Distributed Fault Detection and Correction Using Shortest Path Mechanism in Wireless Sensor Networks

Ankit Arora[1] Sonika Soni[2]
[1] M.tech [2] Assistant Professor
[1,2] Department of ECE, JCDMCOE, Sirsa, Haryana, India.

Abstract—Wireless sensor network is a self-organized network that consists of a large number of low-cost and low-powered sensor devices, called sensor nodes. Recent advances in wireless sensor networks have resulted in a unique capability to remotely sense the environment. These systems are often deployed in remote or hard-to reach areas. Hence, it is critical that such networks operate unattended for long durations. Unlike the cellular networks and ad hoc networks where energy has no limits in base stations or batteries can be replaced as needed, nodes in sensor networks have very limited energy and their batteries cannot usually be recharged or replaced due to hostile environments. Therefore, extending network lifetime through the efficient use of energy has been a key issue in the development of wireless sensor networks. In distributed fault detection (DFD) scheme for wireless sensor networks the status of each sensor node to be either good or faulty is based on the neighboring nodes, but in the DFD algorithm[1],when the sensor fault probability increases the fault detection accuracy decreases and the false alarm rate increases rapidly. In this paper an improved DFD scheme is proposed to detect intermittently faulty sensor nodes and to stringent power budget during fault diagnosis process on sensor nodes in wireless sensor network. Simulation results demonstrates that the improved DFD scheme reduce energy consumption in comparison with previous algorithm. The proposed mechanism is implemented with MATLAB.

Keywords:- Wireless Sensor Networks, Distributed Fault Detection, Fault Diagnosis, Self-Managing Fault Management Mechanism

I. INTRODUCTION

Wireless sensor network (WSN) is widely considered as one of the most important technologies for the twenty-first century [1]. In the past decades, it has received tremendous attention from both academia and industry all over the world. A WSN typically consists of a large number of low-cost, low-power, and multifunctional wireless sensor nodes, with sensing, wireless communications and computation capabilities. Many researchers have been working towards fault detection in WSNs. However, the limited ability of individual sensor nodes and the tremendous scale of a sensor network make detecting failures and ensuring network availability more difficultly. One reason behind the growing popularity of wireless sensors is that they can work in remote areas without manual intervention. All the user needs to do is to gather the data sent by the sensors, and with certain analysis extract meaningful information from them. Usually sensor applications involve many sensors deployed together. These sensors form a network and collaborate with each other to gather data and send it to the base station. The base station acts as the control center where the data from the sensors are gathered for further analysis and processing. In a nutshell, a wireless sensor network (WSN) is a wireless network consisting of spatially distributed nodes which use sensors to monitor physical or environmental conditions. These nodes combine with routers and gateways to create a WSN system. The WSN is made of nodes from a few to several hundred, where each node is connected to one or several sensors. The basic components of a node are:

- Sensor and actuator - an interface to the physical world designed to sense the environmental parameters like pressure and temperature.
- Controller - is to control different modes of operation for processing of data
- Memory - storage for programming data.
- Communication - a device like antenna for sending and receiving data over a wireless channel.
- Power Supply- supply of energy for smooth operation of a node like battery.

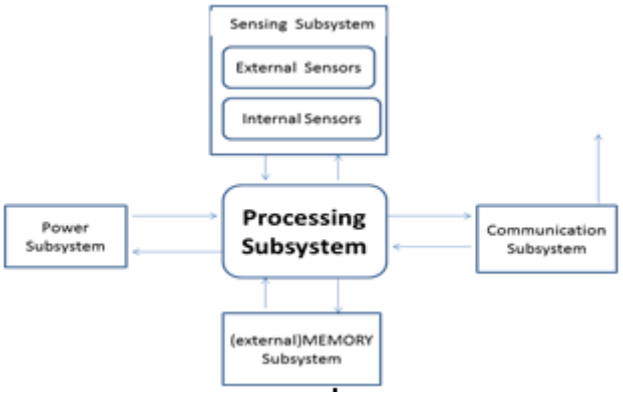

Fig. 1: A Wireless Sensor Node

In this approach, the message of updating the node residual battery is applied to track the existence of sensor nodes. A cell manager employs the self-detection approach and regularly monitors its residual energy status. All sensor nodes start with the same residual energy. After going through various transmissions, the node energy decreases. If the node energy becomes less than or equal to 20% of battery life, the node is ranked as low energy node and becomes liable to put to sleep. If the node energy is greater or equal to 50% of the battery life, it is ranked as high and becomes the promising candidate for the cell manager. Thus, if a cell manager residual energy becomes less than or equal to 20% of battery life, it then triggers the alarm and notifies its cell members and the group manager of its low energy status and appoints a new cell manager to replace it.

