

An Efficient Connectivity Restoration Technique (ECRT) for Wireless Sensor Network

Mahmood ul Hassan^{1,*}, Shahzad Ali², Khalid Mahmood³, Muhammad Kashif Saeed⁴, Amin Al-Awady¹, Kamran Javed⁵ and Ansar Munir Shah⁶

¹Department of Computer Skills, Preparatory year, Najran University, Najran, 66241, Saudi Arabia

²Department of Computer Science, Jouf University, Tubarjal, 74769, Saudi Arabia

³Department of Information Systems, College of Science and Arts, King Khalid University, Mahayil Asir, 63317, Saudi Arabia

⁴Department of Computer Science, Community College Mahayil Asir, King Khalid University, Mahayil, 63718, Saudi Arabia

⁵Department of Computer Engineering, National University of Technology (NUTECH), Islamabad, 44000, Pakistan

⁶Department of Computer Science, Institute of Southern Punjab, Multan, 60000, Pakistan

*Corresponding Author: Mahmood ul Hassan. Email: mahmood.mscs@gmail.com

Received: 03 March 2021; Accepted: 08 April 2021

Abstract: Node failure in Wireless Sensor Networks (WSNs) is a fundamental problem because WSNs operate in hostile environments. The failure of nodes leads to network partitioning that may compromise the basic operation of the sensor network. To deal with such situations, a rapid recovery mechanism is required for restoring inter-node connectivity. Due to the immense importance and need for a recovery mechanism, several different approaches are proposed in the literature. However, the proposed approaches have shortcomings because they do not focus on energy-efficient operation and coverage-aware mechanisms while performing connectivity restoration. Moreover, most of these approaches rely on the excessive mobility of nodes for restoration connectivity that affects both coverage and energy consumption. This paper proposes a novel technique called ECRT (Efficient Connectivity Restoration Technique). This technique is capable of restoring connectivity due to single and multiple node failures. ECRT achieves energy efficiency by transmitting a minimal number of control packets. It is also coverage-aware as it relocates minimal nodes while trying to restore connectivity. With the help of extensive simulations, it is proven that ECRT is effective in connectivity restoration for single and multiple node failures. Results also show that ECRT exchanges a much smaller number of packets than other techniques. Moreover, it also yields the least reduction in field coverage, proving its versatility for connectivity restoration.

Keywords: Cut-vertex; failure recovery; network connectivity; node relocation; wireless sensor network; node failures



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Wireless Sensor Networks (WSNs) consist of various sensor nodes dispersed for monitoring different phenomena. Each node in a sensor network can be equipped with a diverse range of sensors capable of monitoring a wide variety of phenomena. For some applications like wildfire detection or deep-ocean exploration, sensor nodes deployed in a region stay unattended for a long period without any human intervention [1]. For this reason, sensor networks should have self-healing, self-organization, and fault tolerance capabilities for successful operation [2].

Other than the environmental challenges, one of the major challenges in sensor networks originated from the design and characteristics of the sensor node. Generally, sensor nodes are quite small in size and have limited resources in processing power and memory. Sensor nodes use a battery as a central power source [3]. One or more sensors can be attached to a sensor node for measuring different phenomena from the environment. Sensor nodes detect desired phenomena in the environment and can transmit this information to other nodes via radio transceivers.

In general, sensor nodes are often deployed in harsh environments where human access is restricted; therefore, the replacement of node batteries is not possible. After deploying nodes into a network of sensors, those nodes communicate with each other to establish a network. This network coordinates the sensor nodes and transmits the detected information to the end-user [3]. As a sensor network covers a fairly large area, not all sensor nodes are within the transmission range of the end-user. Nodes must rely on multi-hop communications for sending information toward the end user [4–6]. The reliability of the collected information can be increased by increasing the node density in an area. During the period, battery exhaustion of the sensor nodes occurs as a result of message transmission and information processing.

When the battery of a node is fully depleted, this sensor node is considered a dead node. As network nodes begin to disappear during the operation of a sensor network, connectivity holes appear within the network. A connectivity hole is referred to as an area where the nodes are no longer connected. Connectivity holes lead to loss of inter-node connectivity, leading to the inability to send the end-users sensed information. The first step is to detect a failure if a node in the network has failed/died to address such a situation.

Once such a failed node is identified, the next step is to notify the nodes adjacent to the failed node about repositioning. The disconnected nodes become connected again [7]. However, it is possible for network nodes to die at any time due to battery exhaustion or physical damage due to difficult environmental factors.

A failed node may disrupt network connectivity, and the network may develop multiple isolated segments or partitions. This scenario results in a failure of the flow of information between the sensor nodes and the end-user terminal. This scenario may compromise the basic operation of the sensor network and to prevent this, network connectivity should be restored. One potential connectivity solution is the deployment of redundant nodes in place of dead nodes. However, this process is slow and often impracticable, particularly in environments where nodes are more susceptible to physical damage. The connectivity restoration process should be self-organized using existing living nodes within the network. The recovery mechanism needs to be robust, and the overhead on the sensor nodes involved in the recovery process should be minimal because of the resource constraints.

Failure of nodes in a sensor network results in different types of connectivity problems. This work categorized the connectivity problems into four different cases:

1) cut-vertex failure, 2) end node failure, 3) two cut-vertex node failures, and 4) multiple end node failures. Details of such cases are provided in Section 3.

Dealing with the above cases is a challenging task. This paper proposes a novel connectivity restoration technique called An Efficient Connectivity Restoration Technique (ECRT) to handle the issues associated with network connectivity. Our work's major focus is to propose a technique capable of restoring connectivity using the existing nodes in the network; therefore, there is no need to re-deploy new nodes for restoring connectivity. Moreover, ECRT does not substantially reduce the network coverage as observed by the other connectivity restoration techniques. As energy efficiency is one of the major concerns in sensor networks; therefore, ECRT does not rely on the exchange of large amounts of messages for its operation. ECRT can also detect all the identified four connectivity problems and effectively find the solution for each case.

The rest of the paper is organized as follows. In Section 2, we describe the most relevant related work. Section 3 consists of the research method used for our research work, and we also elaborate on the four different cases related to connectivity restoration. Section 4 presents the simulation results, and subsequently, Section 5 concludes this paper.

