International Journal of Computer Trends and Technology
ISSN: 2231-2803 / https://doi.org/10.14445/22312803/IJCTT-V67I3P118
Volume 67 Issue 3, 92-98, March 2019

Original Article

Event-detection Clustering Algorithm for Wireless Sensor Networks

Rajkumar [1], Dr. H. G. Chandrakanth[2]

[1,2]*Sambhram Institute of Technology, Bangalore, VTU Belagavi Karnataka, India*

Abstract - *Wireless sensor networks (WSN) consist of the extensive random deployment of energy-constrained nodes. The sensor has a different capability to sense and send sensed data to Base Station (BS). Sensing and transmitting data to the base station requires a huge amount of energy. In WSNs, saving energy & Extend the lifetime of the network are immense challenges. Several routing protocols have been projected to realize energy efficiency in the heterogeneous situation. In this paper, we propose a clustering algorithm for these kinds of locations: prioritizing the extent of lively sensor nodes by observing the important areas in a network situation. The results of our extensive study for this type of situation show a considerable enhancement in the amount of sensed actions on average by 104.29%, 73.15%, and 50.98%, compared with those of Low Energy Adaptive Clustering Hierarchy (LEACH), Hybrid Energy-Efficient Distributed (HEED), and Effective Distance Cluster Head (EDCH) algorithms, in that categorize.*

Keywords - *WSN, Energy efficiency, Clustering Algorithm, Locational monitoring, Active and inactive sensors nodes*

I. INTRODUCTION

In recent years, the development of WSN has increased at a speedy pace. A classic WSN involves a huge number of small sensor nodes. They monitor or sense physical characteristics such as temperature or pressure, process data, and communicate this data to the base station or sink. Usually, the network is fixed, and battery power is. WSN nodes are adept at bringing themselves together into the supportive network. These nodes are installed in huge numbers to monitor or track events, generally in areas that lack connectivity & wig internet infrastructure. Once set up, these nodes can work without human intervention. WSN can be set up in a structured or unstructured, or haphazard way. The kind of positioning depends on the WSN application and deployment location. Random deployment is the only probability in a strict atmosphere and tough-to-reach spaces, but caution must be taken to dodge uncovered areas. In structured deployment, nodes are positioned at the exact place. Therefore, fewer nodes are essential, and their care, organization, and deployment of nodes become easy [1]. Energy efficiency is essential for this sensor network's effectiveness, as it is difficult to renew sensor node batteries or replace them. Many studies have recommended energy-efficient routing algorithms for WSNs [3], [5]. The key problem of energy consumption for WSNs is their communications [6].

Clustering is one of the greatest methods [2], [7] for dropping energy consumption. In a cluster Wireless Sensor Network, sensor nodes are gathered into a certain number of clusters, each comprising a cluster head (CH) and some non-cluster head nodes (non-CHs). CH gathers data from all the cluster nodes and then heads to other CHs or base stations (BS). In contrast, non-CHs nodes are answerable for detecting location and transferring information to the corresponding CH [4].

The equally distributed CH (cluster head) set can balance the energy consumption among sensor nodes and, lastly, lengthen the network period. In the network with non-uniform sensor node circulation, the mechanisms used to balance the energy consumption and prolong the network period are not in effect. The consistently distributed CH (cluster heads) permit the clusters to have uniform cluster sizes so that the energy utilization surrounded by cluster members or nodes can be well-adjusted. However, the imbalanced energy utilization still occurs among CH (cluster heads) due to the non-uniform sensor node circulation.

In a networking atmosphere, the prominence of distributed sensors might not be equivalent. While certain sensors might be lively due to their nearness to a dangerous area, others might not. Thus the goal of a network organization might comprise prolonging the lifetime of active, rather than inactive, sensors. This paper offers Event-detection EDCH (E-EDCH) for such atmospheres that use the Effective Distance Cluster Head (EDCH) algorithm and can help raise active sensors' lifespans. The remains of this paper are prepared as follows. In Section II, related work is conferred. The particulars of the new location are set out in Section III. E-EDCH algorithm and its difficulty are contained in Section IV. A calculation of the E-EDCH algorithm is put out in Section V, whereas our conclusions can be found in Section VI.