II. FAULT DIAGNOSIS

Detection of faulty sensor nodes can be achieved by two mechanisms i.e. self-detection (or passive-detection) and

active-detection. In self-detection, sensor nodes are required to periodically monitor their residual energy, and identify the potential failure. In this scheme, we consider the battery depletion as a main cause of node sudden death. A node is termed as failing when its energy drops below the threshold value. When a common node is failing due to energy depletion, it sends a message to its cell manager that it is going to sleep mode due to energy below the threshold value .This requires no recovery steps[6]. Self-detection is considered as a local computational process of sensor nodes, and requires less in-network communication to conserve the node energy. In addition, it also reduces the response delay of the management system towards the potential failure of sensor nodes. To efficiently detect the node sudden death, our fault management system employed an active detection mode. In this approach, the message of updating the node residual battery is applied to track the existence of sensor nodes[7]. In active detection, cell manager asks its cell members on regular basis to send their updates. Such as the cell manager sends "get" messages to the associated common nodes on regular basis and in return nodes send their updates. This is called in-cell update cycle. The update_msg consists of node ID, energy and location information. As shown in figure, exchange of update messages takes place between cell manager and its cell members. If the cell manager does not receive an update from any node then it sends an instant message to the node acquiring about its status .If cell manager does not receive the acknowledgement in a given time, it then declares the node faulty and passes this information to the remaining nodes in the cell. Cell managers only concentrate on its cell members and only inform the group manager for further assistant if the network performance of its small region has been in a critical level. Every cell manager sends health status information to its group manager. This is called out-cell update cycle and are less frequent than in-cell update cycle. If a group manager does not hear from a particular cell manager during out-cell update cycle, it then sends a quick reminder to the cell manager and enquires about its status. If the group manager does not hear from the same cell manager again during second update cycle, it then declares the cell manager faulty and informs its cell members. This approach is used to detect the sudden death of a cell manager.

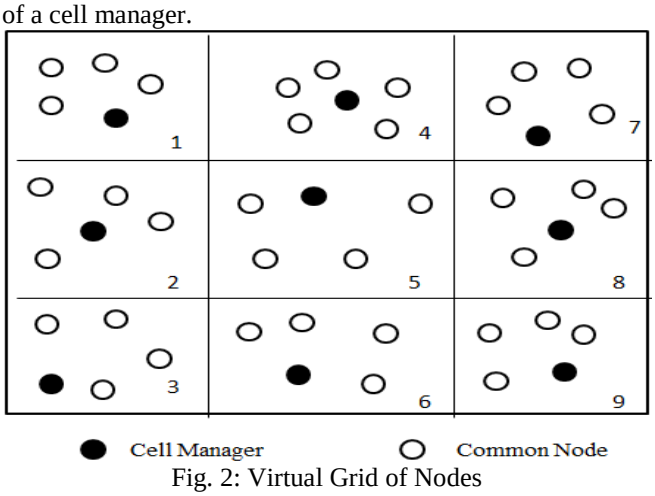

Fig. 2: Virtual Grid of Nodes

Group manager also monitor its health status regularly and respond when its residual energy drops below the threshold value. It notifies its cell members and neighboring group managers of its low energy status and an indication to appoint a new group manager. Sudden death of a group manager can be detected by the base station. If the bases station does not receive any traffic from a particular group manager, it then consults the group manager and asks for its current status. If the base station does not receive any acknowledgement, it then considers the group manager faulty (sudden death) and propagates this information to its cell managers. The base station primarily focuses on the existence of the group managers from their sudden death. Meanwhile, the group managers and cell managers take most parts in passive and active detection in the network.

