International Journal Of Computer Science And Applications Vol. 4, No. 1, April / May 2011 ISSN: 0974-1003

Strategic Intelligent Node Deployment in View of Optimal Scheduling for Sensor Network

Sachin R. Jain
PG Student, Rajiv Gandhi College of Engineering, Research & Technology
Chandrapur
India
sachinjain98440@rediffmail.com

Dr. S. G. Akojwar
Rajiv Gandhi College of Engineering, Research & Technology
Chandrapur
India
sudhirakojwar@rediffmail.com

Dr. Thakur Nileshsingh V.
Shri Ramdeobaba Kamla Nehru Engineering College
Nagpur
India
thakurnisvis@rediffmail.com

ABSTRACT

In this paper, we describe the scheduling approach based on the strategic intelligent node deployment. In general the nodes are deployed randomly or in deterministic way. This paper first discusses the mapping of the randomly deployed nodes and later the concept of the intelligent node. Artificially intelligent nodes are used in scheduling. Status of the node varies according to the scheduling procedure. The node status is either one of the three statuses i.e. standby, work, and turn off. To check the feasibility of the proposed scheduling approach, preliminary results are obtained with some assumptions. Implementation is carried out in C++. An obtained simulation result justifies the feasibility of the proposed approach.

Categories and Subject Descriptors

C.2 [**Computer-Communication Network**]: Network Architecture and Design – *wireless communication*

General Terms

Algorithms, Design, Management, Experimentation.

Keywords

Sensor network, Artificial intelligence, Intelligent nodes, Scheduling.

1 INTRODUCTION

Wireless Sensor Networks (WSN) [1] consist of a large number of small sensor nodes with sensing, data processing, and communication capabilities, which are deployed in a region of interest and collaborate to accomplish a common task, such as environmental monitoring, fire detection, pollution detection, traffic monitoring, industry process control, object tracking and various other application domains. Distinguished from traditional wireless networks, a sensor network has many unique characteristics, such as denser node deployment, higher unreliability of sensor nodes, asymmetric data transmission, and severe power, computation, and memory constraints, which present many new challenges for the development and eventual application of wireless sensor networks.

In particular, sensor nodes are usually battery-powered and should operate without attendance for a relatively long period of time. In most cases [2], it is very difficult and even impossible to change or recharge batteries for these sensor nodes. For this reason, energy efficiency is of vital importance for the operational lifetime of a sensor network. To prolong the lifetime of a sensor network, energy efficiency must be considered. In the last years, wireless sensor networks have gained increasing attention from both the research community and actual users. As sensor nodes are generally battery-powered devices, the critical aspects to face concern how to reduce the energy consumption of nodes, so that the network lifetime can be extended to reasonable times.

This is an approach to conserve the energy and use it effectively and efficiently of sensor nodes using Artificial Intelligence. The advantage of using this approach is to increase the operation time as compared to the actual time. Instead of discovering the nodes from the entire network, the sensor nodes are made intelligent and can be scheduled intelligently, so that they can effectively and efficiently use the available energy.

This paper is organized as follows: Section 2 discusses the related work and the proposed approach is discussed in the section 3. Section 4 presents the experimentation results followed by the conclusion and future scope in section 5; paper ends with the references in section 6.

2 RELATED WORK

We have studied and analyzed the research papers which are related to the scheduling in sensor network. These papers particularly focus on the scheduling mechanisms or approaches in non-hierarchical and hierarchical sensor network.

In [3], Kumar et al. adopt the randomized independent scheduling mechanism to extend network lifetime. It assumes that time is divided into cycles based on a time synchronization method. At the beginning of a cycle, each sensor independently decides whether to become active or go to sleep. It is a very simple self-scheduling mechanism. It does not require location or distance information and has no communication overhead. Nodes do not maintain a neighborhood table because the sensors do not dynamically evaluate their situation. Tian and Georgana proposed a distributed scheduling mechanism to save energy while preserving sensing coverage [4]. This mechanism allows a sensor to turn off only if its sensing area is completely covered by its neighbors' sensing areas to avoid reducing sensing coverage. Each sensor uses its neighbors' location information and sensing range. In this mechanism, nodes need accurate location information and nodes are time synchronized so that they know the beginning of each round. There is a message overhead for advertising location information and scheduling (but only at the beginning of each round). Nodes maintain per neighbor state to keep track of the number of active neighbors. In each round, working nodes never go back to sleep. However, the set of working nodes may be different in different rounds, so energy consumption may still be balanced among the nodes.

Berman et al. [5] formulated the sleep-scheduling problem as a maximization problem with constraints on battery lifetime and sensing coverage. They also presented a centralized and a distributed algorithm to maximize network lifetime. Their distributed mechanism can guarantee a specific degree of sensing coverage (assuming that the sensor density is high enough) whereas the aforementioned mechanism [4] preserves the existing coverage degree. In this mechanism, each sensor is in one of three states: active, idle or vulnerable. In the vulnerable state, if the sensor discovers that part of its sensing area cannot be covered by any of its active or vulnerable neighbors; it immediately enters the active state. Otherwise, it enters the idle state if its sensing area can be monitored by either active neighbors or vulnerable neighbors with a higher energy level. In this algorithm, nodes need accurate location information and nodes need to broadcast their state and energy level in addition to their location. Nodes cannot completely turn themselves off in the idle state and have semi-synchronous monitoring schedules (due to global reshuffles).

In [6], Wu et al. proposed a Lightweight Deployment- Aware distributed scheduling mechanism. The goal here is to provide statistical guarantees on sensing coverage. This mechanism assumes that each working node has a mechanism to know the number of working nodes in its neighborhood. When the number of working neighbors exceeds a threshold determined by the application’s requirement on sensing coverage, the node randomly selects some of its neighbors to turn off and sends tickets to them. When a node collects enough tickets from its neighbors, it may enter the off duty mode after a random back-off period. In this mechanism, nodes are assumed to be randomly and uniformly distributed over the coverage area and each node needs to know how many sensors are within its sensing range. Unlike the previous three mechanisms, nodes have asynchronous sleeping schedules. Energy consumption is balanced among the nodes since the longer a node works, the more tickets it may accumulate and the more likely it will be turned off.