II. RELATED WORKS

Low Energy Adaptive Clustering Hierarchy (LEACH) rules are TDMA-based MAC protocols. The main goal of this protocol is to increase the lifetime of wireless sensor networks by lowering the energy consumption necessary to construct and keep Cluster Heads. The LEACH protocol process involves several rounds with two phases in each [8] [9]: Set-up Phase and the Steady Phase.

In the Setup phase, the main objective is to make a cluster and select the cluster head for each cluster by selecting the sensor node with extreme energy. The steady Phase, which is relatively lengthier in time than the set-up, deals mostly with collecting data at the cluster heads and transmitting collected data to the Base station.

HEED [10] is a protocol that periodically chooses cluster heads, rendering a hybrid of the node remaining energy and a second parameter over constant time repetitions. It practices the primary parameter, i.e., residual energy, to choose the first set of cluster heads. Unlike preceding protocols which need knowledge of the network density or homogeneousness of node dispersion in the field, HEED does not make any assumptions about the network, such as thickness and size. Each node runs HEED separately. At the end of the development, each node either grows into a cluster head or a child of a cluster head. Below are some vital features of HEED:

HEED is hybrid, and clustering is based on two constraints: residual energy of a node is the first constraint in selecting a cluster head and the proximity or node degree.

HEED is scattered: each node runs the heed algorithm independently.

HEED is energy-efficient: the algorithm selects cluster heads that are rich in residual energy, and re-clustering results in allocating energy consumption

EDCH [20] is an additional clustering algorithm that attempts to distribute CHs through the network consistently. EDCH, parallel to HEED, benefits the uniform circulation of CHs in the network. The EDCH algorithm meaningfully optimizes network energy consumption; and subsequently increases lifecycle by up to 251.05% and 150.40% related to LEACH and HEED, respectively [20]. Furthermore, EDCH outperforms HEED in time and message complexities; and has a logical model which offers a believable mathematical basis for this algorithm [11].

None of these algorithms deliberated the diverse gravities of sensor roles in the network to the greatest of our knowledge. Thus, this specific application of WSNs lacks a completely distributed algorithm.

III. LOCATION

We aim to develop a fully distributed clustering algorithm to advance system stability and energy saving in a moveable location. The system stability will be enriched if the selection criteria of CH are well defined. For example, network stability could be advanced if the higher residual energy sensor is privileged. The energy could be saved if the sensor nearest to BS prefers green. Thus the lifetime of a network is extended. Our clustering algorithm must achieve these points without criticizing the network connectivity that certifies the area coverage.

In this specific application of WSNs, the intention of our network design might purely lie in extending the lifespan of active sensor nodes. Doing the same with inactive sensors is not important. To the highest of our knowledge, no proposed algorithm has been for the above scenario. In earlier algorithms, such as [12], [13], [14], and [15], active nodes might use all their power before non-active ones because of their high activity rates. Time being, in a number of others, such as [16], [17], [18], [19], the lifespan of both active and inactive sensors might be virtually equal: because they consider sensors' residual energy and ignore their importance. This study introduces a new algorithm to extend the lifespan of active sensors and thus increase the observing time of critical regions in a network area.

In our proposed algorithm, extending the lifetime of the inactive sensors is not a priority; they could even be surrendered to enlarge active ones.

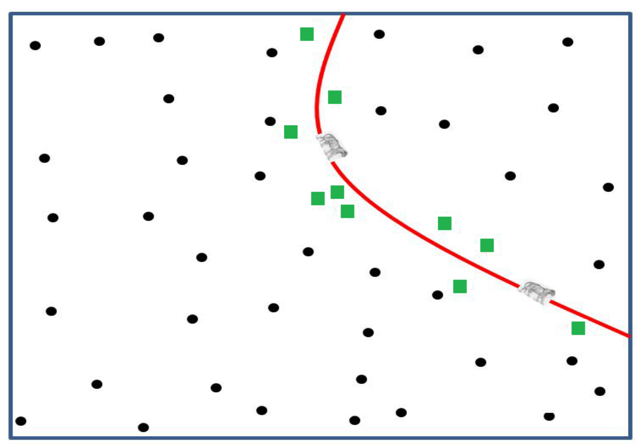

Fig. 1 The example of a network region with active and inactive sensor nodes. Active nodes are shown by green squares, while black circles show inactive sensors.