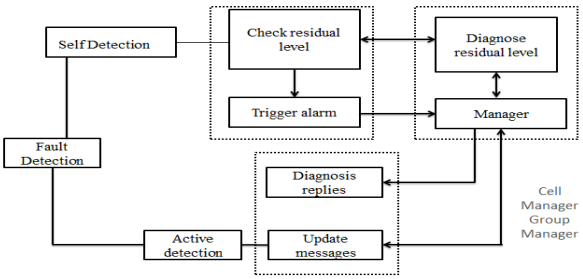

Fig. 3: Fault detection diagnosis

After nodes failure detection (as a result of self-detection or active detection) [3], sleeping nodes can be awaked to cover the required cell density or mobile nodes can be moved to fill the coverage hole. A cell manager also appoints a secondary cell manager within its cell to acts as a backup cell manager. Cell manager and secondary cell manager are known to their cell members.If the cell manager energy drops below the threshold value (i.e. less than or equal to 20% of battery life), it then sends a message to its cell members including secondary cell manager. It also informs its group manager of its residual energy status and about the candidate secondary cell manager. This is an indication for secondary cell manager to stand up as a new cell manager and the existing cell manager becomes common node and goes to a low computational mode. Common nodes will automatically start treating the secondary cell manager as their new cell manager and the new cell manager upon receiving updates from its cell members; choose a new secondary cell manager[5]. The failure recovery mechanisms are performed locally by each cell. In Figure let us assume that cell 1 cell manager is failing due to energy depletion and node 3 is chosen as secondary cell manager. Cell manager will send a message to node 1, 2, 3 and 4 and this will initiate the recovery mechanism by invoking node 3 to stand up as a new cell manager.

III. Self-Managing Fault Management Mechanism For WSN

We will assume a scenario on sensor nodes. Our aim is to transfer information from one node to another. But this path should overcome all the faults in between. A new technique is introduced for detecting faulty nodes and rerouting the path to get the shortest possible path from sender to receiver which is fault free. In this approach a new fault management mechanism was proposed to deal with fault detection. It proposes a hierarchical structure to properly distribute fault management tasks among sensor nodes by heavily introducing more self-managing functions. The proposed failure detection algorithms have been compared with some

existing related algorithm and proven to be more energy efficient.Short-path algorithms generally have polynomial complexity and generally only produce a single path between a source and destination. In shortest path routing, the topology network is represented using a directed weighted graph. The nodes in the graph represent switching elements and the directed arcs in the graph represent communication links between switching elements. Each arc has a weight that represents the cost of sending a packet between two nodes in a particular direction. This cost is generally a positive value that can inculcates such factors as delay, throughput, and error rate, monetary cost etc [7]. A path between two nodes may go through several intermediary nodes and arc. The objective in shortest path routing is to find a path between two nodes that has the smallest total cost, where the total cost of a path is the sum of the arc costs in that path.Shortest-path algorithms can be divided into two classes: distance vector and link state. Distance vector algorithms are based on dynamic programming models and can be implemented in a distributed, asynchronous framework using local cost estimates. The basic distance vector algorithm is known as the Bellman-ford algorithm or the ford-Fulkerson method. Link state algorithms are usually implemented in a replicated fashion with each switch performing an independent route computation. To perform, link cost estimates are required for every link in the network. The basic link state method is Dijkstra's algorithm.[10] The Dijkstra's algorithm builds the local positions and routings of the estimated sensors for applications that require absolute coordinates of nodes, waiting until large number sensor nodes has formed before transforming to absolute coordinates may be a poor choice. Using the method described here, Position estimation using the shortest path method between source node and destination node with low cost in wireless sensor networks that compute absolute coordinates of individual nodes or sub networks independently can be developed. Shortest path method can be extended by applying more advanced MDS techniques. It is the best algorithm to find out shortest path position estimation method in wireless sensor networks.To obtain this,some steps are followed:

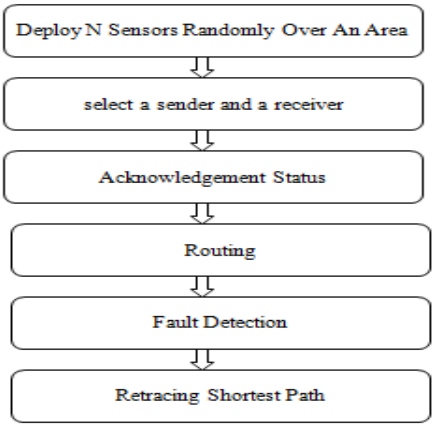

Fig. 4: Self-Managing Fault Management Mechanism

We assume that sensors are randomly deployed in the interested area which is very dense and all the sensors have a common transmission range.[12]The dark circles in the figure represent faulty sensors and the gray circles are good sensors. There might be a failure occurring in a certain area as illustrated in the figure 5.All sensors in this area go out of service.

