IJCTA, 9(27), 2016, pp. 51-59 ISSN: 0974-5572
© International Science Press

Light Weight Distributed Cut Detection Mechanism in Wireless Sensor Network

B. Murali* and T. N. Ravi**

ABSTRACT

Wireless Sensor Network (WSN) experiences an intruded on availability brought about by a few perspectives regularly. For example, the unattended operation is defenseless against surely altering and constrained battery power of a node. In accession to that, the barriers are frequently accessible in touched to network cut, rapid information misfortune contemptible routing choices, and exploitation of strength. A WSN can be slides into several connected divisions. As a result of the disappointment of some of its nodes is termed as "cut". In addition, this research paper is also analyzed about another recognition mechanism called Light Weight Distributed Cut Detection. And it is dealt in the proposed structure to recognize the causes for cuts in the wireless sensor network. Moreover, this is an enhanced re-routing algorithm which is deployed as a part of this framework to distract the route once there is an access to for cut and that has been renowned in the network.

Keywords: Wireless Sensor Networks, Packets, Nodes, Light Weight Distributed Cut Detection, re-routing, Detection value, threshold, ranking.

1. INTRODUCTION

Wireless Sensor Network (WSN) [1] comprises of little nodes with wireless communication, computation and sensing abilities. Power management, information dissemination and numerous protocols have been particularly intended for WSNs where vital mindfulness is a key outline issue. WSNs have an extraordinary point of preference for different applications in our genuine living.

WSNs are a competent in analyzing the widespread areas at chronological decision and high spatial. It is to reveal that the node malfunction is relied upon to be completely standard owing to the generally classified vitality spending structure of the nodes that are restricted by the little batteries. The malfunction of an arrangement of nodes will diminish the capacity of numerous ways in the network [2]. Such failures can expose a split of nodes that have not mistreated to wrap up the disconnected from the rest in bringing about a "cut". Among that two nodes are recognized as disconnected if there is impossibility between them. Wireless Sensor Network encompasses of wide-ranging measures of low-power and low-cost.

2. REVIEW OF LITERATURE

The proposed research deals about the low transparency plan for distinguishing a network packet or cut in a sensor network [3]. Consider a set S of n sensors, which are demonstrated as remote nodes, has been delayed in utilization of numerous applications: military surveillance, medical case and disaster response among others. The special characteristics of WSNs, for example, unattended operation, battery-fueled nodes, and brutal situations posture real difficulties. One of the difficulties is to warranty the network which is connected. The accessibility of the network can be devoid in a large amount of stretch to upset, as a result of that the eccentric wireless channels, early exhaustion of node's strength, and physical altering by aggressive clients.

* Associate Professor & Head, Department of Computer Science, PSG College of Arts and Science, Coimbatore, India.

** Assistant Professor, Department of Computer Science, Periyar EVR College, Trichy, India.

In contrast, the opponent can distress the communication with the objective that sensors on one side of the line can't articulate with sensors on the other side, as well as the base station. At this point it is termed as linear cut a [-cut if in any event [part of the sensors are cut off, where 0 < [<1 is a client indicated parameter. The Distributed Source Separation Detection (DSSD) algorithm [4] is not constrained to [-linear cuts; it can recognize the cuts in different network of numerous segments of discretionary shapes. Moreover, the DSSD algorithms are unlimited to arrangement conveyed in 2D, it doesn't necessitate in sending sentinel nodes, and it authorizes each node to identify, if a cut happens. The algorithm of DSSD includes the closest neighbor communication immediately to extract the condition of map-reading messages to the source node. This module makes the algorithm material to versatile nodes also. Initially, the cut detection issue was measured in a wired network [5].

The disparate to the predictable cut detection issue, in endeavor to identify a network cut between a sender and any node in an arrangement of given destinations. Point-to-Point Cut Detection protocol [6] (P2P-CD) is utilized.

3. PROPOSED FRAMEWORK

The figure 1 represents the proposed framework for Light Weight Cut Detection Algorithm. This framework is used to detect the cuts in the Wireless Sensor Networks.

3.1. Route Discovery

The procedure of route discovery [7] begins when a source node does not have routing data for a node to be communicated with. Route revelation is started by broadcasting a Route Request (RREQ) message. The route is built up when a Route Reply (RREP) message is received. A source node might get different RREP messages with assorted routes. Then, it is upgraded its routing passages if and just if the RREP has a more important sequence number, i.e. crisp data.

