Dynamic Data Dissemination Technique for WSN

Zeeshan Ali Khan, Cécile Belleudy, Michel Auguin
LEAT-CNRS, University of Nice-Sophia Antipolis, France
Email: {zkhan, belleudy, auguin} @unice.fr

Abstract

In wireless sensor networks, there is a need to have energy efficient data dissemination approaches at routing layer due to limited battery capacity of a sensor node. The article proposes an energy efficient virtual grid based data dissemination algorithm in wireless sensor networks. We consider a scenario having multiple sources and sinks. In order to perform the task of packet forwarding in an energy efficient manner, we consider the construction of a virtual grid based architecture. The idea is to divide the network into equally sized grids and each grid takes over the responsibility of data dissemination after periodic interval of time. We perform the simulation of the protocol in ns-2 and results show that this scheme performs better than the static grid based approaches.

1. Introduction

Wireless sensor networks are found in many daily life applications. These networks consist of devices having small size and limited resources thus there is a need to consider energy optimization in these devices at routing, MAC and physical layer. In this article, we target to find an energy efficient approach for data dissemination in a multiple mobile sink/source scenario.

In wireless sensor networks, query is generated by sink destined towards a source. On receiving the query, the source sends the data towards the sink. This query can be any information that will help the sink to take necessary action. For example, in fire fighter application, on receiving the query for locating the geographical location of fire, first responder will send the required location to the node which requested it. This will help the requesting node to decide its trajectory.

We consider an application scenario where there are multiple sources and sinks and we have to propose a robust routing algorithm which not only helps in communication between mobile sink and source but it can also keep track of their position. In the past, this problem has been studied a number of time but the proposed solutions rarely consider the mobility of source as well as sink. There is a need to have a robust backbone or rendezvous architecture to keep track the movement of source as well as sink. This type of problem exists in applications where sink is in need of data from source to find its path in the network such as firefighters application with robots as first responders, battlefield application with tanks in the field aiding soldiers to find their way, etc. In these applications, sink sends a query towards source and based on the information received from source, the sink decides the trajectory of its motion.

The article is organized as follows. In section 2, we will describe the related work that has been done in the field for tackling this problem, section 3 describes our proposed algorithm, and section 4 presents the performance evaluation of the protocol. Finally, section 5 concludes the article.

2. Related Work

The problem for energy efficient data routing from source to sink has been studied in the past a number of times [1, 2, 3, 4, 5, 6, 7]. It involves either static source and sink or static source and mobile sink or mobile source and mobile sink. The requirements for first two scenarios are simple while to manage a mobile source/sink scenario is not a simple problem as it requires a robust routing protocol having rendezvous mechanism to keep track of the position of mobile sink as well as source.

In Grid Based Energy Efficient Routing (GBEER) [2], grid based architecture for data dissemination from multiple sources to multiple sink is proposed. There are multiple sources which announce the presence of data through a data quorum and the sinks access that data through the request quorum but the periodic election mechanism for a cell head costs a lot of energy. Two-tier Data Dissemination (TTDD) [3] describes the grid construction based on the detection of a new event. The nodes of the grid are responsible for forwarding the query towards the source. In comparison with [2], the robustness of grid formation is incredible but this robustness costs a healthy amount of energy because each source has to construct a grid. Also, each time the source moves a specific distance; it has to reconstruct the grid. It can have hotspot problem if source do not move. Our proposition improves the problem in GBEER by periodically shifting the rendezvous points so there is no need to do an election process for removing the hot spot problem. Also, our proposition supports mobility of source by redirecting the query towards new source location once it has moved. Unlike TTDD, hotspot problem is minimized by dynamically changing the grid.

[4] proposes a quad tree based dissemination mechanism but it considers no mobility or minimum mobility which does not make it an ideal solution while [5] describes the railroad infrastructure for the transmission of data to the sink but it assigns the role of data collection to the nodes of rail which create the hotspot problem in the railroad region. The hot spot problem exists in the network when some nodes discharge their energy rapidly as compared to
the other nodes of the network.
Similarly [6] proposes a line based data dissemination virtual infrastructure for communication between source and sink and it has the same problem as [5]. It will have hotspot problem in the line region. Unlike [4], [5] and [6], our approach tackles the hot spot problem by changing the role of rendezvous nodes periodically.
[7] describes the formation of hierarchical routing scheme for the data transfer towards the sink node but just like [2], it costs a lot of energy in election procedure.

We propose a grid based backbone structures for data dissemination which not only helps in minimizing the hot spot problem in the network but also has minimum configuration overhead. The idea of this type of dissemination structure is proposed in [8] and a dynamic grid based algorithm is analyzed in [9]. Here, we present PGSA protocol and its optimizations based on this idea.