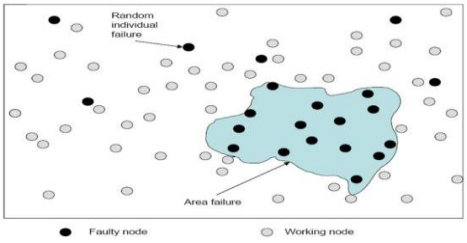

Fig. 5: Sensor nodes randomly deployed over an area[10]

As we are depending on majority voting among the sensors, we assume that each sensor node has at least 3 neighboring nodes. Because a large amount of sensors are deployed into the interested area to form a wireless network, this condition can be easily obtained. Each sensor node is able to locate its neighbors within its transmission range via a broadcast/ acknowledge protocol. Faults can occur at different levels of the sensor network,[6] such as system software, hardware, physical layer, and middleware.In this mechanism, we focus on hardware level faults by assuming the power supply infrastructure. Sensor nodes are still capable of receiving, sending, and processing when they are faulty in the algorithm.

Let n be the total number of sensor Nodes; p be the probability of failure nodes; q be the neighbor sensors; r be the set of all the sensors; $N(r_i)$ be the set of the neighbors of r_i. cor be measurement of r_i; Dis be measurement difference between r_i and r_j at time t; then

$Dis^{t}_{ij}=r_i-r_j$; ……(1)

$\Delta t=t_{l+1}-t_l$; ……(2)

$\Delta Dis^{\Delta tl}_{ij}$ be measurement difference in time t_l to t_{l+1}.

$\Delta Dis^{\Delta tl}_{ij}=\Delta dist^{\Delta tl+1}_{ij}-\Delta dis^{\Delta tl}_{ij}$; ……(3)

$\Delta Dis^{\Delta tl}_{ij}=(cor^{\Delta tl+1}_{i}-coor^{\Delta tl+1}_{j})-(cor^{\Delta tl}_{i}-Cor^{\Delta tl}_{j})$; ……(4)

T_{ij} be test between r_i and r_j, $T_{ij} \epsilon \{0,1\}$, $T_{ij}= T_{ij}$;

θ_1 and θ_2 be two predefined threshold values;T_s be tendency value of a sensor, $T_s \epsilon \{LG, LF, GD, FT\}$.

Sensors are considered as neighboring sensors if they are within the transmission range of each other. Each node regularly sends its measured value to all its neighbors. We are interested in the history data if more than half of the sensor's neighbors have a significantly different value from it. We can find the current measurement is different from previous measurement. If the measurements change over the time significantly, it is more likely the sensor is faulty [3].A test result T_{ij} is generated by sensor r_i based on its neighbor r_j's measurements using two variables and two predefined threshold value. If a sensor is faulty, it can generate arbitrary measurements. If T_{ij} is 0, most likely either both r_i and r_j are good or both are faulty. Otherwise, if T_{ij} is 1, r_i and r_j are most likely in different status.

Each Sensor id r_i and Receiver id $r_j \epsilon N(r_i)$.

Set $T_{ij}=0$ and compute Dis^{t}_{ij}..

IF $|Dis^{t}_{ij}| > \theta_1$ THEN $T_{ij}=1$ and turn to the next node $N(r_i)$

IF $|Dist_{ij}| < \theta_1$ calculate Δs_{ij}

IF $|Dis^{t}_{ij}| > \theta_2$ THEN $T_{ij}=1$ and turn to the next node $N(r_i)$

Repeat above steps until the test results of each node in $N(r_i)$ with r_i all obtained.

