International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 9 Sep. 2022, pp: 1292-1296 www.ijaem.net ISSN: 2395-5252

Wireless Sensor Networks in Cluster Based Fault Tolerant Localization

M. Rajkumar[1], P.Panneer selvan[2]

[1,2]Assistant Professor, Department of Computer Science, Kaamadhenu arts and science college, Sathyamangalam

Date of Submission: 17-09-2022 Date of Acceptance: 28-09-2022

ABSTRACT: Wireless Sensor Network (WSN) has long been regarded as an essential method for many applications. The position of sensor nodes is needed for the majority of WSN applications. The location knowledge of WSN nodes is critical for determining the source of incidents and acting on them. For this reason, several localization algorithms have been created. We suggested CFTL cluster-based fault- tolerant localization in this paper, which is commonly used in WSN applications. These algorithms use a few reference nodes with position information to localize other nodes using these nodes as a reference. However, in practice, certain reference nodes can fail and report incorrect position details to other nodes. This limits the network's overall localization accuracy. As a result, it is critical to recognize and exclude defective reference nodes from the localization phase. However, since both defective nodes and heterogeneous nodes change hop distances in Wireless Sensor Networks (WSN), identifying just faulty nodes within a group of heterogeneous nodes becomes much more difficult. This paper presents a fault filtering approach for fault-tolerant localization that can be used with some of the current hop-based localization algorithms. Under defective conditions, the Fault Filtering algorithm exhibits improved localization accuracy and more robust efficiency.

Keywords: WSN, Fault Detection, CFTL, Cluster Head

I. INTRODUCTION

Many spatially distributed sensor nodes are used in wireless sensor networks (WSNs) to sense or track the physical and environmental environments around it. Temperature, pressure, gas, soil moisture, proximity sensor, light sensor, humidity, ultrasonic sensors, and other sensors are examples of sensor nodes. Per sensor node has a transducer, microcomputer, and transceiver. WSNs have enormous capacity for developing powerful applications due to the diversity of physical and environmental environments [2]. The position of sensor nodes is needed for the majority of WSN applications. This positioning data can be used in navigation, goal detection, place-conscious data collection, and other applications [3]. For localization, the Global Positioning System (GPS) is a common option. However, these technologies necessitate line-of-sight visibility between nodes and satellites, which is not feasible in the majority of implementations [1]. Furthermore, attaching a GPS receiver to each node is not feasible [4]. As a result, localization has become a significant issue in WSNs. It is being thoroughly researched for a broad variety of applications [5]. Since they do not need any extra hardware support for range measurements, range free localization algorithms are more common for large scale WSNs.

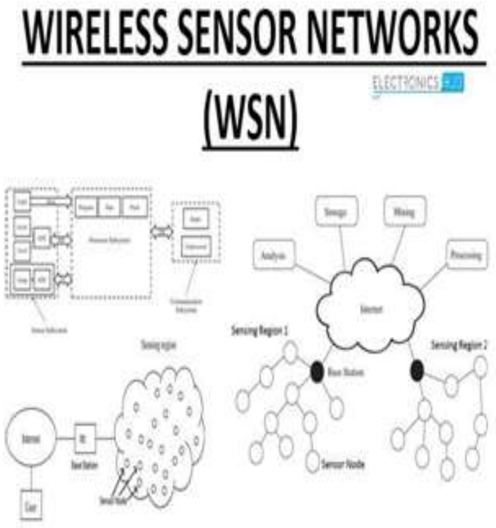

Figure 1: Wireless Sensor Networks

A WSN is typically made up of a large number of sensor nodes with minimal processing and power capacities that communicate over unreliable and low bandwidth radio links [3]. As a result, this resource- constrained environment suffers from repeated node and connectivity failures. The usefulness of a WSN-based programme, on the other hand, stems from providing stable facilities, which necessitates the implementation of fault tolerance techniques. The use of node redundancy is a standard method for providing fault tolerance in WSNs. However, this solution is insufficient to meet the application's specifications. Users ought to identify a certain event (fire identification, tracking) with a certain level of accuracy, for example, no false negatives with or without tolerating false positives. As a result, the optimal responsiveness, i.e., efficiency and timeliness of data transport, often differs across applications. Certain applications, such as wildlife monitoring, may require low responsiveness in extreme situations, while others, such as military applications, might require strong responsiveness [8].

