

## Research Article

# An Efficient Nodes Failure Recovery Management Algorithm for Mobile Sensor Networks

Rab Nawaz Jadoon <sup>1,2</sup>, Adnan Anwar Awan,<sup>3</sup> Muhammad Amir Khan <sup>3</sup>,  
WuYang Zhou <sup>1</sup> and Aamir Shahzad<sup>3</sup>

<sup>1</sup>Key Laboratory of Wireless-Optical Communication, University of Science and Technology China, Hefei 230027, China

<sup>2</sup>Department of Computer Science, COMSATS University, Islamabad, Abbottabad Campus 22060, Pakistan

<sup>3</sup>Department of Electrical and Computer Engineering, COMSATS University, Islamabad, Abbottabad Campus, Pakistan

Correspondence should be addressed to Rab Nawaz Jadoon; [rabnawaz@mail.ustc.edu.cn](mailto:rabnawaz@mail.ustc.edu.cn) and WuYang Zhou; [wyzhou@ustc.edu.cn](mailto:wyzhou@ustc.edu.cn)

Received 17 March 2020; Revised 25 June 2020; Accepted 3 August 2020; Published 19 August 2020

Guest Editor: Hou-Sheng Su

Copyright © 2020 Rab Nawaz Jadoon et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks are not prone to harsh environments and may fail due to various reasons. Failure of sensor nodes causes partitioning of network into various small segments and restricts the communication of nodes. Due to the significant importance of restoration mechanisms, many approaches have been proposed in the literature so far. However, these approaches do not focus on uniform distribution of sensor nodes before the occurrence of failure. This paper fulfills the shortcoming in the literature by proposing a Uniform Distribution and Recovery Algorithm (UDRA) in two parts. The first part (prefailure algorithm) focuses on preparing the mobile sensor nodes to be ready for the failure beforehand by maintaining half of their communication distance between them. Also, it uses a novel method of directional matrix based on one-hop information. By using this method, each mobile node declares itself as cut-vertex (CV), intermediate node, or leaf node. The second part of the algorithm (postfailure algorithm) gives complete recovery procedure in the network by its recovery nodes. The extensive simulations prove that the proposed algorithm supersedes the existing approaches.

## 1. Introduction

Wireless sensor networks (WSNs) are an active area of interest for the researchers and engineering communities. A WSN is typically a network of distributed sensor nodes collecting data from the environment to monitor the physical conditions of that environment. The distributed nodes are of two types, namely, stationary nodes and mobile nodes. Stationary nodes remain stationary and perform sensing activities, routing of data, or act as sink nodes, whereas mobile nodes move around the network to perform sensing and different tasks like replacing failed nodes and may be equipped with different equipment such as fire extinguisher and guns to shoot enemies. These mobile nodes are known as actor nodes and such type of network is called Wireless Sensor and Actor Networks (WSANs). Actors have many capabilities, such as processing of sensed data,

decision making, and performing suitable actions as mentioned above. Robotic Mule [1] is an example of such an actor designed for military use for detection of mines in a battlefield. These distributed nodes are connected to a sink node, which is further connected to the wired world for the usage of the data.

The data provided by the sensor nodes may be used for monitoring the area to make suitable decisions and strategies. For example, in agriculture, this data may be used to make strategies for controlling temperature for different crops or in military decisions this data may be used to make movements of troops and other military equipment according to the enemy strength along the border. These kinds of networks are most suitable for harsh environments in which awareness of the surroundings is critical without compromising human lives. WSNs are very effective for many applications in important domains such as security

systems, surveillance, battlefield reconnaissance, space exploration, search and rescue, and deep sea exploration [2]. The performance of WSNs totally depends upon the fact that the sensor nodes remain connected to the sink node for sending data, as there may be some nodes which are positioned far away from the sink node and need many hops through their neighboring nodes to transmit their data [3, 4] to sink node. Another important fact is that the sensor nodes must cover maximum part of the monitored area for the maximum awareness of the environment.

Achieving connectivity and coverage in an integrated manner is practically a challenging task. One of the challenges is the design of sensor nodes as sensor nodes are small, having limited resources for processing and storage. In order to maintain WSN, mainly batteries are used as the central source of power for these sensor nodes [5], as the batteries used have limited power and get drained with the passage of time resulting in failure of nodes. Another challenging issue is that, in harsh environment such as a battlefield, the sensor nodes may be destroyed by the explosives. Failure of a node or multiple nodes may affect the connectivity and may cause partitioning of the network into multiple disjoint segments, which restricts many nodes from transmitting their data to the sink node. This may also cause connectivity and coverage holes in the area, which compromises monitoring process of the environment. Ineffective monitoring of the environment may cause harm to the entire network or fail to achieve objectives of monitoring because an unattended danger might be present in the coverage hole. Hence, restoration of connectivity and coverage for the network is of utmost importance.

A solution to this problem is to deploy fresh nodes replacing the dead ones which is a costly and sometimes infeasible process. The restoration process must be self-healing, self-organized, and fault tolerant [6] which makes sensor nodes useful in scenarios with minimum or no human interaction. Generally, sensor nodes are deployed in a random manner which creates a highly connected network in some areas and leaves the other areas partially connected. As coverage area may be compromised in such situations, sensor nodes must be capable of adjusting themselves in such a way that their distribution in the area becomes uniform and maximum coverage is attained. Usually, the distribution is supervised by the sink node which can process data and make decisions, but this method increases the overall communication overhead as the decisions must pass the entire network in form of messages and incur delay. Therefore, it is highly recommended that nodes must arrange themselves to decrease the overall messages in the network.

Uniform distribution of actor nodes has already been discussed in [7], but this approach does not give the exact distance by which each node must separate itself from the neighboring nodes, which results in two possible scenarios. First is the isolation of a single node or multiple actor nodes from the entire network. The second situation is the isolation of a network or multiple networks from the rest of network. These two scenarios are not suitable for mission critical networks in a harsh environment like a battlefield and can

cause disconnection of many nodes. To avoid these two scenarios, nodes must be cognizant of the exact distance by which they have to be separated from their neighbor nodes to achieve uniform distribution of nodes in the entire network. Based on our experimentation, we suggest that the distance should be half of the transmission range and nodes must be aware of this fact that they should maintain half of the distance of their communication range. For experimentation, we assume that all nodes have the same sensing and communication range. It is also assumed that transmission range and sensing ranges are also equal. All nodes are deployed in a random manner. Physical constraints are also taken into account in this paper, which are often not considered in many techniques in the literature so far. This paper proposes a new mechanism for the nodes to declare themselves as cut-vertex, intermediate, or leaf node by calculating point-to-point coverage area in all directions, i.e., east, west, north, south, north-east, north-west, south-east, and south-west; details are given later in this paper. After declaration, neighboring nodes are declared as recovery nodes, which are responsible for connectivity in case of failure. The main contributions of this paper are as follows:

- (1) UDRA focuses on proactive approach and stresses more on the pre-failure operations rather than post-failure operations and is most suitable for mission critical applications which need very less time for connectivity and coverage restoration.
- (2) UDRA also considers the physical constraints, which are often not taken into account, hence giving a practical solution.
- (3) UDRA gives the exact distance between two nodes for distribution in area and all nodes try to maintain this distance. This gives advantage that these nodes stay connected in case of failure of any node without movement of such nodes as described earlier.
- (4) UDRA also gives the complete recovery plan, in case of a failure of single node, if some nodes or multiple nodes cannot maintain a distance  $R_c/2$  between their neighbors due to physical constraints.
- (5) UDRA also gives a very simple solution to find cut-vertexes, intermediate, and leaf node by one hop information by node itself.