IF $\sum_{rj \in N(ri)} T_{ij} < [|N(r_i)|/2]$,where $|N(r_i)|$ is the number r_i's neighbor nodes
THEN T_{si}=LG;
ELSE T_{si}=LF;
Communicate T_s to the neighbors.
IF $\sum_{rj \in N(ri) \text{ and } Tsj=LG} T_{ij} < [| N(r_i)_{Tsj=LG}/2|]$
THEN T_{si}=GD;
ELSE
T_{si}=FT;

Communicate T_{si} to the neighbors.[3]. The next step Follows that If there are no neighbor nodes of r_i whose initial detection status is LG, and if the initial detection status Tsi of r_i is LG, then set the status of r_i as normal (GD), otherwise as fault(FT). Check whether detection of the status of all nodes in network is completed or not. If it has been completed, then exit. Otherwise, repeat steps. Once the Faulty Nodes have been detected a shortest path can be created using a Bellman-Ford Algorithm. As there are various algorithms for the shortest path but they not work for the negative edges. For this purpose let us consider that $\delta^l_G(s, t)$ = the shortest weighted path from s to t using at most l edges. We can start by determining $\delta^1_G(s, v)$ for all vϵV, which is infinite for all vertices except s itself, for which it is 0. Then perhaps we can use this to determine G(s, v) for all vϵV. In general we want to determine $\delta^{k+1}_G(s, v)$ based on all $\delta^k_G(s, v)$. The question is how do we calculate this. It turns out to be easy since to determine the shortest path with at most *k* +1 edges to a vertex *v* all that is needed is the shortest path with *k* edges to each of its in-neighbors and then to add in the weight of the one additional edge. This gives us

$$\delta^{k+1}(v) = \min(\delta^k(v), \min_{x \epsilon N\text{-}(v)} (\delta^k(x) + w(x, v))) \quad \text{.....(5)}$$

Here is the Bellman Ford algorithm based on this idea. It assumes we have added a zero weight self loop on the source vertex.

```
1.  % implements: the SSSP problem
2.  function BellmanFord(G = (V, E), s) =
3.  let
4.  % requires: all{Dv = δ^k_G(s, v): vϵV}
5.  function BF(D, k) =
6.   let
7.  D´ ={v ↔min(Dv ,min_{uϵNG-(v)}(D_u + w(u, v))) : vϵV}
8.  In
9.  if (k = |V|) then  ⊥
10. else if (all{D_v = D_v´ v : vϵV})  then D
11. else BF(D , k +1)
12. end
13. D = {v ↔ if v = s then 0 else ∞: vϵV}
14.  in BF(D, 0) end
```

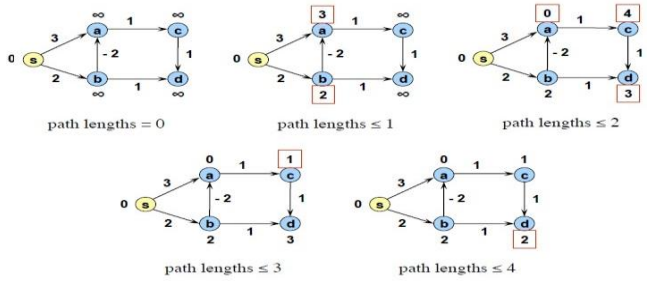

Fig. 6: Steps of the Bellman Ford algorithm. The numbers with red squares indicate what changed on each step.

IV. Simulation And Results

- In this approach a total of 3 scenarios will be implemented the one with no faulty nodes (ideal situation)
- the one where information will stop when a faulty nodes occur (faulty situation),and
- the one where faults will be detected and accordingly shortest path will be made(Proposed work)

The proposed mechanism was implemented with MATLAB. The evaluation results should demonstrate the ability of the mechanism to identify faulty nodes anciently and with limited overheads An example simulation scenario composed of total 100 sensor nodes which are randomly deployed.