3.2. Node Discovery

In distinguishing the nodes [8] and their locations is a fundamental capacity in a WSN after starting and sending, as well as for coordinating recently included nodes. The extent of node revelation is liable to certain tradeoffs in view of the application objectives. In some cases, for expansive networks, asset reserve funds as far as vital in broadcasting competency. And it can be practiced by not sharing a percentage of the topology subtle elements that are regarded superfluous for specific parts of the network.

3.3. Light Weight Distributed Cut Detection Algorithm

This light weight dispersed cut identification algorithm is deployed for considerable amount of sensor nodes are randomly sent in a 2D range. Every sensor node creates tangible information sporadically and every one of these

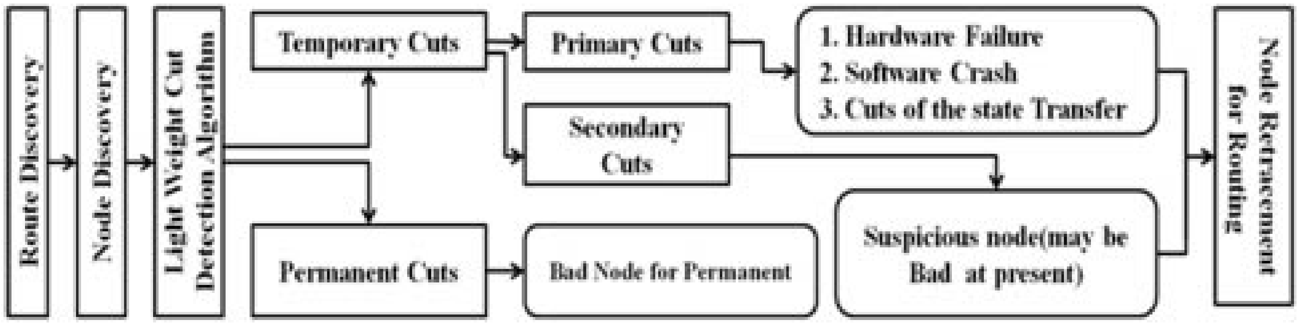

Figure 1: Proposed Framework for Light Weight Cut Detection Algorithm in WSN

nodes works mutually to forward packets in enclosing the information towards a sink. The sink is situated within the network. The sink is accepted and all sensor nodes are generously time harmonized which is required by numerous applications. The sink recognized about the network topology, which can be consummated by cooperative nodes to report their adjacent nodes openly after arrangement.

Stage 1: Initialization

In the starting stage, sensor nodes frame a topology which is a coordinated DAG (Directed Acyclic Graph) [9]. From the DAG, a routing tree is evoked. Once the structure of routing tree is generated, then the reports of information are considered. The reasons for framework initialization is to establish secrete pair wise keys among each general sensor node and the sink. To build up the routing tree and Directed Acyclic Graph (DAG) is to encourage packet sending from each sensor node to the sink. All sensor node v is preloaded the accompanying data:

Secret key solely shared among the nodes which is given by Sv and the sink

Dr – Cycle Duration

Pn: the greatest amount of parent nodes that every node records amid the DAG foundation technique.

Ps¬th packet is numbered Ps _ 1, 0 is used to represent the Ps - 1th packet, and so forth.

Ps: the most extreme packet arrangement number.

Stage 2: Identification of Causing Cut in Network

In each circle, information are transformed all the way to the sink via the routing tree. Every packet forwarder/ sender comprises a small amount of additional bits to the packet. Furthermore, it encodes the packet. In this situation when one cycle achievements, in light of the supplementary bits expressed in the received packets, the sink runs a node arrangement algorithm to recognize the nodes that should be awful nodes and untrusting awful. The algorithm tree is redesigned in each cycle, when a definite number of cycles have passed, sink gathers enough data about the node practices in various routing topologies. The data incorporates which nodes are terrible without a doubt and these nodes are untrusting awful.

Stage 3: Sending of the Packet

In that situation, when a sensor node v has a data item I to report, it generates in sending the supplementary packet to its parent node Pv: <Pv, {Rv, v, Dp MOD Ps, I, pad v, 0 } Sv, pad v, 1> Where Pv - parent node, Rv - accepting node, V-node, Dp - counter node, I - information, pad v, 0 - padding, Sv encryption. Paddings pad v, 0 and pad v, 1 are supplementary to create all packets break even with long, such that sending nodes can't inform packet resources taking into description packet length. In the mean time, the sink can even decrypt the packet to ascertain the authentic real substance now.