Ye et al. developed a mechanism called PEAS (Probing Environment and Adaptive Sensing) that can extend the lifetime of a high-density sensor network in a harsh environment [7]. Assumptions distinguish this work from the previous studies. First, it assumes that sensor nodes may fail frequently and unexpectedly, which makes synchronized sleeping algorithms infeasible because they depend on the predictability of sensors’ lifetime. Second, it assumes that the sensor network is so dense that the total number of sensors may be orders of magnitude higher than the number of working nodes. As a result, it is infeasible for nodes to maintain per neighbor state. Finally, it assumes that nodes do not have location information. In this mechanism, nodes are assumed to be randomly and uniformly distributed and similar to [6], nodes do not need accurate location information and have asynchronous schedules. Unlike most of the surveyed mechanisms, nodes do not maintain per-neighbor state and working nodes never go back to sleep, which may result in unbalanced energy consumption.

Zhang and Hou [8] proposed a distributed mechanism to maximize the number of sleeping sensors while ensuring that the working sensors provide coverage and connectivity. It tries to minimize the overlapping area between the working sensors. A sensor is turned on only if it minimizes the overlapping area with the existing working sensors and if it covers an intersection point of two working sensors. A sensor can verify whether it satisfies these conditions using its own location and the working sensors’ locations. In this mechanism, nodes need accurate location information and nodes need to maintain time synchronization. There is message overhead for advertising location information and scheduling only at the beginning of each round. In each round, working nodes never go back to sleep, but different nodes may be working in different rounds so energy consumption may still be balanced among all the nodes.

Wang et al. proposed an integrated coverage and connectivity configuration protocol [9] which aims to maximize the number of sleeping nodes, while maintaining both coverage and connectivity. The authors combined connectivity configuration protocol with SPAN [10], a connectivity preserving scheduling mechanism, to achieve their dual objectives in this scenario. Nodes running connectivity configuration protocol are in one of three modes: active, listen and sleep. This protocol requires accurate location information and each node needs to maintain a neighborhood table. Nodes have asynchronous sleep schedules and working nodes may go back to sleep, so that the energy consumption is balanced among all the nodes. Cerpa and Estrin [11] proposed using sensors’ local measurements to automatically configure network topology in a high density sensor network. The goal is to maintain a certain data delivery ratio while allowing redundant sensors to stay asleep in order to conserve energy. In this the sensors measure their connectivity as well as their data loss rate and activate their neighbors based on these local measurements. This approach is similar to [7] in several ways. First, it assumes that there is a high density of sensor nodes. Second, after the sensors are activated, they never go back to sleep. Third, both are decentralized mechanisms that allow sensors to use locally measured information to adjust network topology. In this mechanism, nodes do not need accurate location information and there is no periodic message overhead for neighbor discovery. Nodes maintain per-neighbor state to keep track of the number of active neighbors and working nodes never go back to sleep, which may result in unfair energy consumption.

Gui and Mohapatra [12] proposed an extension of the PEAS protocol proposed in [7]. It has the same environment probing mechanism as PEAS, but it does not let working nodes stay awake indefinitely. Simulation results in [12] achieve a higher quality of surveillance than PEAS. However, the energy saving of it is lower than that of PEAS due to the higher message exchange overhead.

Heinzelman et al. [13] proposed low-energy adaptive clustering hierarchy, a cluster-based protocol utilizing randomized rotation of cluster heads to evenly distribute work load among the sensors. In this approach, the operation is divided into cycles and each cycle includes a cluster head selection, data aggregation and transmission. Since cluster heads have more responsibilities like data aggregation and transmission and hence consume more energy, different sensors become cluster heads in each cycle to prevent the cluster heads from running out of energy first. Cluster heads are self-elected at the beginning of each cycle. To conserve energy, non-head sensors are turned off at all times except during their transmission time. It rotates the cluster heads in a randomized fashion to achieve balanced energy consumption and sensors have synchronized clocks so that they know the beginning of a new cycle. Sensors do not need to know location or distance information.

Heinzelman and Chandrakasan [14] further improved the approach in [13] in two major aspects. First, the authors proposed a cluster head selection algorithm for sensor networks that have non-uniform starting energy level among the sensors. However, this algorithm assumes that sensors have global information about other sensors' remaining energy. Second, the authors determined that, under certain assumptions, the required number of cluster heads has to scale as the square root of the total number of sensor nodes to minimize the total energy consumption.

Bandyopadhyay and Coyle [15, 16] considered a simple strategy to select cluster heads. There are two kinds of cluster heads: volunteer cluster heads and forced cluster heads. Each sensor can become a volunteer cluster head. A volunteer cluster head advertises itself to the neighboring sensors, which then forward the advertisement within k hops. Any non cluster head sensor that receives such advertisements joins the cluster of the closest cluster head. Any sensor not associated with a cluster within t units of time becomes a forced cluster head. The schemes in [15] and [16] focus mainly on reducing the communication cost between sensors and their cluster heads. The cluster heads may run out of energy before the other sensors, but the authors did briefly mention two mechanisms to balance the energy consumption No clock synchronization is needed and sensors do not need to know location or distance information.

He et al. [17] designed the Energy-Efficient Surveillance System, in which the tradeoff between energy consumption and surveillance performance is explored by adaptively adjusting the sensitivity of the system. Two different schemes, proactive control and reactive control, to determine the sleep and wake-up cycle are described in [17]. The reactive scheme is apparently stealthier compared to the proactive scheme. Its drawback is that the clocks of the non-sentry sensors may drift in course of time, and as a result, a sentry may need to transmit an awake beacon repeatedly to wake up the non-sentries. This approach saves energy by putting the non-sentry nodes to sleep most of the time. Clock synchronization is required at the beginning of each cycle and sensors do not need to know location or distance information.