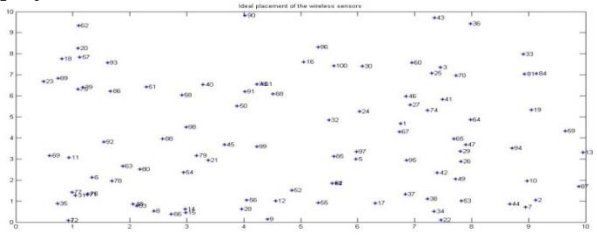

Fig. 7: Ideal Placement Of The 100 Sensor Nodes

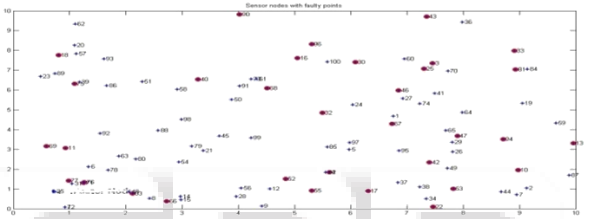

Fig. 8: Faults Detected In The Scenario

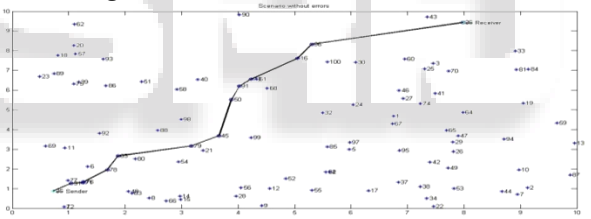

Fig. 9: Scenario 1: No Errors

This is the ideal scenario. Here, we assume that there are no faulty nodes. All the nodes are authentic and fault free. Information is securely transferred from sender to the receiver. We have selected the shortest path from sender to receiver.

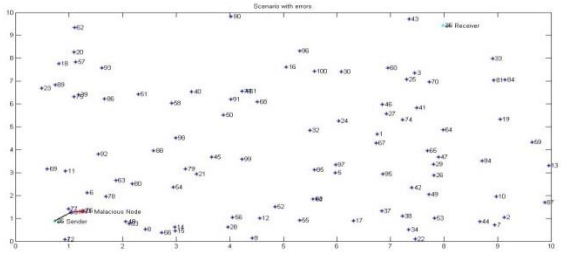

Fig. 10: Scenario 2: With Errors

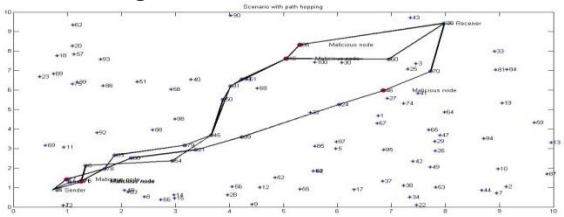

Fig. 11: Scenario 3: Path Hopping

This is the scenario when the sender finds more than 1 route to the receiver. And even after occurrence of a faulty node, the information loss does not intervenes in the route formation. The route is still completed even after a faulty node occurs. This scenario is meant to show the path hopping between sender and receiver. Information can be transferred from more than 1 route also[9].

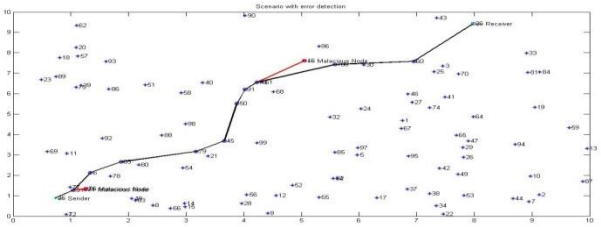

Fig. 12: Retraced Path After Fault Detection

This is the final scenario. Here, the faults are detected and the route will change accordingly. The system will find the shortest path between sender and receiver despite of fault occurrence.