2 Related Work

Connectivity restoration in wireless sensor networks is an area that has been thoroughly studied by researchers [8–11]. Some solutions are based on relocation on-demand, while other solutions rely on post-deployment relocation. The applications requiring sensor nodes to be deployed over large geographic areas use aerial deployment of the nodes [12]. Due to this, node density throughout is not uniform, and some areas may have a higher density of nodes than some other areas. To achieve a uniform distribution of nodes, relocation of sensor nodes is desired so that connectivity can be established between the sensor nodes and the end-user and coverage area can be maximized.

The connectivity aspect is thoroughly studied in the literature, and several approaches are presented [8–12]. Some works are more focused on maximizing the coverage of nodes without affecting connectivity. In [13], a distributed algorithm for the restoration of network connectivity in node failure is proposed. This algorithm relies on the mobility of nodes for restoring connectivity in case of failure of a node, and the algorithm also strives to improve the coverage of the area. In [14], the authors considered robot networks, and a technique similar to [13] is introduced for the process of connectivity restoration. A 2-connected network concept is introduced, meaning that there should exist a minimum of two pathways among each pair of nodes in the network. This approach achieves 2-degree connectivity. For dealing with a node's failure, the algorithm strives to achieve 2-connectivity by moving a pair of sensor nodes. In this way, connectivity is restored. In [15], the authors proposed a technique called C²AP. In this technique, post-deployment of nodes is used for improving coverage and connectivity. A hierarchical architecture is proposed by the authors called COCOLA in [16], where coverage is maximized without forwarding data path to 1-tier node by the incremental relocation of higher-tier nodes. However, both the proposed solutions, C²AP and COCOLA, are incapable of dealing with the failed nodes' implications.

In [17], a solution based on node's cascaded movement is introduced for connectivity restoration due to failed nodes. According to this technique, a nearby node replaces a failed node, which is then replaced by another node, and this process continues until finding a redundant node. C³R (Coverage Conscious Connectivity Restoration) [18], RIM (Recovery through Inward

Motion) [19], and AUR (Autonomous Repair) [20] are techniques that have a similar focus as our work. C³R addresses connectivity restoration in case of one or more failed nodes. Once a failed node is detected, then it is substituted by each neighbor node temporarily for a limited amount of time. After a specified time, the neighbor node gets back to its original position. The other neighbors repeat this process. The main issue with this approach is that it involves much movement of neighboring nodes to tackle a node's failure. It results in a large overhead in energy consumption as sensor node movement consumes a substantial amount of energy [21]. In [13], the authors proposed a new method called DARA. DARA uses a scheme based on probability for detecting cut vertices and selecting an appropriate neighbor node to the failed node for relocation. The appropriateness of the neighbor is decided based on the number of communication links. In [19], the authors proposed another technique called RIM. RIM is based on recovery by the inward motion, and the neighboring nodes of a failed node take part to relocate the failed node temporarily.

RIM requires all nodes in the network to have 1-hop information. Once a failed node is detected, the neighbors of that failed node move inward toward its position. Inward movement is to ensure that the neighbors will stay in the range of each other. The main idea is that as these neighbors are directly affected by the node's failure, therefore, after inward movement, if they can reach each other again, the network connectivity would be restored to its pre-failure state. This relocation procedure is applied recursively for handling any node that gets disconnected due to one of their neighbors' movements. Through extensive simulations, it was proved that RIM is effective for connectivity restoration. However, in the case of a highly dense network, RIM suffers because it tends to move a large number of neighboring nodes, resulting in a substantial increase in the total traveled distance overhead. This also results in high energy usage, decreasing the network lifetime.

In [20], the authors proposed an approach called Autonomous Repair (AuR). It uses a concept similar to electrostatic interaction based on Coulomb's law between modeling connectivity charges among neighboring nodes. For the recovery process, only localized information about the immediate neighboring nodes is used. Neighbors initiate the recovery process by first detecting the failed nodes. Then, the neighbors start to move toward the failed nodes, causing the intra-segment topology to be stretched. If connectivity is still unrestored, the segment is then moved as a block toward the deployment area's center. Moving all segments close to the center increases node density in the center point's vicinity, leading to an increase in the probability of connectivity getting reestablished. Geometric Skeleton based Reconnection (GSR) is proposed in [22]. GSR employs a geometrical skeleton-based approach to logically partition networks into different segments. A group of nodes having maximum connectivity becomes the geometrical skeleton backbone. Each segment keeps track of all skeletal backbone nodes because it plays an important part in network partitioning. In the case of network partitioning, each segment tries to join the geometrical skeleton backbone. This process leads to the restoration of connectivity. However, GSR assumes that each node knows the locations of all other nodes in the network. Second, it is also a pre-requisite that all nodes in the network must be aware of all nodes present in the geometrical skeleton backbone. These assumptions are impractical, particularly for large networks, because keeping all this information in a network with mobile nodes can cause massive overhead. Another problem that may arise during the network's operation is that the skeleton backbone may exhaust its energy soon, decreasing such nodes. After a while, the lack of presence of such nodes may result in compromising the recovery mechanism. All the above-mentioned works do not consider connectivity, coverage, and energy efficiency collectively. Our proposed work can be

distinguished from the above-mentioned works because it addresses connectivity restoration, better coverage, and efficient energy use in an integrated manner.

An energy-efficient technique, Intelligent On-Demand Connectivity Restoration for Wireless Sensor Networks (IDCRWSN) [23], has been presented that efficiently uses the sensor nodes' residual energy. IDCRWSN restores the connectivity through redundant nodes, which are managed by Slave Keeper nodes. The Slave Keeper nodes are managed and controlled by the Master Keeper nodes. The Permanent Relocation Algorithm for Centralized Actor Recovery (PRACAR) and Self-Route Recovery Algorithm (SRRA) [24] addresses the connectivity restoration of failed actor nodes. The PRACAR restores failed actor nodes' connectivity, and SRRA provides an optimal path to the relocated sensor nodes.

Efficient Solution for Connectivity Restoration (ESCR) [25] is an energy-efficient technique. The technique restores the network with an efficient consumption of residual energy and least node movement. Only such node can participate in network restoration, which is near the faulty node and has more energy. It stopped unnecessary cascade movement of the nodes during the restoration process. ESCR consists of two algorithms, i.e., Assigning Backup Nodes (ABN) and Connectivity Restoration Process (CPR) are designed to handle actor node failure and connectivity problems in wireless sensor and actor networks. ESCR compared with other well-known techniques and found better than others. It is evaluated in the environment where sensor nodes are stationary, and only actor nodes can move. Its result can be varied if we consider the mobility of sensor nodes as well.

