value we can find the most possible location of the sensor node. This centralized localization algorithm is based on the colour theory to perform positioning in mobile wireless sensor networks.

B)GPS free localization Algorithms

GPS Based localization algorithm requires some GPS system which only suitable for outdoor environment and has several disadvantages: first, the availability of GPS signals, second, the availability of global positioning systems (GPS) which requires additional hardware at additional costs, third, size of the sensor nodes, fourth, power of the sensor nodes and last, the availability of a number of fixed-point reference nodes, or anchors, with globally known locations. Due to these reason we require an algorithm which is GPS free. The directional or GPS-free localization algorithm is the first localization algorithm given by Akcanet. al. [31].

a) Directional Localization Algorithm

The directional localization algorithm is a GPS-free localization algorithm [31], which has the following assumptions: first, each node has a compass pointing to north, second, Node can measure distance to their neighbors using a well known measurement method (i.e. TOA Time of Arrival [39] or signal strength [40]), third, Motion actuators allow each node to move in a specific direction (with respect to North), fourth, Actuator, compass and distance measurements are subjected to errors caused by various real world disturbances and fifth, Other than the above, we do not need any additional equipments or infrastructure.

b) GPS less, outdoor, self positioning algorithm

Hung-Chi Chu et. al. [39] has been proposing a GPS less localization algorithm. In this algorithm some sets of nodes are known as reference points (RP’s). These RP’s are deployed in a region, called deployment area and broadcast some packets, these packets contain the localization data. The other nodes in the networks receive these packets, process these packets and can easily localize it. The characteristics of self positioning algorithm are given as; it is a distributed self positioning algorithm, and the energy consumption is low because sensor nodes simply use the connectivity information.

3. PROPOSED ALGORITHM

This chapter contains the proposed algorithm for wireless sensor nodes that solves some existing problems like

- Load Balancing
- Energy Efficiency
- Node Life Time
- Network Life Time
- Handling Mobility situation for Wireless Sensor Nodes

To solve these problems we have followed the hierarchical algorithm’s concept and have assigned a level to each node as per their Euclidian distance from Base Station. In this algorithm we have taken two types of devices:

- Base Station
- Sensor Nodes
**Base Station:** It is responsible for transmitting and receiving of data/control packets.

**Sensor Nodes:** Sensor Node is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network.

### 3.1 Assumptions made for proposed algorithm:
- All the nodes in the network area will be laid in Hierarchical Levels.
- This Hierarchical level will be equivalent to a tree (data structure) with some node levels. Level 0 will be treated as the highest level.
- And all the node of the tree has some degree constraint which is depend on the level of the node.
- All the nodes are using GPS free algorithm to find their location.

**Step 2:** After assigning level, each node broadcast the joining request packet as shown in the figure 3.2, which contains the
- node id
- Level of the node.

**Figure 3.2 Request Messages to higher level nodes**

**Step 3:** When any node receives a join request:
- It first checks its parent and the level of the request. If the parent of the node is null (no parent) or level is higher than or equal to the node itself then node discard this request packet, otherwise the node check its node degree if the node degree is greater than or equal to the degree constrain and also discard the request,
- Otherwise response with a positive acknowledgement (response) as shown in following figure.

**Figure 3.3 Responses to the Request**

### 3.2 A proposed Hierarchy Based Algorithm
This hierarchical algorithm works in three stages:
- Hierarchical structure formation stage
- Data gathering and transmission stage
- Refinement stage.

#### 3.2.1 Stage1: Hierarchical structure formation stage:
The Hierarchical structure formation stage has the following five steps:

**Step 1:** In this stage,
- The base station broadcast an initial *init* message which contains the information about the position of the base station.
- After receiving the *init* message all the nodes first calculates the Euclidian distance from the base station. And according to this distance assign a level to itself.

**Figure 3.1 Sensor Nodes forming a Hierarchy**
Step 4: The requested node joins the node from which it gets the first request acknowledgement and as a child node to itself, then the parent node adds this node to its child list and increases the node degree count by 1.

Step 5: Steps 2-4 are repeated until the whole Hierarchy has been formed.

