Efficient Algorithm for Maximizing Life Time of Wireless Sensor Networks

Ram Baksh[1]**, Ranjan Kumar Thakur**[2]**, Arabind Kumar**[3]**, Aditya Pratap Singh**[4]

[1]Delhi University, Department Of Mathematics, Dyal Singh College, Lodhi Road, New Delhi - 110003

[2]Delhi University, Department Of Mathematics, Deshbandhu College, Kalkaji, New Delhi - 110019

[3]Delhi University, Department Of Mathematics, Kamala Nehru College, August Kranti Marg, New Delhi – 110049

[4]Delhi University, Department Of Mathematics, PGDAV College (Evening), Nehru Nagar, New Delhi - 110065

Abstract: ***In this paper we give domatic partition algorithm for obtaining the maximum number of disjoint dominating sets. At a time only one dominating set activated and other puts into rest mode. We also explore grid-based coordinated routing in wireless sensor networks and compare the energy available in the network over time for different grid sizes. A test area is divided into square-shaped grids of certain length. Fully charged battery powered nodes are randomly placed in the area with a fixed source and sink nodes. One node per grid is elected as the coordinator which does the actual routing. The source node starts flooding the network with every coordinator joining in the routing. Once the flooding reaches the sink node, information is sent back to the source by finding the back route to the source this process is continued until a node (coordinator) along that route runs out of energy. New coordinators are elected to replace the depleted ones. The source node refloods the network so that the sink can find a new back route to send information. This entire process continues until the network is partitioned and the connectivity between the source and the sink nodes is lost. In this work we compare domatic partition algorithm with grid based algorithm.***

Keywords: Domatic; Dominating; Sensor; wireless; flooding;

1. Introduction

Wireless Sensor Networks are a result of the combination of advances made in the field of analog and digital circuitry, wireless communications and sensor technology. A wireless sensor network typically consists of small devices called sensor nodes that are capable of sensing the environment around them. The sensor nodes are devices that are capable of sensing, gathering, storing and transmitting information. The main advantage of these nodes is their self-organizing capability. Large networks of such small nodes are therefore growing in use. The sensor nodes can be deployed anywhere without actually having to install or deploy them manually. In remotely inaccessible areas, these sensor nodes are just strewn across the desired sensor field. The self-organizing capability of the sensor nodes enables the nodes to form a cooperative network and gather information. This information can then be retrieved. Thus, sensor networks enable intelligent monitoring of inaccessible areas with ease and accuracy.

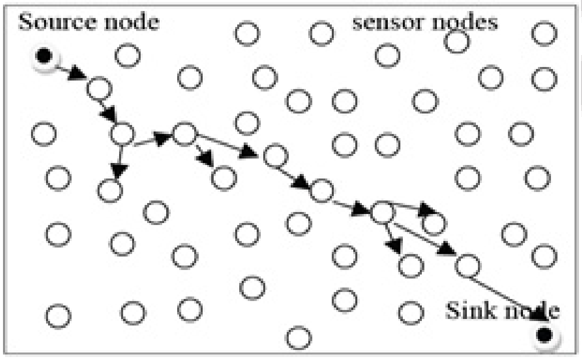

Figure 1: shows a wireless sensor network with sensor nodes

Wireless sensor network (WSN) is a term used to describe an emerging class of embedded communication products that provide redundant, fault-tolerant wireless connections between sensors, actuators and controllers. Figure 1 shows a wireless sensor network with sensor nodes, a source node and a sink node.

1.1 The Sensor Node

Like all other technologies, wireless sensor networks are also subject to constraints. One of the major challenges is the energy constraint of the sensors nodes. Sensor nodes are driven by a battery and have very low energy resources, which in turn affects the network lifetime, O.Kasten [1]. Figure 1 shows the architecture of a sensor node.