The rest of the paper is organized as follows. Section 2 gives the comparison of the proposed algorithm with the related work done in the past. Section 3 has the details of the proposed algorithm along with the pseudocode. Validation of results is reported in Section 4, and Section 5 concludes the paper.

## 2. Related Work

Preserving connectivity while maintaining overall coverage intact by using limited number of mobile nodes is a difficult task, especially when a single or multiple nodes fail. It is the problem of utmost importance which has been studied in the literature in recent years. Various solutions have been proposed regarding this issue. There are two kinds of

solutions. The first kind of solutions depends upon relocation on demand, whereas the second kind of solutions is based on postdeployment relocation. In general, sensor nodes are deployed randomly by aerial deployment. This kind of deployment may cause nonuniform distribution of mobile nodes in the AOI; i.e., some areas may have higher density and some areas may have lower density of mobile nodes. Therefore, relocation is desired in this situation so that the connectivity is assured between mobile nodes and the end user to maximize the coverage area. Connectivity issue has been targeted in the literature and many solutions have been provided in this regard [7–13]. To solve connectivity issue, coverage holes must be avoided in some areas. Therefore, a balance between connectivity and coverage restoration is important to keep both intact. Several approaches have been used in recent years.

In [8], the authors proposed a distributed algorithm for the network connectivity restoration, when a node is failed. For the restoration of network connectivity and improvement of coverage, the technique uses mobile nodes at the time of failure. A concept of 2-connected network was introduced in [13]. This means that among each pair of nodes there must exist at least two connected pathways in the network. This approach has 2-degree connectivity and requires 2-hop information for each node. In [7], coverage-aware and connectivity constrained actor positioning (C2AP) algorithm is proposed. In this approach, post-deployment of nodes is discussed. In [14], a cascade movement of nodes was given as a solution when some nodes fail. According to this technique, the neighboring node of the failed node replaces it. Then, such node is replaced by another node and replacements continue until being replaced by redundant node.

In [9], there is an algorithm known as Recovery by Invert Motion (RIM). In RIM, the recovery of the failed node was done by its neighboring nodes by invert movement towards the failed node. The main idea is that, through missing of HEART BEAT message, neighbors will know about the node failure. Therefore, neighbors make inward movements to the failed node for restoration of the network connectivity.

Prefailure cut-vertex nodes determination is done by Partition Detection and Recovery Algorithm (PADRA), which is presented in [10]. Failure of cut-vertex nodes causes disjoints in the network. The suitable neighbor is selected for each cut-vertex node to handle its failure. Hence, in this way, prefailure connectivity and coverage are restored. The connectivity restoration process involves relocations of all the involved neighboring nodes, which causes cascaded movements of such neighbors.

In [11], coverage-aware connectivity restoration (C3R) in mobile sensor networks technique gives the solution by replacing the failed nodes by their neighboring nodes. The solution suggests the to and fro movement of the neighboring nodes between the position of failed nodes and their original position. The schedule of changing turns of these nodes was made by the node which reaches the failed node first.

To restore the connectivity of the network, an algorithm known as Volunteer-instigated Connectivity Restoration

(VCR) is proposed in [12]. In this algorithm only, the immediate neighbors of the failed actor node take part. The selection criterion of the neighboring nodes is based on nearness. The cascaded movement is limited in this algorithm.

A provision of connected path planned by controlling a robot is described in [13]. Minimizing distance with minimal number of hops counts for mobile robots in such a way that connectivity remains intact which is the supreme goal of this algorithm. The proposed approach contains two algorithms. For allocation of a robot to an event location, the nearest robot to the event is identified by the first algorithm. Then, the nearest nonconnected robot is searched by this algorithm. The nearest robot is then directed to bring itself forward to connect to any of the joined section with allocated robot. This procedure is performed until the network becomes completely connected. For minimization of the hop count between the base station and the event area, the second algorithm is used. It optimally provides such locations where there are minimum numbers of hops.

However, a Backbone Polygon (BP) is built around the center of the disjoint network area in Connectivity Restoration with Assured Fault Tolerance (CRAFT) algorithm presented in [14]. To connect each outer partition to the BP, by two nonoverlapping paths, Relay Nodes (RNs) are positioned in the targeted area having low cost, resulting in the restoration of the connectivity of the network.

In [15], the authors give an exceptional connectivity restoration technique called Geometric Skeleton based Reconnection (GSR). GSR uses a geometrical skeleton based approach to partition network logically into different segments. A geometrical skeleton backbone consists of a group of nodes that have the maximum connectivity with other nodes. Each segment has record of all the skeletal backbone nodes, so, in case of network partitioning, each segment attempts to join the geometrical skeleton backbone. In this way, connectivity can be restored.

To observe the influence of realistic terrain, hybrid recovery strategy based on random terrain (HRSRT) is proposed in [16]. The Area of Interest (AOI) is divided into a grid of cells. All cells have equal size. Mapping of these cells is used to plan the terrain. The weight " $\omega(c)$ " represents each cell  $c$ . The weight of each cell " $\omega(c)$ " can be calculated by the summation of weight of each path " $\omega(P)$ ." A complete graph  $K_n$  is created by considering the minimum weights of paths between segments. A tour " $T$ " for MDCs (mobile data collectors) is done by a random terrain-based path planning algorithm, which is established on  $K_n$  for initialization of connectivity restoration. The weight of tour " $\omega(T)$ " is directly proportional to the cost of data collection and aggregation. There may be one or more than one MDCs. This results in different relay nodes deployment approaches. So, for connectivity restoration, an "optimized relay node deployment" and "relay nodes deployment" are responsible. Minimal energy and cost for data aggregation and collection are the goals of these algorithms.

In [17], Survivability-Aware Connectivity Restoration (SACR) is given. Data load levels of disjoint parts are considered for connection of the partitioned segments

between different disconnected segments. A set of moveable nodes are located in the proper location of these inaccessible segments. A relay partition is generated for every isolated segment, immediately after isolated segments are traced. This causes restoration of connectivity from the AOI to the base station.

Another technique for calculating cut-vertex is presented in [18]. For this purpose, this technique uses two localized and distributed algorithms. For the detection of cut-vertex and noncut-vertex nodes, a local subgraph having two hops with connected dominating set (CDS) knowledge is used in the first algorithm. The second proposed algorithm depends entirely on the first one. For unrecognized parts of the network, a limited distributed depth-first search algorithm is used without searching the whole network. This algorithm discovers the states of all nodes by comprehensive test bed experiments with the presence of a CDS. Cut-vertex nodes are detected with lower energy consumption. Simulations results are the witness of this fact.

A method for maintaining confidentiality of location of roaming position-based services (PBSs) users using machine learning techniques is suggested in [19]. For roaming PBS users, the authors suggest a triphase process. It identifies user position by merging decision trees and  $k$ -nearest neighbor and estimates user destination using hidden Markov models along with position track sequence. In addition, the proposed model follows a mobile edge computing service strategy which will ensure the timely delivery of PBSs. The advantages of a mobile edge service policy offer confidentiality of position and low latency through networking and computing services in the vicinity of roaming users.