Stage 2: Data gathering and transmission stage:
This data gathering and transmission stage has three steps:

Step 1: After Hierarchical structure formation stage,
- Each node sends its child list to its parent node (immediate higher level node).
- According to the child list, the father node sends a TDMA schedule to its child node. In its schedule, the child node can send its data to the father node.

Step 2: In this step if
- If the child node has the data then it forwards its data to its parent node in its TDMA time slot.
- Otherwise, it sends a nack (no data) data to its parent node.
- CSMA/CA approach is used by the node to send the data as shown in figure 3.5.

Step 3: In this step if
- The parent node aggregates the data from its children and sends the aggregated data to its parent node.
- Finally, the node near to the base station sends its data to the base station. As shown in figure 3.5.

Figure 3.4 Data from lower level nodes to higher level nodes

Stage 3: Refinement stage
Hierarchical structure changes if following situations occur such as
- Failure of node
- Movement of the parent or child node.
- Energy level of the nodes goes below the certain level.

Step 1: When a node moves from one location to another, it changes its position. There are two possibilities regarding the movement of the nodes.
- The node either moves within the same level or
- One level above or below

When the position of the node changes, it localizes itself by localization algorithm. After calculating its position, the node calculates its distance from the base station and re-calculates its level.

Step 2: In this step if
- The level of the node does not change then the node checks that it is in the range of its parent or not. If it is within the range of its parent node then no need to re-join the Hierarchy.
- Otherwise, the node changes its level according to the distance from the base station and broadcasts a join request and adds the new node in its father node list and removes the old.

Step 3: Handling the node invalidation: The node invalidation condition arises in two ways

(a). Child node invalidation: when the data transmission takes place if the parent node does not get any response from any of its child, it adds this node to the invalid list and waits for the next time slot. In the next time slot, the node not receiving any response, this node will be inferred as an invalid child and removed from its child list.

(b). Parent node invalidation: Each node transmits the data after receiving the data request packet to its parent node. If any node does not receive the data request from the long time (approximately 2 time slot), it deletes the father node from its list. And sends the join request packet. And re-joins the Hierarchy.

Step 4: Handling the energy constrains: There are two possible values of the energy level of the node.
- A node with an energy level higher than half of the original battery capacity.
- A node with an energy level lower than half of the original battery capacity but higher than the average energy level.

(a) If the node energy level is lower than half of the original battery capacity but higher than the average energy level (Threshold value) then move the node one level lower and increase the level count by 1.
Otherwise if the node energy level is lower than the threshold value then move this node to the lowest level.

(b) If the leaf node has the energy higher than the battery capacity then move this node one level above and decrease the level count by 1.

4. SIMULATION TOOL AND METHODOLOGY

WSNs have the potential to become significant subsystems of engineering applications. The simulation software’s help in accurately modeling the real world scenario. The practical testing of the design in wireless sensor networks requires large quantity of hardware and involves high cost. Using simulators is an intelligent way to determine whether testing of the proposed methodology can be fruitful or not. In summary, simulation software’s provide us a low cost way to test the scalability, dynamicity, and ease of implementation of the proposed design. There are many simulation tools for WSNs. These include Network Simulator-2 (ns-2), OMNeT++, OPNET, GloMoSim, etc. NS-2 being the most popular among them. It uses C++ and Otcl, and follows an object oriented approach. But this simulator does not scale well. So, ns-2 is not advisable for simulation of large networks.

OMNeT++ was designed with a view to provide the researchers with a better discrete event simulation environment. OMNeT++ an open source simulator was used for simulation of proposed algorithm and existing algorithms its primary application area is the simulation of communication networks, but because of its generic and flexible architecture, is successfully used in other areas like the simulation of complex IT systems, queuing networks or hardware architectures as well.