3. Periodic Grid Shuffling Algorithm (PGSA)
As discussed in the related work, there are a number of protocols [2, 7] which spend their resources in changing the role of backbone communication node. This periodic search and election process although helps in choosing the right node to perform the head role but at the same time, it consumes a lot of energy. The other types of protocols [3, 4, 5, and 6] have rendezvous nodes, where a certain node within the hierarchy of the network is responsible for backbone communication. Our strategy tries to take the benefit of these two types of approaches combining the positive features of both. We divide the network into equally spaced grids where nodes of each grid are responsible for communication after periodic interval of time. Thus, we rotate the task of traffic forwarding between different grids as well as we do not waste our energy in election process. Below, we mention the way in which virtual grid is formed and how we periodically change the grid position.

3.1 Grid Construction
In order to realize the operation of PGSA, N grids are formed in the network. Each grid has a cell size of c and distance between two neighboring grids is delta_c.
Only one grid is active at a time and it has the responsibility of backbone communication. We explain this idea in figure 1 where the network is divided into four grids and nodes of each grid are shown by circle, triangle, pentagon and square nodes respectively.

3.2 Algorithm for Periodic Grid Shift Topology
The position of virtual grid will change with respect to a predefined time period. The role of cell head will change as a function of virtual grid position at a particular time. The nodes nearest to the crossing points of the grid lines will be chosen as cell heads for the back bone communication. For example, in figure 1 if we consider values of T to be 10, 20, 30 … seconds, then:
\[
\begin{align*}
  t0 &= 0 \leq t \leq 2.5 \text{sec} \\
  t1 &= 2.5 \leq t \leq 5 \text{sec} \\
  t2 &= 5 \leq t \leq 7.5 \text{sec} \\
  t3 &= 7.5 \leq t \leq 10 \text{sec}
\end{align*}
\]
For time interval T+t0, circle nodes are active and responsible for backbone communication. Similarly, triangle nodes, pentagon nodes and square nodes are responsible for backbone communication between time intervals T+t1, T+t2 and T+t3 respectively.
The nodes of currently active grid will act as cell heads. If a node wants to communicate, it will first communicate with its cell head which is the nearest node on the currently active grid. We will refer to this grid/network as backbone communication grid/network or rendezvous grid/network in the rest of this article.

For example, if a query q1 is generated by the sink at time t1 then it will follow the grid responsible for communication at time t1 (triangle nodes) and the data generated as a result of this query will also follow the same grid. Since the role of cell head is changed periodically, this approach will help in balancing the energy consumption of the whole network.

3.3 Periodic Handover Approach
When a query q0 is generated at time t0, it will find the path towards the source on grid g0. The data generated by the source as a result of this query is given by d0. Now if the data d0 is being transmitted towards the sink and the role of virtual grid is changed from grid g0 to grid g1. In this case, d0 will follow the path on g0 despite the fact that active grid is g1 at the present moment. This is to simplify the problem and to reduce the overhead of handover of packets from one grid to another in case of handover.

3.4 Query/Data Forwarding
In order to send the query from sink to the source, the query is first sent towards the local cell head which is active at the present moment. Once it is reached at cell head, the backbone grid will forward the query towards the destination cell where the source is located. The cell head of source cell will forward the query towards it. On receiving the query, the source will analyze it and send the data towards the sink. The data forwarding will follow same mechanism as that of query forwarding. First of all, it will be send towards the source’s cell head. The source’s cell head will forward the data towards the sink’s cell head. Once the data is reached towards the sink’s cell head, it will be forwarded towards the sink. In order to forward the query/data on the grid, XY/YX routing mechanism is used. In this mechanism, the node which has to forward the packet will analyze its geographical coordinates with the destination node’s coordinates. First, the packet is forwarded in x-direction until x-coordinates of the current node equals x-coordinates of destination. Then, it will be forwarded in y-direction, until the y-coordinates of the current node equals the y-coordinate of...
the destination. Thus, packet reaches at the destination node’s cell head. That is the query/data forwarding mechanism used in our protocol.

3.5 Source Tracking

The source registers itself with the cell head and once source changes its position, it will re-register itself with the nearby cell head. In sub-section 3.8, we also propose an overlay network which can be used to store the control information such as current position of source or sink. This will reduce the burden on the dynamic grids that are used for data dissemination.

3.6 Sink Tracking

In applications where the source changes its position like battlefield application (with tanks as source and soldiers as sinks) or firefighter application (with robots as source giving location of fire and firefighters as sinks), the sink normally changes its position when it receives the data from the source. Because it decides its trajectory based on data it receives from source, so it can be either considered static or with a speed with which it does not change its current cell.