Deng et al. [18] proposed a sleep-scheduling algorithm, a scheme for cluster-based high density sensor networks. The goal is to reduce energy consumption while maintaining adequate sensing coverage capabilities [18]. To achieve this goal, the scheme selects sensors farther away from the cluster head to sleep with higher probabilities. The rationale behind this scheme is based on the assumption that each sensor's radio transceiver is capable of changing its transmission power in continuous steps to achieve different transmission ranges; a farther away sensor needs more power to communicate with the cluster head, and therefore, has higher energy consumption. This scheme only considers static clusters. In other words, cluster heads are not changed once they are selected.

Deng et al. proposed Balanced-energy Sleep Scheduling (BS) in [19]. It extends the scheme [18] by evenly distributing the sensing and communication tasks among the non-head sensors so that their energy consumption is similar regardless of their distance to the cluster-head.

3 PROPOSED APPROACH

In this section, the proposed approach and its relevant basics are discussed.

3.1 Node Deployment Strategy

In general, without having any fixed strategy, we can able to deploy the sensors by aircraft as shown in Figure 1 and these sensors spread out in random fashion, but there may be the problem of trees and landscapes. To avoid these types of problems, the researchers tried to identify the feasible node deployment strategies. Key issues in the node deployment strategies are: the coverage area of the nodes, finding the dead nodes in the coverage area as well as the nodes on the boundary of the other nodes coverage area. Due to this, more attention of the researches moved to the node deployment methods with and without the constraints according to the applications and in general.

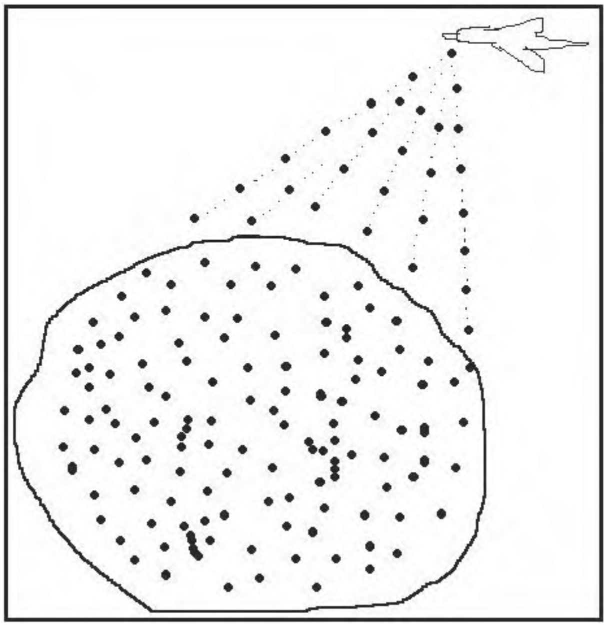


Figure 1: Node deployment

3.2 Node Deployment Related Work

Potential-field-based approach for node deployment is deeply discussed in [20], in which nodes are treated as virtual particles, subject to virtual forces. These forces repel the nodes from each other and from obstacles, and ensure that an initial, compact configuration of nodes will quickly spread out to maximize the coverage area of the network. In addition to these repulsive forces, nodes are also subject to a viscous friction force. This force is used to ensure that the network will eventually reach the state of static equilibrium, i.e. all nodes will ultimately come to a complete stop. The viscous force doest not, however, prevent the network from reacting to changes in the environment; if something is moved, the network will automatically reconfigure itself for the modified environment before return once again to a static equilibrium. Thus, nodes move only when it is necessary to do so, saving a great deal of energy. A hybrid approach based on clustering in [21] is used for load balancing, where the 2-D mesh is partitioned into 1-D arrays by row and by column. Two scans are used in sequence: one for all rows, followed by the other for all columns. Within each row and columns, the scan operation is used to calculate the average load and then to determine the amount of overload and under load in clusters. Load is shifted from overloaded clusters to under load clusters in an optimal way to achieve a balanced state. Each cluster covers a small square area and is controlled by cluster head, knows the information about cluster's position in the 2-D mesh and the number of sensors in the cluster. Limited motilities based approach is discussed in [22], where sensor can flip (or hop) only once to a new location and the flip distance is bounded. In this framework, the problem is to determine the optimal way for flip based sensors to maximize the coverage in the network. After detecting the coverage holes, the sensors move to new position to prevent coverage hole. Such movement can be realized in practice by propellers that are powered by fuel, coiled springs that unwinds for flipping. In this model, sensors can flip only once to a new location. Sensor node deployment method based on a centralized virtual force [23], which combines the idea of potential field and disk packing. In this a powerful cluster head, which communicate with all the other sensors, collect sensor position information, calculate forces and desired position for each sensor. The distance between two adjacent nodes when all nodes are evenly distributed is defined as a threshold to distinguish attractive or repulsive force between two nodes. The force between two nodes is zero if their distance is equal to the threshold, attractive if less than and repulsive if greater than. The total force on a node is the sum of all the forces given by other sensors together with obstacles and preferential coverage in the area. In [24], three protocols are evaluated for sensor network to maximize the sensor coverage with less time, movement distance and message complexity. These protocols first discover the existence of coverage holes in the target area based on the sensing service required by the application. After discovering a coverage hole, the protocols calculate the target positions of these sensors, where they should move. These three protocols are VEC (VECtor-based), VOR (VORonoi-based) and Minimax based on the principle of moving sensors from densely deployed areas to sparsely deployed areas.

For static environment, deterministic deployment is used since the location of each sensor can be predetermined properly. The stochastic deployment is used when the information of sensing area is not known in advance or is varied with time, that is the position for sensor deployment cannot be determined [25, 26].

In [27], a centralized deterministic sensor deployment method, DT-score is the basis. Given a fixed number of deployable sensors, DT-score aims to maximize the area coverage of sensing area with obstacles. In the first phase of DT-score, a contour-based deployment to eliminate the coverage holes near the boundary of sensing area and obstacles. In the second phase, a deployment method based on the Delaunay Triangulation is applied for uncover regions. Before deploying a sensor, each candidate position generated from the current sensor configuration is scored by a probabilistic sensor detection model.