3 Research Method

We identified different cases that may come up if single or multiple nodes in a network die. Our work's objective is to propose a solution capable of recognizing all of the identified cases and then taking necessary actions to restore connectivity. For our proposed algorithm, we assume that all sensor nodes are randomly deployed in the deployment area. After their deployment, all nodes discover their neighbors by exchanging HELLO beacon messages. For the initial relocation of nodes, the mechanism used in [18] is used. Algorithm 1 presents the steps that are used for initial node relocation. Fig. 1 shows the initial relocation scenario. It is assumed that all network nodes are homogenous and have the same processing and communication capabilities. For each node, it is assumed that the sensing range and communication range are equal. In this paper, we use R_c for representing the sensing and communication range.

Algorithm 1: Initial node relocation for ECRT

N_i : Sensor Nodes, Nbr_j : Neighboring Nodes of N_i , R_c : Communication Range

Input: Area, N_i , R_c

Output: Initial node relocation successfully done

```

1  BEGIN
2  SET  $N_i$ 
3  FOR EACH Node  $N_i$  DO
4  SET  $DS_i \leftarrow$  Distance between the Sink and  $N_i$ 
5  IF ( $DS_i \geq R_c/2$  &&  $DS_i < R_c$ ) THEN
6  MoveTowardsSink( $N_i, DS_i$ ) // Function
7  ELSE IF ( $DS_i \leq R_c/2$ )
```

(Continued)

```

8      //Do not move
9      ELSE
10     //this node is present outside the range of the Sink
11     FOR EACH Neighbouring Node  $Nbr_j$  of  $N_i$ 
12     SET  $DN_i \leftarrow$  Distance between  $N_i$  and  $Nbr_j$ 
13     IF( $DN_i > R_c/2$ )
14     MoveTowardsNeighbor( $Nbr_j, DN_i$ )
15     ELSE
16     //Do not move
17 END
18 MoveTowardsSink( $N_i, DS_i$ )
19 BEGIN
20 While( $DS_i > R_c/2$ )
21 Keep moving towards Sink
22 Update position and notify neighbors via HEARTBEAT message
23 SET  $DS_i \leftarrow$  Distance between  $N_i$  and Sink
24 END
25 MoveTowardsNeighbor( $Nbr_j, DN_i$ )
26 BEGIN
27 While( $DN_i > R_c/2$ )
28 Keep moving towards the node
29 Update position and notify neighbors via HEARTBEAT message
30 SET  $DN_i \leftarrow$  Distance between  $N_i$  and  $Nbr_j$ 
31 END

```

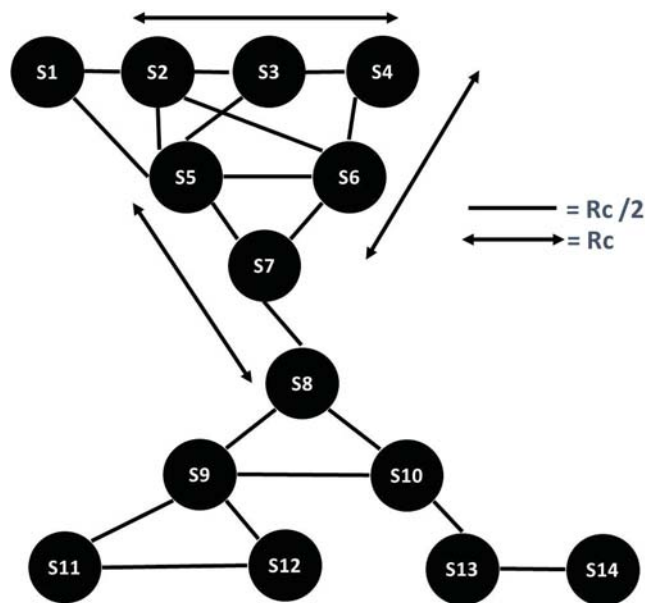


Figure 1: Network topology after initial relocation

Once the nodes are deployed in an area, each node will broadcast HELLO beacon messages with a transmission range of $Rc/2$ for providing its location information to other nodes in the network [18]. Each node share information, including its ID and current position in acknowledgment (ACK), to all the neighboring nodes. The transmission range of $Rc/2$ is used for the transmission of ACK. Also, each node periodically sends a broadcast message for synchronization called SYN message. The transmission range used for transmitting SYN messages is also $Rc/2$. SYN messages are used for the identification of failed nodes. For example, let us consider the scenario presented in Fig. 1. If the node S7 has failed, nodes S5, S6, and S8 will not receive SYN messages from S7. The absence of a SYN message means the failure of a node. Upon detecting a failed node, nodes S5, S6, and S8 will send a HELLO message with a transmission range Rc toward the failed node direction. Upon receiving the HELLO message, in reply, each node transmits an ACK message. In this way, each node updates the list of neighbors, and they initiate mobility to restore the network, as depicted in Fig. 2.

Upon detecting the failure, it is important to understand the impact of failure on the network topology. In this work, we have categorized four scenarios that can occur due to single or multiple node failure.

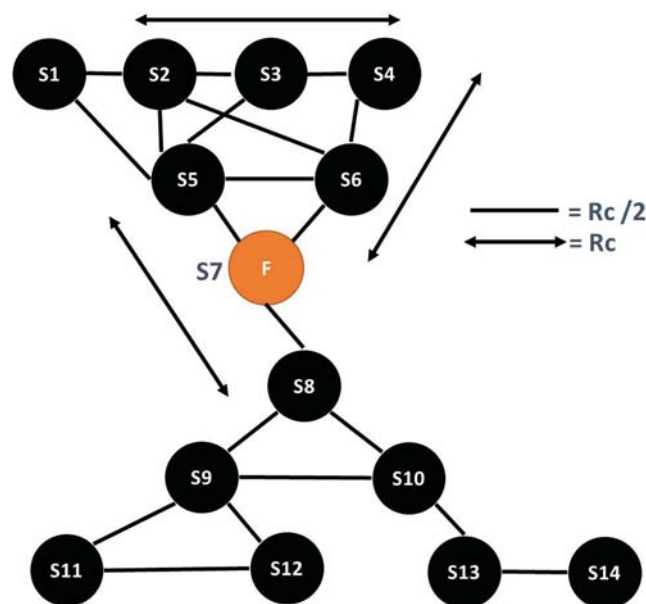


Figure 2: Single cut-vertex failure

3.1 Scenario 1: Single Cut-Vertex Failure

The cut-vertex scenario is illustrated in Fig. 2. This happens when a node's failure divides the connected network into multiple disjoint partitions [21]. In Fig. 2, it can be observed that the failure of node F divides the network into two partitions resulting in cut-vertex failure. Failure of node F is detected by nodes S5, S6, and S8 due to the absence of SYN messages from node F. Algorithm 2 illustrates the single cut-vertex failure detection and restoration process. For connectivity restoration by our proposed algorithm, nodes S5, S6, and S8 send a broadcast HELLO (also known as Heartbeat) message with a communication range of Rc . If each node receives an