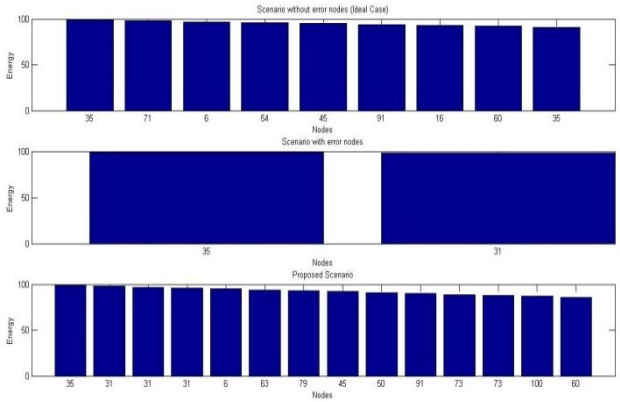

Fig. 12: Comparison Between No. Of Nodes And Energy Distribution

V. Conclusion And Future Scope

In the Distributed Fault Detection (DFD), there are various algorithm to determine the faulty nodes. We assume the case of power failure as there is no recovery techniques in those area. Therefore we have to change the direction of information when transmitted from a Sendor Node to the Receiver Node. This paper has presented a new strategy for power control in WSNs where operational longevity is an issue. As the deployment of Thousand Numbers of Sensor Nodes in Area needs Energy Performance.The new approach provides a methodology for the Retracing of Optimal Path with an Energy Efficiency and Accuracy[1]. This assessment becomes the power performance booster among the previous workout as it automatically determines the shortest path after path hopping is traced A self Management approach links the sensor nodes from the source node to the destination node with in a shortest path and shows distributed accuracy. It is estimated that by the year 2020 more than 100 billion wireless sensors will be deployed for applications as diverse as environmental monitoring, agricultural monitoring, machine health monitoring, surveillance, and medical monitoring.[11-13] These networks, which connect the physical world with the digital world, provide us with a richer understanding of our environment and with the ability to more accurately control our surroundings. However, there are many challenges that must be addressed before the full potential of these networks are realized. Wireless sensor networks must be reliable and scalable to support large numbers of unattended wireless sensors; they must last for extended periods of time using limited battery power; they must be secure against outside attacks on the network and on data fidelity; they must be accurate in providing required information while performing in-network processing to reduce data load; and they must interface with existing networks.

References

[1] Jinran Chen, Shubha Kher And Arun Somani. Distributed Fault Detection Of Wireless Sensor Networks, Dependable Computing And Networking Labiowa State Universityames,Iowa .Pages75-84,Computer Network,2006.

[2] A.A. Abbasi, M. Younis, "A Survey On Clustering Algorithms For Wireless Sensor Networks", Computer Communications, Vol. 30, Issue 14, Vol. 15, Pp. 2826-2841, October 2007.

[3] N. Vlajic, D. Xia, "Wireless Sensor Networks: To Cluster Or Not To Cluster", International Symposium On A World Of Wireless, Mobile And Multimea Networks(Wowmom06), Pp. 258-268, July 2006.

[4] Benahmed Khelifa1, H. Haffaf , Merabti Madjid, And David Llewellyn-Jones," Monitoring Connectivity In Wireless Sensor Networks", Vol. 2, No. 2, June, 2009

[5] B. Krishnamachari And S. Iyengar. Distributed Bayesian Algorithms For Fault-Tolerant Event Region Detection In Wireless Sensor Networks. Ieee Transactions On Computers, 53(3):241–250, March 2004.

[6] M. Yu, H. Mokhtar, And M. Merabti, "A Survey On Fault Management In Wireless Sensor Network," Inproceedings Of The 8th Annual Postgraduate Symposium On The Convergence Of Telecommunications, Networking And Broadcasting Liverpool, Uk, 2007.

[7] S. Marti, T. J. Giuli, K.Lai, And M. Baker, "Mitigating Routing Misbehaviour In Mobile Ad Hoc Networks," In Acm Mobicom, 2000, Pp. 255-265.

[8] F. Koushanfar, M. Potkonjak, And A. Sangiovannivincentelli, "Fault Tolerance Techniques In Wireless Ad-Hoc Sensor Networks," Uc Berkeley Technical Reports 2002.

[9] W. L. Lee, A. Datta, And R. Cardell-Oliver, "Winms: Wireless Sensor Network-Management System, An Adaptive Policy-Based Management For Wireless Sensor Networks," School Of International Journal Of Wireless & Mobile Networks (Ijwmn) Vol.2, No.4, November 2010.

[10] Http://En.Wikipedia.Org

[11] Chee-Yee Chong, Member, Ieee And Srikanta P. Kumar, Senior Member, Ieee "Sensor Networks: Evolution, Opportunities,And Challenges", Proceedings Of The Ieee, Vol. 91, No. 8, August 2003.

[12] A.Nippun Kumaar, Kiran.G, Sudarshan Tsb," Intelligent Lighting System Using Wireless Sensor Networks" Vol.1, No.4, December 2010

[13] Zahra Rezaei , Shima Mobininejad," Energy Saving In Wireless Sensor Networks", Vol.3, No.1, February 2012.