3.3 Node Deployment Mapping

When the nodes are deployed randomly, the nodes organize themselves; this is known as self organizing nodes as shown in Figure 2. As the node positions are not remaining fixed, still the nodes organized themselves. Randomly deployed nodes can be organized by using the defined geometry, for example, triangularly, squarely, circularly organized. For this, we have to map the randomly deployed nodes by using the defined geometry and this geometry can be used to design scheduling and routing procedure or algorithm. These mappings are shown in Figure 3, Figure 4 and Figure 5 respectively.

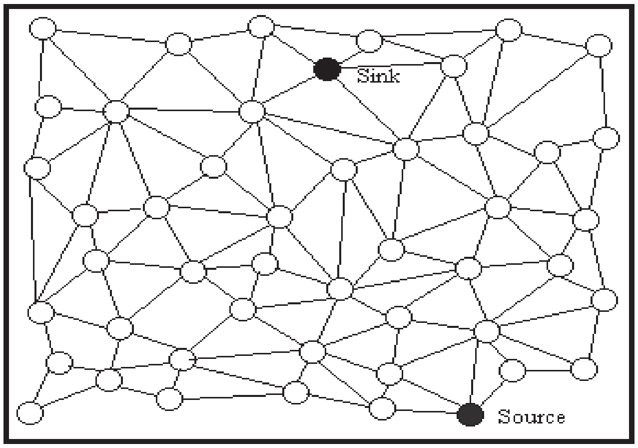


Figure 2: Self organizing nodes

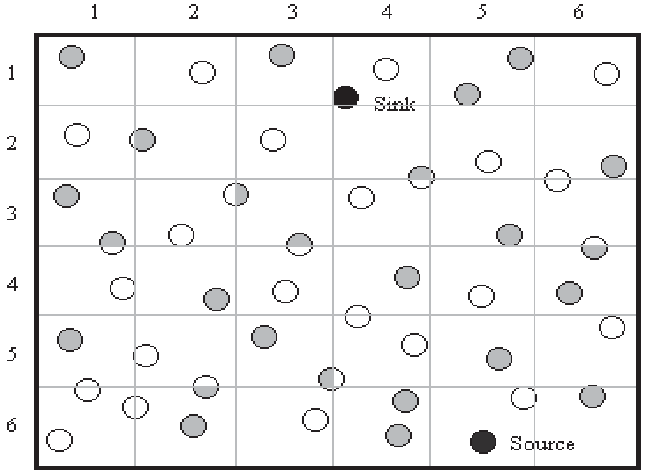


Figure 3: Square mapping

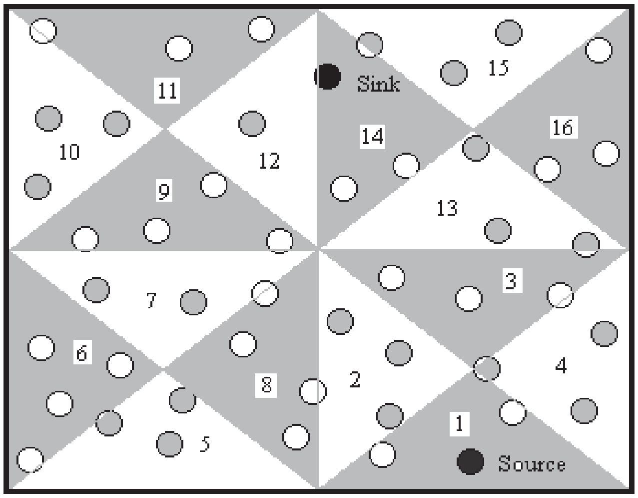


Figure 4: Triangular mapping

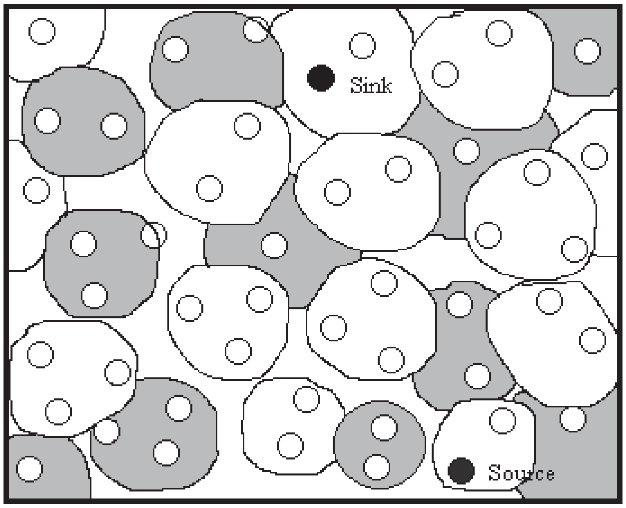


Figure 5: Circular mapping

In the presented work, we have adopted the square region mapping. For this, first we have identified the region of interest which is of square shape.

3.4 Intelligent Nodes

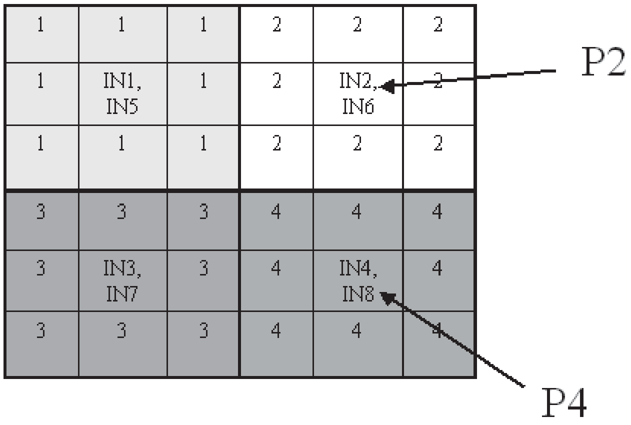
Main focus of this work is on the ability and utility of deploying intelligent nodes on a sensor network. Computational power along with the ability to communicate is often diminished significantly due to the severe power constraints put on a node when in a rugged environment. A wireless radio transmitter is almost always the biggest power consumer for a node. Thus reducing communications often provides a significant boost in the lifespan of a node. This requires an intelligent node to effectively balance the need to send data with the amount of power available to transmit and receive data.