Other intermediate responsiveness groups may be defined, such as applications that do not need high delivery reliability [6] but do need delivery timeliness, i.e., if any data is missing, application output would not suffer. Still, data should arrive within the time bounds provided by the application.

This work contributes to the state of the art by demonstrating that we can balance the ease of constructing a centralised fault management system to reduce power usage usually offered by dispersed fault management systems. As a result, the proposed device achieves high precision in fault detection while also using little resources [9].

The following is how this paper is structured. Section II discusses related literature concerned with fault control schemes. The suggested fault management scheme is then presented in Section III, and the experiments are described in Section IV. Section V reports on simulation performance, and Section VI concludes the paper with finaldiscussions.

II. BACKGROUND STUDY

Alavi, S. M. M., & Saif, M. [1] The process of data fusion is one of the most effective strategies for reducing the dimension of the dataset while simultaneously improving overall device efficiency. Since the data sensed is periodic and has redundancy and reduced energy and storage space, it becomes an essential challenge to do best with these limited resources. The proposed model employs a mixture of the Kalman Filter and the ELM (Extreme learning machine) to detect flawed data against regular data based on the data pattern. To begin, the Kalman filter reduces the total datasets into a series of patterns, i.e., instead of broad data sets, it offers a collection of regular and defective data patterns. Second, this collection of data is used to train the ELM model in order to achieve high classification accuracy. The proposed model outperforms previous models in terms of precision and processing time.

A fault tolerant localization algorithm known as clustering-based DV-Hop is defined in [3] Bhat, S. J., & Santhosh, K. V.. To weed out flawed nodes, this algorithm employs basic K-means clustering and voting techniques. In localization, only non-faulty nodes are included.

Liu, S. et al. [5] describes a data aggregation approach for homogeneous WSNs that combines fault node self- checking and data fusion. A new weighting approach based on statistics theory is proposed, and the weighting coefficient is dependent on the divergence between thecalculated meaning.

Long, H. et al. [6] describes the dynamic consensus expanded Kalman filter, a sensor control technique for dynamic goal monitoring. The goal is initially positioned using weighted least squares. Each move is then tracked to establish a complex monitoring cluster, with cluster participants collaborating to conduct target detection and distributed state estimation. The algorithm performs state estimation for nonlinear structures using the consensus filter of a distributed extended Kalman filter.

S. Arjunan etal [7].But, in clustering method, a CH bears additional workload i.e., getting sensed information transmitted by data dissemination, member sensors, and data aggregation to the BS .

Wang Su, & Yang Bo. [10] The easy and efficient object identification, position, and tracking algorithms are developed to meet the battlefield detection environment's requirements, completely using multi-sensor data fusion. In the laboratory, the identification, position, and monitoring of a man, a shooter, and a vehicle are accomplished using a small-scale network. The experimental results indicate that the device can achieve a high identification rate and accurate target position and monitoring trajectories in a specific detection environment.

III. SYSTEM MODEL

When all of the network's nodes have been deployed, they must organise themselves into clusters before sending the sensed data to BS. The nodes in the distance-based distributed design transfer data over shorter distances, so the energy expended is much lower than the energy used to transmit directly to the BS. Some network parameters are considered in order to group the nodes into clusters:

Node grouping-phase I: This segment addresses the proposed method for clustering randomly deployed nodes based on inter-node size. Using this algorithm, nodes with inter-node distances less than the optimum distance are clustered into a single cluster. Depending on the node density of the WSN, can be varied to maintain the number of nodes per cluster (m) and hence the number of clusters

(k) at an optimum. A large number of nodes can be positioned within a short distance in a random deployment. In this scenario, the scale of the cluster is determined by the number of nodes in the clusters, which might or may not be the same, because certain clusters may have a greater number of nodes in size) than others. This condition results in the creation of clusters of varying sizes and the overloading of the cluster head of large density clusters. The traffic caused by these clusters causes the network to become unbalanced and becomes a hindrance to routing. The amount of energy consumed by intra-cluster processing ranges in direct relation to the number of nodes in the cluster. In a heavily deployed WSN, may be reduced to restrict the m, which raises k. In a sparsely deployed WSN, may be increased to have m ideal, which reduces k. The proposed algorithm limits the number of nodes in a cluster to m, and the data rate and usable bandwidth determine its value. Thus, the interval between nodes and cluster size are the two parameters used as algorithm termination conditions, and it shapes distributed and traffic balanced clusters.

