ISSN 22773061

Fault Tolerant Event Detection in Distributed WSN via Pivotal Messaging

Chhavi Gupta, Rashmi Sharma, Neha Agarwal, Yashwant Singh
Department of Computer Science, Jaypee University of Information Technology, Waknaghat, Solan(HP), India
{chhavigupta.cs, rashmi.nov30,lko.neha }@gmail.com, yashu_want@yahoo.com

ABSTRACT

To mitigate the problem of loosing data, which can be very crucial to prevent environmental disasters, we proposed a fault tolerant event detection algorithm in a distributed environment. We considered the data loss because of node failure in sensor network due to some technical fault or energy issues. This algorithm works by backing up the data of cluster heads at some secondary nodes. If the secondary node that is backup cluster head notices the failure of primary or vital cluster-head, it will inform this to all non cluster-head nodes through pivotal messages. Subsequently, backup cluster-head will start working as vital cluster-head with new backup cluster-head. We have taken all parameters with the promising optimal use of energy. We have compared our results with the existing Dynamic Static Clustering Protocol (DSC) and fault tolerant Dynamic Clustering Protocol (FT-DSC). We have compared results on the basis of various parameters like energy consumption over time, number of data packet transmission and network life time based on network remaining energy.

Keywords

Wireless Sensor Networks; Event Detection; Fault Tolerance; Clustering; Energy Efficiency.

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 7, No 1
editor@cirworld.com
www.cirworld.com, member.cirworld.com

INTRODUCTION

Wireless Sensor Networks proved its worth in real time environment as it plays a vital role in predicting weather conditions. Advance estimate of the weather condition can be useful in preventing disasters to take place. The Wireless Sensor Network consists of a number of tiny nodes called the sensor nodes placed at distributed location in the networks. Each sensor nodes are incorporated with the sensing, processing and communication efficiencies. These sensor nodes are divided into clusters to improve the manageability. Each cluster is regulated by a cluster head.

The selection of the cluster head can be adaptive, deterministic or hybrid. Various algorithms are present for each of the categories like LMSSC [4], LEACH-C [5], LEACH-F [6], CBCDACP [7], and FZ-LEACH [8] are the examples of adaptive strategy. ACE-C [5], ACE-L [9], RCLB [10] are some techniques of deterministic cluster head selection whereas M-LEACH [11], EAMC [12], UCR [13], GCA [14], PEGASIS [15] works as a hybrid scheme.

We have used LEACH [1] algorithm for cluster head selection. Leach algorithm works on the concept of dynamic head in order to improve the energy efficiency. Energy is the important bottleneck point in sensor network. LEACH algorithm keeps changing the cluster heads after a time interval in order to maximize the lifetime of sensor nodes.These cluster heads sends the periodic information about their respective cluster to the base station. Then, Base station on the basis of past experiences takes the intelligent decision that whether the event has occurred or not. If the event occurs then the base station takes the prescribed action like generation of alarms [25].

As we know, in real scenario everything is error prone or fault prone. What if the cluster head got failed? There will be no data to send to the base station. Suppose at this mean time weather condition degrades then it can lead to disaster. Before going into further discussion let us know, what actually is the fault? The fault is an unexpected failure of the component which leads to the blockage of the system. So, the fundamental challenge in WSN is dealing with the failure of the cluster head. For this we have proposed an algorithm which will maintain the backup of every cluster head that is vital cluster-head at another secondary cluster head called backup cluster-head. Nearest node to the vital cluster head will be declared as the backup cluster head so that the communication overhead can be minimized. As the vital cluster head will receive any data, it will be backed up at the secondary location. If at any point backup cluster head doesn't get any update, it will wait for the threshold time. If in this mean time it doesn't get any update it will declare itself as vital cluster head and will send the pivotal message to each non cluster-head node in the cluster. Now onwards, all nodes will start treating the backup cluster-head as the vital cluster-head.