The term intelligent nodes indicate that, our sensor nodes in WSN have some decision making capability, they can work together according to situation intelligently, so that the work can be done effectively and efficiently. The operation time can be increased; we can carry out the same operation for longer period of time with same number of sensor nodes. Intelligent nodes offer many interesting advantages to sensor networks that cannot be received otherwise.

3.5 Scheduling of Nodes

Scheduling of the sensor nodes means the selection or election of the particular nodes, which may take part in the actual communication to form the path between the source node and the sink or sinks. Which nodes will be in active or wake-up mode and which nodes will be in inactive or sleep mode that will be decided. The active nodes will take part in actual data transmission or communication and once their battery is low or they fail due to some reason, any one node in the same region, which is in inactive or sleep mode, will be made wake-up or active. The communication will be continued, until all the nodes in that region are dead.

The routing procedure will take care of the other parameters or constraints of the nodes. We can use any predefined routing protocol or we can make our own. Different scheduling procedures are possible with existing node deployment (Simple Scheduling) and with automatic change in node deployment due to the environmental change (Adaptive Scheduling). Scheduling of nodes is shown in Figure 6.



1	1	1	2	2	2
1	IN1, IN5	1	2	IN2, IN6	2
1	1	1	2	2	2
3	3	3	4	4	4
3	IN3, IN7	3	4	IN4, IN8	4
3	3	3	4	4	4

Figure 6: Scheduling of Intelligent Nodes in WSN

3.6 Initial Assumptions

To design the algorithm, we have made following assumptions:

1. The area of the region must be known before deploying the Intelligent Nodes (**INs**) in the WSN (here square area).
2. Take finite number of Intelligent Nodes (**INs**) to be deployed in WSN.
3. Take finite lifetime for the batteries used for the **INs** that are being deployed in the WSN.
4. Each Sensor **IN** is covering a specified area in WSN (here 8 neighbors).
5. Each **IN**, which is active, is communicating directly with the sink node.
6. Let P1, P2, P3…. denote Locations at which the INs are deployed.
7. Let Locations may be Labeled as following: -
    a. EMPTY at the beginning.
    b. PASS if work at that Location is complete.
    c. FAIL if work at that Location is incomplete.

8. Let the **INs** may be in the following Modes: -
    a. TURNOFF mode if the battery of **IN** is Low or it is not working.
    b. STANDBY mode if any other **IN** is working at that Location.
    c. WORK mode if it is working at that Location.

3.7 Algorithm (For Square Region)

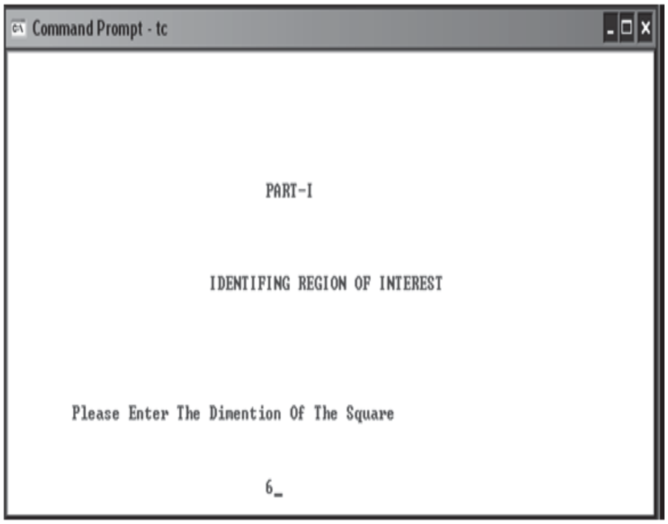
Following are the steps of execution of the algorithm:

1. START.
2. Initially keep all the available **INs** to STANDBY mode and LABEL all the Locations (P1, P2, P3….) to EMPTY at which the **INs** are to be schedule.
3. Schedule all the required **INs** at the required Location and keep them in WORK mode.
4. Schedule the remaining **INs** at the Location where already an **IN** exists and keep them in STANDBY mode.
5. For each Location in the WSN repeat the steps 6 to 7 until they are LABELED either PASS or FAIL.
6. If work at the Location over, LABEL that Location as PASS.
7. ELSE IF the **IN** fails or battery of the **IN** is low then
    a. Make that **IN** to TURNOFF mode.
    b. IF any other **IN** is available at that Location that is in STANDBY mode then change the mode of the **IN** to WORK mode.
    c. ELSE Label that Location as FAIL.
8. If all the Locations in WSN are LABELED PASS then signal SUCCESS.
9. ELSE signal FAILURE.
10. STOP.

4 EXPERIMENTAL RESULTS

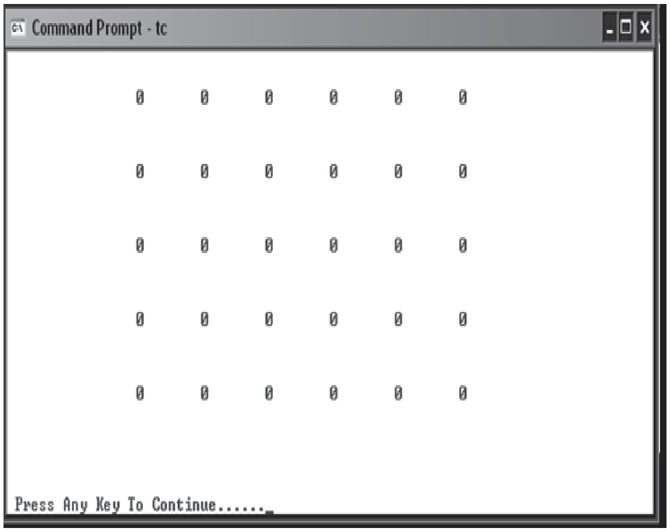
Experimentation is carried out with C++. The proposed algorithm is implemented in C++ with certain assumptions. Obtained results are preliminary where we have not considered the actual power consumption of the nodes. Here we have assumed the linear happening which may not be the actual case. So, we are calling the obtained results are preliminary. Various implementation results are shown through the screenshots in Figure 7 to Figure 16.