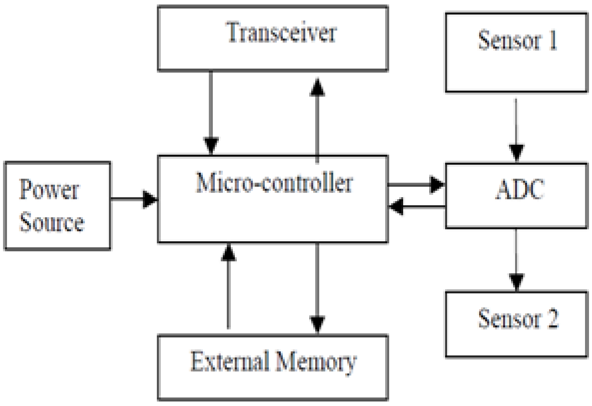

Figure 2

1.2 Components of a Sensor Node

From the Figure 2, the components of a sensor node are a microcontroller, a transceiver, external memory, a power source and sensors. Microcontroller: The microcontroller performs tasks, processes information and controls the functionality of the other components in the node. Microcontrollers are best suited for sensor nodes due to their flexibility to connect to other devices, less power consumption and the fact that the microcontroller can enter a sleep state wherein only a part of the controller is active, thus saving energy.

1.3 Transceiver

Sensor nodes operate in the Industrial, Scientific and Medical (ISM) band, which provides free radio, huge spectrum allocation and global availability. Radio frequency communication is best suited for sensor networks. Sensor networks use the communication frequencies between 433 MHz and 2.4 GHz. The transceiver provides the functionality of both the transmitter and the receiver in the sensor node.

1.4 External Memory

Mostly, the on-chip memory of the microcontroller and flash memory are used, and off-chip Random Access Memory (RAM) is barely used. Depending on the type of storage, two kinds of memory are used: user memory for storing application related or personal data, and program memory for programming the device.

1.5 Power Source

The sensor node consumes power for sensing the environment, gathering information, storing and processing the information gathered. Most of the energy is spent for communication with other nodes in the network and staying active. Batteries are the main source of power for sensor nodes.

1.6. Sensors

A sensor is a device that responds to a change in its surroundings in a measurable manner. The continuous analog signal that is sensed by the sensors is digitized by analog-to-digital converters and sent to the controllers for further processing. Sensors should be small, adaptive to the environment, operate and configure on their own, and consume low power. There are two types of sensors, active and passive. Active sensors gather data by probing into the environment, while passive sensors gather data without actually disturbing the environment.

2. Domatic Partitions

In this section we will discuss an algorithm which will gives a maximum number of disjoint dominating sets, Islam[2], of a graph.

2.1 Model and Definitions

We assume that sensors are deployed in the plane and model a sensor network by an undirected graph $G = (V, E)$ where the vertex set V denotes the set of sensors and E represents the links $(u, v) \in E$ between two sensor nodes $u, v \in V$ if they are within their transmission range (recall that every sensor node $u \in V$ has the same transmission range normalized to 1). A subset $D \subseteq V$ is a dominating set if every vertex $v \in V \setminus D$ has at least a neighbor in D. Our algorithm to find disjoint dominating sets operates in rounds where at each round i we construct a dominating set D_i, where $D_i \setminus D_j = Ø$; $i \neq j$ and $i, j \geq 1$. We color the nodes in D_i with color class i.

2.2 Compatible and Incompatible node

A node $v \in N[u]$ is called an uncovered neighbor of u at iteration (round) i if there is no node $w \in D_i$ that is a neighbor of v (we call it v is not covered by any node in D_i).The total number of uncovered neighbors of u (including u) in iteration i is denoted as, $uncov_i(u) = |\{ v \mid v \in N[u], vw \notin E , \forall w \in D_i \}|$. The dom number of a node u denoted by dom(u) is the number of neighbors (including u) that belong to some dominating sets, that is, $dom(u) = |\{ v| \; v \in N[u], v \in D_j , j \geq 1\}|$. Each node u maintains a triple at round i, $t_i(u) = 1/uncov_i(u), dom(u), id$ which is updated in each round. If $uncov_i(u) = 0$, then we assign $uncov_i(u) = 1/(n + 1)$ where $n = |V|$. We call a node u compatible if $uncov_i(u)$ # $1/(n+1)$, else it is called incompatible. For any two compatible nodes, u and v we say u is smaller than v in round i if $t_i(u)$ is lexicographically smaller than $t_i(v)$ and denote the order by $t_i(u) < t_i(v)$. Note that only compatible nodes are compared.

