Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9702650, 10 pages
https://doi.org/10.1155/2018/9702650

Research Article

Intelligent On-Demand Connectivity Restoration for Wireless Sensor Networks

Khalid Mahmood,[1] **Muhammad Amir Khan**,[1,2] **Mahmood ul Hassan**,[1]
Ansar Munir Shah,[1] **Shahzad Ali**,[3] **and Muhammad Kashif Saeed**[1]

[1] *Department of Computer Science, IIC University of Technology, Cambodia*
[2] *Department of Electrical Engineering, COMSATS Institute of Information Technology, University Road, Tobe Campus, Abbottabad 22060, Pakistan*
[3] *College of Science and Arts, Department of Computer Science, Al Jouf University, Tabarjal, Saudi Arabia*

Correspondence should be addressed to Khalid Mahmood; khalid.next@gmail.com

Received 1 December 2017; Revised 14 March 2018; Accepted 12 April 2018; Published 23 May 2018

Academic Editor: Javier Prieto

Wireless sensor networks are envisioned to play a very important role in the Internet of Things in near future and therefore the challenges associated with wireless sensor networks have attracted researchers from all around the globe. A common issue which is well studied is how to restore network connectivity in case of failure of single or multiple nodes. Energy being a scarce resource in sensor networks drives all the proposed solutions to connectivity restoration to be energy efficient. In this paper we introduce an intelligent on-demand connectivity restoration technique for wireless sensor networks to address the connectivity restoration problem, where nodes utilize their transmission range to ensure the connectivity and the replacement of failed nodes with their redundant nodes. The proposed technique helps us to keep track of system topology and can respond to node failures effectively. Thus our system can better handle the issue of node failure by introducing less overhead on sensor node, more efficient energy utilization, better coverage, and connectivity without moving the sensor nodes.

1. Introduction

Wireless sensor networks (WSNs) provide deployment of low cost wireless nodes which are useful to collect and measure data about a certain environment [1]. Wireless sensor networks are designed to perform the task of sensing in a specific environment and this requires the sensor nodes to communicate with each other and with base station via the sink node. As sensor nodes are generally deployed over a very large area, so, direct communication between sensor nodes and the end user is impractical. Therefore, nodes rely on multihop communications for sending the sensed information to the base station. Wireless sensor networks are widely used in many application areas such as rescue operations, habitat monitoring, structural monitoring, and wild fire detection [1]. The sensor nodes are considered to work in collaboration; thus the connectivity of sensor nodes is given prime importance and node failure can cause the network's malfunction. Due to this the network can get disconnected and as a result information cannot be sent to the end user which compromises the basic aim of the wireless sensor network. In order to address this issue, researchers have proposed various mechanisms to restore the network connectivity. One possible solution to the problem is to replace the failed nodes with new nodes. However, this solution is impractical in most of the cases as sensor nodes are usually deployed in areas where human intervention is minimal. Therefore, replacing the failed nodes with the new nodes is often not possible. The second solution that is proposed in the literature [2–5] is that the sensor nodes relocate themselves in order to restore connectivity. We also consider the second solution and propose a new algorithm that is capable of restoring the network connectivity by relocation of nodes.

Usually, WSNs are deployed in harsh environments; therefore, the connectivity and coverage of sensor nodes play a vital role during the lifespan of the network. Since the energy, computation, and communication resources of a sensor node are typically constrained, a large set of sensor nodes are required to cover the targeted area and increase the reliability of collected data. After successful deployment of sensor nodes in an area of interest, nodes are expected to stay connected to each other and form a network. Network connectivity guarantees sensor nodes to perform their task and to forward their data to sink nodes or to a base station (BS) that serves as a gateway to remote centers. The effectiveness of WSNs mainly relies on intersensor and sensor-BS connectivity which should sustain all the time. However, node failure causes an interruption in the network and degrades its operations. A node may fail due to external circumstances, hardware malfunction or because of constrained energy and power resources. This failure can limit the communication of some nodes, disconnect them from the network, and make them unreachable from some of their neighboring nodes.