In [20], an energy-conscious green adversary paradigm for its use in intelligent industrial environments by maintaining confidentiality is proposed. Although the hardware and software parts of cyber physical systems can be mutually improved to reduce their energy consumption, this paper focuses on aspects of position conservation and confidentiality of the information. Based on the literature findings (assumptions, adversary targets, and capabilities), the authors include some testimonials to support cyber physical security practitioners and researchers. The proposed model is running on real-time, anticipatory position-based query scheduling to reduce the cost of communication and computation for each query, thereby encouraging minimization of energy consumption. In addition, to avoid degrading slots, we measure the transfer/acceptance slots that are needed for each question. The experimental findings indicate that, in comparison with current methods, the proposed solution will reduce energy consumption by up to five times.

The key objective explained in [21] is to extend the lifespan of the network by extending the lifespan of the working sensors as well as moving the collected data from the super node into the sink. Bat Algorithm (BA) is used in this article to pick the optimal monitoring sensor node and corresponding direction to minimize energy consumption. Results of simulation and comparison with other algorithms suggest the superiority of the algorithm proposed. The simulation results of the proposed algorithm show that the

proposed algorithm was able to reduce the network's power consumption and increase the network's service life. The proposed algorithm may also outperform the comparable algorithms by 27 per cent on average.

In [22], the whale optimization algorithm (WOA) is used in this paper to solve the resource allocation (RA) problem in Internet of things (IoT) with the goal of optimum RA and to reduce the overall cost of communication between resources and gateways. There is a comparison of the proposed algorithm with other current algorithms. Results suggest that the proposed algorithm works properly. The proposed process, based on different metrics, is better than others in terms of "absolute cost of communication."

The solution to this problem has three optimization objectives. They are as follows:

- (a) Coverage restoration
- (b) Connectivity restoration
- (c) Both coverage and connectivity in an integrated manner

In order to achieve the above optimization objectives, three approaches are used.

- (a) Reactive approach: in this approach, a recovery mechanism is developed after the occurrence of failure. References [9–11] are the examples of such approach.
- (b) Proactive approach: in this approach, recovery mechanism is established before the occurrence of failure. It means that sensor nodes are prepared well before the failure. References [7, 18, 23] are the examples of such approach.
- (c) Hybrid approach: in this approach, both reactive and proactive approaches are used. This approach is used according to the requirement. Yihanli, Shivendra, 2006 [10, 24], etc. are the examples of such approach.

Table 1 shows optimization objective and nodes' mobility and either algorithm is reactive, proactive, or hybrid and shortcomings of the techniques presented in the literature since 2007.

The proposed (UDRA) gives the solution to achieve connectivity restoration objective by using hybrid approach. The extensive simulations show that the proposed algorithm outperforms its baseline algorithms. Table 2 gives the details of optimization objectives, approach, and strengths of the proposed algorithm.

### 3. Details of the Proposed Algorithm

**3.1. The Scenario and Assumptions of UDRA.** The proposed algorithm considers all sensor nodes as mobile sensor nodes, deployed randomly in an Area of Interest (AOI). It is assumed that all nodes have capability of sensing the environment to find the physical constraints. This gives them the knowledge of their capability for the decision of maintaining  $R_c/2$  distance with the neighbors. This can be useful for their availability to become recovery node for failure later. It is also assumed that, at the time of startup, all nodes discover

TABLE 1: Overview of related work algorithms.

Reference	Optimization objective	Nodes' mobility	Proactive, reactive, or hybrid	Shortcomings
C2AP [7]	Connectivity and coverage restoration	Actor nodes	Proactive	No exact distance b/w sensor nodes for maintaining coverage and connectivity is given
DCRA [8]	Connectivity restoration	Actor nodes	Hybrid	Cut-vertexes are determined by depth-first search trees (DFS) and CDS which are time consuming
RIM [9]	Connectivity restoration	Mobile sensors	Reactive	Cascade movements of sensor nodes consume a lot of energy
PADRA [10]	Connectivity restoration	Actor nodes	Hybrid	Exact distance of distribution is not given
C3R [11]	Connectivity and coverage restoration	Mobile sensors	Reactive	This requires a lot of movements of the nodes, which reduces network lifetime
VCR [12]	Connectivity restoration	Robots	Reactive	This takes a lot of time for convergence
DEEHC [13]	Connectivity restoration	Robots	Proactive	There is lack of multirobot communication and cooperation
CRAFT [14]	Connectivity restoration	Mobile sensors	Reactive	Detection of orphan nodes in multimode failure is not addressed
GSR [15]	Connectivity restoration	Mobile sensors	Reactive	The assumptions made in this technique are not practical which can increase overhead of network
HSRT [16]	Connectivity restoration	No mobility (replacement by relay nodes)	Reactive	No solution for coverage replacement of relay nodes may not be feasible in hostile environment
SACR [17]	Connectivity restoration and network lifetime	Mobile sensors	Hybrid	There is no solution for multinodes and simultaneous nodes failures
CDS [25]	Connectivity restoration	No mobility	Proactive	This does not present any plan of recovery in case of node failure
DCHS-WS [26]	Coverage restoration	Mobile sensors	Reactive	Particle swarm algorithm (PSA) is used which takes a lot of time
MMLPP [27]	Connectivity restoration	Mobile vehicles	Proactive	Path query algorithm needs a lot of time
AUR [24]	Connectivity restoration	Mobile sensors	Hybrid	CDS is complex to implement
TEEN [23]	Scalability and network lifetime	Mobile sensors	Proactive	There is no solution for coverage and connectivity

TABLE 2: Overview of the proposed algorithm.

The proposed technique	Optimization objective	Nodes mobility	Proactive, reactive, or hybrid	Strengths
UDRA	Connectivity restoration	Mobile sensors	Hybrid	(i) Having less time for connectivity restoration. (ii) Giving the exact distance between two nodes for distribution in area. (iii) Giving complete recovery plan. (iv) Giving very simple solution to determine cut-vertex, intermediate, and leaf nodes.

each other to form a network [8]. The proposed algorithm assumes that mobile nodes can determine their location relevant to their neighbors [9]. Each mobile node makes a list of its first hop neighbors, which reduces the messaging overhead in the network as compared to having two-hop neighbor information. This paper mainly focuses on proactive approach for the mobile nodes in such a way that it maintains connectivity as well as coverage. If the mobile nodes are not prepared before the failure, it will take some time to react, hence sacrificing the connectivity and coverage for a certain amount of time. URDA gives permanent

relocation of mobile nodes in case of failure which assures both connectivity and coverage without wasting a lot of time.