ACK from another neighboring node, node's mobility needs to restore network connectivity. Upon receiving the ACK from neighboring nodes, and all nodes update the routing list. The solutions presented in [13,17,22] rely on the cascaded relocation of neighboring nodes in the given scenario. It is proven that cascaded relocation of nodes leads to more energy consumption leading to a quick drainage of sensor node's batteries [4]. Cascaded relocation also leads to the shrinking of network coverage. Our proposed algorithm improves energy efficiency by avoiding the un-necessary mobility of neighboring nodes and improving network coverage. Essentially, this algorithm also prefers coverage; therefore, these coverage holes will be fulfilled by the neighbors by measuring overlapped distance according to [18].

Algorithm 2: Single cut-vertex failure detection and restoration

$N(SN_i)$: Sensor Nodes, $N(CV)$: Cut –

Vertex Nodes, $Nbr(CV)$: Neighboring Nodes of CV, ACK: Acknowledgement, R_c : Communication Range

Input: Area, $N(SN_i)$, $N(CV)$, $Nbr(CV)$, R_c

Output: Single cut-vertex failure detection and restoration

```

1  BEGIN
2  SET  $N(SN_i)$ ,  $N(CV)$ 
3  FOR EACH cut-vertex node in  $N(CV)$  do
4  IF ( $Nbr(CV)$  detects that  $N(CV)$  has failed) THEN
5   $Nbr(CV)$  broadcast HEARTBEAT with a range equals to  $R_c$ 
6  IF ( $Nbr(CV)$  received ACK) THEN
7  Update_Routing_Table( $N(SN_i)$ );
8  //As connectivity is maintained under range  $R_c$ , therefore, no need for movement
9  ELSE
10  Cascaded_Movement();
11  Update_Routing_Table( $N(SN_i)$ );
12  END IF
13  END IF
14  END

```

3.2 Scenario 2: Single End Node Failure

End nodes are also referred to as leaf nodes. These are nodes normally present at the edge of the network. Failure of end-nodes does not affect inter-node connectivity. However, their failure affects the coverage area. Upon detecting end node failure (due to the absence of HELLO messages), the neighboring nodes will calculate the overlapped coverage area with the failed node using the mechanism presented in [18]. A neighboring node with more overlapping distance with the failed node is a suitable candidate for moving toward a failed node. During the movement, the node will continue to send HELLO messages and receive ACK messages to neighboring nodes with a communication range of R_c . This process ensures that the node mobility by a suitable candidate node does not cause network disconnectivity. For further illustration of this process, let us consider Fig. 3a. Let us assume that node S4 has failed. The failure of S4 will be detected by its neighbors, i.e., S3, and S6 due to the absence of SYN messages from S4. Both the neighbors S3 and S6 will compute the relative overlapped sensing area with the failed node S4. Node S3 has more overlap areas with the failed node than S6. Therefore, S3 will be selected as a suitable neighbor responsible for moving toward the failed node S4. According to [18], S3 will move a

maximum distance of $R_c/4$ toward the direction of S4. This process is illustrated in Fig. 3b, and the steps are presented in Algorithm 3. During the movement, S3 will continue sending and receiving HELLO and ACK messages with R_c 's transmission range with neighbors to ensure connectivity. End nodes are also referred to as leaf nodes. These are nodes normally present at the edge of the network. Failure of end-nodes does not affect the inter-node connectivity [26–28].

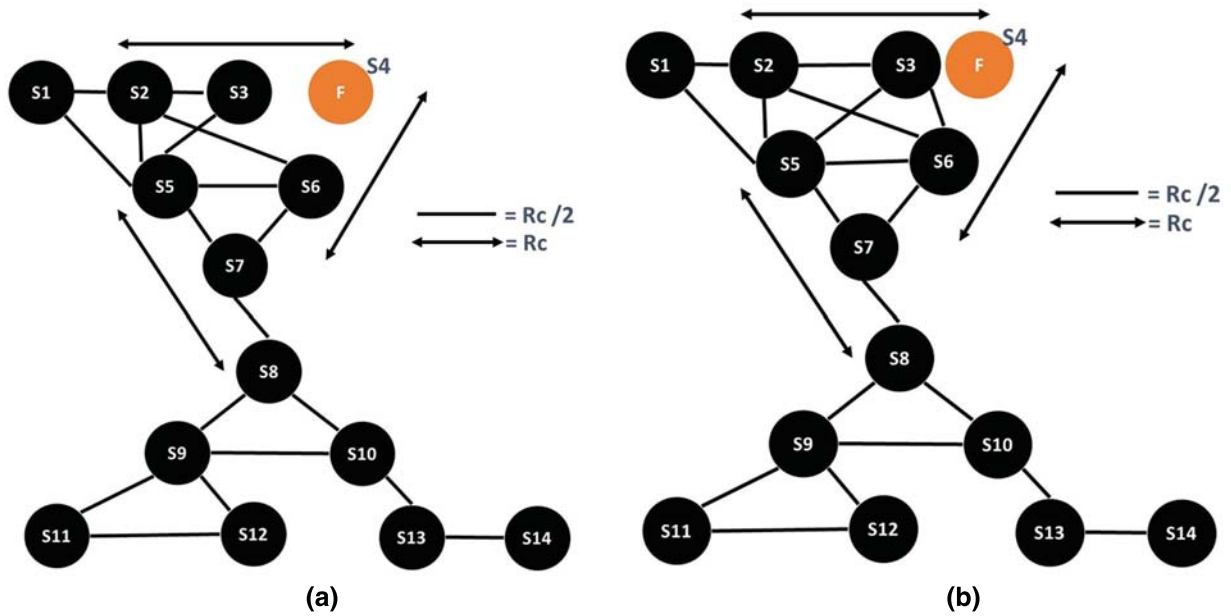


Figure 3: (a) End node failure (b) Recovery from end node failure

Algorithm 3: Single-end node failure detection and restoration

$N(SN_i)$: Sensor Nodes, $N(EN_i)$: End Nodes, $Nbr(EN_i)$: Neighboring Nodes of EN_i , R_c : Communication Range