In this paper we present a hybrid approach IDCRWSN (Intelligent on-Demand Connectivity Restoration for Wireless Sensor Networks), which consists of a model that performs in collaboration between two sections: CT (Care Taker) and ODS (On-Demand System). The neighboring nodes of each sensor node are care taker for that node. Therefore, each sensor node maintains a list of 1-hop neighbors. ODS consists of master keeper and slave keeper nodes. Master keeper is a powerful node having more processing, storage and energy resources. Slave keepers are capable of moving on demand and each one has a powerful sensor node in the shutdown state at the time of deployment. These nodes possess distinguished features in terms of power, energy, and processing that increase the lifespan of the network. Our proposed technique contains one master keeper node which acts as a gateway to the base station. Three slave keeper nodes are deployed which are also powerful nodes as compared to sensor nodes, having more processing, power, and energy resources. The sensor nodes are considered to be heterogeneous; that is, it is assumed that some of the nodes are relatively powerful in terms of energy and transmission range. Sensor nodes make partial utilization of their transmission range. Whenever a sensor node becomes energy deficient, it broadcasts an alert message. As a result, the neighboring node having sufficient energy increases its transmission range in order to temporarily make up for the energy deficient node. In the case that the entire neighboring nodes become energy deficient and there is no more neighboring node, having sufficient energy, the affected node broadcasts a message to trigger ODS module. ODS directs the master keeper and the slave keeper to take part in the recovery process. The master keeper controls and monitors the slave keepers. ODS will perform recovery process in two circumstances. Firstly, on receiving out of the energy message from the slave keeper, this in turn will alert the master keeper about failed node. Secondly, when the node is declared failed due to any environmental circumstances, slave keeper will send recovery request to the master keeper. The major objective of the proposed technique is to achieve less overhead, prolong network lifetime, and better energy utilization.

The contribution of this paper can be summarized as follows:

(i) We introduce a new energy and coverage-aware connectivity restoration technique that ensures the connectivity by making use of an adjustable transmission range of the neighboring nodes.

(ii) We compare our approach to other baseline approaches and performed extensive simulations. By analyzing the results, we observed that the proposed technique outperforms the baseline approaches in terms of connectivity and coverage restoration.

To the best of our knowledge, IDCRWSN is the efficient model which adopted hybrid approach. The baseline techniques make use of either distance or partially utilized transmission range only. However, our approach is hybrid because it makes use of both of the mentioned mechanisms and addresses the problem of energy deficiency and node failure by maintaining the connectivity and coverage. The rest of the paper is organized as follows. Section 2 explains the existing works done so far. Section 3 presents research method, Section 4 describes the results and analysis, and finally Section 5 concludes the article.

2. Literature Review

A considerable amount of work has been done in different aspects of connectivity restoration in WSN. Existing research works show that the connectivity restoration techniques so far introduced primarily deal with failure detection and recovery. In [2], the authors proposed a connectivity restoration technique called RIM (Recovery through Inward Motion). RIM detects the failure of the nodes on the basis of periodic hello messages from the neighboring nodes. All the neighboring nodes participate in the recovery process and move towards the failed node by using inward motion. Though RIM is simple, it has a few limitations. For instance, cascaded movement of all the neighbors towards the failed node may cause the reduction in the coverage area. Moreover, the movement of all neighboring nodes towards the failed node is also an energy consuming task. This not only leads to more energy consumption but also results in substantial reduction in the coverage area.

Another coverage-aware connectivity restoration technique, C^3R (Coverage Conscious Connectivity Restoration) proposed in [3], guarantees the connectivity restoration by temporarily repositioning the neighboring nodes to the failed node's position, one at a time. C^3R makes schedule for neighboring nodes in order to participate in the recovery process. The neighboring nodes follow the prescribed schedule and each neighbor moves closer to the failed node's position and serves for some time. Before repositioning the failed node, each neighbor directs its neighboring nodes to follow the alternate path, if available, for routing sensed data, otherwise buffer the data and wait for it to come back. After giving services for some time, each neighbor returns back to its original position, letting other neighbors of the failed node to take its position. This process executes recursively, and the network remains connected. Although the solution given in C^3R, seems promising, it has few limitations. The participating neighboring nodes move back and forth to restore the connectivity, which causes more energy consumption, message overhead, and also causing some nodes to be in waiting state.