**3.2. Maintaining  $R_c/2$  Distance Distribution.** As mentioned in Section 1, the mobile nodes are deployed in random manner in some AOI. This may lead to a situation when there might be an abundance of mobile nodes in one area and shortage of mobile nodes in some other areas. Therefore, all mobile nodes must be uniformly distributed throughout AOI. To maintain uniform distribution, this paper suggests

that these mobile nodes should try to maintain  $R_c/2$  distance between them. If all the neighbors of a mobile node succeed in maintaining  $R_c/2$  distance, then there are two advantages of this distance. The first is that all the mobile nodes will be uniformly distributed in AOI, as mentioned earlier. The second advantage is that, in case of some node failure, there is no need for the neighbors of the failure node to move towards that failure node and restore coverage and connectivity. The reason is that neighbors will stay connected to the immediate neighboring nodes of the failed node. The first advantage is obvious, as all nodes have the same distance between them, so the mobile nodes will be distributed in uniform manner. However, the second advantage can be explained by the example depicted in Figure 1. In the example shown in Figure 1, mobile nodes A, B, C, D, E, F, and G are successful to maintain  $R_c/2$  distance between them. Half of the distance is taken by bisecting the distance between them by using dashed line segments just in case of failure of nodes D or F. Their neighboring nodes C and A will be stayed connected with nodes E and G, respectively. Their overlap coverage area, shown by shaded areas, remains intact. So, there is no need of movement for mobile nodes C and A for the restoration of connectivity and coverage. This saves extra movement in case of failure, hence saving extra energy spending on such movements, as the distance between the mobile nodes is well known by the localization methods as given in [9, 10]. The relationship between coverage area and distance is inversely proportional; i.e., the coverage area will be more when the distance between the nodes is less. The coverage area will increase with the decrease in distance. In Figure 2, it is depicted that there is the overlap area between nodes N1 and N2.  $R_c$  is their communication range, whereas  $\alpha$  and  $\beta$  are the angles, which helps in solving the area under arc "pq" with centers N1 and N2. So, the area of the arc "pq" when center is N1 can be calculated by the equations given in [15]

$$\text{area} = \left(\frac{\beta}{\pi} R_c^2\right) - \left(\frac{d}{2} R_c\right) \times \left(\sin\left(\frac{\beta}{2}\right)\right). \quad (1)$$

Equation (1) can also be used to find out the area under the same arc "pq" with the center N2. Hence, the overall coverage area can be found by the equation [15]

$$\text{overlap} = 2 \times \frac{\text{area}}{\pi R_c^2}. \quad (2)$$

**3.3. Determining Cut-Vertex, Intermediate, and Leaf Nodes.** For determining cut-vertex (CV), intermediate node (IN), and leaf node (LN), each mobile node is in a network. A novel method of directional matrix is proposed. All of the mobile nodes try to maintain  $R_c/2$  distance between them. If all nodes maintain such a distance, then there will be no cut-vertices. But still there is a chance that there might be some nodes which cannot maintain  $R_c/2$  distance due to some physical constraints. In this case, the concerned node maintains a matrix showing the directions of its neighbors by their relative positions and the compass mounted on each

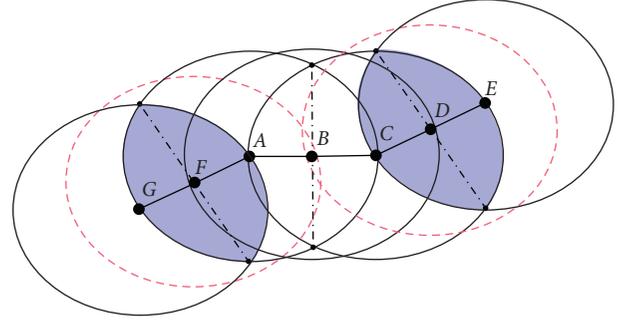


FIGURE 1: A and E will be still connected with G and C, respectively, if F or D fails.

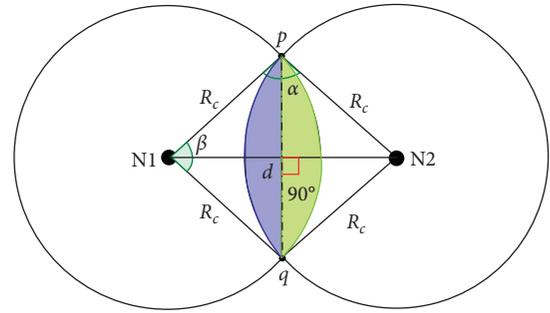


FIGURE 2: Overlap area between N1 and N2.

node. By this matrix, a node declares itself as cut-vertex, immediate, or leaf node. This matrix is based on consisting of three rows and three columns as shown in

$$A = \begin{bmatrix} a11 & a21 & a31 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}. \quad (3)$$

Here, in this matrix, the position of the concerned node is in the middle of the matrix and is given by element a22, whereas the element a23 shows number of one-hop nodes in east direction. The element a21 shows number of one-hop nodes in west direction. The element a21 shows number of one-hop nodes in north direction. The element a32 shows number of one-hop nodes in south direction. The element a11 shows number of one-hop nodes in north-west direction. The element a31 shows number of one-hop nodes north-east direction. The element a31 shows number of one-hop nodes in south-west direction and the element a33 shows number of one-hop nodes in south-east direction. Directions are mapped on matrix according to directions shown in Figure 3.

$$A = \begin{bmatrix} \text{nw} & \text{N} & \text{ne} \\ \text{W} & 1 & \text{E} \\ \text{sw} & \text{S} & \text{se} \end{bmatrix}. \quad (4)$$

Initially, matrix A has the following values:

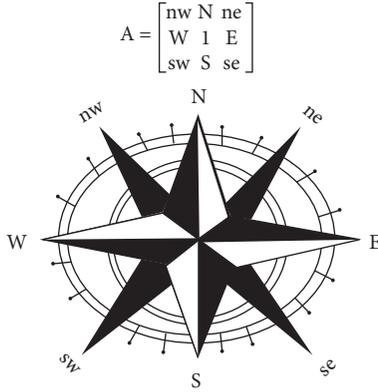


FIGURE 3: Directions of mobile nodes.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (5)$$

i.e., 1 for the position of the concerned node and 0 for all other directions. As far as the concerned node finds its neighbors by overlap area, it finds their direction by its navigational compass and by their relative positions. The concerned node increments the number of first hop nodes in their respective directions. The cases of determining cut-vertex, intermediate, and leaf node can be best explained with the help of Figure 4. In Figure 4, consider node A; it has a node let's say K, which cannot maintain the distance of  $R_c/2$  with A. Node A has B in east, K in north, E at south, F at south east, and C at south west, so the directional matrix for A will be

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \quad (6)$$

Node A will broadcast the message with the IDs of its neighbors that it has B, C, E, F, and K with it at its first hop. Do they have any of these connected with them? So, all of the neighbors receive this message, and answer back to A with "NO CONNECTION" message, if none of them are connected to each other. So, A makes its value 0 in the matrix and checks for connections of its neighbors:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \quad (7)$$

It will find that none is connected without it, so it declares itself as cut-vertex and nominates its neighbors as its recovery nodes for the time of failure. Similarly, K has A node which cannot maintain  $R_c/2$  with K, so the K directional matrix will be

$$K = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (8)$$

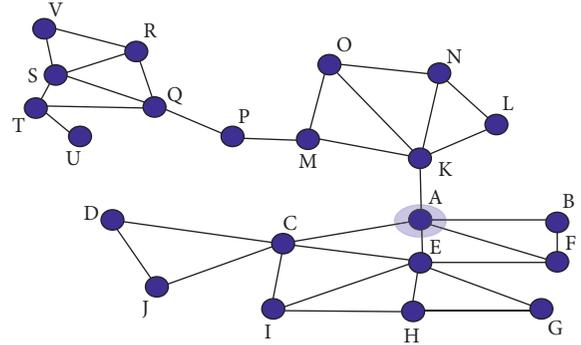


FIGURE 4: Scenario for determining CV, IN, and LF.