Stage 4: Forwarding of the Packets

At this position when a sensor node w gets encrypted packet hw, nj it forms and progresses the associated packet to its parent node Pw: <Pw, { Rw, n'}Sw where n0 is obtained by frilling the furthest right log (Pn) bits off n. Then, Rw, which has log Pn bits, is supplementary to the front of n'.

Stage 5: Receiving of the Packets at the Sink

The sink attempts to determine a child node for each guardian node by decoding where it brings about a string. On the inedible possibility that the endeavor fizzles the packet is changed and it should be dropped. In the event that it thrives the packet is sent from the regarded node.

Algorithm 1: Receipt of Packet at the Sink

```
Step 1: Packet <0; n> as Input.
Step 2: Decrypt the packet if Success Attempt = false
Step 3: if decryption fizzles then proceed, else
Step 4: The sequence is recorded if Success Attempt = genuine
Step 5: v ← w, Success Attempt = false; proceed from step 4;
Step 6: if Success Attempt = false then
Step 7: Then the packet is dropped.
```

Algorithm 2: Light Weight Distributed Cut Detection

```
Step 1: Input: Tree R, with every node v set apart by “+” or “_,”and its dropping proportion dv.
Step 2: for every leaf node v in R discover parent node until the sink node classify the nodes
Step 3: consider v as positive limit and w as negative edge.
Step 4: if w.mark = “”_”” then until w.mark = “”+”” or w is Sink, Set nodes from a to c as awful without a doubt;
Step 5: if w is Sink then Set v as awful without a doubt;
Step 6: if w.mark =’’+’ and if w is not awful for beyond any doubt then Set v and w as untrusting terrible else
Step 7: if dw - dv > 5θ then
Step 8: Set w as awful without a doubt;
Step 9: if distinction dv – dw > 5θ then Set v and w as untrusting awful
```

- Pv, max – the series number which is mainly just seen.
- Pv, flip - the amount of progression number flips
- Pv, rcv - number of established packets.

The tumbling proportion in every cycle is premeditated as follows:

$$d_v = \frac{p_{v,flip} * P_s + p_{v,\max} + 1 - p_{v,rcv}}{p_{v,flip} * P_s + p_{v,\max} + 1}$$

To recognize mainly apt bad nodes from untrusting nodes:

$$S_j = \{\langle v_i, w_i \rangle | \langle v_i, w_i \rangle is\ a\ suspicious\ pair\ \langle v_i, w_i \rangle = \langle w_i, v_i \rangle$$

It is believed with the aim of nodes are not purposefully fallen by sending nodes, then tumbling quantity of this node should be lower than θ. Note that θ ought to be more remarkable than 0. At this time let us accept θ quality is 0.5. The arrangement of nodes can be taken in any of the accompanying cases (i) packet droppers without a doubt. (ii) Untrusting packet droppers. (iii) No packet droppers without a doubt. The proposed Light Weight Distributed Cut Detection algorithm deals with respect to each sink node in T and the accompanying cases exist.

Case 1: $+\{+\}^{+}$ If the dropping proportion is not exactly è , then a node do not have dropped packets (called useful without a doubt) or the node is suspected to have dropped packets (called untrusting terrible).

Case 2: $+\{-\}^{+}$If the dropping proportion worth is equivalent to 0, then the node does not have dropped packets.

Case 3: -{+}$^+$If the dropping proportion is not exactly θ, however more noteworthy than zero means, then the node is supposed to have dropped packets.

Case 4: -{-}$^+$If the tumbling proportion is more prominent than θ, then a node more likely than not dropped packets (called awful without a doubt). The dropping packets might be malignant node, traffic and collision. In view of the above cases, I added to a node order algorithm to discover nodes that whether the node is terrible without a doubt, untrusting awful, or useful without a doubt. The tree deployed to forward information is progressively changed from cycle to cycle and every sensor node might have an interchange parent node which is called tree redesign happens.

Stage 6: Prioritize the nodes after Light Weight Distributed Cut Detection algorithm

The tree deployed to forward information is progressively distorted progressively from cycle to cycle, which empowers the sink to watch the conduct of every sensor node in a huge assortment of routing topologies. For every one of these situations, node arrangement algorithm is connected to distinguish sensor nodes that are awful without a doubt or untrusting terrible. After different cycles, sink additionally recognizes awful nodes from those are applying in relation to untrusting terrible a few proposed heuristic strategies.