Fig 1 shows an example of this specific network region. The green line demonstrates a popular path repeatedly used by the animals in this location. The green squares are active sensors due to their close affiliation to this critical region; the black circles are inactive sensors due to their distance from the tiger animal paths.

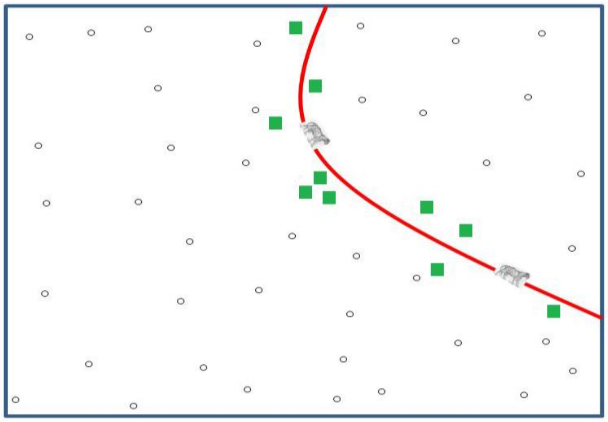

Fig. 2 The example of a network region including active and inactive sensor nodes. All inactive nodes have died while all active sensor nodes are alive.

In the above-styled scenario, the ideal pattern of dying may be related to those in Fig. 2. The inactive sensors in Fig. 2 have died, whereas all active sensor nodes are alive. Indeed, the inactive sensors have been sacrificed to extend the lifespan of active ones.

IV. THE E-EDCH ALGORITHM

The location discussed in Section III requires an algorithm that proficiently extends the lifespan of active sensors, and the significance of our location observation is dependent upon its active nodes. Thus this section proposes.

E-EDCHalgorithm: which uses an EDCH clustering algorithm and prioritizes extending active sensor nodes that observe important parts of the location

The first step uses EDCH: a novel clustering algorithm for WSNs [20]. We have chosen EDCH because of its uniqueness and efficiency [5].

In clustering algorithms, the part of CHs is very important. They gather all sensed events from other sensors, perhaps combine them, and send the data to the BS base station. Thus their duty might be heavier, and their batteries drain faster than others. Using the E-EDCH algorithm, we aim to give inactive sensor nodes a substantial duty task because they are not a priority. This would free active sensor nodes from CH duties, meaning their batteries could be preserved and used only for sensing.

In the first step, the EDCH algorithm is implemented. Thus, all sensors are assembled into a number of clusters, with each cluster having a CH. In the next step, all clusters analyze their inner procedures individually from the others. Inferring does not require synchronizing clusters and can therefore save advanced energy. After a small-time, the CH can find its most inactive CM because it receives every data from sensors in its cluster. Therefore, the most inactive sensor is nominated as the CH in the second step. CHs are not revolved occasionally through all clusters. Thus each CH can remain in its heading role all over its lifetime. Once this CH (Cluster Head) is about to run out of control, it selects the most inactive CM as the new CH (Cluster Head). It can also send all required information to the newly nominated CH (Cluster Head).

The novel CH (Cluster Head) presents itself to all its CMs (Cluster Members), informing them of its novel role. Other CMs (Cluster Members) do not need to register with the novel CH(Cluster Head). Because it has received all information concerning them from the prior CH, this process can be repetitive for every sensor node in all clusters unless everyone dies. The most active sensor nodes are nominated final, enabling the network to advantage of them as much as likely.

The pseudo-code of the E-EDCH algorithm is obtained in Algorithm 1. *N sensor* nodes, shown by *n*[0], *n*[1], ..., *n*[*N* − 1] are cluster green; the part of each node is clear green by its color attribute. Current CHs are shown in green color, dead CHs in white, and others are shown in black.