As K has two nodes in north-east, i.e., N and L and two nodes in north-west, O and M. So, node K checks its neighbor connections by setting its value to zero:

$$K = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (9)$$

As N relates to L and O and O is connected with M and there is node A which lacks the connection with these four nodes, so node K declares itself as cut-vertex. Similarly, node E will make directional matrix as

$$E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \quad (10)$$

As there is connection between H to G and F to A, so, E will declare itself as intermediate node. For the leaf node, examples are nodes D, V, N, and O. Leaf nodes can be found if there is triangle of zeros above, below, or both in directional matrices:

$$\begin{aligned} D &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \\ V &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \\ N &= \begin{bmatrix} 0 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \\ Q &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (11)$$

Similarly, all the nodes in the networks declare themselves as CV, IN, or LF. There might be a chance that there will be an IN and it declares itself as CV. After declaration,

the concerned nodes assign their neighbors as recovery nodes and are prepared for the node failure situation.

### 3.4. UDR Algorithm

**3.4.1. Prefailure UDR Algorithm.** As mentioned in Section 1, nodes are deployed in random manner. So, there is a chance of profusion of nodes in one area and lack of nodes in the other. This rises to a coverage problem as all the area is not attended in uniform manner. Uniform distribution of nodes can be done by the sink node, taking instructions from the end user. But this type of strategy is not suitable as overall messaging overhead of the network increases, because sink node must transmit its messages to all the network nodes and must wait for their acknowledgment. So, nodes must rearrange themselves, without adding extra messaging overhead to the network. So, first, a node after deployment tries to find its neighbors by coverage overlap of their  $R_c$ . Then, each node looks around its environment to find out any physical constraints. If it finds any unavoidable obstacle, it just sets CM (cannot move) flag, by sending message to its neighbors along with its ID. Then, that node calculates directional matrix to find out either it is cut-vertex (CV), intermediate node (IN), or leaf node (LF) which is already discussed in detail in previous section. The neighbors update their neighboring list by putting CM and CV or IN or LF with that node's ID. If there are no unavoidable obstacles, then a node compares its ID with neighboring node IDs. If its ID is greater than all other IDs, it sends message "distribution admin" (DA) along with its ID. After this, DA calculates each neighboring node distance, which may be less than, equal to, or more than  $R_c/2$ . If all neighboring distances are equal to the required distance, then prefailure algorithm ends. So, there is no need to move to the failed node's location. If all distances are less than  $R_c/2$ , then DA asks its neighbors to move away from the DA to maintain the mandatory distance in all directions. If all distances are greater than the required distance, then DA asks its neighbors to move towards the DA to maintain such distance. There may a situation. When some nodes may be at distance equal to  $R_c/2$ , some nodes may be at distance less than  $R_c/2$ , and some have distance more than  $R_c/2$  from DA, so, in such situation, DA broadcasts a single message having three parts: the first part of the message starts with "start," then node IDs of those which have distances equal to  $R_c/2$ , and ends at "Eq." Then, the second part of message starts with IDs of those nodes having distance less than  $R_c/2$  and ends with "less" and the third part contains IDs of those nodes which have distance with DA more than  $R_c/2$ . After this, there is a footer of "more" and its ends with "end" which means end of a message. This message is broadcast by DA and all neighboring nodes read this message looking for their ID and responding as accordingly. If a node has distance equal to  $R_c/2$ , then it stays there and simply discards the message. If node falls in the category of having distance less than  $R_c/2$ , so it moves away from DA to make it  $R_c/2$  and sends "done" message to DA. If distance from DA is greater than  $R_c/2$ , then that node calculates possible movement

distance towards the distribution admin (DA) node by calculating overlap coverage area and how much it can move towards admin node by finding the least coverage area with its neighbors. After calculating possible to move distance, the node moves to that possible distance and sends "done" message to DA. When all the neighbors adjust their distances, DA updates distance list and finds DM for node failure situation. However, there might be a chance of a node having more than one DA, so such node will find possible distance with the DA. It selects the DA with lesser possible distance with itself. There might be another situation that there are two or more admins in the neighborhood in this situation; none of DAs moves towards each other, as their other neighboring nodes require moving towards them. Detailed steps are explained in the proposed prefailure pseudocode in Algorithm 1.

**3.4.2. The Proposed Postfailure Algorithm.** Mobile nodes in a network become uniformly distributed in AOI with the help of prefailure algorithm, to guarantee connectivity and coverage. There might be some nodes which cannot maintain  $R_c/2$  with their neighbors, so recovery nodes are assigned to them. Now, for such nodes, which cannot maintain  $R_c/2$ , a problem occurs when some nodes fail in the network creating connectivity and coverage hole behind. To resolve this issue, a proposed node failure algorithm is presented in this paper. When a node finds one of its neighboring nodes has failed, for example, consider a scenario given in Figure 4. Suppose that node A finds node F which has failed. Node A will have the status of failed node F in its list that either F is a cut-vertex (CV), intermediate node (IN), or leaf node (LN). So, node A will react to this situation according to the status of failed node F. If failed node F is CV, then it has at least four recovery nodes. If it has four or two recovery nodes, then all of them move their possible movement distance to restore connectivity. If they still not find each other, then they will increase their communication range and send message to sink node for redundant node as there is no other possibility to restore both coverage and connectivity. If there is only one node for as recovery node for CV, then it will move its possible distance to restore connectivity; if recovery nodes still do not find any node to restore connectivity, then they increase their range and send message to sink node for redundant node. If failed node F is IN or LN, then the same procedure will be adopted; for IN, two recovery nodes or only one recovery node will move their possible distance or increase their range. For LN, only one recovery node will follow the above procedure. Pseudocode of this is given in Algorithm 2.

**3.5. Energy Model.** An energy model which we have used in this paper is illustrated in [25]. Equations (12) and (13) show the relationship of required energy for transmitting and receiving a B-bit data packet. Energy per bit consumption of receiver circuitry is given by (13). For more details, the readers are recommended to read details in [25]

Input: AOI and  $R_c$

- (1) Begin
- (2) Find ( $N_i$ )// $i = 1, 2, 3, \dots$  number of neighbors
- (3) Check for PC = physical constraint
- (4) If PC = true
- (5) Send (owned ID, CM)//CM = cannot move
- (6) Go to (step 48)
- (7) End if
- (8) If receive (some IDs, CM)
- (9) Update (some IDs, CM)//in neighbor's list
- (10) End if
- (11) If PC = false
- (12) Compare (owned ID with  $N_i$ 's IDs)
- (13) End if
- (14) If owned ID > all  $N_i$ 's IDs
- (15) Send (owned ID, distribution admin) to rest of neighbors
- (16) Calculate (distance with each neighbor)
- (17) If all neighbors' distances =  $R_c/2$
- (18) End if
- (19) If distance <  $R_c/2$
- (20) Send (move away)//to maintain  $R_c/2$  distance
- (21) End if
- (22) End if
- (23) Else
- (24) Send (calculate possible movement distance)//to those neighbors whose distance is >  $R_c/2$
- (25) Wait
- (26) Receive (possible movement distances of all neighboring nodes)
- (27) Calculate (actual distance—possible distance to move)//to find out which node can maintain  $R_c/2$
- (28) Send (adjust, distance to move)
- (29) End if
- (30) If receive (move away)
- (31) Move//to maintain  $R_c/2$
- (32) Else if receive (calculate possible movement distance)
- (33) If distribution admin > 1
- (34) Send (CM)//to distribution admins
- (35) Else
- (36) Calculate (possible movement distance)//keeping in mind  $d \leq 2 \times R_c$  and by averaging possible movement distance of each neighboring node
- (37) Send (adjustable distance)//to distribution admin
- (38) End if
- (39) End
- (40) Else if receive (adjust, distance to move)
- (41) Move//to adjust that  $R_c/2$
- (42) End if
- (43) Update all neighboring node distances ( $d_i$ )
- (44) If all  $d_i \neq R_c/2$
- (45) Go to (step 36)
- (46) End if
- (47) Else if all  $d_i = R_c/2$
- (48) Determine directional matrix = DM
- (49) End if
- (50) End
- (51) Determine DM
- (52) If DM = LN//leaf node
- (53) Go to (step 66)
- (54) Else if DM = IN//intermediate node
- (55) Go to (step 64)
- (56) Else if DM = CV//CV = cut-vertex
- (57) Check availability of four recovery nodes
- (58) If availability = false
- (59) Go to (step 64)

```

(60) If availability = true
(61) Set at least four neighboring nodes as recovery nodes
(62) End if
(63) End
(64) Set at least two neighboring nodes as recovery nodes
(65) If availability of two nodes = false
(66) Set at least one neighboring node as recovery node
(67) End if
(68) End