- Tree Reshaping: The tree utilized for sending information from sensor nodes to sink progressively distorted from cycle to cycle [10]. For example each sensor node might have an alternate parent node from cycle to cycle. To let the nodes and the sink have a reliable perspective of their parent nodes. After this the reshaping of the tree is designed later. Towards the initiation of every cycle (j = 1, 2 …), node v picks the [hcj(Sv) MOD pn;v] the parent node as its parent node for this cycle, where hc is a hash capacity and hcj(Sv) = hc(hcj^{i-1} (Sv))**.** Note that, the parents are chosen and foreordained by both the preloaded secret Sv and the rundown of parents traced in the stage of tree establishment. The determination is known by the sink. In this way, the proceeding roguishly node can't self-assertively choose its parent for its attacks.

$$S_j = \{\langle v_i, w_i \rangle | \langle v_i, w_i \rangle \textit{ is a suspicious pair } \langle v_i, w_i \rangle = \langle w_i, v_i \rangle$$

- Recognizing the majority of probable Bad Nodes from Untrusting Bad Nodes: After cycle closures, the sink figures the falling rate of each node, and scuttles node classification algorithm to distinguish nodes that are terrible without a doubt or untrusting awful. Subsequent to the amount of untrusting awful nodes are possibly huge. To explore this research depicts that how to distinguish in all probability terrible nodes from the untrusting awful nodes as takes after. By looking at the guidelines in Case 3 and Case 4 for recognizing untrusting terrible nodes. The proposed strategy analyzes, in each of these cases (i) there are two nodes, meant as v and w, which have the same likelihood to be the awful nodes and (ii) no less than one of them must be awful. The entitled these nodes as an untrusting pair. For each cycle j, all recognized untrusting sets are recorded in an untrusting set indicated

Therefore, after m cycles of discovery, we can acquire a progression of untrusting sets: S1; S2; … ; Sm.

Algorithm 3: Integration of Global Based and Stepwise Based Ranking Algorithm for organize the nodes in the network

The Global based Ranking (GR) technique is able to recognize the most awful nodes with some misallegations while the Stepwise Ranking (SR) strategy has less miscalculation though may not identify the similar quantity of dreadful nodes as the GR (Global Ranking) strategy. To adjust the tradeoff, an innovative proposal in the Hybrid Ranking (HR) strategy, this is formally furnished. In hybrid ranking, the node with the most noteworthy charged record worth is immobile and foremost picked as no doubt awful node. After an undoubtedly terrible node has been picked, the one has the most astounding charged record esteem among the rest is picked just if the node has not generally been blamed together with the awful nodes that have been recognized as of now. Along these lines, the allegation account worth is considered as a vital foundation in recognizable proof, as in the GR technique; then, the likelihood that a guiltless node being confined by terrible nodes is likewise measured by not picking the nodes who have continually being alleged as one with officially distinguished awful nodes, as in the SR (Stepwise Ranking) strategy.

```
Step 1: Sort every single untrusting node into Queue L as indicated by the slipping request of their blamed
        record values, $\varnothing$ = null.

Step 2:

Step 3: While $V_{j=1}^{m} S_j \neq \varnothing$

Step 4: v ← deque(L)

Step 5: if there exists $\langle v,* \rangle V_{j=1}^{m} S_j$ then

Step 6: $\bar{S} \leftarrow \bar{S} \bigcap \{ v$

Step 7: expel all
```

3.4. Node Retracement for Re-Routing the Suspicious Node

After it has been set up that specific usefulness is not accessible anymore because of a disappointment in the essential routing [11] nodes, another administration supplier must be chosen. In this, a new framework called Node Retracement model for re-routing the suspicious node in the Wireless Sensor Network to be proposed as the future work.