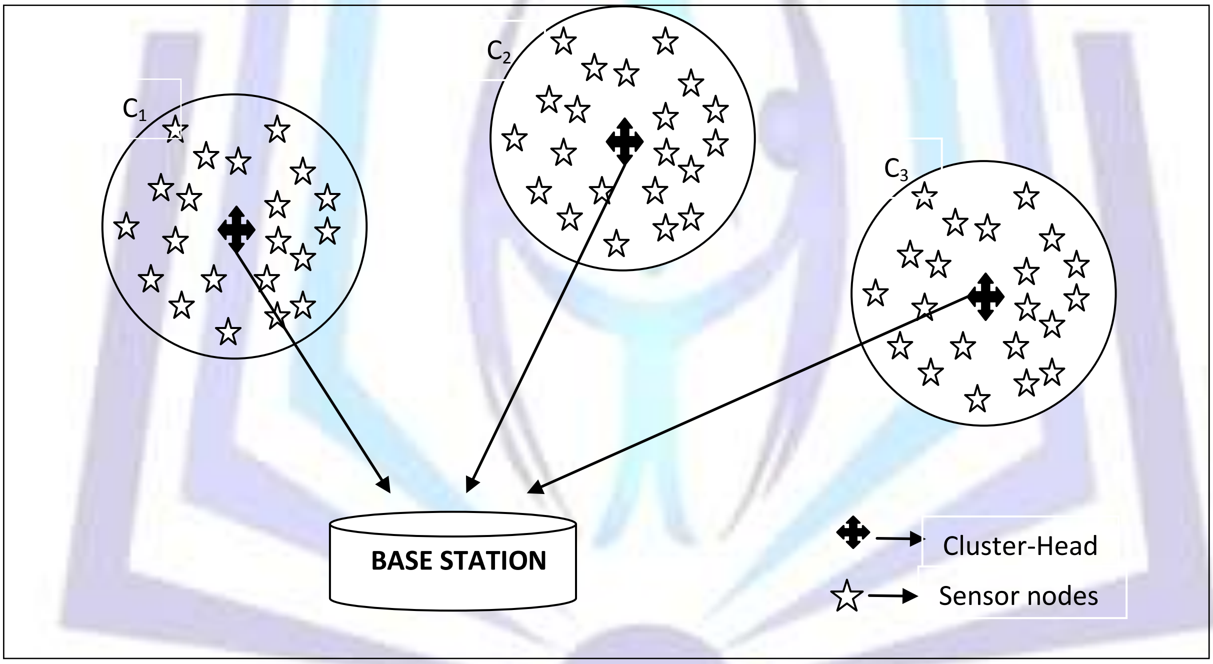


Figure 1 Working of clusters in WSN

RELATED WORK

As we all know, the fault never occurs after the announcement. So, we should be always ready to tackle the situation in an optimal and efficient manner. There can be various reasons for node failure like energy running down, failure of any hardware component of the node, security attacks, software errors, communication errors or environmental interferences [3]. Many researchers have been working for various algorithms to deal with fault tolerance in sensor network. There are basically two approaches for fault detection- Centralized approach and Distributed approach [2].

In a centralized approach, a geographically centralized node in the network behaves as a manager. This node controls and manages all the events and faults in the network. The central node detects the fault in the network by broadcasting a heartbeat message or query message to know the alive status of the various nodes in the network. Correspondingly, each node responds the heartbeat message to inform the manager that all nodes are working properly. If the manager doesn't get any update from any single node manager can detect the problematic node. This approach works very well and efficiently to detect faults in the network. But the manager node becomes the single point of failure. If it crashes, the whole network will be blocked. This approach also doesn't work well in large scale network as all the traffic will be concentrated on the manager. It will be so complex to manage such a large volume of the traffic and it will lead to the abrupt energy consumption of manger node.

Jessica Steddon et al. [16] proposed the algorithm by making available the knowledge of neighbor nodes of each node to the base station. Thus, base station constructs the network topology with this available information. After a fixed interval of time, base station keeps updating his network topology on the basis of the message received by various nodes. These messages contain the information about their neigbours along with their ids. As the base station knows the network topology, it can easily trace the faulty node by divide and conquer method based on adaptive route update messages.

R. Szewczyk et al. [17] presented a protocol called SPINs: Security Protocol for sensor networks. They proposed a routing protocol for detecting the faulty or the defective node in the network. This is done through the approach of routing discovery and update. In this scheme, every node sends the update message to the base station after every constant amount of time. A fault in the network is detected on the basis of this information only. This scheme is very expensive in terms of energy. This is because as we know, every node has limited energy and energy is the most critical factor for consideration in WSN.

Sapon Tanachaiwiwat et al. [18] introduced the concept of fault detection through probe sensors. In this scheme it is assumed that every node is aware of its geographical location. Initially sink node send the message to its trusted neighbour to the intended receiver. These neighbours forward the message to their trusted neighbours and so on. In this way the message is received at its destination through the trusted path. If the packet drops excessively or sink node notices any compromised data, it starts searching for the fault node through the route and takes corresponding action.

In distributed approach, every node has some level of decision power or say every node works as a manager to some extent. Each node sends fault information to the base station only when there is really a fault. Various algorithms are there for this approach like Node fault self detection and self- correction On its hardware physical malfunction, Failure detection by neighbour coordination, Utilization of WATCHDOG to detect misbehaving neighbor, use of group technology to distribute fault detection into the network, etc. This type of approach is very energy-efficient and works well for data centric sensor applications.

S Harte et al. [19] introduced a self detection to examine the fault in hardware as well as software of the nodes. Each node in the network contains an accelerometer which works as a sensing component to examine the direction and impact on the node. Software components like ADCC, timers are adopted with TinyOs as operating system to take sensor node's reading. While Farinaz koushanfar et al. [20] proposed a simpler method in which every node observes its sensors and compare the recorded value with past experienced data.

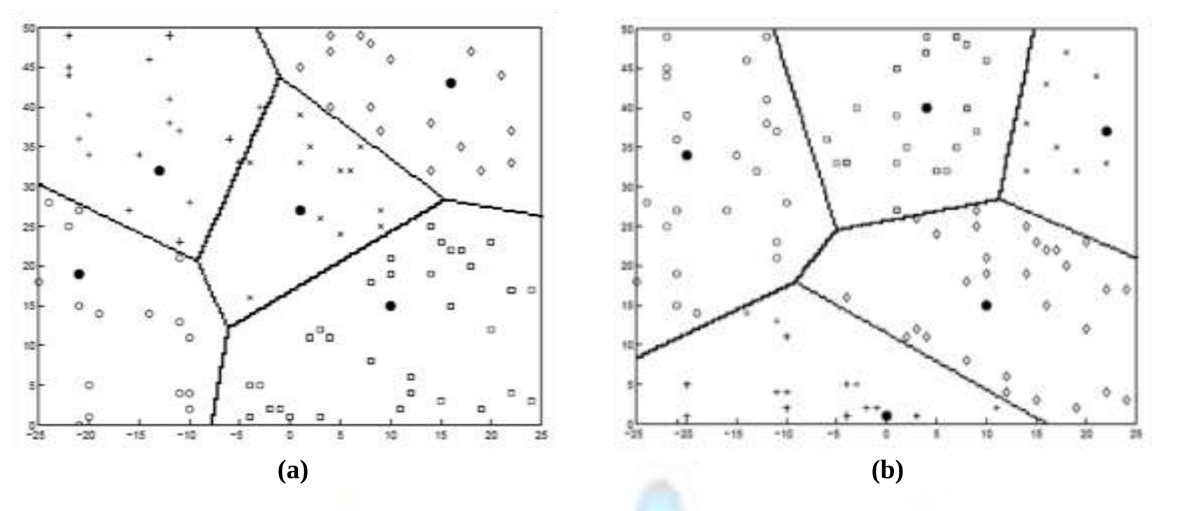
Sergio Marti et al. [20] introduced a fault tolerance scheme by WATCHDOG approach. In this approach sender maintains a watchdog buffer of sending messages. Each overheard message is matched with the buffered one. If a match occurs it is removed from the buffer and forgotten. If the message lies in the buffer for longer than a threshold time then it is considered that a fault has been occurred. Then the corresponding measures are taken in order to deal with the faulty node.