After running the program the first output screen appears is shown in Figure 7. The input required is the dimension of the square region. The Input provided here is 6. It will create the region by interest. Figure 8 shows the region of interest before deploying the intelligent nodes (IN). Now the number of nodes to be deployed is to be entered. Now we have to enter the number of sensor nodes that we want to deploy in the region of interest. Here we have taken 9 nodes to be deployed in the region as shown in Figure 9. Once the nodes are deployed in the network, the next task is location discovery of sensor nodes in the region of interest as shown in Figure 10. After the location discovery, the nodes are self-organized, as they are programmed intelligently. Figure 11 shows the nodes, their status and whether they are on, off, or standby mode. Figure 12 shows the communication start screen. Actual working starts self organization. The nodes have discovered their location. Now depending on the situation the working is started. Initially the nodes which are having the highest priority are turned to work mode and all other nodes will be in sleep mode. Value in the status variable gives the actual status of the nodes. If it contains value 0, it means it is in turn off mode and the node is dead. The value 1 indicates that the node is in standby mode and if the value is 2 it means the node is in work mode. Figure 13 shows, first node of every part is now in the active mode, rest of all is in the standby mode. It also indicates the working of the every first node in the parts of the region. The initial energy is for every node is 10 joule and the location of every node is fixed. As the communication starts, the node power decreases as shown in Figure 14 i.e. 1st node battery power decreases. Figure 15 shows the communication start for IN4. The 1st IN fails then 2nd IN become active, 2nd IN fails 3rd become active and so on…. Figure 16 represents the work done in the particular region.



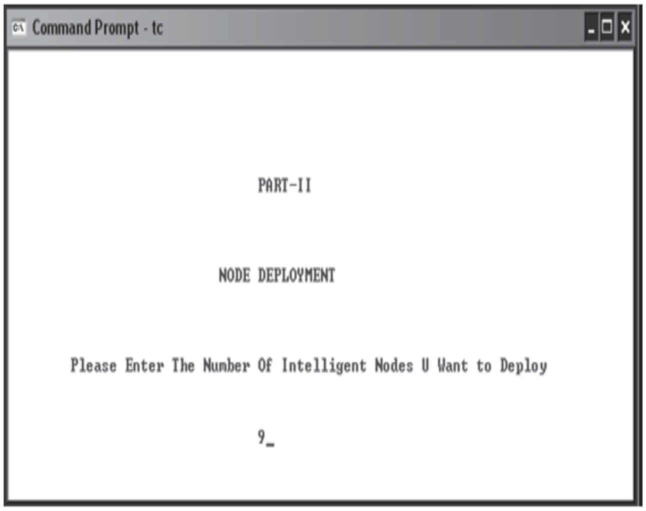
```
PART-I

IDENTIFING REGION OF INTEREST

Please Enter The Dimention Of The Square

6_
```

Figure 7: Identifying region of interest



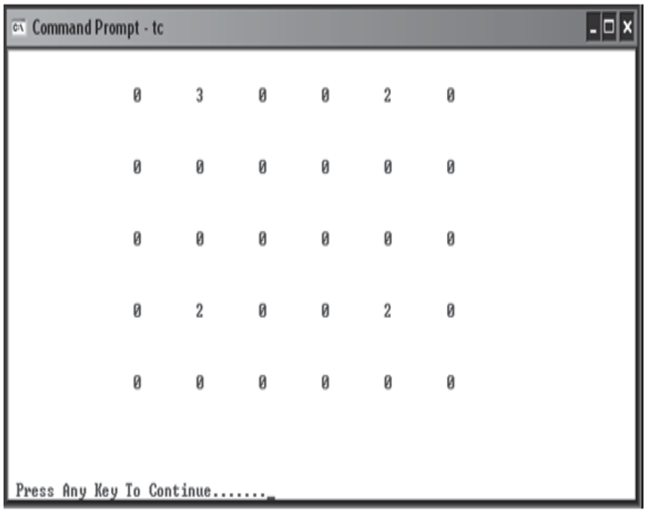
```
0     0     0     0     0     0

0     0     0     0     0     0

0     0     0     0     0     0

0     0     0     0     0     0

0     0     0     0     0     0

Press Any Key To Continue......_
```

Figure 8: Region of interest



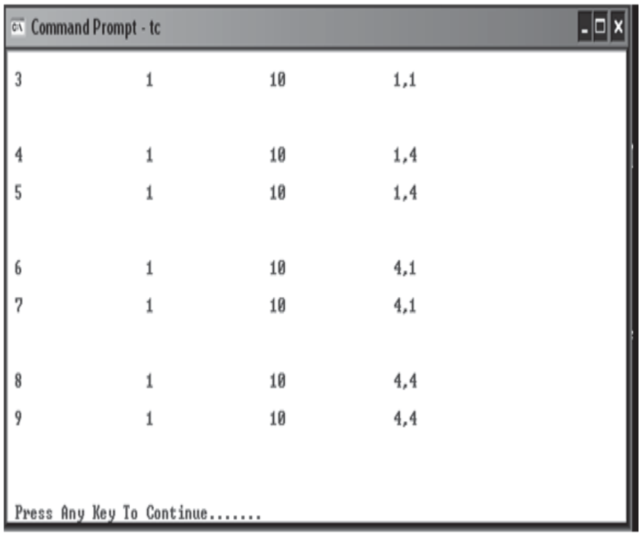
```
                    PART-II


                NODE DEPLOYMENT


     Please Enter The Number Of Intelligent Nodes U Want to Deploy


                    9_
```