In the first line of Algorithm 1, all sensors gather green in smaller clusters using the EDCH algorithm [20]. In Lines 2 to 22, if the remaining energy of the CHs is equal to or less than their 0.01 of preliminary power; or merely the first CH (Cluster Heads) has been nominated in all clusters, it catches its less active sensor by using *Find Less Active Member* process, and transfers the CH role to this less active node. If the residual energy persists, its color attribute changes to white, showing dead sensor nodes.

The novel CH undertakes its novel role and presents itself to other sensors in its cluster. In Line 20, the Steady-Phase of each footstep is running.

A. *Complexity of E-EDCH*

In this subsection, we consider the complexity of the E-EDCH algorithm.

Algorithm 1 E-EDCH Clustering Algorithm
Step1: EDCH();
Step2: while (*SetCH.Count* > 0) do
Step3: for ($i \in SetCH$) do

Step4: if ($n[i].Energy < 0.01 \times JN$) or (*FirstRound* == true) then
Step 5: *SetCH.erase*(i);.
Step 6: if (*FirstRound* == false) then
Step 7: $n[i].colour \leftarrow White$;
Step 8: end if
Step 9: $t \leftarrow FindLessActiveMember(i)$;
Step 10: if (t _= −1) then
Step 11: *SetCH.insert*(t);
Step 12: $n[t].colour \leftarrow Green$;
Step 13: $t \leftarrow TransferData(i)$;
Step 14: $n[t].advertise - message()$;
Step 15: for ($j \in Cluster(t)$) do
Step 16: $n[j].receices - message()$;
Step 17: end for
Step 18: end if
Step 19: end if
Step 20: Steady-Phase();
Step 21: end for
Step 22: end while
Step 23: STOP

Lemma 1. *E-EDCH has $O(N2)$ time complexity in its lifetime.*
Proof. The first line of Algorithm 1 illustrates the EDCH algorithm, which has a time complexity of O(q) per sensor node [11]. C1×q×N illustrate*s this* for the network, where $C1$ is a predetermined number, q is the number of CHs (Cluster Heads), and N is the number of sensor nodes.

As of Lines 2 to the stop, we have q clusters; the number of sensor nodes in all clusters is N/q average. Finding most inactive sensor nodes in all clusters, selecting them, and transferring the information from the earlier CH(Cluster Head) to the novel one can be completed in a fixed number of commands. Then in every cluster, the sum executing commands can be shown by $C2\times(N/q-1)$ and for every of the network is $q\times C2\times(N/q-1) \leq C2 \times N$ per CH (Cluster Head).
As we have $N/q-1$ CH (Cluster Head) transfer method averagely, the entire of executable commands is less than $C1 \times q \times N + (N/q-1)\times C2 \times N/q \leq C1 \times q \times N + C2 \times N2/q$. Therefore, the entire time complexity of E-EDCH is $O(N2)$ for the network life span.

Lemma 2. *E-EDCH has $O(N2)$ message complexity.*
Proof. The primary column of Algorithm 1, running the EDCH algorithm, has O(N) message complexity. This is illustrated by $C1\times N$, where N is the number of sensors [11].
From Lines 2 to the last part, there are ($N/q-1$) CH (Cluster Head) transfer events: with an average transfer method in every cluster. For all CH (Cluster Head) transfers, one message is used to transfer every required information from the previous CH (Cluster Head) to the novel one. Moreover, $N/q-1$ messages are used for the novel CHs (Cluster Heads)' advertisements. Consequently, less than N^2/q^2 messages go by through every network's set-up phase in its life span. Thus the total passing messages for E-EDCH is less than $C1 \times N + C2 \times N^2/q$, and the message complexity of E-EDCH is $O(N2)$ for its lifespan.

V PERFORMANCE EVALUATION OF E-EDCH

In this part, an extensive performance study was directed to assess the performance of the E-EDCH by the simulation software used in [20] and [11]. The performance of E-EDCH is matched in contrast to LEACH, HEED, and Novel EDCH algorithms. LEACH, HEED, and EDCH are used as suggestion lines to evaluate E-EDCH due to the following:
• LEACH, HEED, and EDCH are renowned clustering algorithms for WSNs.
• All LEACH, HEED, and EDCH algorithms are entirely distributed.
• In all algorithms, every sensor is a CH or joins to precisely a single CH.
• In HEED and EDCH, the possibility that two nearby nodes stay elected as CHs, is lesser [20], [10].
E-EDCH will be matched with LEACH, HEED, and EDCH regarding:
1) The number of events recognized by the network.
2) The share of live active sensors to all live sensors over the network area.
The sensors' internal computational measures do not consume energy; all of their energy is castoff for message passing only. In this study, the energy prototype is the same as the one working in [7]. Furthermore, if parameter M presents the network edge, the base station is 10×M meters away from the network's edge. Additionally, the initial energy of every sensor is 10 J.