4. SIMULATION RESULTS AND DISCUSSIONS

4.1. Simulation Environment

The proposed light weight distributed cut detection plan is actualized in both ns2 simulator and java to assess the efficiency and effectiveness of the proposed plan. The execution of the proposed plan is measured from two perspectives: the rate of detection, characterized as the proportion of effectively distinguished terrible nodes, and the false positive likelihood, characterized as the proportion of mis-accused honest nodes over every single pure node. The proposed analysis runs the simulation on a 400 × 400m^2 network with haphazardly produced network topology. Unless generally expressed, the proposed set is the rate of awful nodes to 10%, 100 sensor nodes is the size of the network, the per-node packet reporting interim to 3 seconds, and the length of each cycle to 300 seconds. Additionally, when an awful node chooses to drop packet in a cycle, it drops 30% of the packets.

4.2. Results and Discussion

Effect of the Number of Cycles: We think about the quantity of cycles expected to gather data such that a high detection and stable also a low false positive likelihood is accomplished. The attack models 1-1, 1-2, 2-1, 2-2, 3-1 and 3-2 is the transition from node to node. We utilize HR (Hybrid Ranking) algorithm here and first plot the detection rate under the six attack models in each cycle in table 1. From the table 1, we can see every single terrible node can be distinguished after 8 cycles paying little respect to the attack model. Among them, under attack model 1-2, the terrible nodes will be identified rapidly after 5 cycles. This is on the cycles that an awful node does not drop packets from its downstream nodes at a few interims, which brings about the + {“} + case and the awful nodes can be identified promptly as per the proposed guideline. Despite what might be expected, under attack model 3-2, more cycles are expected to accomplish a higher detection rate. For this situation, awful nodes are shrewd and their self-generated packets are not dropped. Subsequently, they are just arranged as suspiciously terrible nodes. More cycles are required before they are in the long run identified by means of a ranking algorithm. Subsequent to the attack model 3-2 is the most troublesome one; we consider the standard deviations of the detection rate and the false positive likelihood under this attack model. The information is used to register the standard deviations is received from the simulation keep running more than 50 arbitrary network topologies.

Table 1
Number of Cycles vs. Detection Value and Standard Deviation of Detection Value & False Positive

	Detection Value							
No of Cycle	*AM 1-1*	*AM 1-2*	*AM 2-1*	*AM 2-2*	*AM 3-1*	*AM 3-2*	*SD of Detection Value*	*SD of False Positive*
1	0.77	0.87	0.71	0.73	0.73	0.5	0.4	0.15
2	0.89	0.98	0.88	0.86	0.85	0.71	0.625	0.2
3	0.95	0.981	0.95	0.95	0.94	0.82	0.78	0.3
4	0.98	0.99	0.96	0.97	0.97	0.88	0.83	0.25
5	0.98	1	0.97	0.985	0.98	0.92	0.89	0.25
6	0.99	1	0.975	0.979	0.99	0.93	0.915	0.25
7	0.975	1	0.97	0.98	0.99	0.935	0.92	0.26
8	0.99	1	0.98	1	1	0.95	0.93	0.26
9	1	1	1	1	1	0.965	0.94	0.29
10	1	1	1	1	1	0.98	0.935	0.28

The simulation results are appeared in table 1. As should be obvious, the standard deviation of detection rate gets to be little as the quantity of cycle’s increments. It gets to be steady after 8 cycles at a cycle 0.125. The standard deviation of the false positive likelihood is higher than that of detection rate, yet it is still as low as 0.15.

Effect of the Percentage of Bad Nodes: Table 3 demonstrates the detecting execution as the rate of terrible nodes changes. By and large, the less the quantity of awful nodes and the less demanding is used to distinguish these nodes. Be that as it may, after a numerous cycles of identification, the identification rates under various rate of terrible nodes get to be comparable, and every one of them accomplish high detection rate. In the table 3, Case 1 represents 2% bad nodes, Case 2 gives 6% bad nodes, Case 3 represents 10% bad nodes and Case 4 gives 14% bad nodes.

Table 2
Number of Cycles with Bad Nodes Detection Values and Bad Nodes False Positive

	Detection Value of Bad Nodes				*False Positive Value of Bad Nodes*			
No of Cycle	*Case 1*	*Case 2*	*Case 3*	*Case 4*	*Case1*	*Case2*	*Case3*	*Case4*
1	0.7	0.58	0.575	0.57	0.001	0.002	0.003	0.0035
2	0.96	0.79	0.75	0.72	0	0.002	0.006	0.0075
3	1	0.92	0.88	0.8	0.001	0.002	0.004	0.014
4	1	0.95	0.92	0.85	0.001	0	0.01	0.017
5	1	0.95	0.95	0.92	0.001	0.001	0.009	0.017
6	1	0.95	0.95	0.93	0.001	0.007	0.008	0.016
7	1	1	0.97	0.92	0.001	0.007	0.007	0.016
8	1	1	0.98	0.93	0.001	0.004	0.0145	0.016
9	1	1	0.98	0.93	0.002	0.01	0.016	0.021
10	1	1	0.98	0.95	0.002	0.016	0.0155	0.017

Effect of Threshold: (1) Differentiating nodes "-" and "+" by threshold. So as to implement the proposed plan to endure regular packet misfortune, we utilize an threshold µ when denoting every node with″+″ or ″-″.