3.7 Query Retransmission

The protocol proposes the query retransmission mechanism. This mechanism activates itself when we have generated a query and after a specific period of time, the data is not received at the sink. At this time, sink will regenerate the query to get the data. This retransmission depends on the time to live limit of the query packet and its round trip time. Once the time limit is over, there is a high probability that the packet is lost in the network and a retransmission is required.

3.8 Example of PGSA Packet Forwarding Mechanism

In order to explain the functionality of the algorithm, we are going to describe an example of the protocol with the help of a figure. For this we consider at time \( \tau_1 \), the nodes on the grid which are responsible for communication (figure 2 (a-b)) are denoted by \( g_0 \) and the nodes responsible for communication at \( \tau_2 \) are denoted by \( g_1 \). When \( g_0 \) is responsible for communication, then the packet forwarding mechanism is described in figure 2(a). The query is send from sink to its cell head of grid \( g_0 \) as depicted by step (1). In step (2) and (3), the query is sent from sink’s cell head to the source’s cell head. In step (4), the query is forward from source’s cell head towards the source. The same process is repeated to forward the data from the source to the sink.

In this manner our protocol tries to send equal number of data and query packets on each grid. This way it evenly distributes the communication energy on all the nodes of the network. As a result, the probability of sudden death of a node decreases which helps in removing the hot spot problem. Also there is not much configuration overhead during run time because the grid selection and periodic rotation of grid happens at the initial configuration of protocol.

3.9 PGSA with Overlay Network (PGSA_O)

Several optimizations are possible to PGSA in order to increase its efficiency. When source has high mobility rate, we propose PGSA with overlay network having directional antennas. We will refer to this variation as PGSA_O in the rest of the paper. As we know that PGSA uses geographical routing in order to forward a packet and it needs to know the location of source or sink if they move during the course of query/data transmission. So, once a source or sink moves within the network, it updates this information to overlay network. The overlay network consists of nodes that are located in the same plane. The z-coordinates of this network are different from that of sink or source. So, they will use directional antennas that will reduce the transmission interference from other network thus it will take less amount of energy to broadcast the control information in the network. This is explained in figure 3 where the black nodes belong to overlay network and their z-coordinates is different from other nodes.

For example in fire fighter application, nodes that measure the temperature are installed on the roof while robots and firefighter move on the ground floor. So these temperature sensing nodes can function as the overlay nodes. They can use directional antennas for broadcasting data in overlay network which have minimum interference that reduces packet collisions resulting in an energy efficient broadcast.

![Figure 1](image.png)

**Figure 1** Periodic Change of Rendezvous Nodes in PGSA
4. Performance Evaluation

For the purpose of performance evaluation, we have performed extensive simulations in ns-2. The objective is to analyze various parameters of PGSA protocol and to compare PGSA with algorithms having a static grid for backbone communication. XY routing protocol is implemented in ns-2 that is used at routing layer. Table 1 presents the simulation parameters used for simulation. It is assumed that each node knows its position and the grid used is a perfect grid. The idle energy consumption of a node is not considered and the transmission and reception energy for each packet at network layer is taken into account.

Table 2 shows the energy consumed per backbone node and it shows that the nodes of static grid algorithm consumes maximum. This is due to the fact that dynamically changing grid does the load balancing by forwarding the packets on different grid architectures in order to remove the hot spot problem. Figure 4 plots following three scenarios with cell size = 80m and number of grids to be 4, 2 and 1. It is to be noted that the energy consumption of a node lying on (x, y) location is plotted on z-axis. These figures also indicate the superiority of dynamic grid based schemes over static ones. Table 2 gives the average energy consumed by a backbone node in order to forward the traffic. This shows that we get 7 times gain in energy while forwarding the data on dynamic grid (figure 4c). This is due to the fact that for experimentation purpose, we send equal number of queries from sinks towards randomly placed sources after periodic interval of time. We also implemented query retransmission mechanism at routing layer. In static grid case (#grids = 1), there is a lot of traffic load and buffer overflows at MAC layer resulting in query retransmissions. This increases the traffic significantly which resulted in energy consumption for static grid case. Also, in dynamic grid based scenario, decreasing the grid displacement size increases the lifetime of the network. If the cell size is changed keeping the grid displacement factor and receiver threshold values constant, it will increase the energy consumption of overall network because number of hops from source to sink and vice versa are increased. This is confirmed by energy values given in table 3 and the distributed energy plot for this case is shown in figure 5 (a) & (b). If the number of sources in the network is increased, it will increase the energy consumption of the network due to increase in the network traffic. This is confirmed by figure 5 (c) and (d). The average energy consumption values for each node of rendezvous network are listed in table 4.