2.3 Algorithm for Domatic Partition

In this section we present a description of our algorithm. Our algorithm works in rounds starting from round 1 and in each round i we compute a dominating set D_i such that $D_i \cap D_j = Ø$, $i \neq j$; i; $j \geq 1$. The algorithm runs for $\delta + 1$ rounds. However, it stops in round j if we cannot form a new disjoint dominating set D_j or we have $j = \delta + 1$. If $j = \delta + 1$ then we have reached the optimal (maximum) number of disjoint dominating sets. Otherwise, $j - 1$ is the maximum number of disjoint dominating sets $D_1, D_2, - - - , D_{j-1}$ returned by our algorithm.

Let $D_1' = Ø$. Starting at round 1, first, we compute a triple for each node in G and select the smallest node among the compatible nodes. From the triple it is easy to see that we always find such a single node. Let u be the node. We color u 1, add u to D_1 (note that D_1' is a temporary set that grows as dominating nodes are added to it). We update $t_1(v)$, $v \in N[u]$. Then we pick the smallest compatible node in $V \setminus D_1'$, color it 1, insert into D_1' and update the nodes triples. At each selection of a node u in D_1' we make at least one node incompatible. We repeat the process until all nodes become incompatible. Assign $D_1 = D_1'$. We claim that nodes in D_1 form a dominating set. At this stage $\forall u \in G$, $uncov_i(u) = 1/(n + 1)$ and $\forall u \in G \setminus D_1$, $dom(v) \geq 1$. We now describe in general how a dominating set D_i is formed at round $i \geq 2$. First, initialize $\forall u \in G$, $uncov_i(u) = deg(u)$ and let $D_i' = Ø$. Select the smallest node among the compatible nodes in $V \setminus (D_1 \cup D_2 - - - D_{i-1} \cup D_i')$, color it i, add to D_i'. Update the triples of the corresponding nodes. Then find the smallest compatible node in $V \setminus (D_1 \cup D_2 - - - D_{i-1} \cup D_i')$, color it i and insert into D_i' and update the triples of the nodes whose $uncov_i(.)$, $dom(.)$ values are changed. Continue this process until one of following cases occurs

(I) all nodes become incompatible or
(II) $V \setminus (D_1 U D_2 - - -D_{i-1} U D_i') = Ø$ and there exists at least one compatible node in G or
(III) $V \setminus (D_1 U D_2 - - - D_{i-1} U D_i') \neq Ø$ and there exists a node u such that $\forall u' \in N[u], u' \in D_1 U D_2 - - - D_{i-1} U D_i'$.

If case (I) is satisfied then assign $D_i = D_i'$ and we have already constructed the i^{th} dominating set D_i and next round i + 1 is invoked to generate D_{i+1}. On the other hand, if $V \setminus (D_1 U D_2 - - - UD_{i-1} U D_i') = Ø$ and there is at least a compatible node (Case (II)) then we cannot form the current dominating set and the algorithm reports that i - 1 is the maximum number of disjoint dominating sets and the execution ceases. This is because there is no node to be included in D_i' and some nodes are still uncovered. Similarly if Case (III) is satisfied then the algorithm terminates and reports i - 1 is the maximum number of disjoint dominating sets. This is because there is no neighborhood N[u] of u from which we can select a node to cover u. Let j represent the maximum number of dominating sets returned by our algorithm, that is, the algorithm constructs $D_1, D_2, - - - ,D_j$ disjoint dominating sets.