Deborah Estrin et al. [21] proposed the clustering technique for developing scalable and energy efficient application in WSN world. Ann T.Tai et al. [22] gives a node failure detection solution by making use of a hierarchy of cluster based communication to have scalability, completeness and accuracy at a single time. The whole network is divided into groups called clusters and each cluster has some fault management schemes. Faulty nodes are identified in each network by diffusing intracluster heartbeat message. Cluster makes the decision of any event on the basis of these collected heartbeat messages and the past experienced data. While, Ruiz et al.[23] proposed a manager- based model for event driven detection. In this scheme every cluster head is present in the network with an agent and manager is a single entity works externally on the network. Each agent sends the periodic state change information to the manager. Manager builds the network topology and energy model to detect the faulty nodes in the network. If any node responds with residual energy, then this node will be declared as failed.

There are numerous of approaches that work for fault detection in WSN but our approach is different from above stated approaches. This is because we proposed a hybrid scheme that works very well with the large scale network. As we all know, whenever we talk about the WSN first thing comes to our mind is Energy. That is, how to maximize the lifetime of the node with fulfilling other requirements like fault tolerance at the same time. So, our work gives the solution for all these questions. This algorithm works very efficiently with optimal communication overhead.

LOW ENERGY ADAPTIVE CLUSTERING HIERARCHY (LEACH)

LEACH [1] is a self organizing clustering technique based on the randomization of the cluster - head. This approach is very useful in maximizing the lifetime of the nodes in the network. In LEACH, cluster-head is changed after every time interval t. This is because the cluster - head consumes much amount of energy is dissipated message to all other nodes and receiving messages from other nodes in the cluster. The working of this algorithm is divided into number of rounds- advertisement phase, set up phase and steady-state phase [1].



(a) (b)

Figure 2 Working of LEACH Algorithm

In the above diagram, Figure 2 (a) gives a view of clustering at time t_1 and figure 2(b) represents the clusters at time t_2. Nodes of the same cluster are represented by the similar symbol and cluster-head is represented by ●.

1. Advertisement Phase

In this phase, nodes in the cluster show their interest to be the cluster-head of the cluster for the current round. Nodes make this decision based on the suggested percentage of the cluster head and the number of times a node has been a cluster-head till now. A random number between 0 and 1 is chosen by a node n. If this number is less than the threshold value T(n), it will be declared as a cluster-head. T (n) can be calculated by [1]-

$$T(n) = \begin{cases} \frac{P}{1-P*(r \bmod \frac{1}{P})} & If\ n \in G \\ 0 & otherwise \end{cases}$$

Where P is the desired percentage of cluster-heads, r in the round number and G is the set of nodes that have been the cluster-head four previous 1/P rounds.

When a node elects itself as a cluster-head, it sends the advertisement message to all other nodes in the cluster.

2. Set-up Phase

In this phase, every non cluster-head node decides the cluster-head to which they belong. This decision is based on the basis of the signal strength of the advertisement message. It is because the strength of the signal decreases with the increase in the distance of the cluster-head. Every node informs the cluster-head after making their decision of belongings. On the basis of the received requests of various nodes, cluster-head creates TDMA schedule and transmit it to the interested nodes [1].

3. Steady-state Phase

In this phase, all cluster-heads of different clusters in the network send the collected data of their clusters to the base station. LEACH assumes that every node has some data to send, so when they get their turn according to the TDMA slot they transmit data to the cluster-head of their cluster respectively. After collecting data from all non cluster-head nodes; cluster-head perform a signal compressing function to reduce the energy required for transmission. This compressed data is sent to the base station for further processing and next round begins [1].

PROPOSED WORK

Fault tolerance mechanism is used is WSN to provide reliability to the network. Energy is the most concerned topic of discussion whenever we talk about WSN. Every node in sensor network has limited energy. So, our objective is to implement a fault tolerant network with minimum energy drainage.