```

ALGORITHM 1: Prefailure algorithm.

```

Input: DM = CV or DM = IN or DM = LF
(1) If (node A detects, node F has failed)
(2) If distance =  $R_c/2$ 
(3) Find (new neighbors)
(4) If (new node found)
(5) Restore (connectivity)
(6) Else if (new node not found)
(7) Move (possible movement distance)
(8) Else if (new node not found)
(9) Go to (step 39)
(10) Restore (connectivity)
(11) Else if distance  $\neq R_c/2$ 
(12) Go to (step 18)
(13) End if
(14) End if
(15) End if
(16) End
(17) Else
(18) If DM = CV
(19) Move (towards node F by possible movement distance)
(20) Find (other recovery nodes)
(21) If new node found
(22) Restore (connectivity)
(23) Else if (new node not found)
(24) Go to (step 39)
(25) End if
(26) Else if DM = IN
(27) Move (towards node F by possible movement distance)
(28) Find (other recovery nodes)
(29) If new node found
(30) Restore (connectivity)
(31) Else if new node not found
(32) Go to (step 39)
(33) End if
(34) Else if DM = LF
(35) Move (towards node F by possible movement distance)
(36) Find (new node)
(37) Restore (connectivity)
(38) If new node not found
(39) Increase  $R_c$ 
(40) Send (redundant node)//to sink node
(41) End if
(42) End

```

ALGORITHM 2: Postfailure algorithm.

$$E_{Tx}(B, d) = \begin{cases} (E_{elec} + \varepsilon_{fs}d^2)B, & d < d_0, \\ (E_{elec} + \varepsilon_{mp}d^4)B, & d \geq d_0, \end{cases} \quad (12)$$

$$E_{Rx}(B) = E_{Rx-elec}B, \quad (13)$$

$$E_{reng}(n) = E_{max} - E_{Tx}(B, d) - E_{Rx}(B). \quad (14)$$

Different terms used in this model are explained in Table 3.

#### 4. Experimentation and Results Discussion

In order to validate the effectiveness and better performance of UDRA, UDRA is compared with other baseline approaches already present in the literature. The protocols of geometric skeleton based reconnection (GSR) approach [15], autonomous repair (AUR) [24], and UDRA have been simulated in OMNET++ platform. For the comparison of these three protocols, the performance parameters of total distance covered after relocation, average number of nodes moving during relocation, percentage reduction in field coverage, and number of packets exchanged have been set in the simulation for protocol evaluation.

**4.1. Simulation Setting.** This section discusses the simulation setup, performance metrics, and results of the simulations. All the simulation parameters are summarized in Table 4.

UDRA and other protocols of the category have been implemented and evaluated in OMNET++, INET simulation framework. Simulation parameters and their values are defined in Table 4.

Simulations for each algorithm are conducted in OMNET++ separately and simulation logs are imported in MATLAB for result presentation. All the simulation results are taken within 10% of simple mean, whereas the confidence analysis interval is taken as 90% for all simulations. To evaluate the performance of the proposed algorithm against baseline algorithms, the following metrics are used:

Total distance moved reports the total distance collectively travelled by the nodes involved during recovery. This may be envisaged as a network-wide assessment of the efficacy of the recovery schemes applied.

Number of nodes relocated records the number of mobile nodes moved during recovery. This metric measures the breadth of the network connection restoration process.

Number of messages exchanged tracks the total number of messages exchanged between nodes. This measure measures the overhead relating to contact imposed by the recovery process.

range of communication ( $R_c$ ), all experiment nodes have the same range of communication. Initial WSN topology is affected by the value of  $r$ . While small  $r$  creates a sparse topology, and a large  $R_c$  boosts network connectivity overall, very few nodes need to be involved

in the cascaded movement in a highly connected WSN and the restoration process will converge rapidly with little overhead; a large value of  $R_c$  would increase the number of neighbors of a failed node and thus increase the overhead incurred by these neighbors.

Figure 5 presents the total distance by movement for relocation versus increase in number of nodes. From the results, UDRA performs well in comparison to the other baseline algorithms because during its operation, only recovery nodes are moved. Hence, the cascaded relocations are minimized. As a result, the average distance moved for UDRA is less as compared to the other considered algorithms. UDRA performs consistently well as compared to other considered algorithms as the number of nodes in the network has increased. The major reason behind this is that the movements of the nodes which cause further partitioning of the network are restricted by UDRA. One fascinating thing to note from the above figure is that the performance of GSR, unlike other algorithms, improves with increase in number of nodes. The major reason behind increase in performance is the working methodology of GSR. As the number of nodes increases, the backbone of the network becomes stronger and stronger, resulting in making recovery mechanism be more proficient and stable and causing improved performance. Besides this, the assumptions taken by GSR are unrealistic and, in case of topology change, they cause enormous overhead in terms of packet exchanges. It can still be seen from Figure 5 that our protocol still performs reasonably well as compared to GSR.

The average number of nodes moved versus increase in the total number of nodes is present in Figure 6. A smaller number of nodes moved as compared to AUR and GSR algorithms can be observed from the figure. The avoidance of continuous cascaded relocations of nodes is the main reason behind this phenomenon. Furthermore, it can also be observed from Figure 6 that, as the number of nodes increases, number of nodes moved in UDRA increases less than other considered algorithms. This shows that UDRA is more scalable, when compared to other considered protocols.

The effect of recovery process on the coverage is shown in Figure 7. As the nodes are relocated because of failure of nodes, a reduction in the coverage area is observed. For all the considered protocols, it can be realized from Figure 7 that when there is decrease in the percentage field coverage, the communication range increases. However, percentage reduction in the field coverage of UDRA is least as compared to the GSR and AUR protocols. It is obvious from Figure 7, that the field coverage reduction under UDRA is much lesser as compared to the baseline algorithms. Among the considered protocols, GSR results in the most field coverage reduction. Cascaded relocation of nodes is the main reason behind this increase of field coverage reduction. Similarly, for the other considered algorithms, cascaded relocation is the main reason for more reduction in field coverage. In UDRA, more appropriate recovery nodes are chosen to take part in recovery process. These nodes have not moved at all or moved a smaller distance, resulting in higher overlapped coverage of these recovery nodes.

TABLE 3: Meaning of different terms in the energy model.

Term	Definition
$E_{Tx}$	Energy consumed to transmit data
$E_{Rx}$	Energy consumed to receive data
$E_{reng}$	Residual energy
$E_{elec}$	Energy consumed per bit by the transmitter circuitry
$E_{max}$	Initial energy of sensor nodes
$E_{Rx-elec}$	Energy consumed per bit of a receiver
$\epsilon_{fs}$	Energy required by radio frequency (RF) amplifier in free space
$\epsilon_{mp}$	Energy required by radio frequency (RF) amplifier in multipath
$d_0$	Threshold distance

TABLE 4: Simulation parameters.