Algorithm for Domatic Partition which is given by, Islam [2]

Input: A unit disk graph G = (V, E)

Output: j disjoint dominating sets $D_1, D_2, - - -, D_j$.

```
Compute $t_1 (u) = <1/uncov_1(u), dom(u), id>$ , $\forall u \in V$.
$D'_1 = Ø, i = 1$
while $i \leq (\delta + 1)$
    Find the smallest compatible
    node u in $V \setminus (D'_1 U D'_2 U - - - D'_I )$
    if there is such u then
        $D'_i = D'_i U \{u\}$
    else
        if Case (II) or Case (III) is satisfied then
            $j = i - 1$
            break
        else
            if all nodes are incompatible then
                $D_i = D_i'$
                $i = i + 1$
                $j = i$
                $D'_{i+1} = Ø$
            endif
        endif
    endif
    update $\forall u \in V$ $t_i(u) = <1/uncov_i(u), dom(u), id>$
end while
```

3. The Flooding Algorithm

The flooding algorithm is one of the most simple and widely used algorithms in a point to- point communication network. In the flooding algorithm, the source first broadcasts information to all its neighboring nodes. Each receiving node, in turn broadcasts the information it receives to all its neighboring nodes, other than the source node. The information thus traverses from the source node to the destination node and through all the nodes in the network. The basic algorithm for flooding is as shown below which is given by Ganesan[3] :

```
Algorithm 1:
For the source $r_0$, do:
Send the message on all outgoing links.
For vertex $v \neq r_0$ do:
If the message is received for the first time:
    1. Store the information in an output buffer.
    2. Forward the message to every other node in its own vicinity.
If the message is received again, discard the message.
End
```

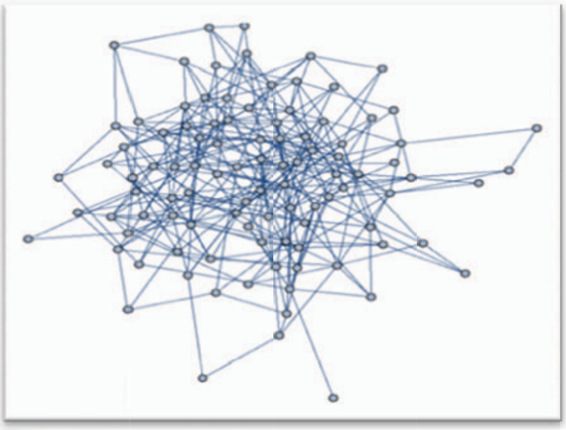

Figure 3: Simulation topology showing the traditional flooding algorithm

Though flooding algorithm is easy to implement, it is still complex and poses its own challenges. Flooding is inefficient in terms of network bandwidth utilization, because each node transmits and receives multiple packets of data, thus wasting network bandwidth. Real world flooding is considerably complex as care has to be taken to avoid duplication of data packets, infinite loops and clearing the output buffers of multiple entries of data. Another flooding algorithm that shows a flooding-based tree construction protocol for avoiding duplicate deliveries Uttara[4] is as shown below:

```
Algorithm 2: FLOOD (Node S)
if Node n receives the packet for the first time then
    Mark Node n as received
    Parent S ←
    Source n ←
    Increment Level Field
    Rebroadcast packet
end if
```

In this algorithm, the node which sends the packet to a node for the first time is the parent of the node. The *Level* field is incremented by one, and then the packet is rebroadcast. The *Level* field is to denote the number of hops the node is away from the source node. The main advantage of this flooding is that a node can receive a packet only if it has not yet received a packet. Therefore, the duplication of data packets is reduced and hence the network bandwidth is efficiently utilized to a certain extent. Every node in this algorithm has a unique parent and as many children as the nodes it can reach. This flooding algorithm is an improvisation of the traditional flooding algorithm.