In table 5, we show the average energy consumed by the overlay network (PGSA_O) for broadcasting a packet having cell size equal to 80m. Average energy required to broadcast the same broadcast packet on a PGSA type grid is also estimated. It clearly shows that this type of network reduces the load on a rendezvous network thus increasing the lifetime of the rendezvous network. The results verify that increasing the number of virtual grids decreases the energy consumption for each backbone node of the network. Also, grids having smaller cell size consumes more due to increase in number of hops.

5. Conclusion

We present the comparison of dynamic grid based structure (PGSA) with static configuration of grid based backbone structure for source/sink communication. PGSA not only minimizes the hot spot problem but it also does not have large static configuration overhead. When source/sink nodes are mobile in a sensor network, the efficient way of locating them is either to periodically announce their position on a virtual structure. This virtual structure is also used as backbone network for packet forwarding. As a result of this approach, nodes of backbone network will die out soon thus creating the hotspot problem. PGSA resolves this issue by forming more than one virtual structure and periodically changing their role for packet forwarding.

For future work, we propose the comparison of static backbone networks with dynamic networks for rail-structure based or line-based structure with static configuration. This will efficiently balance the energy consumption across the network. The idea is to change the backbone rendezvous structure periodically to view its impact on energy balancing of the network.
Figure 4 Distribution of Communication Energy Consumption among Sensor Nodes (a) PGSA(c=80m, delta_c=20m & #grids=4) (b) PGSA(c=80m, delta_c=40m & #grids=2) (c) Static Grid Based Communication Mechanism PGSA(c=80m, delta_c=0m & #grids=1)

Figure 5 Distribution of Communication Energy Consumption among Sensor Nodes (a) PGSA with c = 80m (b) PGSA with c = 40m (c) PGSA with No. of Sources = 2 (d) PGSA with No. of Sources = 4
6. References


<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Protocol</td>
<td>802.11</td>
</tr>
<tr>
<td>Routing Protocol</td>
<td>XY Routing Protocol with retransmissions</td>
</tr>
<tr>
<td>Sensor Field Size</td>
<td>300 * 300 m²</td>
</tr>
<tr>
<td>Number of Sinks</td>
<td>4</td>
</tr>
<tr>
<td>Number of Sources</td>
<td>2-4</td>
</tr>
<tr>
<td>Packet Size</td>
<td>1000 bits</td>
</tr>
<tr>
<td>Per Bit TX/RX Energy Consumption</td>
<td>17.5nJ</td>
</tr>
<tr>
<td>Query Generation Rate</td>
<td>1 query per second</td>
</tr>
<tr>
<td>Cell Size, c</td>
<td>40, 80 m</td>
</tr>
<tr>
<td>Grid Displacement, delta_c</td>
<td>20, 40 m</td>
</tr>
</tbody>
</table>

Table 1 Simulation Parameters

<table>
<thead>
<tr>
<th>Type of Rendezvous Network</th>
<th>Average Energy Consumed by a RDV Nodes (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGSA (c = 80m, delta_c = 20m, #grids=4)</td>
<td>29.7</td>
</tr>
<tr>
<td>PGSA (c = 80m, delta_c = 40m, #grids=2)</td>
<td>45.54</td>
</tr>
<tr>
<td>PGSA (c = 80m, delta_c = 0m, #grids=1)</td>
<td>424.6</td>
</tr>
<tr>
<td>Static Grid Approach</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Average Energy consumed by grids of same cell size

<table>
<thead>
<tr>
<th>Type of Rendezvous Network</th>
<th>Average Energy Consumed by a RDV Nodes (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGSA (c = 80m, delta_c = 20m, #grids=4)</td>
<td>29.7</td>
</tr>
<tr>
<td>PGSA (c = 40m, delta_c = 20m, #grids=2)</td>
<td>33.6</td>
</tr>
</tbody>
</table>

Table 3 Average Energy consumed by grids of different sizes

<table>
<thead>
<tr>
<th>Number of Sources (c = 80m)</th>
<th>Average Energy Consumed by a RDV Node for Broadcasting a Packet (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sources = 2</td>
<td>16.55</td>
</tr>
<tr>
<td>Sources = 4</td>
<td>29.74</td>
</tr>
</tbody>
</table>

Table 4 Average Energy consumed as a function of number of sources

<table>
<thead>
<tr>
<th>Type of Rendezvous Network</th>
<th>Average Energy Consumed by a RDV Node for Broadcasting a Packet (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGSA</td>
<td>9.92578</td>
</tr>
<tr>
<td>PGSA_O</td>
<td>1.34247</td>
</tr>
</tbody>
</table>

Table 5 Average Broadcast Energy consumed by different versions of PGSA