The VCR (Volunteer-instigated Connectivity Restoration) proposed in [4] detects the failure of actor nodes by sending heartbeat messages. It engages the immediate neighbors of the failed actor node to restore the connectivity of the network, based on proximity and partially utilized transmission range. The VCR only selects the nearest neighbors during the connectivity restoration process, which limits the cascaded movement of the neighboring nodes towards the failed actor node. Though, the VCR limits the cascaded movement of neighboring nodes in static environment, but in a more dynamic environment, it cannot guarantees the reduction in the cascaded movement. Moreover, the VCR only favors the nearest neighbors to take part in the connectivity restoration process, regardless of checking their energy levels. This can lead the low energy nodes to take part in connectivity restoration process, which can result in more node failures in the network.

The PADRA (Partition Detection and Recovery Algorithm) proposed in [5] detects the cut-vertex nodes in advance that can cause possible partitioning in the network. Once all the cut-vertex nodes are identified, appropriate neighbor is selected for each cut-vertex node to handle its failure. The designated neighbor replaces the failed node with the node whose replacement does not partition the network into disjoint segments. The replacement causes cascaded movement of all the neighboring nodes which take part in the connectivity restoration process. PADRA causes more energy consumption due to the fact that all participating nodes move during the connectivity restoration process.

The aforementioned techniques involve all neighboring nodes of the failed node to participate in the connectivity restoration process, which causes the large number of nodes to be moved. As a result the coverage area is reduced and more energy is consumed. To overcome this issue, our proposed technique IDCRWSN efficiently exploits the neighboring nodes by considering the energy level and partially utilized transmission range together; hence it is capable of restoring the network connectivity with minimal movement of nodes and efficient utilization of energy.

As discussed earlier, WSNs are deployed in a harsh environment and there is a more chance of large scale damage and security attacks [6], which causes disjoint segments in the network. As an example, sensor nodes deployed in the battlefield area can be partitioned into disjoint segments by explosives. Thus, the remaining sensor nodes in the vicinity are unable to communicate due to disjoint partitions. Therefore, to make the WSN operational again, restoring intersegment connectivity is crucial. In [7], two classifications of heuristics to restore the connectivity of disjoint segments are discussed; the first type places extra relay nodes to establish the connectivity, whereas, the second one uses mobile agents to transport data between disjoint segments.

A distributed restoration algorithm called ORNP (Optimal Relay Node Placement) proposed in [8] efficiently minimizes the relay node (RN) count while restoring the connectivity of segmented WSNs. ORNP makes use of Voronoi partition to subdivide the area of interest into clusters. Cluster head and cluster members are chosen in each Voronoi cell. While the entire network becomes segmented, the proposed ORNP technique finds a sensor node that is covered by at least two distinct cluster heads. Sensor node, now acts as a relay node to associate disjoint segments. The number of disjoint segments is further decreased by applying the 3-star technology which recursively deploys one RN for every three disjoint segments. Finally, the center of mass (CoM) is calculated for the smallest polygon which contains all remaining disjoint segments. The RNs are then placed towards the CoM to accomplish the connectivity of the entire network. The OAAS (obstacle avoidance connectivity restoration strategy based on Straight Skeleton) connectivity restoration technique, while considering the existence of obstacles for WSNs, works in three phases [9]. During the first phase, the modified minimum spanning tree (MST) construction algorithm, namely, *Ad-Prim*, is applied for the selection of representatives in order to establish the intercomponent connection. Obstacle-avoid algorithm (OA) is executed in phase two, to construct the shortest paths around obstacles. In the last phase, minimum numbers of RNs are deployed with the help of Straight Skeleton Based SMT (Steiner Minimal Tree) Construction Algorithm (SSIC). A novel technique, namely, BIND (Boundary-aware optimized Interconnection of Disjoint segments) proposed in [10], overcomes the shortcomings of SMT (Steiner Minimum Tree) based solution to find the least number of relay nodes to restore the connectivity of a partitioned network. The Connectivity Restoration with Assured Fault Tolerance (CRAFT) algorithm proposed in [11] attempts to form a Backbone Polygon (BP) around the center of the damaged network area where no partition lies inside. Relay Nodes (RNs) that provide enhanced coverage and connectivity in the targeted area at low cost are then deployed to connect each outer partition to the BP through two nonoverlapping paths to restore the connectivity of the network.