We directed three sets of experiments to match the performance of E-EDCH alongside that of LEACH, HEED, and EDCH. Every sensor has a haphazard activity in the first set: from 0 to 100%. Figures 3 and 4 indicate the number of recognized events for different numbers of nodes for E-EDCH, EDCH, HEED, and LEACH. In Figure 3, the network edge is M = 200 meters once the number of sensors differs from 100 to 500 nodes all through the networks. In Figure 4, the number of sensors is 300 nodes, whereas the network edge differs from M = 50 to 200 meters. In both figures, the horizontal axis demonstrates the number of sensors in the network; and the vertical axis demonstrates the number of recognized events for every algorithm. These figures show that E-EDCH senses additional events than EDCH, HEED, and LEACH on average by 104.28, 73.14, and 50.97 percent.

The first set of experiments displays significant over-performance of E-EDCH when related to EDCH, HEED, and LEACH for the number of identified events.

In the second set of experiments, we present the fraction of live active nodes to every live sensor node in the network. Every sensor node has a random activity rate from 0 to 100%, like in the first set of experiments. Figures 5 and 6 display the percentage of live active sensors to every live sensors node in the network for E-EDCH, EDCH, HEED, and LEACH algorithms. In this set, 200 sensors are divided haphazardly in a 200×200 square meter network. In Figure 5, we deliberated all sensor nodes as an active sensors if their activity rate was equivalent to or more than 90%. In contrast, in Figure 6, we deliberated all sensors as active if their activity rate was equivalent to or more than 50%. In both figures, the horizontal axis presents the number of deceased sensor nodes; and the vertical axis presents the proportion of live active sensor nodes to all live sensor nodes.

The second set of our study establishes that in E-EDCH, the active sensor nodes be alive, on average, longer than the others. On the other hand, in LEACH, HEED, and EDCH, particularly in the former, the active sensor nodes live, on average, not as long as the others.

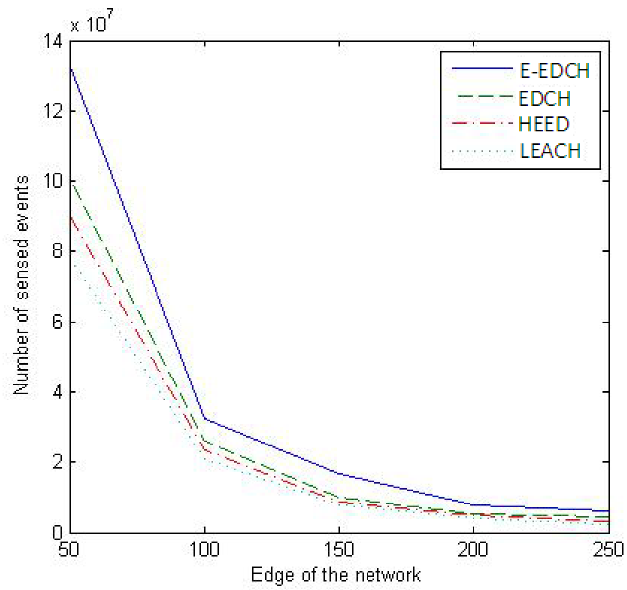

Fig. 4 Number of sensed actions for different numbers of network edges for N=300 sensor nodes and the sensor node's random activity rates.

Finally, in the last set of experiments, we want to present the live active and inactive sensor nodes in the network for LEACH, HEED, EDCH, and E-EDCH. As a result, 100 sensor nodes are divided haphazardly in a network range; 56 are active, and sensor nodes haphazardly select themselves as active or inactive sensor nodes in the network.