Input: Area, $N(SN_i)$, $N(EN)$, $Nbr(EN)$, R_c

Output: Single-end node failure detection and restoration

```

1  BEGIN
2  SET  $N(SN_i)$ ,  $N(EN_i)$ 
3  FOR EACH end node in  $N(EN_i)$  do
4  IF( $Nbr(EN_i)$  detects that  $N(EN_i)$  has failed) THEN
5  Calculate_Overlapped_Area ( $Nbr(EN_i)$ ,  $Nbr(EN_j)$ )
6  IF ( $Overlapped\_Area(Nbr(EN_i)) > Overlapped\_Area(Nbr(EN_j))$ ) THEN
7   $Nbr(EN_i)$  moves towards  $Nbr(EN_j)$  with a distance  $R_c/4$  from its origin
8  // Broadcast HEARTBEAT message to all neighboring nodes
9  Relocation();
10 Update_Routing_Table( $N(SN_i)$ );
11 END IF
12 END IF
13 END

```

3.3 Scenario 3: Two Cut-Vertex Node Failure

Two cut-vertex node failure is shown in Fig. 4, where multiple nodes fail simultaneously causing a network partition. The implications of two cut-vertex node failures are substantial. In this case, merely broadcasting HELLO messages with a transmission range of R_c will not be received by the nodes toward a failed node as the distance between nodes is greater than R_c (as illustrated in Fig. 4a). The absence of ACK will mean no immediate neighbors of the failed node in the R_c range. Dealing with this problem, a new type of message called coordination message will be broadcast with a transmission range of $R_c/2$ among the neighbors as they start to move toward the failed node. The coordination message ensures that no moving node goes out of range of its neighbor, causing further disruption in the network which shows simultaneous node failure. These moving nodes will continue to transmit HELLO messages and wait for ACK messages. Receiving the ACK message will mean a node in the failed node's vicinity capable of restoring connectivity. This process is illustrated in Fig. 4b, and the steps are presented in Algorithm 4. Fig. 4b shows the original position of the relocated nodes as well as the position after relocation.

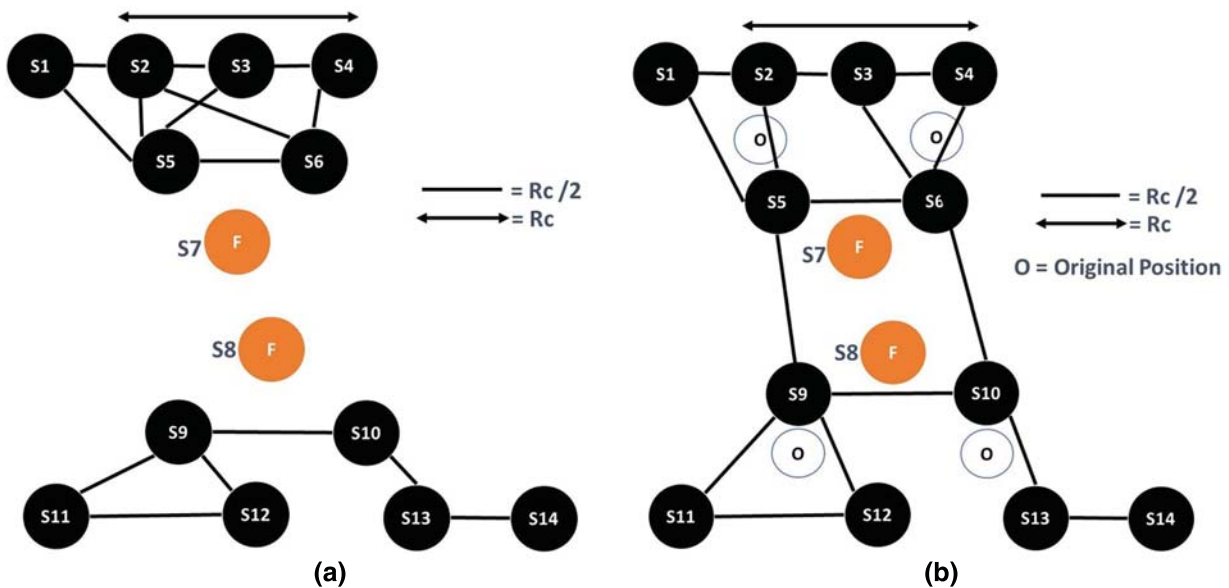


Figure 4: (a) Two cut-vertex nodes fail. (b) Recovery of two cut-vertex node

Algorithm 4: Two cut-vertex failure detection and restoration

$N(SN_i)$: Sensor Nodes, $N(CV)$: Cut –

Vertex Nodes, $Nbr(CV)$: Neighboring Nodes of CV, R_c : Communication Range

Input: Area, $N(SN_i)$, $N(CV)$, $Nbr(CV)$, R_c

Output: Two cut-vertex failure detection and restoration

```

1 BEGIN
2   SET  $N(SN_i)$ ,  $N(CV)$ 
3   FOR EACH cut-vertex node in  $N(CV)$  do

```

(Continued)

```

4     IF(Nbr(CVi) && (Nbr(CVj) detects that N(CVi) and N(CVj) has failed) THEN
5     Nbr(CVi) and Nbr(CVj) will move toward N(CVi) and N(CVj) with respect to Rc/2
    to its origin
6     Nbr(CVi) and Nbr(CVj) will broadcast HEARTBEAT message
7     Cascaded_Movement();
8     Update_Routing_Table(N(SNi));
9     END IF
10    END
    
```

3.4 Scenario 4: Multiple End Node Failure

Multiple end node failure is illustrated in Fig. 5. As WSNs operate in harsh environments, therefore multiple end node failure is a possibility. Multiple end node failure can cause a big coverage hole, and for various applications, it is undesirable. To deal with such a situation, the failed nodes' neighbors start to move toward the failed nodes and exchange SYN messages for reporting the change in location to all neighbors. The maximum movement toward the failed nodes by neighbor nodes is $R_c/4$ (as assumed in most baseline works such as [13,17,18]). Fig. 5b illustrates the movement of nodes toward the failed nodes to cope with multiple node failures. Multiple end node failure detection and recovery process are presented in Algorithm 5.