A hybrid recovery strategy based on random terrain (HRSRT) to restore the connectivity of damaged WSNs is proposed in [12]. The influence of realistic terrain is planned by mapping the area of interest into a grid of equal sized cells. A weight $\omega(\mathrm{c})$ is assigned to each cell c that represents the corresponding terrain influence within. The weight $\omega(\mathrm{P})$ of each path P is calculated by accumulating the weight of each cell $\omega(\mathrm{c})$ along path P. In this way, the weighted complete graph Kn is constructed by considering minimum weight paths between segments. A random terrain based path planning algorithm (RTPP) is developed on Kn to initiate connectivity restoration tour T for mobile data collectors (MDCs). During the connectivity restoration, $\omega(\mathrm{T})$ is proportional to the cost for data collection and aggregation. The number of MDCs may be one or more than one, resulting in two different relay nodes deployment strategies, that is, an optimized relay node deployment (ORND) and RND (Relay Nodes Deployment) to take part in connectivity restoration. The main objective of these strategies is to minimize the energy cost for data collection and aggregation. In [13], the authors contributed a connectivity restoration technique SACR (Survivability-Aware Connectivity Restoration) for partitioned WSNs with the help of mobile nodes. The proposed SACR technique connects the partitioned segments of the damaged network by considering data load levels of disjoint segments. A group of moveable nodes are moved between different disjoint segments and an appropriate location of these isolated segments is found in order to stop the moveable nodes. Once the location of isolated segments is traced, a relay partition is created for every isolated segment to restore the connectivity. In [14], the authors designed a robot control strategy to provide a connected path from the area of interest to the base station. The main objective is to discover a minimum distance with less number of hop counts for mobile robots, such that the network connectivity remains unbroken. The proposed strategy consists of two algorithms. The first algorithm identifies the nearest robot to the event area and allocates it to the event location. Then, the algorithm searches the nearest nonconnected robot and forwards it to the transmission range of any of the connected component with already allocated robot. The algorithm executes recursively until the entire network is connected. The second algorithm optimally finds the locations for minimizing the hop count from the base station to the event area.

In this paper, we focus on restoring the connectivity of the network with a minimum number of nodes to be moved during relocation along with efficient energy utilization. The baseline schemes which are discussed above require a large numbers of nodes to be moved towards the failed node's position. This results in consuming more energy. So to the best of our knowledge IDCRWSN is the efficient model which adopted hybrid approach and addresses the problem of failure detection and recovery by maintaining the connectivity and coverage intact.

3. Research Method

3.1. Problem Formulation. IDCRWSN can be applied to the system having features like stationary sensor nodes surrounded by slave keepers having a master keeper controlling unit. Slave keepers are supposed to move on demand and are mobile. They are connected to master keeper as well as with each other and neighboring nodes. We take the following set of assumptions in our model:

(i) N sensor nodes are randomly deployed in an area of interest.

(ii) Deployed sensor nodes remain static in the monitoring area. It means that all the sensor nodes in the network other than slave keepers are static, so, they do not move from their initial position.

(iii) The communication range of a sensor node is "rc", whereas rc/2 is half of the communication range.

(iv) Although deployment will be random, as slave keeper can move on demand, so each slave keeper is supposed to be at an equal distance of rc/2 from its neighboring nodes.

(v) Interslave communication will not affect the communication of sensor nodes.

(vi) Master keeper will have the knowledge of the exact location of each node and slave keeper at the time of deployment and when any node is declared dead; it will update its network topology information.

(vii) Slave keeper can move only on demand, when commands are given by master keeper and the alert message from neighboring sensor nodes.

(viii) Adjacent nodes having rc/2 as distance between them are supposed to be the caretaker of each other and are supposed to help in temporary relocating node.

(ix) The initial energy of deployed sensor nodes will be same, that is, $E_{\max} = 20$ Jouls (J).

(x) The energy consumption of the deployed sensor nodes is not uniform.

(xi) In case one node is running out of energy and the neighboring node failed to fulfill the recovery, then the query goes to master keeper, and it will give priority to failed node relocation.