In the table 4 and table 5 the following are used to represent the cases. Case 1 is used to represent 2 cycles, case 2 gives 4 cycles, case 3 represents 8 cycles and case 4 represents 8 cycles. Table 4 demonstrates the effect of this threshold on the detection execution. As delineated in table 4, the bigger is the threshold; the lower is the detection rate. This is on the cycles that, less nodes will be set apart as ″-″ as the threshold increments; consequently,

Table 3
Threshold with Detection Value of Bad Nodes and False Positive of Bad Nodes

	Detection Value of Bad Nodes					*False Positive Value of Bad Nodes (in %)*			
Threshold	*Case 1*	*Case 2*	*Case 3*	*Case 4*	*Threshold*	*Case 1*	*Case 2*	*Case 3*	*Case 4*
1	0.75	0.92	0.95	0.98	1	0.02	0.02	0.025	0.03
2	0.75	0.92	0.95	0.98	2	0.01	0.02	0.02	0.01
3	0.75	0.92	0.95	0.98	3	0.01	0.01	0.02	0.01
4	0.75	0.92	0.95	0.98	4	0.01	0.02	0.025	0.015
5	0.75	0.92	0.95	0.98	5	0.02	0.04	0.06	0.06
6	0.75	0.92	0.95	0.98	6	0.022	0.06	0.09	0.09
7	0.75	0.92	0.95	0.98	7	0.03	0.065	0.1	0.12
8	0.75	0.92	0.95	0.98	8	0.04	0.065	0.11	0.14
9	0.74	0.92	0.95	0.98	9	0.035	0.06	0.12	0.16
10	0.69	0.88	0.94	0.95	10	0.075	0.14	0.18	0.24
11	0.65	0.86	0.91	0.94	11	0.12	0.19	0.225	0.255
12	0.57	0.79	0.85	0.92	12	0.078	0.12	0.165	0.19
13	0.52	0.75	0.83	0.88	13	0.048	0.095	0.14	0.155
14	0.46	0.62	0.72	0.82	14	0.05	0.11	0.13	0.17
15	0.42	0.59	0.66	0.78	15	0.04	0.07	0.11	0.15

Table 4
Identifying Bad Nodes using Detection Value and False Positive value on Threshold with Various Dropping Rates

	Detection Value					*False Positive Value*			
Threshold	*Case 1*	*Case 2*	*Case 3*	*Case 4*	*Threshold*	*Case 1*	*Case 2*	*Case 3*	*Case 4*
1	0.82	0.97	0.98	0.98	1	0.58	0.72	0.8	0.83
2	0.81	0.98	0.99	0.99	2	0.41	0.58	0.68	0.72
3	0.82	0.99	0.99	0.98	3	0.26	0.42	0.5	0.58
4	0.83	0.98	1	1	4	0.18	0.28	0.35	0.4
5	0.825	0.98	1	1	5	0.11	0.18	0.22	0.25
6	0.81	0.97	1	1	6	0.06	0.1	0.12	0.14
7	0.81	0.97	0.99	1	7	0.02	0.05	0.02	0.05
8	0.805	0.96	1	1	8	0.01	0.03	0	0.02
9	0.8	0.94	0.99	1	9	0	0.01	0	0.01
10	0.78	0.92	0.98	0.98	10	0	0	0.01	0.01
11	0.57	0.7	0.78	0.88	11	0	0	0	0
12	0.4	0.5	0.58	0.62	12	0	0	0	0
13	0.3	0.38	0.38	0.4	13	0	0	0	0
14	0.2	0.3	0.28	0.32	14	0	0	0	0