Step II of cluster head selection: For resource- constrained and fault-prone WSN, the cluster head election algorithm should require low communication overhead and message complexity in terms of time and messages. The algorithm ensures that only one cluster head is active at any given moment, and non-cluster nodes must be aware of their CH. Energy usage is assumed to be evenly distributed by spontaneously spinning the cluster-head. In order to prolong the cluster's lifespan, the re-election of a new CH selection process must be performed after the crash loss of a current CH. The algorithms that choose the CH at random must look for the fittest nodes around the entire network, which increases the cluster setup time and energy as a function of the number of nodes in the network, while the suggested algorithm chooses a node inside the cluster as the CH, which greatly reduces time and energy and improves scalability.

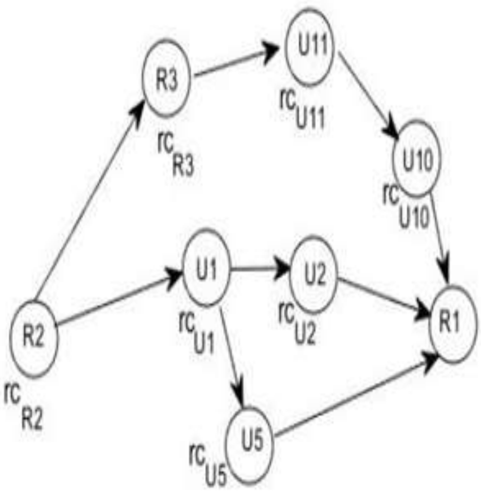

Figure 2: Multipath Communication

Data aggregation-phase III: When a node is elected as a CH, the notification is broadcast to all nodes in the cluster. The non-CH nodes then submit their sensed data to the CH during the data aggregation process. It establishes a Fault traffic pattern and distributes time slotsamong cluster participants to prevent a collision at CH. CH aggregates the data and sends it to the BS at the end ofeach round. The third function of the CFTL algorithm is fault tolerance, which is achieved in a straightforward way in the proposed framework. The health parameter measurement in each cluster updates the node-id, and a nominee list consisting of nodes with greater node-ids outside the actual CH node-id is maintained. As a CH fails, a new node with the highest node-id is elected as CH and restores the cluster by sending its own data to the BS for the current round.

FAULT FILTERING ALGORITHM

To enhance the localization accuracy in erroneous situations, we have developed a fault filtering This algorithm assumes that the cumulative number of defective nodes is less than L=2. First, the distance between nodes is increased by calculating the distance as a function of node contact radius. The average communication distance (ACD) is a feature of node communication size..

Algorithm: Fault Filtering Algorithm Input: List of CFTL values, consistency threshold. Output: Consistent CFTL values.

```
1.  While (ACDmax - ACDmin) > consistency threshold.
2.  Sort CFTL values.
3.  Jenks Step1:
4.  Find ACDmean, which is mean of CFTL
5.  Find sum of squared deviation SSD as shown in (3)
6.  Jenks Step2:
7.  For i D 1 to (number of CFTL values - 1)
8.  Classify CFTL into two groups ; (1 V ACDi) and (ACDiC1 V
ACDend )
9.  Find class mean for Group1 and Group2, ACDGmean1and ACDGmean2
10. Find SCDi from (4)
11. Find GF(i) from (5)
12. End for
13. The value with highest GF is the best classi_cation and I is the classi_cation index
14. Retain cluster with majority of values
15. End While
```

If the strongest cluster node is node it sends data to the base station and completes the data routing operation. Otherwise, it sends the data to the best candidate among its neighbours, who then repeats the routing procedure. This procedure is repeated until a node decides it is the best candidate and sends directly to the base station. Since the best candidate will have an indirect direction that is safer than direct delivery, this localised decision-making mechanism results in a monotonic reduction in energy costs over time..

IV. DISCUSSIONS

Model of Faults

WSNs are vulnerable to a broad variety of processing and networking faults. WSNs are often disrupted due to cheap machinery, scarce services, and harsh environmental factors [1]. To obtain the optimal tolerance, faults must be identified appropriately. Our fault classification is dependent on data transfer protocols' capacity to withstand the consequences of these faults [10]. We classify all faults observed during data transport as either unacceptable or tolerable.