3.2. Energy Model. We have assumed the energy model depicted in [15] to transmit and receive a β-bit data packet over distance d. The energy consumption of a sensor node when it transmits a β-bit data packet over distance d is calculated as

$$E_{Tx}(\beta, d) = \begin{cases} \left(E_{\text{elec}} + \varepsilon_{fs} d^{2}\right)\beta & d < d_{\circ} \\ \left(E_{\text{elec}} + \varepsilon_{mp} d^{4}\right)\beta & d \geq d_{\circ}, \end{cases} \qquad (1)$$

where β is the size of the data packet expressed in bits, ε_{fs} is the energy required by the radio frequency (RF) amplifier in free space, and ε_{mp} is the energy required by the radio frequency (RF) in multipath. E_{elec} is the energy consumption per bit of the transmitter circuitry.

The energy consumption of a sensor node to receive a β-bit data is given by

$$E_{Rx}(\beta) = E_{Rx-\text{elec}}\beta, \qquad (2)$$

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

The remaining residual energy of a sensor node is given by

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

3.3. System Model. We consider fifteen stationary sensor nodes, three slave keepers, and one master keeper along with one sink node as shown in Figure 1. The sink node behaves as a gateway to the base station. Slave keepers are capable of moving on demand and each one having a powerful sensor

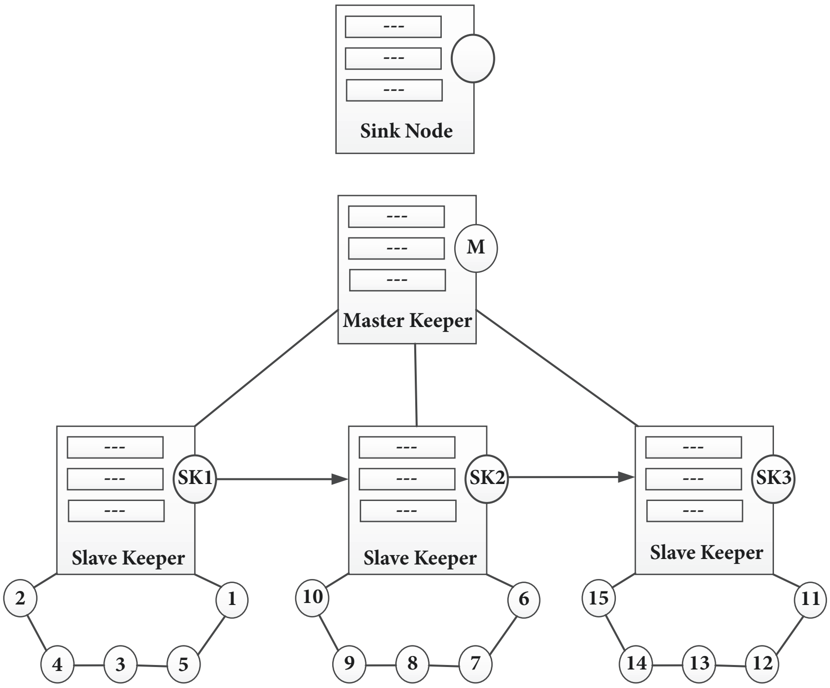


Figure 1: Design of IDCRWSN.

node in the shutdown state at the time of deployment. The master keeper controls and monitors the slave keepers. We have applied centralized approach for the slave keepers with respect to sensor nodes due to the fact that each slave keeper monitors the sensor nodes associated with it.

Once sensor nodes are randomly deployed, the slave keeper nodes are subject to arrange themselves in predefined order with sensor nodes. This is done after the command *arrange_soon* is broadcasted by the master keeper to the slave keepers. As soon as the slave keepers receive the message, they start moving and come at a fixed distance of rc/2 from a maximum of five sensor nodes. So in this way, three clusters are formed. After the formation of the network, slave keepers will send the *arranged_order* message to the master keeper containing the network topology of their associated sensor nodes. Master keeper will now track and save the network topology until any change occurs. Note that master keeper can also move on demand and deployed sensor nodes partially utilize their transmission range. Sensor nodes will send heart beat messages to each other in order to know about the status of each other and to make sure that the network is alive.