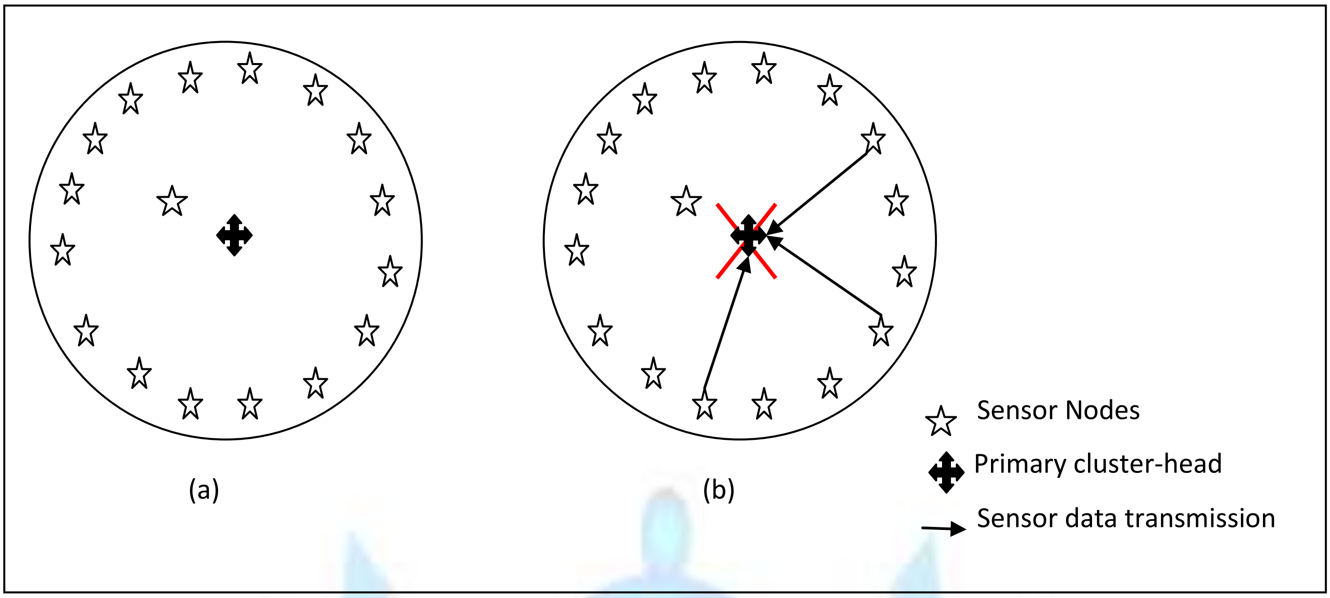


Figure 3 Non fault Tolerant sensor network

Consider the scenario, as shown in figure 3, cluster-head fails after receiving data from three sensor nodes of the cluster. This data is very crucial as disaster can happen any time. Suppose this is that mean time when the disaster is going to happen. No alternative is present to retrieve the lost data in the network. This situation can lead to the major catastrophe. So, our proposed algorithm aims to never happen these types of situations and to take preventive measure as we all know prevention is better than cure.

Functioning of our proposed algorithm is discussed in two phases – Vital phase and backup phase.

3.1 Algorithm VITAL FUNCTION ()

```
BEGIN
    1.  Apply LEACH to get vital cluster-head V and assign $V_{id}$=id [0];
    2.  For i = 1 to n-1
    3.          id[i] = i;
    4.          d[i] ← distance between i and $V_{id}$
    5.  For i =1 to n-1
    6.          Do min = i;
    7.          For j= i+1 to n-1
    8.            Do if  d [j] < d [min]
    9.                  then min = j;
    10.                         Swap (d [i], d[min]);
    11.                         Swap (id[i], id[min]);
    12. Backup cluster-head id ($B_{id}$) ← id[1];
    13. $V_{id}$ receives sensor recorded data
    14. Apply Backup function to update $B_{id}$ if new information is received.
END
```

3.2 Algorithm BACKUP FUNCTION (V_{id}, B_{id}, T)

```
BEGIN
    1.  W_t=0;
    2.  IF $B_{id}$ is receiving updates from $V_{id}$ && w_t <= T
    3.        $V_{id}$ is working properly.
    4.  Else
    5.  For i=2 to n-1
    6.        $B_{id}$ will send pivotal message Pv to i for notifying that $V_{id}$ has failed.
    7.        $V_{id}$ =$B_{id}$;
    8.  Repeat step 2 to 6 of Vital function
END
```

TABLE I. Symbols and Definition

SYMBOL	DEFINITION
V_{id}	Vital cluster-head id
B_{id}	Backup cluster-head id
n	Number of nodes in a cluster
d[i]	Array containing the distance between the vital cluster - head and each non cluster-head node in the cluster
id[i]	Array containing the ids of each node in the cluster
W_t	Waiting time of backup cluster-head for getting updates
T	Threshold time

Vital Phase

Vital phase deals with the primary operations of the distributed WSN i.e. selection of cluster head and maintain records of different sensors in the cluster. In the very first step of our proposed approach, we used LEACH for selecting the vital cluster-head. On the basis of some probability function and energy level of sensor nodes as discussed above, LEACH will return the vital cluster-head id. Now, the distance between the vital cluster-head and each non vital cluster-head nodes of the cluster is calculated and stored in the array d[i]. In the next step sorting of d[i] is done in order to have the min distance. With the swap of the distance values in the array, ids are also swapped so that we can have the node's id with the distances. We are interested in finding least distant node as if we declare it as backup node, communication delay will be optimal and energy requirements as well will be optimal. One another idea behind this selection is minimum delay in updating the data at backup node. So, in the third step of the algorithm we will have the nearest node to the vital cluster-head and it will work as backup cluster-head. As the vital cluster-head will get any update from any sensor node of the cluster, it will call Backup function of the algorithm.