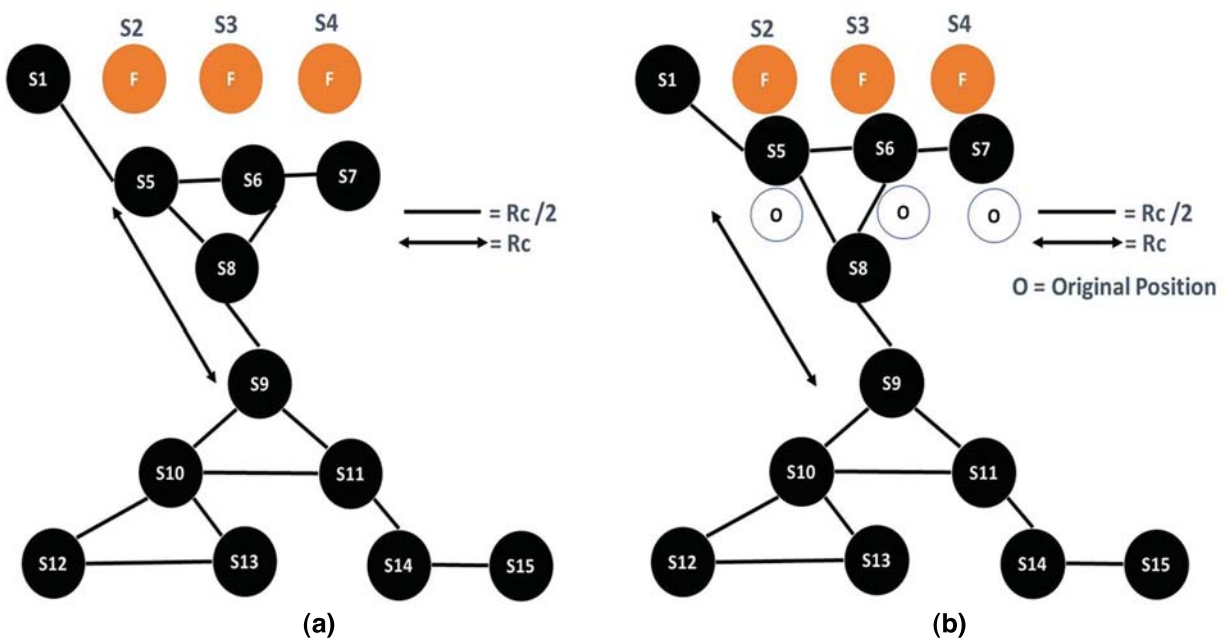


Figure 5: (a) Shows the multiple end nodes failure (b) Shows the recovery of multiple end nodes failure

Algorithm 5: Multiple end node failure detection and restoration

$N(SN_i)$: Sensor Nodes, $N(EN_i)$: End Nodes, $Nbr(EN_i)$: Neighboring Nodes of EN_i , R_c : Communication Range

Input: Area, $N(SN_i)$, $N(EN)$, $Nbr(EN)$, R_c

Output: Multiple end node failure detection and restoration

```

1  BEGIN
2    SET  $N(SN_i)$ ,  $N(EN_i)$ 
3    FOR EACH end node in  $N(EN_i)$  do
4      IF( $Nbr(EN_i)$  detects that  $N(EN_i)$  have failed) THEN
5         $Nbr(EN_i)$  will move towards  $N(EN_i)$  with respect to  $R_c/4$  to its origin
6         $Nbr(EN_i)$  will broadcast heartbeat message
7        Coverage_Enhancement();
8        Update_Routing_Table( $N(SN_i)$ );
9      END IF
10  END

```

3.5 Energy Model

The proposed technique ECRT transmits and receives a β -bit data packet over distance d by using the energy model described in [29]. While transmitting a β -bit data packet over distance d , the sensor node's energy depletion is calculated as:

$$E_{Tx}(\beta, d) = \begin{cases} \beta E_{elec} + \beta \epsilon_{fs} d^2 & d < d_0 \\ \beta E_{elec} + \beta \epsilon_{mp} d^4 & d \geq d_0 \end{cases} \quad (1)$$

where ϵ_{fs} is the required energy in free space at the Radio Frequency (RF) amplifier and ϵ_{mp} is the required energy in multipath at the Radio Frequency (RF) amplifier. E_{elec} is the energy depletion per bit for the sender or receiver circuit.

While receiving a β -bit data packet over distance d , the energy depletion of a sensor node is calculated as:

$$E_{Rx}(\beta) = E_{Rx-elec} \beta \quad (2)$$

where $E_{Rx-elec}$ is the energy consumed per bit by the receiver circuitry.

The remaining residual energy of a sensor node is given by:

$$E_{reng}(n) = E_{max} - E_{Tx}(\beta, d) - E_{Rx}(\beta) \quad (3)$$

4 Results and Discussion

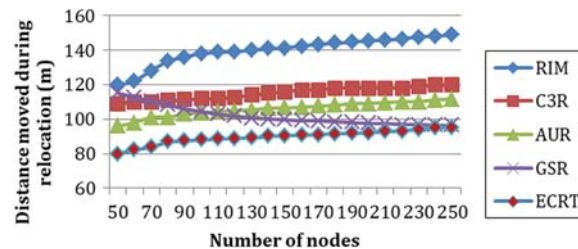
For the simulations, we have used NS2 (Network Simulator 2). During all simulations, at time $T = 0$, sensor nodes are randomly deployed in a field with dimensions of 950×950 m². The communication range is varied between 25 and 150 m. Node density is varied in the simulation area by varying the number of nodes between 100 and 250. Tab. 1 illustrates the simulation parameters. Each point in the graph is calculated by running simulations with random seeds ten times. The results for the proposed algorithm are compared with baseline algorithms RIM [18], C3R [19], GSR [22], and AUR [20]. The following sections explain the results obtained by doing extensive simulations.