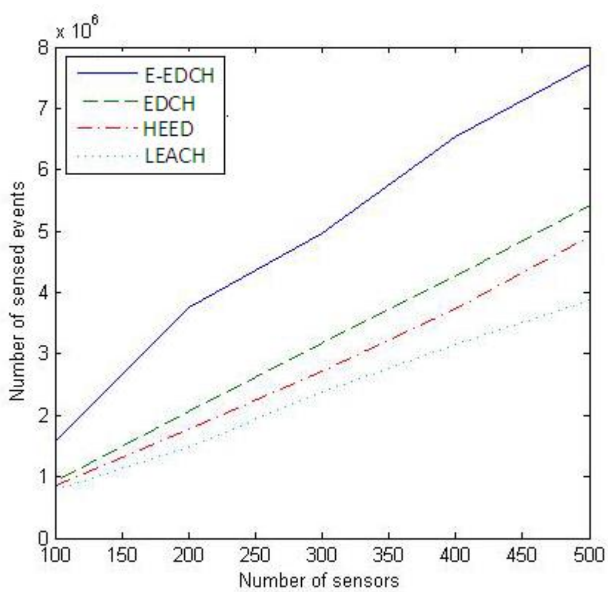

Fig. 3 Number of sensed actions for different numbers of sensor nodes for M=200 meters and the sensor node's random activity rates.

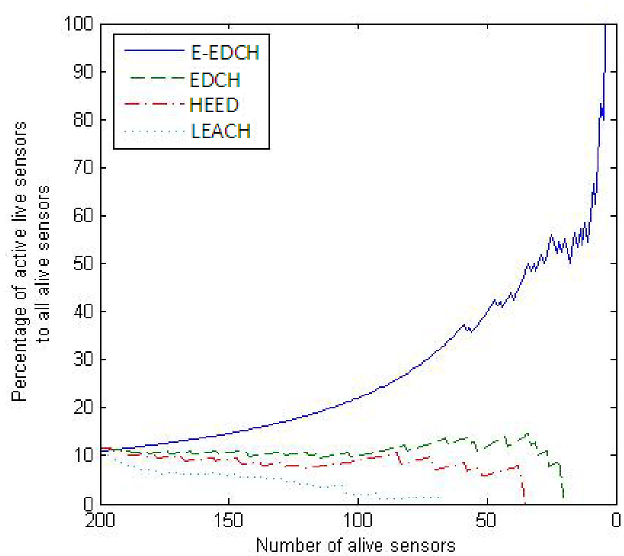

Fig. 5 The percentage of live active sensor nodes to all live sensor nodes in different numbers of died sensor nodes with random activity rates. Sensor nodes are measured as active if their activity rate is ≥ 90%.

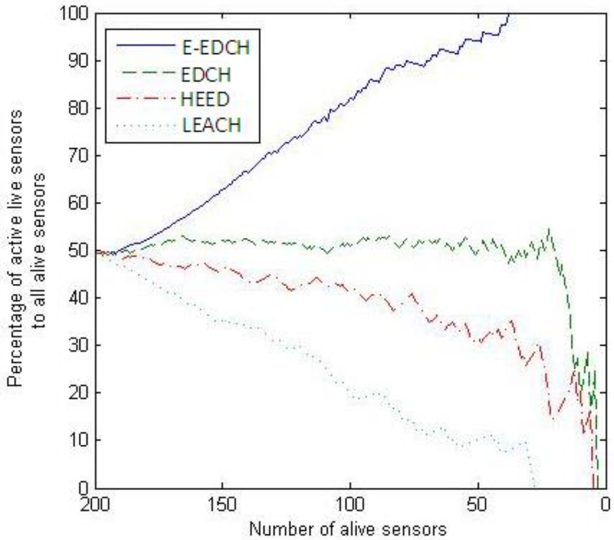

Fig. 6 The percentage of live active sensor nodes to every live sensor node in different numbers of died sensor nodes with random activity rates. Sensor nodes are considered active if their activity rate is ≥ 50%.