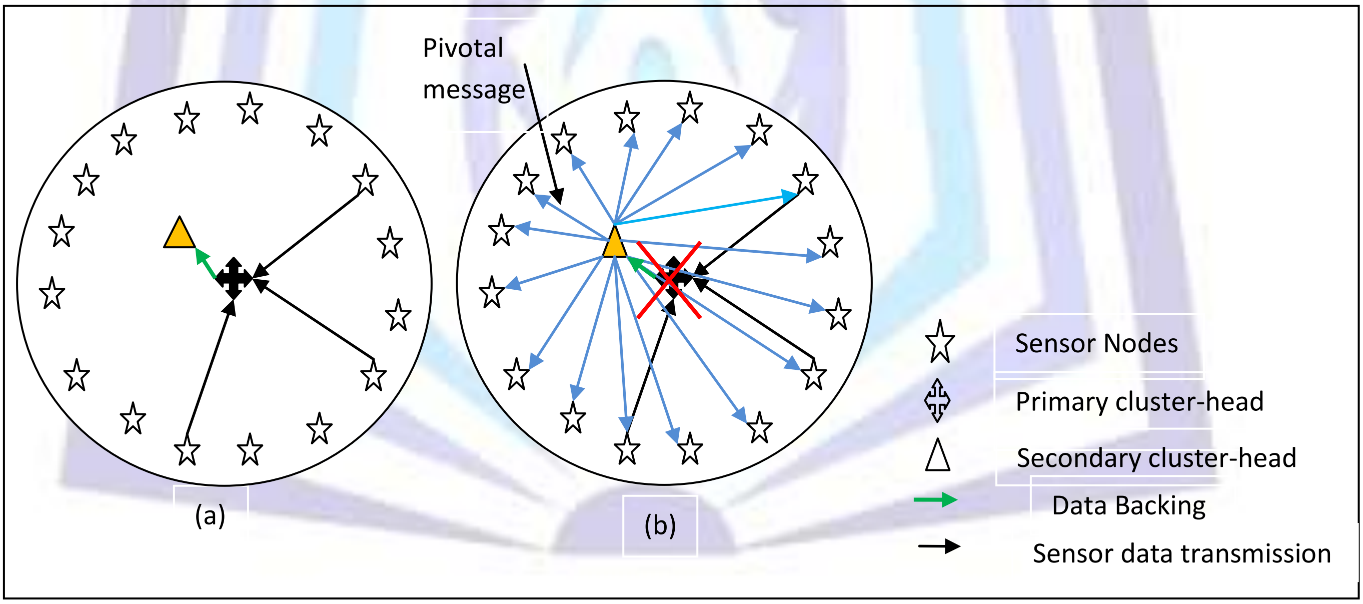


Figure 4 Working of Vital and Backup cluster-head

Backup Phase

Backup phase deals with backing up the data at secondary nodes to provide reliability in the odd times. As the vital cluster-head receive any data from any cluster node, backup function will be called. If the backup cluster is getting updated from vital cluster-head regularly i.e if the waiting time of backup cluster-head is less than the threshold time T, means that vital cluster-head is working properly. T is predefined threshold time parameter. If the waiting time of backup cluster-head exceeds T, then it will consider the vital cluster-head to be failed. In correspondence of that backup cluster-head will send the pivotal message Pv to all non cluster-head nodes of the network informing them that the vital cluster-head has failed. Thus, Backup cluster-head will declare him as the virtual cluster-head and again it will search for the nearest node to declare that as his backup cluster-head.

Example

The problem stated in figure 3 can be solved by our proposed algorithm in the simple and efficient manner with optimal energy usage. As shown in figure 4, every update is backing up at backup cluster-head represented by a yellow triangle. If the above stated problem occurs, nothing to worry as all data is preserved at backup node. The backup cluster - head will pass the information regarding the failure of vital cluster-head to all other nodes through pivotal messages, which is represented with blue arrows in the figure.

RESULTS AND DISCUSSIONS

We have implemented our proposed algorithm with defined network parameters, and compared our results with existing algorithms DSC [26] and FT-DSC [24]. We have measured our performance on the basis of communication overhead, network lifetime and energy consumption. During the transmission and receiving of data, sender and receiver consumes energy. First we simulate mathematically followed by simulation set-up along with results and discussion.

A. Mathematical Analysis

Let us assume sensor nodes of the cluster are generating data of α_{data} bytes and consumes ε_{data} amount of energy during transfer that covers ∂ distance from sender to receiver (from normal node to Cluster-head node). Hence total energy consumption in transfer of data is

$$TotalEnergy(\alpha_{data}, \partial) = \alpha_{data} \in_{data} + \alpha_{data}\, \partial \in_{data} \quad (1)$$

Above equation computes the total energy consumption in sending data (energy consumption in generation of data along with energy consumption in distance coverage) at sender side.

As we have mentioned that in our proposed work vital node forwards received data towards Back-up node. Hence Energy consumption of vital node in forwarding data is very minute (due to small distance between them)

$$Energy_{VN}(\alpha_{bkdata}, \partial_{vnBk}) = \alpha_{bkdata} \in_{bkdata} + \alpha_{bkdata}\, \partial_{vnbk} \in_{bkdata} \quad (2)$$

Where, α_{bkdata} is back up data

∂_{vnbk} is distance between vital and Backup node

$\in_{bkdata}$ is energy consumption is sending back-up data.

The energy consumption in transmission ($Energy_{TX}$) and receiving ($Energy_{RX}$) of data set on both sides (in between cluster-head and non cluster-head node). Probability of sending data from non-cluster head node to vital node and further from vital to Back-up node is $\rho_{\alpha data}$ and $\rho_{\alpha bkdata}$

$$Energy_{TX} = \rho_{\alpha dat\ a} \times TotalEnergy(\alpha_{data}, \partial) + \rho_{\alpha bkdata} \times Energy_{VN}(\alpha_{bkdata}, \partial_{vnBk}) \quad (3)$$

And

$$Energy_{RX} = \rho_{\alpha data} \times \alpha_{data} \in_{data} + \rho_{\alpha bk\ data} \times \alpha_{bkdata} \in_{bkdata} \quad (4)$$

We have simulated our work on cluster level only, hence we are not computing any kind of computation related to Base station.

Let us compute the actual size of data including data backing message

$$\alpha_{data} = \beta \times \alpha_{bkdata} \quad (5)$$

Moreover, probability of sending data including Back-up data probability

$$\rho_{data} = \aleph \times \rho_{\alpha bkdata} \quad (6)$$

By putting the values of equation (5) and (6) , equation (3) becomes

$$Energy_{TX} = \rho_{data} \times TotalEnergy(\alpha_{data}, \partial) + \frac{\rho_{data}}{\aleph} \times \frac{1}{\beta} \times TotalEnergy(\alpha_{data}, \partial)$$

$$= TotalEnergy(\alpha_{data}, \partial)(\rho_{data} + \frac{\rho_{data}}{\aleph\beta}) \quad (7)$$

From above equation we calculate the transmission energy consumption of our proposed algorithm

$$TotalEnergy(\alpha_{data}, \partial) = \frac{Energy_{TX}}{(\rho_{data} + \frac{\rho_{data}}{\aleph\beta})} = D \times Energy_{TX} \quad (8)$$

From equation (8), we can also compute the transmission energy consumption of DSC and FT-DSC algorithms.

Where,

$$D = (\rho_{data} + \frac{\rho_{data}}{\aleph\beta})^{-1} \quad (9)$$

The values of $\aleph$ and β is computed from equation (9).

Similar to energy consumption of transmission method we can compute the values of $Energy_{RX}$ of proposed, DSC and FT-DSC algorithms.