Simulation parameters	Values		
Area of simulation	1000 × 1000 m <sup>2</sup>		
No. of nodes	50–250		
Size of data packet	900 bits		
$E_{elec}$	70 nanojoules/bits		
$E_{max}$	25 J		
$\epsilon_{fs}$	15 picojoules/bits/m <sup>2</sup>		
$\epsilon_{mp}$	0.0012 picojoules/bits/m <sup>2</sup>		
$d_0$	70 m		
$R_c$	25–150 m	Simulation tool	OMNET++
Simulation tool	OMNET++		

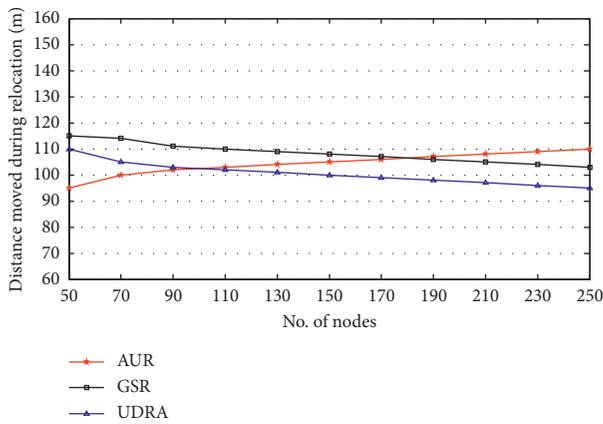


FIGURE 5: No. of nodes versus distance moved.

In Figure 8, the average number of packets exchanged during the procedure of connectivity restoration is shown. UDRA results in the minimum number of packets exchanged, while the maximum number of packets is exchanged by GSR. The major reason behind this is that, in UDRA, the emphasis is on minimizing the number of neighbors involved during the recovery process, which results in reduction of cascaded movement of nodes, hence reducing the total number of messages exchanged. On the other hand, due to excessive movement of nodes, GSR results in the maximum number of packets exchanged. So, it can be concluded that UDRA is more energy efficient as compared to AUR and GSR algorithms. Another important feature is the scalability associated with the reduction of the

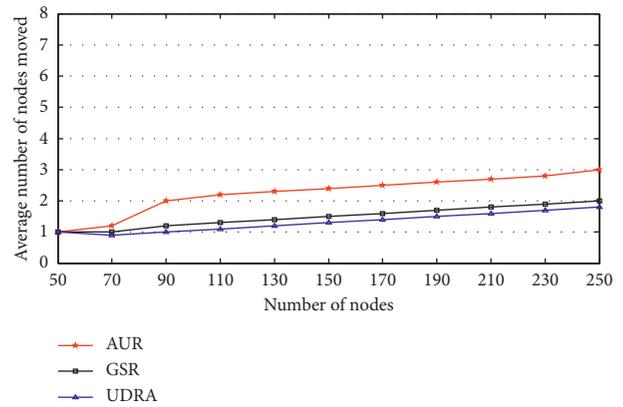


FIGURE 6: Average no. of nodes moved during recovery.

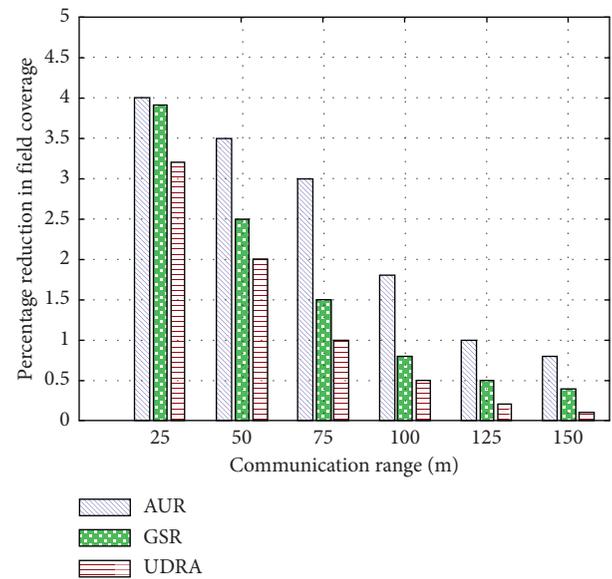


FIGURE 7: Percentage reduction in field coverage.

transmission/receiving of messages. Due to least number of messages exchanged by UDRA, it proves to be more scalable as compared to AUR and GSR protocols. In sensor networks, connectivity restoration is of immense importance and a technique must be capable of restoring the connectivity. During this work, our major emphasis was to design a

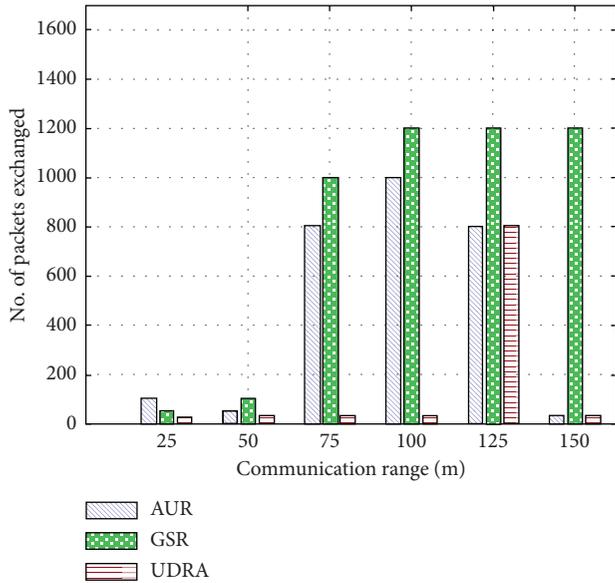


FIGURE 8: Total number of exchanged packets.

TABLE 5: Results summary.

Comparison of protocols for 250 nodes	Average distance travelled (m)	No. of messages exchanged (average)	No. of nodes relocated (average)	Age reduction in field coverage % (average)
UDRA	3000	700	70	03
RIR	13000	3000	200	10
AUR	15000	5500	700	15

prefailure and postfailure connectivity restoration protocol capable of achieving effective connectivity restoration by relying on minimal node mobility, minimal reduction in field coverage, and consuming minimal energy of nodes by reducing the overall number of exchanged messages. The solution proposed by UDRA achieves all the above goals. The performance of the UDRA was evaluated by using extensive simulations and by comparing its performance with AUR and a GSR technique; its effectiveness is proven.