Figure 9: Sensor node deployments



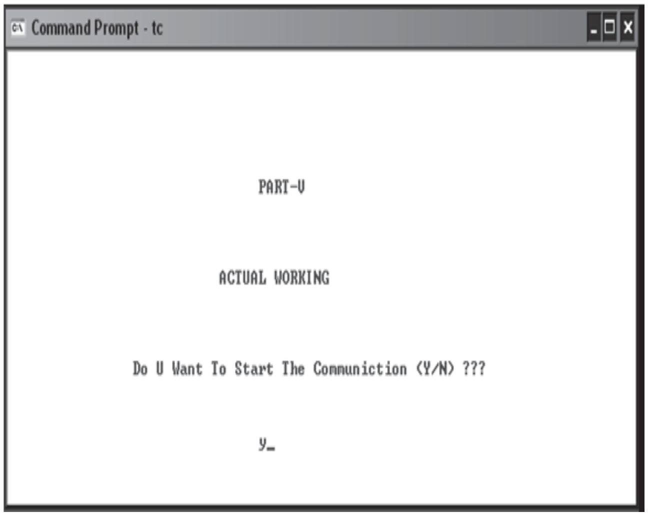
```
        0    3    0    0    2    0

        0    0    0    0    0    0

        0    0    0    0    0    0

        0    2    0    0    2    0

        0    0    0    0    0    0



Press Any Key To Continue......._
```

Figure 10: Location discoveries of sensor nodes



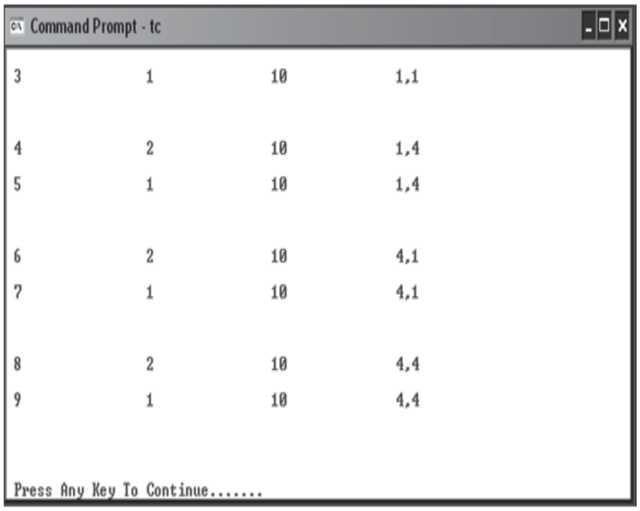
```
3          1          10          1,1


4          1          10          1,4
5          1          10          1,4


6          1          10          4,1
7          1          10          4,1


8          1          10          4,4
9          1          10          4,4



Press Any Key To Continue.......
```

Figure 11: Self organizations of sensor nodes



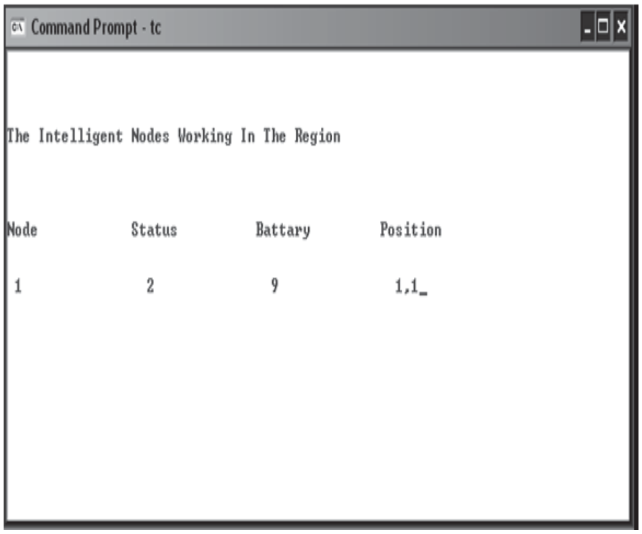
```
                    PART-V


                ACTUAL WORKING


     Do U Want To Start The Communiction (Y/N) ???


                    y_
```

Figure 12: Actual working



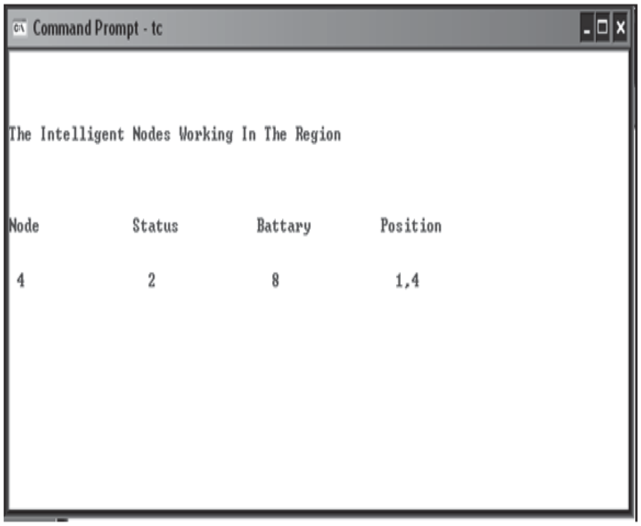
```
3          1          10          1,1


4          2          10          1,4
5          1          10          1,4


6          2          10          4,1
7          1          10          4,1


8          2          10          4,4
9          1          10          4,4



Press Any Key To Continue.......
```

Figure 13: Status of nodes at the initial stage.



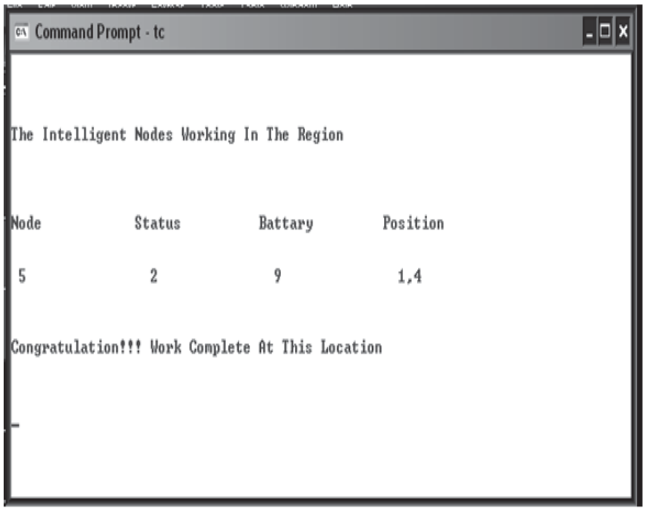
```
The Intelligent Nodes Working In The Region


Node          Status          Battary          Position

 1              2               9               1,1_
```