From above analysis, we can conclude that the proposed algorithm is more energy efficient than existing DSC and FT-DSC algorithm. In our proposed work whenever a vital node receives any update from non cluster-head; a back-up message is communicated towards Back-up node. Here the sender (Vital node) and receiver (Back-up node) is fixed but in FT-DSC protocol, a non-Cluster head node transmits a special packet to the cluster head to notify that it is still alive. In this way Fault tolarence is provided in our proposed algorithm. Main advantage of our algorithm is after detection of faulty cluster head we are having an another option which can be a cluster head.

B. Simulation Set-up

We simulate proposed work along with DSC and FT-DSC algorithms on TinyOs and compute performance on the basis of following defined values. With subsequent parameters we have computed the energy consumption, communication overhead and network lifetime.

TABLE II. Simulation Parameters

PARAMETERS	***VALUE***
Network Size	100×100
Number of Nodes	Maximum 250
Number of Clusters	Maximum 20
Base Station Position	90 X 170
Data Packet Size	512 Bytes
Special Packet Size	16 Bytes
Energy Consumption for Sending Data Packets	40 joule
Energy Consumption in free space/air	0.01 pJoule
Initial Node Energy	2 J
Cluster Head Probability	3%
Energy Consumption for Sending Data Packets	20 pJoule

Users are required to input number of nodes whose maximum limit is 250, number of clusters to be formed which can be 20 at a maximum and number of rounds. We have rum this algorithm for different valued parameters at each run. This algorithm also runs at the complexity of O(n^2) which is equal to the Ft-DSC with better performance and efficiency.

C. Simulation Results and Discussions

Based on the above-mentioned parameters and their standards in the Table II, we perform a simulation of the DSC , FT-DSC and our proposed work and quantify their performance in terms of energy utilization, communication overhead and network lifetime . We run the simulation over 30 times and taken the average values. Energy consumption is considered in selecting clusterheads, sending updates, backing data or sending pivotal messages.Figure 5 shows that our proposed algorithm is taking much less energy as compared to the DSC and FT-DSC algorithms.

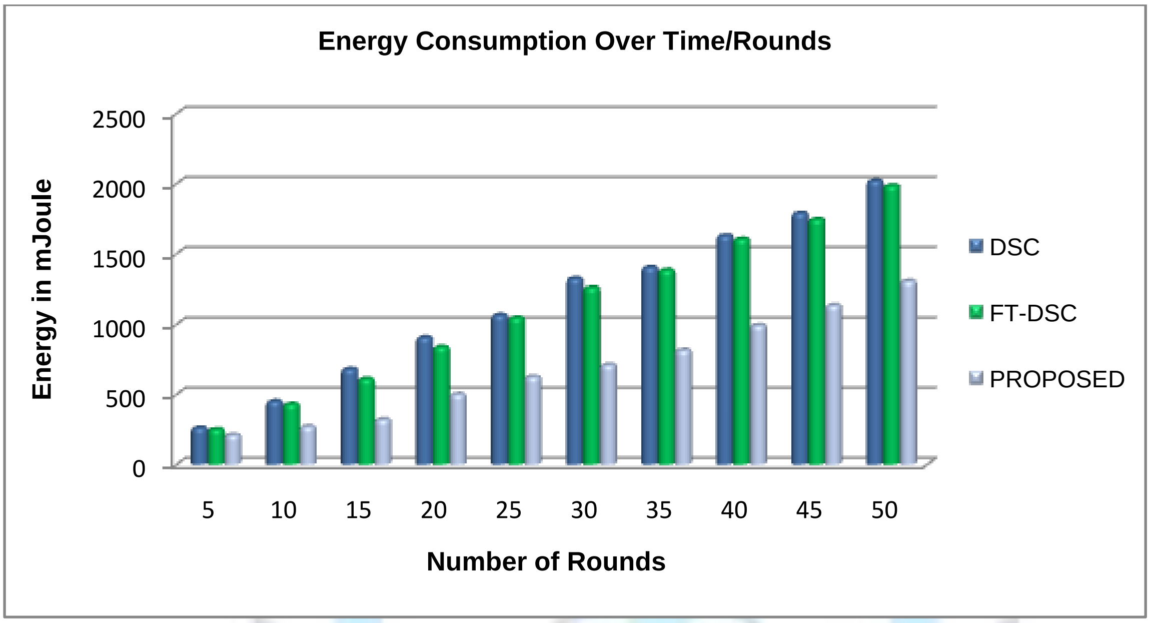


Figure 5 Energy Consumption over Rounds

On the other side, Communication overhead is a vast problem to study in WSN. So, Figure 6 is representing the efficient performance of our proposed algorithm in terms of communication overhead. Existing DSC protocol broadcast only data packets that considers communication overhead only in terms of data packet transmission. In the FT-DSC protocol, non cluster head node sends a special packet to cluster head when it has no data to transmit. This special data towards cluster head is for reconfirmation of cluster heads availability. But in proposed algorithm, after reception of data from non cluster-head, Vital node forwards those data-packets toward Back-up cluster-head. And here the distance between vital and back-up cluster -head is very small. In place of any special message, Back-up cluster head wait for a limited amount of time (threshold). After that threshold value back-up node declares itself as a vital cluster head.