Table 1: Simulation parameters

Simulation parameters	Values
Simulation area	950 × 950 m ²
Number of nodes	50–250
Rc	25–150 m
Simulation tool	NS-2.34

4.1 Total Distance Moved During Relocation

Fig. 6 shows the effect of increasing the number of nodes on the total distance moved by nodes for connectivity restoration. It can be observed that our proposed protocol ECRT performs well compared to all the other baseline algorithms. The major reason behind this is that ECRT moves just the critical nodes near the failed nodes. Alternatively, all the other baseline algorithms rely on non-critical nodes' movement, resulting in cascaded relocation. Therefore, the average distance moved by all baseline protocols is much more as compared to our proposed algorithm. Cascaded relocation increases the average distance moved by the nodes during recovery and average energy consumption. As ECRT reduces cascaded relocation compared to other protocols, therefore it proves to be more efficient.

**Figure 6:** Nodes vs. distance moved

4.2 Number of Nodes Moved

The number of nodes moved by the considered protocols by increasing the total number of nodes in the network is presented in Fig. 7. As the number of nodes increases; the number of nodes moved by all protocols increases. However, our proposed protocol ECRT outperforms all the considered baseline protocols as the increase in the number of nodes moved is lesser. Cascaded relocation is the main reason for more nodes moving on average for all considered baseline algorithms.

4.3 Reduction in Field Coverage

The percentage reduction in field coverage concerning different communication range is shown in Fig. 8. Two factors contribute to the reduction in field coverage. First, the nodes that die due to complete drainage of their batteries; second, the node movement to restore connectivity. Our proposed protocol aims to achieve connectivity restoration by reducing the number of exchanged messages (for achieving energy efficiency) and moving only critical nodes (for the restoration of connectivity and improving field coverage in case of failed nodes). It is evident from the figure that with an increase in the communication range of nodes, the percentage reduction in the field

coverage decreases for all considered protocols. The percentage reduction in the field coverage by our proposed protocol ECRT is lesser than all other baseline protocols. Among all considered protocols, RIM yields the largest percentage reduction in the field. The major reason behind this observation is using excessive cascaded relocation for connectivity restoration. Other protocols also move non-critical nodes for connectivity restoration, leading to the increased energy consumption of nodes due to movement leading to the failure of more nodes in the network. This ultimately decreases the coverage area.

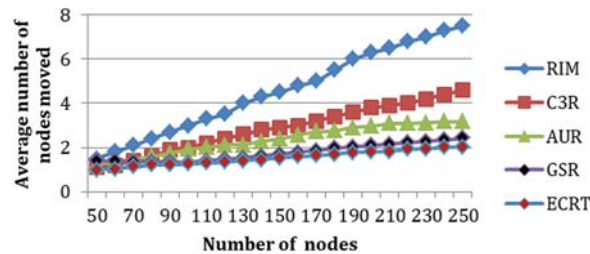


Figure 7: The average number of nodes moved during recovery

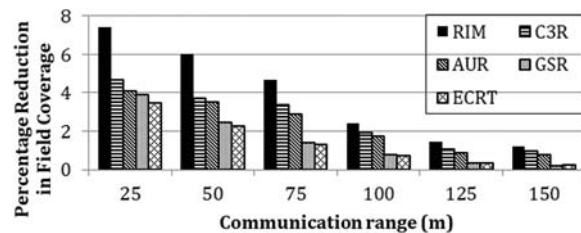


Figure 8: Percentage reduction in field coverage

4.4 Number of Exchanged Packets

The average number of packets exchanged by all considered protocols is illustrated in Fig. 9. For the working of each protocol, several different types of packets are transmitted. For Fig. 9, the communication range is varied between 25 to 150 m. It can be observed from the figure that with the increase in the communication range, the number of packets exchanged increases for all considered protocols. The operational details of each protocol impact the number of packets that need to be exchanged. Moreover, the decisions made regarding the movement of nodes also play a key role. Whenever a node is relocated, it needs to exchange different control packets with its neighboring nodes. The more nodes a protocol relocates, the more packets are exchanged. It can be seen from the figure that RIM exchanges the maximum number of packets compared to all other protocols. Cascaded relocation is one of the major factors resulting in increased packets for all baseline protocols. As our protocol moves just the critical nodes; therefore, it avoids unnecessary relocation of nodes. This results in a lower number of exchanged packets. Due to this, our proposed protocol exchanges the least number of packets. Exchanging the least number of packets also makes our proposed protocol more energy efficient as packets' exchange requires energy. Therefore, ECRT proves to be the most energy-efficient protocol among all considered protocols.

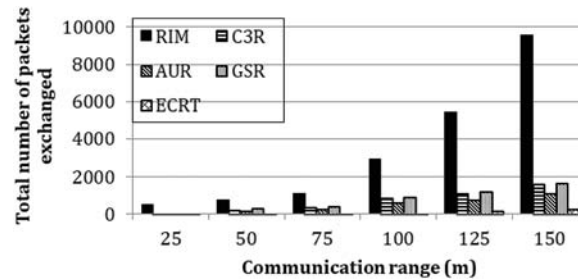


Figure 9: Total number of exchanged packets

For the continuous operation of sensor networks, connectivity restoration is of immense importance, and a technique capable of restoring the connectivity is crucial for smooth operation. A connectivity restoration technique should be self-organizing, coverage-aware, and energy-efficient. By studying the literature, it was observed that most of the solutions for connectivity restoration focused on only one of the above features but not all of them collectively. This research aimed to design a novel connectivity restoration mechanism that effectively restores connectivity by moving fewer nodes than existing techniques. Another focus was to keep the connectivity restoration technique energy efficient by exchanging a minimal number of control messages. Last but not least, the technique should minimize the reduction in field coverage. Our proposed connectivity restoration mechanism achieves all the above objectives. Extensive simulations proved the effectiveness of our proposed protocol.

5 Conclusion

Due to Wireless Sensor Networks (WSN)'s inherent characteristics, network partitioning is one of the fundamental problems. It is important to have an efficient connectivity restoration mechanism capable of restoring connectivity in single or multiple node failures. In this paper, we proposed a novel connectivity restoration technique called ECRT. ECRT can detect single and multiple node failures and efficiently restore connectivity by relocating a minimal number of nodes. It does not face the problem of cascaded relocation, and by doing extensive simulations, it is proved that it achieves an energy-efficient operation by exchanging a minimal number of packets. Moreover, it moves much lesser nodes on average compared to the other baseline approaches. It also improves the field coverage as it results in a minimum percentage reduction in the field coverage compared to the other approaches.