Figures 7, 8, 9, and 10 present the leftover nodes after 50 sensor nodes in the network have deceased. These figures show that in LEACH, 20 sensor nodes, HEED 26 sensors, EDCH 31 sensors, and E-EDCH 47 sensors amongst the leftover 50 nodes are active. So this experiment approves the first two sets of experiments, enlightening that E-EDCH might extend the life span of active sensor nodes in the network when related to EDCH, HEED, and LEACH.

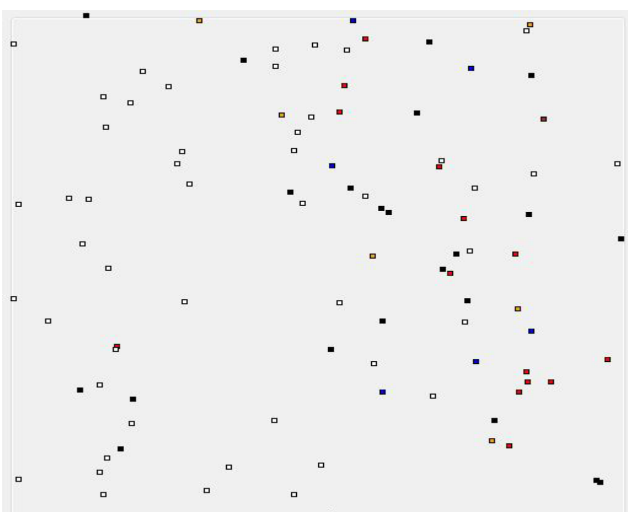

Fig. 7 Live and died sensor nodes for LEACH algorithm after dying of 50 sensor nodes. 20 be alive active, and 30 be alive inactive remains in the network.

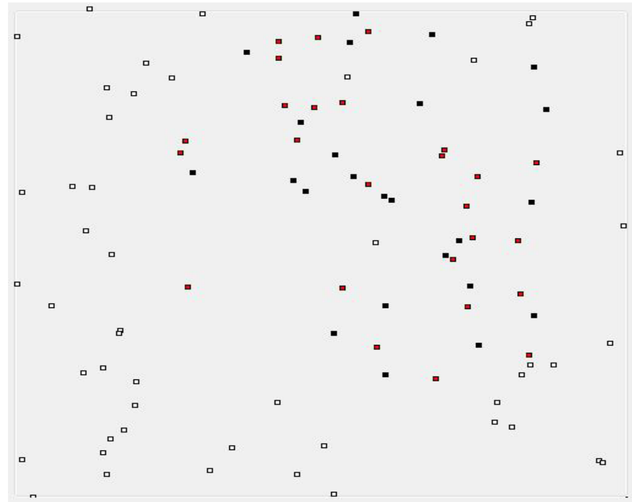

Fig. 8 Live and died sensor nodes for HEED algorithm after dying of 50 sensor nodes. 26 be alive active, and 24 be alive inactive remains in the network.

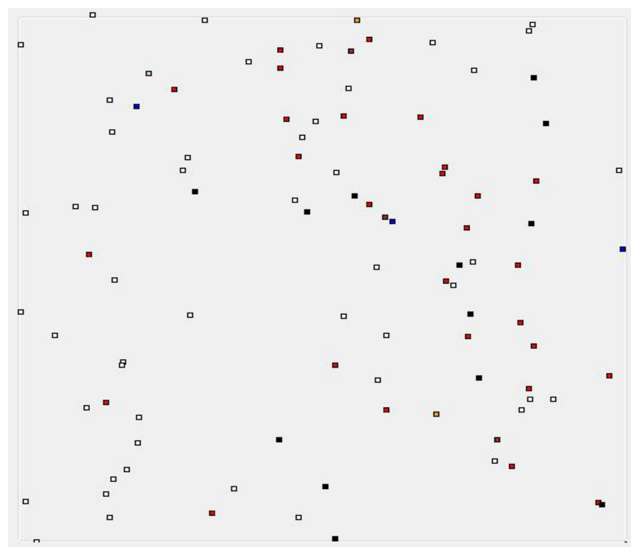

Fig. 9 Live and died sensor nodes for EDCH algorithm after dying of 50 sensor nodes. 31 be alive active, and 19 be alive inactive remains in the network.