4.1 Parameters used for Simulation

The parameters used for the simulation are given in table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>0.0001 ms</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>Data Message</td>
<td>500 bytes long</td>
</tr>
<tr>
<td>Mobility speed</td>
<td>0 - 100 unit.sec</td>
</tr>
<tr>
<td>Transmission range</td>
<td>100 units</td>
</tr>
<tr>
<td>Initial energy of the node</td>
<td>2500.00 joule</td>
</tr>
<tr>
<td>Energy consumed in Transmission</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td>Energy consumed in Computation</td>
<td>5 nJ/bit</td>
</tr>
<tr>
<td>Energy consumed in Mobility</td>
<td>100/unit</td>
</tr>
</tbody>
</table>

Table 4.1 Parameters used for simulation purpose

5. SIMULATION RESULTS

5.1 Average Time for First Node to die vs. Number of nodes

The figure 5.1 shows the time taken for the first node to die. The graph is drawn between the average times taken by the first node to die vs. the numbers of nodes in a network. Figure 5.1 shows that in case of HB algorithm average time take is more as compared to LEACH & TEEN. Figure 5.1 also shows that the first node of HB algorithm dies first when 20 nodes were taken for simulation. But as the number of nodes increases the performance of the HB algorithm improves. This is because when less nodes are deployed, nodes were scattered very far from each other, so in order to form a hierarchical structure more control messages were required and node had to move in order to be a part of a tree. Because of the mobility and more control messages transmitted by node, more energy was consumed hence the first node in the network with less number of nodes die early in case of HB. But as the number of nodes are increases then in tree formation phase the control message energy is consume and energy in mobility is less hence the performance of HB algorithm improve.

5.2 Network Life Time vs. Number of nodes

Simulation results show that the performance of HB algorithm is better than LEACH and TEEN algorithms. The simulation results also show that in HB Algorithm, as number of nodes increases the network life time also increases. The performance of HB algorithm is better when the number of nodes increase because of the fact that as the number of the nodes increases, less mobility of nodes is required to form the tree structure and the number of control messages also decreases hence the performance of HB Algorithm increases.
5.3 Average Data Rate vs. Number of node for HB Algorithm

The figure 5.3 shows the average data rate vs. number of nodes for HB Algorithm. HB algorithm is a hierarchy based (tree) routing algorithm and each node in the tree has some level. The average data rate of node at different level is shown in figure 5.3. Data rate for HB algorithm in the nodes near to the base station is high. As the level of nodes increases the average data rate decreases. The figure 5.3 also shows that when the number of nodes increases the data rate also increases. The energy consumption at level 0 is more than in case of level 2, because of more data rate. Since in HB algorithm when the nodes reach a critical energy level they move to the next higher level where energy consumption is less hence the life time of the nodes as well as network increases.

6. CONCLUSION

In WSNs, the nodes closer to the sink always lose their energy first, thus causing the overall network to "die". In this work HB Algorithm has been proposed to build an optimum mobility pattern for maximum energy efficiency. The other advantage of HB Algorithm is that it is better targeting because sensor nodes are deployed randomly, therefore there is often a requirement to move the sensor nodes for better sight or for close proximity to the physical activity. Mobility in HB Algorithm helps in better quality of communication between sensor nodes. HB Algorithm is also suitable for Mobile Wireless Sensor Nodes. HB Algorithm improves nodes and network life time by moving the node to the next higher level. Simulation results show that the nodes in level 0 consume more energy than at higher level. When these nodes at lower level reach a critical level of energy, they move to next higher level, where energy consumption is less thus improving the life time of the nodes and network.

Simulation results show that because of mobility in HB Algorithm energy dissipation is more efficient. Simulation results also show that the HB Algorithm for Wireless Sensor Networks performs better in some aspects than the existing algorithms such as LEACH and TEEN. The HB Algorithm is energy efficient than LEACH and TEEN.

7. Future work

There are various possible applications scenarios for traditional wireless sensor networks, which are envisaged at the moment. These applications include environmental monitoring, military surveillance, digitally equipped homes, and health monitoring, manufacturing monitoring, conference, vehicle tracking and detection and monitoring inventory control. Since, Mobility in Sensor Networks is relatively a new concept; its specific, unique application areas are yet to be clearly defined. Most of its application scenarios are the same as that of traditional wireless sensor networks, with the only difference of mobility of the nodes.

References