Figure 14: 1st node battery power decreases



```
The Intelligent Nodes Working In The Region


Node            Status          Battary         Position

 4                2               8               1,4
```

Figure 15: IN4 start communication in the network



```
The Intelligent Nodes Working In The Region


Node            Status          Battary         Position

 5                2               9               1,4


Congratulation!!! Work Complete At This Location


_
```

Figure 16: Work done in the region

5 CONCLUSION AND FUTURE SCOPE

As the main limitation of the sensor network is its lifetime and the energy consumption. Sensor nodes are equipped with the limited battery life. To utilize the battery in optimal way, the nodes should be used in proper way to manage the energy consumption. Therefore, the nodes should be scheduled optimally. This paper presented the approach where the intelligent nodes are used which is capable to manage the energy consumption in optimal way and consequently increases the lifetime of the sensor network. In implementation, we have considered only one hop communication and experimentation is carried out with C++. Presented results are the preliminary results with certain assumptions. These results can be enhanced by performing the exact simulation of the proposed approach in NS-2. Even, there is a scope to go for the heterogeneous network where the nodes are of different battery life. In presented work, we have assumed that each node is having the same battery life. Scope is there to analyze the experimentation results by varying the assumptions.

6 REFERENCES

[1] D. Culler, D. Estrin and M. Srivastava, “Guest Editors' Introduction: Overview of Sensor Networks”, IEEE Computer, vol. 37, issue 8, pp. 41-49, Aug. 2004.

[2] K. L. Mills, “A brief survey of self-organization in wireless sensor Networks”, Wireless Communication and Mobile Computing, vol. 7, issue 7, pp. 823-834, Sept. 2007.

[3] S. Kumar, T. H. Lai and J. Balogh, “On K-coverage in a mostly sleeping sensor network”, Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (Mobicom ’04), Philadelphia, Pennsylvania, pp. 144–158, 2004.

[4] D. Tian and N. D. Georganas, “A coverage-preserving node scheduling scheme for large wireless sensor networks”, Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA ’02), Atlanta, Georgia, pp. 32–41, 2002.

[5] P. Berman, G. Calinescu, C. Shah and A. Zelikovsly, “Efficient energy management in sensor networks”, Y. Xiao & Y. Pan (Eds.), Ad hoc and sensor networks, Nova Science, 2005.

[6] K. Wu, Y. Gao, F. Li and Y. Xiao, “Lightweight deployment-aware scheduling for wireless sensor networks”, ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on “Energy Constraints and Lifetime Performance in Wireless Sensor Networks” , vol. 10, issue 6, pp. 837–852, Dec. 2005.

[7] F. Ye, G. Zhong, J. Cheng, S. Lu and L. Zhang, “PEAS: A robust energy conserving protocol for long-lived sensor networks”, Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS ’03), pp. 28–37, 2003.

[8] H. Zhang and J. C. Hou, “Maintaining sensing coverage and connectivity in large sensor networks”, Proceedings of NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, Feb. 2004.

[9] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks”, Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (Sensys ’03), Los Angeles, California, pp. 28–39, 2003.

[10] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, “SPAN: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks”, Wireless Networks, vol. 8, issue 5, pp. 481–494, 2002.

[11] A. Cerpa and D. Estrin, “ASCENT: Adaptive Self-Configuring Sensor Networks Topologies”, Proceedings of IEEE INFOCOM 2002, New York, New York, June 2002.

[12] C. Gui and P. Mohapatra, “Power conservation and quality of surveillance in target tracking sensor networks”, Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (Mobicom ’04), Philadelphia, Pennsylvania, pp. 129– 143, 2004.

[13] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient communication protocols for wireless

microsensor networks", Proceedings of Hawaaian International Conference on Systems Science, Jan. 2004.

[14] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "An application-specific protocol architecture for wireless micro-sensor network", IEEE Transactions on Wireless Communications, vol 1, issue 4, pp. 660–670, Oct. 2002.

[15] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks", INFOCOM, 2003.

[16] S. Bandyopadhyay and E. J. Coyle, "Minimizing communication costs in hierarchically-clustered networks of wireless sensors", Computer Networks, vol. 44, pp. 1–16, 2004

[17] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru et al., "Energy-efficient surveillance system using wireless sensor networks", Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys '04), Boston, Massachusetts, pp. 270–283, 2004.

[18] J. Deng, Y. S. Han, W. B. Heinzelman and P. K. Varshney, "Scheduling sleeping nodes in high density cluster-based sensor networks", ACM/Kluwer Mobile Networks and Applications, vol. 10, issue 6, Springer Netherlands, pp. 825–835, Dec. 2005.

[19] J. Deng, Y. S. Han, W. B. Heinzelman and P. K. Varshney, "Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks", Proceedings of the 4th Workshop on Applications and Services in Wireless Networks (ASWN '04), Boston, Massachusetts, 2004.

[20] A. Howard, M. J. Mataric and G. S. Sukhatme, "Mobile Sensor Networks Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem," the 6th International Symposium on Distributed Autonomous Robotics Systems, Japan, June, 2002.

[21] J. Wu and S. Wang, "Smart: A scan-based movement – assisted deployment method in wireless sensor networks", Proceedings of IEEE Conference on Computer Communication (INFOCOM), Miami, March, 2005.

[22] S. Chellappan, X. Bai, D. Ma and D. Xuan, "Sensor Deployment Using Flip-Based Sensors".

[23] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," Proceedings of INFOCOM, March, 2003.

[24] G. Wang. G. Cao and T. La Porta, "Movement- assistd sensor deployment", Procedding of IEEE Conference on ComputerCommunication (INFOCOM), Miami, March 2005.

[25] S. Meguerdichian , F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Network," IEEE INFOCOM'01, April, 2001.

[26] C. Huang, and Y. Tseng, "The Coverage Problem in a wireless Sensor Network," Proceeding of WSNA '03, pp.115-121, 2003.

[27] C-H Wu, K-Chuan and Y-C Chung, "A Delaunay Triangulation Based Method For Wireless Sensor Netwrok Deployment," Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS'06), IEEE, 2006.