VI CONCLUSION AND FUTURE WORK

In this application of (wireless sensor network) WSNs, the importance of distributed sensors node might not be equal due to how close they may or may not be too critical areas. In this paper, we projected the E-EDCH algorithm for this exacting WSNs, an Event-detection algorithm that extends the lifetime of active sensor nodes by sacrificing inactive sensor nodes. E-EDCH finds the least active sensor node in every cluster and chooses this sensor node as the CH (Cluster Head) to free the active sensors node from the heavy duties of cluster heading. The projected algorithm has been examined by comparing its effectiveness with these clustering algorithms LEACH, HEED, and original EDCH algorithms. Our evaluations showed that the E-EDCH outperformed the other three algorithms quite considerably.

REFERENCES

[1] K. Akkaya& M. Younis, "A survey of Routing Protocols in Wireless Sensor Networks," Adhoc network Journal, Elsevier, 2005, pp 325-349.

[2] F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.

[3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” in 33rd Annual Hawaii International Conference on System Sciences, 2000.

[4] Al-Baz A, El-Sayed A. Cluster head selection enhancement of LEACH protocol in the wireless sensor network. Minufiya J Electron Eng Res (MJEER). January 2017;26(1):153-169.

[5] M. Mathew and N. Weng, “Quality of Information and Energy Efficiency Optimization for Sensor Networks via Adaptive Sensing and Transmit-ting,” IEEE Sensors Journal, vol. 14, pp. 341–348, February 2014.

[6] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, “Performance Measurements of Motes Sensor Networks,” in 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), (Venice, Italy), pp. 174–181, ACM, 2004

[7] Karl, H., &Willig, A. (2005). Protocols and architectures for wireless sensor networks. Hoboken, NJ: Wiley.

[8] S. Lindsey, C.S. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 2002.

[9] Rajesh Patel, Sunil Pariyani, Vijay Ukani," Energy and throughout Analysis of Hierarchical Routing Protocol(LEACH) for Wireless Sensor Networks," International Journal of Computer Applications Volume 20- No. 4 (April 2011).

[10] Younis, O.and Fahmy, S., 2004. HEED A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. IEEE Transactions on Mobile Computing, 2004, 3(4), 660–669.

[11] Rajkumar and Dr. H G Chandrakanth Mathematical Modeling of EDCH Clustering Algorithm for WSNs. In International Journal of Computer Science and Engineering, Vol.7, Issue.3, March 2019.

[12] A. Maizate and N. El Kamoun. Efficient survivable self-organization for a prolonged lifetime in wireless sensor networks. International Journal of Computer Applications, 58(16):31–36, November 2012.

[13] C. S. Nam, H. J. Jeong, and D. R. Shin. The adaptive cluster head selection in wireless sensor networks. In Semantic Computing and Applications, 2008. IWSCA '08. IEEE International Workshop, pages 147–149, Incheon, South Korea, July 2008.

[14] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. Wireless Communications, IEEE Transactions on, 1(4):660–670, October 2002.

[15] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. Communications Magazine, IEEE, 43(3):8–13, March 2005.

[16] I. Gupta, D. Riordan, and S. Sample. Cluster-head election using fuzzy logic for wireless sensor networks. In Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual, pages 255–260, Washington, USA, May 2005.

[17] G. Ran, H. Zhang, and S. Gong. Improving on leach protocol of wireless sensor networks using fuzzy logic. Journal of Information & Computational Science, 7(3):767–775, 2010.

[18] M. J. Handy, M. Haase, and D. Timmermann. Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In Mobile and Wireless Communications Network, 2002. 4th International Workshop, pages 368–372, Stockholm, Sweden, September 2002.

[19] G. Smaragdakis, I. Matta, and A. Bestavros. Sep: A stable election protocol for clustered heterogeneous wireless sensor networks. Technical report, Boston University Computer Science Department, 2004.

[20] Rajkumar, Dr. H G Chandrakanth," EDCH: A Novel Clustering Algorithm for Wireless Sensor Networks," in EJERS, European Journal of Engineering Research and Science, Volume-4, Issue- 3., March 2019, 10.24018/ejers, ISSN: 2506-8016.