Formalized paraphrase Unacceptable Flaws

The consequences of intolerable errors are those that data transfer protocols cannot address. WSNs may be used in harsh conditions to identify fires and monitor citizens in disaster zones. These conditions have the potential to permanently kill nodes on a massive scale or the whole WSN, which is clearly unmanageable. Other unacceptable flaws include sink crash loss and network partitioning. The sink is crucial since it serves as a connection between the consumer and the WSN. Consequently, if the sink fails, the network would be unable to connect with the customer, resulting in an unacceptable error. Since source nodes and sink nodes can belong to separate network partitions, network partitioning is regarded as an unacceptable flaw. If the WSN can be maintained, these unacceptable defects can be turned into tolerable ones.

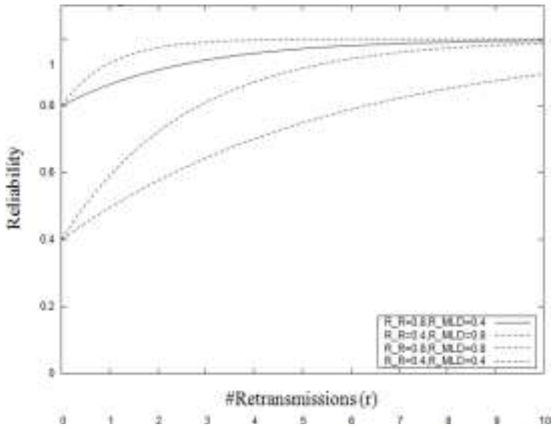

Figure 3: Retransmission during Fault occurrence

Communication Errors: The most common failures in the WSN are communication failures. Message loss and longer message delays are examples of data transport failures. These errors have a strong effect on the WSN's responsiveness.

Node Failures: Node failures trigger a shift in network topology, which can affect the WSN's responsiveness. Nodes can often begin to misbehave. The event detection performance suffers as a result of these errors.

V. CONCLUSION

To achieve fault tolerance in wireless sensor networks, the cluster forming process is an acceptable option (WSN). This implements a used CFTL scheme that discovers the linked neighbour sets and starts the clustering phase based on connectivity. Each node determines its energy usage to determine the most energy-efficient route for data transmission from node to node. The suggested method has many benefits. To shape clusters, for example, all that is needed is awareness of one-hop neighbours. Furthermore, the clustering scheme is resistant to topological changes induced by node collapse, mobility, CH transition, and node addition or removal.

REFERENCES

[1] Alavi, S. M. M., & Saif, M. (2013). Fault Detection in Nonlinear Stable Systems Over Lossy Networks. IEEE Transactions on Control Systems Technology, 21(6), 2129–2142. doi:10.1109/tcst.2012.2236839

[2] Biswas, P., Charitha, R., Gavel, S., & Raghuvanshi, A. S. (2019). Fault Detection using hybrid of KF-ELM for Wireless Sensor Networks. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). doi:10.1109/icoei.2019.8862687

[3] Bhat, S. J., & Santhosh, K. V. (2020). Fault Tolerant Localization Based on K-means Clustering in Wireless Sensor Networks. 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). doi:10.1109/conecct50063.2020.9198415

[4] Shaikh, F. K., Khelil, A., & Suri, N. (2007). On Modeling the Reliability of Data Transport in Wireless Sensor Networks. 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing (PDP’07). doi:10.1109/pdp.2007.55

[5] Liu, S., Tan, Z., Fan, X., Long, H., Li, Y., & Liu, L. (2012). Fault-tolerant data aggregating using median filtering in cluster-based homogeneous WSN. Proceedings of the 10th World Congress on Intelligent Control and Automation. doi:10.1109/wcica.2012.6359229

[6] Long, H., Fan, X., & Liu, S. (2012). Nonlinear Target Tracking Schedule Based on Dynamic Consensus for WSN. 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control. doi:10.1109/imccc.2012.210

[7] S. Arjunan etal.F5N-based unequal clustering protocol (F5NUCP) for wireless sensor networksInt. J. Commun. Syst.(2018)

[8] Tsang-Yi Wang, Li-Yuan Chang, & Pei-Yin Chen. (2009). A Collaborative sensor-fault detection scheme for robust distributed estimation in sensor networks. IEEE Transactions on Communications, 57(10), 3045–3058. doi:10.1109/tcomm.2009.10.080244

[9] Vieira, T. P., Almeida, P. E. M., & Meireles, M. R. G. (2017). Intelligent fault management system for wireless sensor networks with reduction of power consumption. 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE). doi:10.1109/isie.2017.8001471

[10] Famila et al.Improved artificial bee colony optimization based clustering algorithm for SMART sensor environmentsPeer Peer Netw. Appl.(2020)