3.1 Grid-based Coordinated Routing

The main focus of Grid Based Coordinated Routing (GBCR) is on partitioning the network into square shaped grids to extend network lifetime. The entire network is divided into equally shaped grids, and in each grid an active node, the coordinator is elected, like in the Span algorithm. The underlying routing algorithm used in GBCR is similar to level flooding. The following algorithm is used by grid-based coordinated routing protocol given by Sawant [5]:

Algorithm 3 Grid-based coordinated routing protocol

```
C <= set of coordinator nodes
 while network is not partitioned do
 while C ≠ Ø or sink node not yet reached do
 Pick a node C_i randomly from C
 FLOOD(C_i )
 end while
 send information from the sink node back to the source node
 elect new coordinator nodes C'
C <= C'
```

In grid-based coordinated routing, information reaches only selected nodes in the field instead of to all the nodes in the network. The main idea of dividing the network into grids is to make only one node alive for each grid, while the rest of the nodes in that grid are sleeping so as to conserve their battery life. In each grid, the coordinator participates in routing as long as the amount of energy in that coordinator is above a certain threshold value. When the energy drops below the threshold, a new coordinator is elected for that grid. The source transmits information to the sink through the active coordinators, and the sink traces a route back to the source. The process of flooding continues till the nodes participating in the routing run out of energy, when new coordinators are elected and a new route back to the source from the sink is calculated. The source starts flooding by sending a query message to all the neighbor coordinators, which flood other coordinators in the network till the message reaches the sink node. Each coordinator node in grid-based routing has three states, namely, routing, warning and depleted states. When coordinator nodes in a particular route die, or run out of energy, new coordinators are elected to replace the old nodes. All nodes in the network are randomly assigned IDs. In each grid, the node with the highest ID becomes the coordinator. When the node with the highest ID runs out of energy, the node with the next highest ID becomes the coordinator for that grid. Each time the coordinator node changes the sink node traces back a route to the source node.

4. Comparison of Domatic Partition Algorithm with Grid Based Algorithm

We have written a Computer Program of Domatic Partition Algorithm. If we take input of a graph in computer program it will give dominating sets.

Input: A graph of 14 vertices

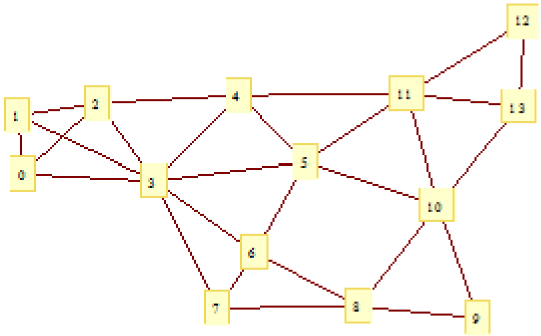

Figure 4: Graph

Output

dominating sets are:
first { 3, 10, 12}
second { 1, 5, 8, 13}
third {0, 7, 9, 11 }

We have chosen 14 nodes in test area of size 100X100m2 and suppose each node have 100 units of energy. We have taken energy consumption ratio is idle: receive: transmit 1.0: 1.5: 2.0 from Rovert[5]. Now we divides the networks into different types of non uniformly grids.

If area of networks divides like figure below then nodes can make one time communication but we know from Rovert[5] nodes can use 75% of its energy and transmission nodes spent two units per unit time therefore networks alive 37.5 unit time.

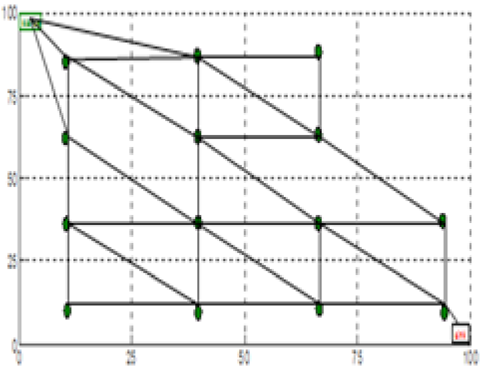

Figure 5: The uniform grid structure with 14 sensor nodes deployed uniformly in the field.