### 5. Conclusions and Future Work

A novel algorithm “Uniform Distribution and Recovery Algorithm (UDRA)” is proposed in this paper. This algorithm provides both connectivity and coverage restoration solution for wireless sensor networks in an integrated manner. In Table 5, it is clear that average distance travelled by the sensor nodes is only 3000 meters. This distance is very less than 13000 meters and 15000 meters. So, having less average distance travelled proves that the proposed algorithm is an energy efficient solution. Most of the energy stored in batteries is consumed by travelling. If an algorithm is not energy efficient, then batteries of nodes will be drained off with the passage of time. This makes the problem worst and decreases the network lifetime. Furthermore, these results show that the number of messages exchanged by the

proposed algorithm is very less as compared to RIR and AUR. This decreases the message overhead in the network. Greater message overhead causes delay in recovery process as it takes a lot of time by communication messages. Another finding from Table 5 is that in the proposed technique only 70 nodes are relocated; this gives an upper hand to the proposed algorithm over RIR and AUR as more relocation of nodes needs more energy used. The other advantage of a smaller number of relocated nodes is that relocation of nodes can cause coverage loss. So, the lesser relocated nodes are, the lesser the coverage loss will be. The last most important finding of Table 5 is the percentage reduction in the field coverage. In UDRA, the percentage in field coverage reduction is only 3% after recovery process. This shows that UDRA outperforms its competitor baseline algorithm. Less percentage in field coverage reduction ensures that most of the network or AOI is monitored and less information from AOI is lost. The cost of these advantages would be that increased number of nodes is required to cover certain AOI, which is still affordable for mission critical applications. The detailed analysis would be done in the future.

### Data Availability

No data were used to support this study. We have conducted the simulations to evaluate the performance of the proposed protocol. However, any query about the research conducted in this paper is highly appreciated and can be asked from the principal authors (Rab Nawaz Jadoon (rabnawaz@mail.ustc.edu.cn) and Muhammad Amir Khan (amirkhan@cuiatd.edu.pk)) upon request.

### Conflicts of Interest

The authors declare that there are no conflicts of interest.

### Acknowledgments

Dr. Rab Nawaz Jadoon personally thanks COMSATS University, Islamabad, Abbottabad Campus, and School of Information Science and technology, USTC Hefei, China, that extended their full support by providing all key resources during the implementation and all afterward phases of this project. Also, special thanks are due to Prof. WuYang Zhou, because of his continuous encouragement and massive support academically, financially, and socially during this research work. This work was financially supported by the National Natural Science Foundation of China (Grant no. 61631018).

### References

- [1] R. N. Jadoon, W. Zhou, I. A. Khan, M. A. Khan, and W. Jadoon, “EEHRT: energy efficient technique for handling redundant traffic in zone-based routing for wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 7502140, 12 pages, 2019.
- [2] S. Lee, M. Younis, and M. Lee, “Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance,” *Ad Hoc Networks*, vol. 24, pp. 1–19, 2015.

- [3] V. Ranga, M. Dave, and A. K. Verma, "Node stability aware energy efficient single node failure recovery approach for WSANs," *Malaysian Journal of Computer Science*, vol. 29, no. 2, pp. 106–123, 2016.
- [4] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619–632, 2012.
- [5] R. Jadoon, W. Zhou, W. Jadoon, and I. Ahmed Khan, "RARZ: ring-zone based routing protocol for wireless sensor networks," *Applied Sciences*, vol. 8, no. 7, p. 1023, 2018.
- [6] R. N. Jadoon, W. Zhou, I. Ahmed Khan, M. Amir Khan, S. Akhtar Abid, and N. Ali Khan, "Performance evaluation of zone-based routing with hierarchical routing in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 7152858, 10 pages, 2019.
- [7] K. Akkaya and M. Younis, "C2AP: coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks," in *Proceedings of the 2007 IEEE International Performance, Computing, and Communications Conference*, pp. 281–288, IEEE, New Orleans, LA, USA, April 2007.
- [8] A. Abbasi, K. Akkaya, and M. Younis, "A distributed connectivity restoration algorithm in wireless sensor and actor networks," in *Proceedings of the 32nd Conference on Local Computer Networks*, Dublin, Ireland, October 2007.
- [9] M. Younis, S. Lee, S. Gupta, and K. Fisher, "A localized self-healing algorithm for networks of moveable sensor nodes," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'08)*, New Orleans, LA, USA, November 2008.
- [10] K. Akkaya, A. Thimrnapuram, F. Senel, and S. Uludag, "Distributed recovery of actor failures in wireless sensor and actor networks," in *Proceedings of the IEEE 2008 IEEE Wireless Communications and Networking Conference*, pp. 2480–2485, Las Vegas, NV, USA, April 2008.
- [11] N. Tamboli and M. Younis, "Coverage-aware connectivity restoration in mobile sensor networks," *Journal of Network and Computer Applications*, vol. 33, no. 4, pp. 363–374, 2010.
- [12] M. Imran, M. Younis, A. Md Said, and H. Hasbullah, "Volunteer-instigated connectivity restoration algorithm for wireless sensor and actor networks," in *Proceedings of the 2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, pp. 679–683, Beijing, China, June 2010.
- [13] P. Chanak, I. Banerjee, and R. S. Sherratt, "Energy-aware distributed routing algorithm to tolerate network failure in wireless sensor networks," *Ad Hoc Networks*, vol. 56, pp. 158–172, 2017.
- [14] Y. K. Joshi and M. Younis, "Exploiting skeletonization to restore connectivity in a wireless sensor network," *Computer Communications*, vol. 75, pp. 97–107, 2016.
- [15] X. Wang, L. Xu, S. Zhou, and W. Wu, "Hybrid recovery strategy based on random terrain in wireless sensor networks," *Scientific Programming*, vol. 2017, Article ID 5807289, 19 pages, 2017.
- [16] X. Liu, "Survivability-aware connectivity restoration for partitioned wireless sensor networks," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2444–2447, 2017.
- [17] A. Wichmann, T. Korkmaz, and A. S. Tosun, "Robot control strategies for task allocation with connectivity constraints in wireless sensor and robot networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1429–1441, 2018.
- [18] O. Dagdeviren, V. K. Akram, and B. Tavli, "Design and evaluation of algorithms for energy efficient and complete determination of critical nodes for wireless sensor network reliability," *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 280–290, 2019.
- [19] Y. Zeng, C. J. Sreenan, N. Xiong, L. T. Yang, and J. H. Park, "Connectivity and coverage maintenance in wireless sensor networks," *The Journal of Supercomputing*, vol. 52, no. 1, pp. 23–46, 2010.
- [20] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.
- [21] A. K. Sangaiah, D. Vishwasrao Medhane, G.-B. Bian, G. Ahmed, M. Alrashoud, and M. Shamim Hossain, "Energy-aware green adversary model for cyberphysical security in industrial system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3322–3329, 2020.
- [22] A. K. Sangaiah, M. Sadeghilalimi, A. A. R. Hosseinabadi, and W. Zhang, "Energy consumption in point-coverage wireless sensor networks via bat algorithm," *IEEE Access*, vol. 7, pp. 180258–180269, 2019.
- [23] A. Abro, Z. Deng, K. Ali Memon et al., "Minimizing energy expenditures using genetic algorithm for scalability and longevity of multi hop sensor networks," in *Proceedings of the 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 183–187, IEEE, Beijing, China, July 2019.
- [24] Y. Zhang, J. Wang, and H. Guan, "An autonomous connectivity restoration algorithm based on finite state machine for wireless sensor-actor networks," *Sensors*, vol. 18, no. 1, p. 153, 2018.
- [25] H. Essam, M. Younis, and E. Shaaban, "Minimum cost flow solution for tolerating multiple node failures in wireless sensor networks," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC)*, pp. 6475–6480, London, UK, June 2015.
- [26] O. Moh'd Alia, "Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm," *Information Sciences*, vol. 385–386, pp. 76–95, 2017.
- [27] Q. A. Arain, I. Memon, Z. Deng, M. H. Memon, F. A. Mangi, and A. Zubedi, "Location monitoring approach: multiple mix-zones with location privacy protection based on traffic flow over road networks," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 5563–5607, 2018.