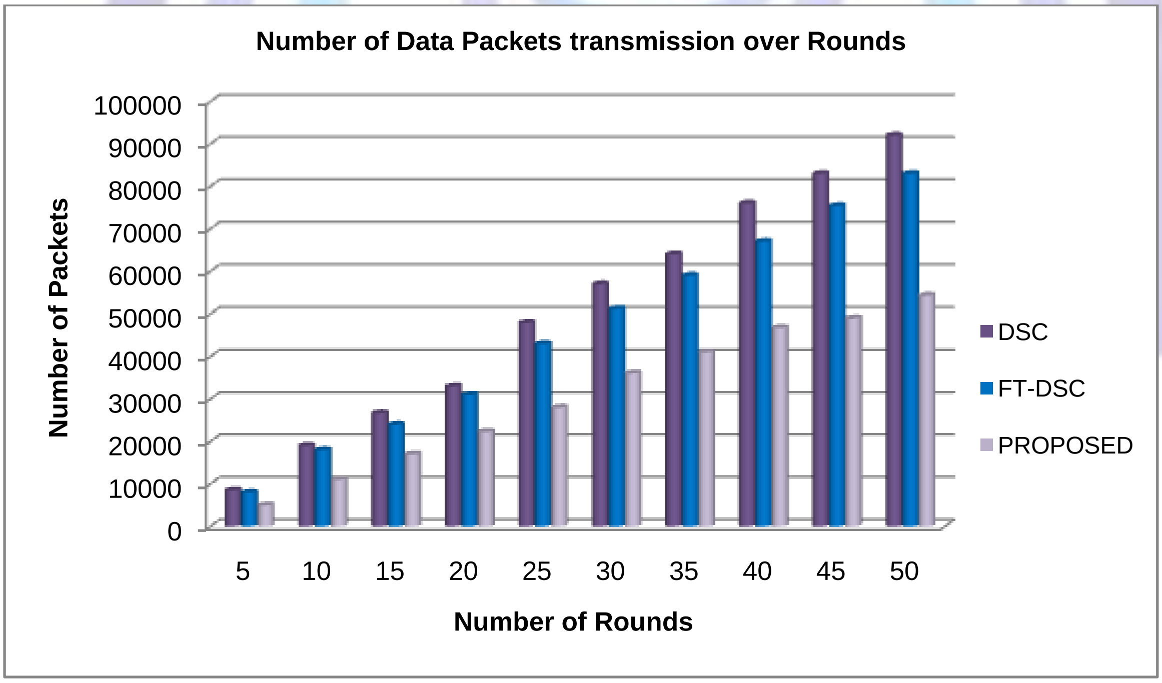


Figure 6 Communication Overhead over Rounds

Furthermore, in our proposed work energy dissipation is lesser than the existing DSC and FT-DSC algorithm. Due to the threshold limit of Back-up cluster head energy in sending or receiving data is less. Therefore, the network lifetime of our proposed work is larger than the existing algorithms.

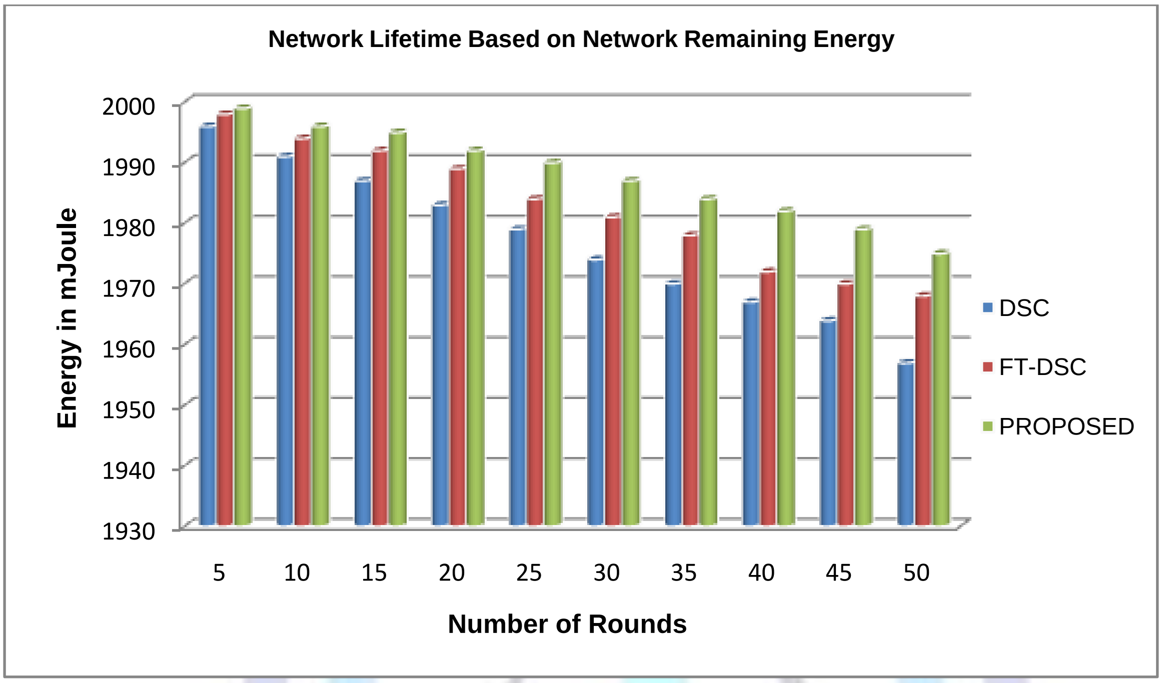


Figure 7 Network Lifetime Over Rounds

Finally, from above-stated comparisons we can say that proposed algorithm works better than the DSC and FT-DSC algorithms.

CONCLUSION

In this paper, we designed a fault tolerant algorithm which works for detecting events in the distributed wireless sensor network. It provides reliability if any cluster-head faces some critical condition or failure. No more devastation will be faced now due to cluster-head failure in the network. We have considered energy as the main issue, and tried to use it in an optimal manner. The algorithm performs very well as compared to the DSC and FT-DSC with maintaining the same complexity level of O (n^2). We have shown our results for less communication overhead, maximized network lifetime and less energy depletion of nodes over the rounds. In future, we can further compare it with other existing algorithms. We can also work upon it for improving reliability in terms of other parameters.

REFERENCES

[1] W..Heinzelman et al., 2000 Energy-Efficient Communication Protocol for Wireless Microsensor Networks, 33rd Hawaii international Conference on System Sciences, vol. 2, pp.1-10.

[2] M.Yu, H.Mokhtar, and M.Merabti, 2007 A Survey of Fault Management in Wireless Sensor Networks, Journal of Network and Systems Management, Volume 15, Issue 2 , pp 171-190.

[3] Che-Aron, Z., Al-Khateeb, W. F. M., Anwar, F., 2010 An Enhancement of Fault-Tolerant Routing Protocol for Wireless Sensor Network, International Conference on Computer and Communication Engineering (ICCCE), pp. 1-3.