If test area divides like below the networks alive 37.5 unit time

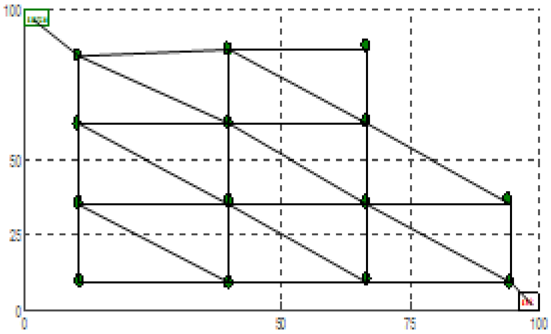

Figure 6: Flooding between the coordinator nodes in a 100 × 100m2 networks for a grid structure with the source node in nodes of size 50m and sink node in grids of size 25m each

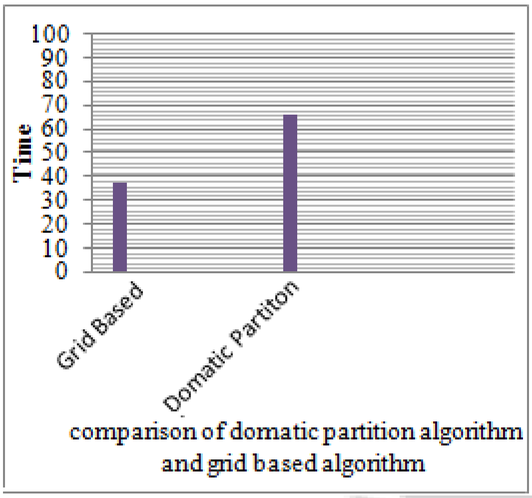

Figure 7

5. Conclusion

When we divided networks into grids we gets networks partition after 37.50 unit time but in domatic partition algorithm networks gets partitions after 65.62 unit time because from figure 5.0 our C code gives three dominating sets which is activated one by one first dominating set running 37.50 unit time second dominating set running 18.75 unit time and third one running 9.37 unit time therefore total running time is 65.62 unit. This implies that domatic partition algorithm for maximizing networks lifetime is better than grid based algorithm. The comparison has shown in figure 5.12.

6. Future Research

- **Implementation on NS2:** The non-uniform grid-based routing protocol is simulated in MATLAB. This can be extended onto NS2.
- **Mobility of nodes:** The nodes in our network are stationary. Mobility may be added to the nodes in the network and the working of the protocol can be observed.

References

[1] O.Kasten,"EnergyConsumption,"2001,http://www.inf.ethz.ch/personal/kasten/research/bathtub/energy consumption .html

[2] Kamal Islam, Selim G. Akl, Hebk Meijer, "*Maximizing the lifetime of a sensor network through domatic partition*", Proc. of the 34th IEEE conference on 2009 cs.

[3] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of lowpower wireless sensor networks," *Technical Report UCLA/CSD-TR 02- 001*, 2002.

[4] Robert Akl, Uttara Sawant, "Grid-based coordinated routing in wireless sensor networks," *Consumer Communications and Networking Conference*, pp. 860-864, 2007.

[5] Robert Akl,1 Priyanka Kadiyala,1 andMohamad Haidar2, "*NonuniformGrid BasedCoordinated Routing in Wireless Sensor Networks*" Hindawi Publishing Corporation Journal of Sensors Volume 2009, Article ID 491349, 11 pages doi:10.1155/2009/491349

Author Profile

Ram Baksh received the M.Sc degree in Mathematics from Banaras Hindu University Varanasi and M. Tech. degree in Computer Science and Data Processing from Indian Institute of Technology Kharagpur 2012.