a portion of terrible nodes will escape from being recognized. As appeared in table 4, when the threshold builds, the false positive likelihood expands first and then it diminishes after the threshold achieves a specific quality (turning point). Thus, this research selects the threshold to be 0:1, with which high detection rate and low false positive can be accomplished all the while, as appeared in table 4. (2) Threshold for Identifying Nodes with Dropping Rates. Considering that accidental impacts might bring about two nodes to have distinctive dropping rates, we utilize a threshold to separate the case that two nodes truly have diverse dropping rates with the case there are coincidental crashes. Table 5 demonstrates the effect of this threshold on the detection rate and the false positive. We can observe that, the bigger the threshold, the lower the detection rate and the false positive likelihood. This is on the cycles that, the distinction in dropping rates between two nodes is a critical parameter for our ranking algorithms to separate the practices between nodes of parent-child. On the off chance that the threshold is too expensive, the proposed algorithms can't locate the anomalous practices that are exclusively pondered the dropping rate contrast. In the event that the threshold is too little, the false positive likelihood will be expanded, which is appeared in table 5. In view of the simulation, it discovered threshold 0:1 can render a high detection value and low false positive likelihood.

6. CONCLUSION

The proposed research illustrates in addressing the issues of cuts in the wireless sensor network. The proposed light weight distributed cut detection method which is a simple and effective technique for classifying the nodes into good, bad and suscipious that causes a cut in the networks. In different scenarios, the proposed method is effectively verified by means of conducting simulations and extensive analysis was made based on obtained results. In the future, the level of the suspicious node can be predicted by using Fuzzy Inference System (FIS).

REFERENCES

[1] Sookyoung Lee, Mohamed Younis, Meejeong Lee, "*Connectivity restoration in a Partitioned wireless sensor network with assured fault tolerance*", Adhoc Networks -Elsevier, pp.1-19,2014.

[2] Mariam Alnuaimi, Khaled Shuaib, Klaithem Alnuaimi and Mohammed Abed-Hafez, "*Data Gethering in Wireless Sensor Networks with ferry nodes*", Proceeding of 2015 IEEE 12th International Conference on Networking, Sensing and Control, pp.221-225, 2015.

[3] David Naranjo-Hernández, Laura M. Roa, Javier Reina-Tosina, Miguel A. Estudillo-Valderrama, and Gerardo Barbaro, "*Low power Platform and Communications for the development of wireless body sensor networks*", International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation, pp.1-13, 2015.

[4] Ganesan Subramanian, Zaheeruddin Ahmed, Niyi Okelola, Arumugaselvi Murugan, "*LEACH Protocol based Design for Effective Energy Utilization in Wireless Sensor Networks*", Science and Technology (TICST), 2015 International Conference, pp. 385-389, November 2015.

[5] Mr. D. Shiva Rama Krishna, K. Sridhailam, "*Cut Detection in Wireless Sensor Networks*", International Journal of Research (IJR), pp.438-448, December 2015.

[6] Ka-Lun Lee, Jie Li, Chien Aun Chan, N. Prasanth Anthapadmanabhan, Hungkei (Keith) Chow, "*Energy-Efficient technologies for point to point fiber access*", Optical Fiber Technology - Elsevier, pp.71-81, December 2015.

[7] Adnan Ahmed, Kamalrulnizam Abu Bakar, Muhammad Ibrahim Channa, Abdul Waheed Khan, "*A Secure Routing Protocol with Trust and Energy Awareness for Wireless Sensor Network*", Mobile Networks and Application-Springer, pp.1-14, 2016.

[8] Muhammad Muneer Umar, Nabil Alrajeh and Amjad Mehmood, "*SALMA: An efficient State based Hybrid Routing Protocol for Mobile nodes in Wireless Sensor Networks*", International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation, pp. 1-12, 2016.

[9] Nicklas Beijar, Oscar Novo, Jaime Jimenez, Jan Melen, "*Gateway Selection in Military Networks*", 5th International Conference on Internet of Things (IoT), pp.90-97, 2015.

[10] Xu Xu, Weifa Liang, Xiaohua Jia and Wenzheng Xu, "*Network throughput maximization in unreliable wireless sensor networks with minimal remote data transfer cost*", Wireless Communications and Mobile Computing, pp.1-17, 2015.

[11] Md Azharuddin, Pratyay Kuila, Prasanta K. Jana, "*Energy Efficient Fault Tolerant Clustering and routing algorithms for wireless sensor networks*", Computer and electricals Engineering-Elsevier, pp.1-14, 2014.