[4] P. Tillaport et al., 2005 An Approach to Hybrid Clustering and Routing in Wireless Sensor Networks, IEEE Aerospace, pp.1-8.

[5] W. Heinzelmanet al., 2002 An Application-Specific ProtocolArchitecture for Wireless Microsensor Networks, IEEE Transactions on Wireless Communications, vol. 1, no.4, pp. 660-670.

[6] W. Heinzelman, 2000 Application-Specific Protocol Architectures for Wireless Networks, Ph. D thesis, Massachusetts Institute of Technology,.

[7] Jannatul Ferdous et al., 2010 A Comprehensive Analysis of CBCDACP in Wireless Sensor Networks, Journal of Communications, vol. 5, no. 8.

[8] Vivek Katiyar and Narottam Chand, 2011 Improvement in LEACH Protocol for Large-scale Wireless Sensor Networks, In proceedings of ICETECT, pp. 260-264.

[9] C. Liu, C. Lee and L.Chun Wang, 2007 Distributed clustering algorithms for data gathering in wireless mobile sensor networks, Elsevier Science Journal of Parallel Distributed Computing, vol.67, pp.1187 -1200.

[10] N.Kim, J.Heo, H.Kim and W. Kwon, 2008 Reconfiguration of clusterheads for load balancing in wireless sensor networks, Elsevier Science Journal of Computer Communications, Vol. 31, pp. 153-159.

[11] L.Ying and Y. Haibin, 2005 Energy Adaptive Clusterhead Selection for Wireless Sensor Networks, 6th international Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), pp. 634-638.

[12] Xinfang Yan et al., 2008 An Energy-Aware Multilevel Clustering Algorithm for Wireless Sensor Networks, international Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp. 387-392.

[13] Guihai Chen et al., 2007 An unequal cluster-based routing protocol in wireless sensor networks, Springer Science + Business Media, LLC.

[14] Sang Hyuk Lee et al., 2011 Gradual Cluster Head Election for High Network Connectivity in Large-Scale Sensor Networks, ICACT,vol. 13, pp.168-172.

[15] Lindsey et al., 2002 PEGASIS: Power-Efficient gathering in sensor information systems, IEEE conference on Aerospace, vol. 3, pp.1125-1130.

[16] Jessica Staddon, Dirk Balfanz, Glenn Durfee, 2002 Efficient Tracing of Failed Nodes in Sensor Networks. Ist ACM International Workshop on Wireless Sensor Networks and Applications, USA.

[17] A.Perrig et al., 2001 SPINS: Security protocols for sensor networks, ACM MobiCom'01, Italy.

[18] Sapon Tanachaiwiwat et al., 2003 Secure Locations: routing on trust and isolating compromised sensors in location-aware sensor networks, Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03), pp. 324-325.

[19] S Harte, A Rahman, K M Razeeb, 2005 Fault Tolerance In Sensor Networks using Self-Diagnosing Sensor Nodes, IEE International Workshop on Intelligent Environments.

[20] Sergio Marti, T.J.Giuli, Kevin Lai, Mary Baker, 2000 Mitigating Routing Misbehavior in Mobile Ad Hoc Networks, 6th International Conference on Mobile Computing and Networking, USA.

[21] Deborah Estrin, Ramesh Govindan, John Heidemann, Satish Kumar, 1999 Next Century Challenges: Scalable Coordination in Sensor Networks, ACM/IEEE International Conference on Mobile Computing and networking.

[22] Ann T. Tai, Kam S. Tso, William H. Sanders, 2004 Cluster-Based FailureDetection Service for Large-Scale Ad Hoc Wireless Network Applications, International Conference on Dependable Systems and Networks DSN ' 04, pp. 805-814.

[23] Linnyer Beatrys Ruiz et al., 2004 Fault management in event-driven wireless sensor networks, International Workshop on Modeling Analysis and Simulation of Wireless andMobile Systems, Italy.

[24] Karim, L et al., 2009 A Fault Tolerant Dynamic Clustering Protocol of Wireless Sensor Networks, IEEE conference of Global Telecommunications (GLOBECOM).

[25] Yashwant Singh, Sumah Saha, Urvashi Chugh, Chhavi Gupta, 2013 Distributed Event Detection in WSN for Forest Fires, Internation Conference of Computer Modelling and Simulation (UKSIM'13).

[26] F. Bajaber and I. Awan, 2008 Dynamic/Static Clustering Protocol for Wireless Sensor Network, Second UKSIM European Symposium on Computer Modeling and SimulationSecond, pp. 524-529.

Author' biography with Photo

Chhavi Gupta holds B.Tech in computer science from Uttar Pradesh Technical University, India. Currently pursuing M.Tech in computer science from Jaypee University of Information Technology, Waknaghat, Solan(H.P, India). She is working in the Research area of Event detection in Wireless Sensor Network.

Rashmi Sharma holds M.Sc, M.Tech in computer science from Banasthali University, Rajasthan (India). Currently pursuing Ph.D in the field of Real Time Distributed system in computer science from Jaypee University of Information Technology, Waknaghat, Solan(H.P, India).

Neha Agarwal has done her B.Tech in Computer Science. Currently pursuing M.Tech. from Jaypee University of Information Technology, Waknaghat, Solan (H.P, India).

Dr. Yashwant Singh is Assistant Professor in the Department of Computer Science and Engineering and Information Communication Technology, Jaypee University of Information Technology, Waknaghat, (H.P.) India. He has completed his Ph.D. in Computer Science from Himachal Pradesh University Shimla, year 2011. He received his M. Engg from Punjab Engineering College Chandigarh (Now PEC University) in year 2006. He has done his B. Engg. From Sant Longowal Institute of Engineering and Technology, Sangrur, Punjab in year 2004. He has done three and half year Diploma in Computer Engineering from Government Polytechnic Kangra in Year 2001. He is Life Member of ISTE.