According to (3), the deployed sensor nodes will calculate their remaining residual energy $E_{\text{reng}}(n)$ and if the energy $E_{\text{reng}}(n)$ of any sensor node n_i is less than $E_{\text{threshold}}$, it will broadcast the *energy_alert* message to its neighboring nodes. The neighboring node which has sufficient energy will broadcast *recovery_offered* message and will increase its transmission range instead of moving towards the energy deficit sensor node. In this way the neighboring node will temporarily recover the energy deficit sensor node. When the recovery node itself is running out of energy it will broadcast *energy_defficiency* message to slave keeper. The slave keeper will then alert the master keeper by sending *activity_time* message. Now master keeper will direct the concerned slave keeper to replace the failed node by its redundant node. Slave keeper will perform this activity when it will receive *activity_granted* message from master keeper. The energy deficit node will send *recovery_time* message to its redundant node so that the redundant node can change its state from shutdown to active state and slave keeper will replace this node at the failed node's location. After every fixed time interval "t" the slave keepers are responsible to update their status to master keeper. Figure 2 depicts failure detection and the recovery process, whereas the steps involved in IDCRWSN technique are given in Algorithm 1.

3.4. Complexity Analysis

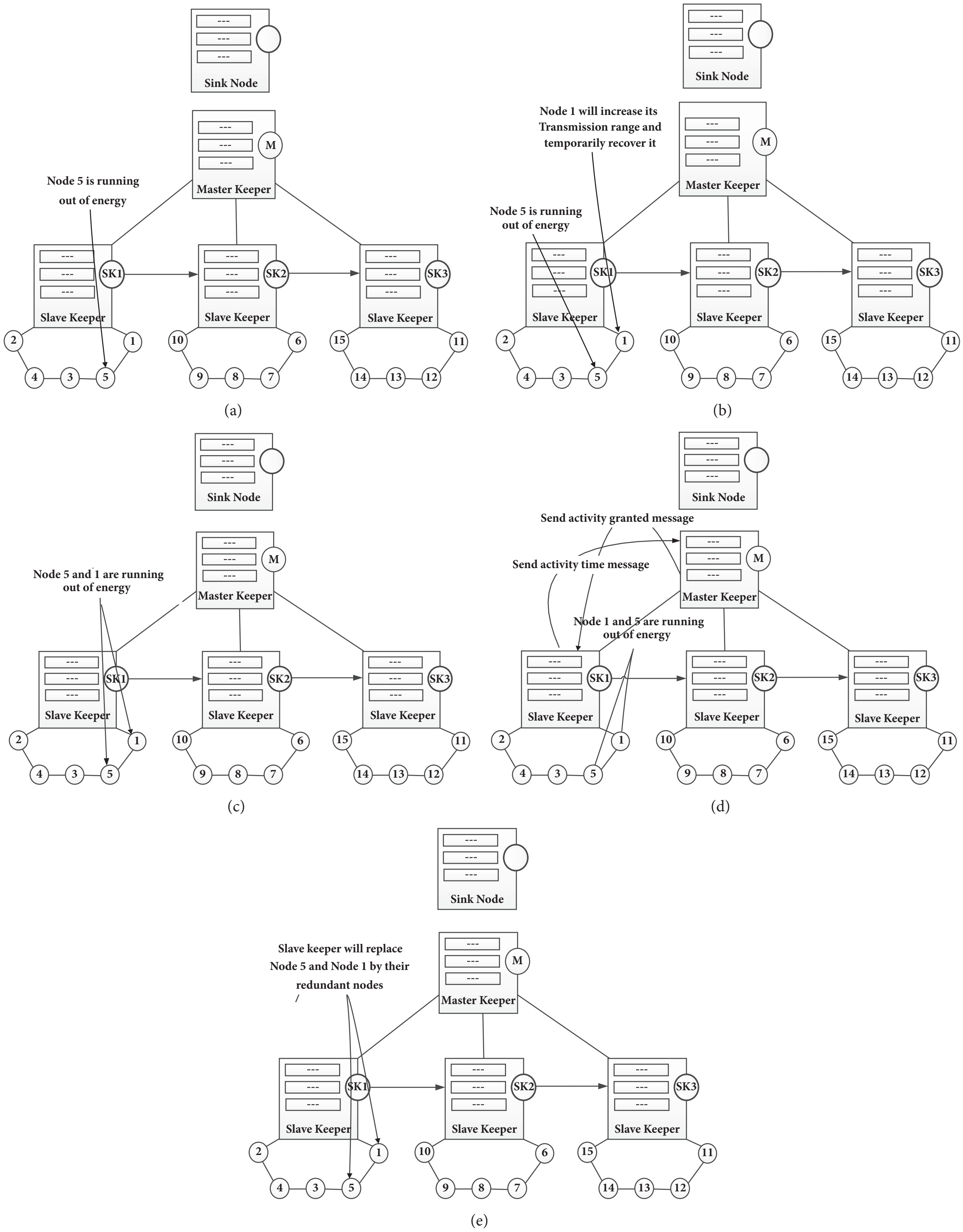
3.4.1. Time Complexity. The time complexity of Algorithm 1 depends on the number of slave keepers, sensor nodes, and the process of energy calculation for each sensor node. Suppose that