Future work can be extended in two possible dimensions. The first dimension involves developing an analytical model for the proposed solution under different mobility models. The second dimension involves implementing the proposed solution on real sensor nodes and doing extensive performance evaluations.

Acknowledgement: The authors extend their appreciation to the Jouf University for funding this work through Research Groups under Grant Number (40/117).

Funding Statement: This research is funded by Jouf University Saudi Arabia, under the research Project Number 40/117. URL: www.ju.edu.sa.

Conflicts of interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. Lee, M. Younis and M. Lee, "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance," *Ad Hoc Networks*, vol. 24, no. 9, pp. 1–19, 2015.
- [2] M. Z. A. Bhuiyan, G. Wang, J. Cao and J. Wu, "Deploying wireless sensor networks with fault-tolerance for structural health monitoring," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 382–395, 2015.
- [3] M. Hassan, A. A. Awady, K. Mahmood, S. Ali and M. K. Saeed, "Node relocation techniques for wireless sensor networks: A short survey," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 11, pp. 323–329, 2019.
- [4] V. Ranga, M. Dave and A. K. Verma, "Node stability aware energy efficient single node failure recovery approach for WSNs," *Malaysian Journal of Computer Science*, vol. 29, no. 2, pp. 106–123, 2016.
- [5] M. Hassan, M. A. Khan, S. Ali, K. Mahmood and A. M. Shah, "Distributed energy efficient node relocation algorithm (DEENR)," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 95–100, 2018.
- [6] M. R. Senouci, A. Mellouk, L. Oukhellou and A. Aissani, "WSNs deployment framework based on the theory of belief functions," *Computer Networks*, vol. 88, no. 4, pp. 12–26, 2015.
- [7] B. Ishibashi and R. Boutaba, "Topology and mobility considerations in mobile ad hoc networks," *Ad Hoc Networks*, vol. 3, no. 6, pp. 762–776, 2005.
- [8] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14–15, pp. 2826–2841, 2007.
- [9] K. Akkaya and M. Younis, "Coverage and latency aware actor placement mechanisms in WSNs," *International Journal of Sensor Networks*, vol. 3, no. 3, pp. 152, 2008.
- [10] K. Akkaya and M. Younis, "C2AP: Coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks," in *IEEE Int. Performance, Computing, and Communications Conf.*, New Orleans, LA, USA, pp. 281–288, 2007.
- [11] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.
- [12] M. Younis, S. Lee and A. A. Abbasi, "A localized algorithm for restoring internode connectivity in networks of moveable sensors," *IEEE Transactions on Computers*, vol. 59, no. 12, pp. 1669–1682, 2010.
- [13] A. A. Abbasi, K. Akkaya and M. Younis, "A distributed connectivity restoration algorithm. Wireless sensor and actor networks," in *Proc. of the 32nd IEEE Conf. on Local Computer Networks (LCN 2007)*, Dublin, Ireland, pp. 496–503, 2007.
- [14] P. Basu and J. Redi, "Movement control algorithms for realization of fault-tolerant ad hoc robot networks," *IEEE Network*, vol. 18, no. 4, pp. 36–44, 2004.
- [15] K. Akkaya and M. Younis, "Coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks," in *Proc. of the 26th IEEE Int. Performance Computing and Communications Conf. (IPCCC 2007)*, pp. 363–374, 2010.
- [16] M. K. Saeed, M. Hassan, A. M. Shah and K. Mahmood, "Connectivity restoration techniques for wireless sensor and actor network (WSAN), a review," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 9, pp. 139–145, 2018.
- [17] G. Wang, G. Cao and T. F. L. Porta, "Movement-assisted sensor deployment," in *Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies*, Hong Kong, pp. 640452, 2012.
- [18] 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.
- [19] M. Younis, S. Lee and G. Gupta, "A localized self-healing algorithm for networks of moveable sensor nodes," in *Proc. of the IEEE Global Telecommunications Conf.*, New Orleans, LA, pp. 1–5, 2008.
- [20] Y. K. Joshi and M. Younis, "Autonomous recovery from multi-node failure in wireless sensor network," in *IEEE Global Communications Conf.*, Anaheim, CA, pp. 652–657, 2012.

- [21] K. Yan, G. Luo, L. Tian, Q. Jia and C. Peng, "Hybrid connectivity restoration in wireless sensor and actor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, pp. 1366, 2017.
- [22] Y. K. Joshi and M. Younis, "Exploiting skeletonization to restore connectivity in a wireless sensor network," *Computer Communications*, vol. 75, no. 12, pp. 97–107, 2016.
- [23] K. Mahmood, M. A. Khan, M. Hassan, A. M. Shah, S. Ali *et al.*, "Intelligent on-demand connectivity restoration for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 10, 2018.
- [24] K. Mahmood, M. A. Khan, M. Hassan, A. M. Shah and M. K. Saeed, "Permanent relocation and self-route recovery in wireless sensor and actor networks," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 83–89, 2018.
- [25] M. K. Saeed, M. Hassan, K. Mahmood, A. M. Shah and J. Khan, "Efficient solution for connectivity restoration (ESCR) in wireless sensor and actor-networks," *Wireless Personal Communications*, vol. 117, no. 3, pp. 2115–2134, 2020.
- [26] B. Chen, H. Chen and C. Wu, "Obstacle-avoiding connectivity restoration based on quadrilateral Steiner tree in disjoint wireless sensor networks," *IEEE Access*, vol. 7, pp. 124116–124127, 2019.
- [27] L. Liu, M. Ma, C. Liu, W. Qu and G. Zhang, "ATCFS: Effective connectivity restoration scheme for underwater acoustic sensor networks," *IEEE Access*, vol. 7, pp. 87704–87715, 2019.
- [28] Y. Zhang, Z. Zhang and B. Zhang, "A novel hybrid optimization scheme on connectivity restoration processes for large scale industrial, Wireless Sensor and Actuator Networks," *Processes*, vol. 7, no. 12, pp. 939, 2019.
- [29] 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, no. 3, pp. 158–172, 2017.