the number of slave keepers = s,

the number of sensor nodes associated with each slave keeper = n,

process of energy calculation for each sensor node = E.

Once the master keeper passes the signal to the slave keepers, they send the signal to the associated sensor nodes. Now the sensor nodes will compare the residual energy with threshold energy. If all the sensor nodes of every slave keeper



Node 5 is running out of energy

(a)

Node 1 will increase its Transmission range and temporarily recover it

Node 5 is running out of energy

(b)

Node 5 and 1 are running out of energy

(c)

Send activity granted message

Send activity time message

Node 1 and 5 are running out of energy

(d)

Slave keeper will replace Node 5 and Node 1 by their redundant nodes

(e)

Figure 2: Node failure detection and recovery in IDCRWSN: (a) node 5 failure; (b) node 1 offering recovery by increasing its transmission range; (c) recovery node running out of energy; (d) master keeper directs slave keeper to recover node 5 and node 1 failure; (e) replacement of failed nodes 5 and 1 by their redundant nodes.

```
Input: Area, Rc, n
Output: Connectivity restored
Begin
    Set $n_i$
        For each node $n_i$ do
            Calculate ($E_{\text{reng}}(n_i)$)
            if ($E_{\text{reng}}(n_i)$) < $E_{\text{threshold}}$ then
                //Send energy_alert message to neighboring nodes ($Nbr_j$)
                energy_alert ($Nbr_j$)
            For each neighboring node $Nbr_j$ of $n_i$
              //Neighboring node will increase its transmission range
                recovery_offered ($Nbr_j$)
              end if
            if $E_{\text{reng}}(Nbr_j) < E_{\text{threshold}}$ then
                Send energy_defficiency message to slave keeper
              Send activity_time message to master keeper
              Send activity_granted message from master keeper
              //Slave keeper will replace failed nodes with their redundant nodes
              Replace_Failed_Node ($n_f$)
            end if
```

Algorithm 1: Node failure and recovery in IDCRWSN.

have sufficient energy, then the total number of iterations (best score) required to execute the algorithm is ($\mathbf{n} \times \mathbf{E}$). If the number of slave keepers is s, then the total executions required will be $\{(\mathbf{n} \times \mathbf{E}) \times \mathbf{s}\}$. In worst case, if the energy of each sensor node is less than threshold energy, then the total number of iterations (worst score) required to execute the algorithm is $\{(\mathbf{n} \times \mathbf{E}) \times (\mathbf{n} - \mathbf{1})\} \times \mathbf{s}$.

The nature of our proposed algorithm is polynomial. The outer loop executes for each slave keeper and nested loop is executed for every sensor node assigned to current slave keeper. In terms of Big-O notation, the complexity of the Algorithm 1 is $O(n^2)$.

4. Results and Analysis

The effectiveness of IDCRWSN is validated through extensive simulations. This section depicts the simulation environment, performance metrics, baseline approaches, simulation results, and analysis.

4.1. Simulation Environment, Performance Metrics, and Baseline Approaches. The simulation experiments are performed on a discrete event simulator, called OMNeT++. All the protocols are implemented in a simulation framework of OMNeT++, called INET [16]. The experiments are conducted in 600 m × 600 m square area. The number of nodes deployed in an area of interest ranges from 20 to 200 nodes. We have simulated several configurations with varying the number of nodes and communication ranges.

4.1.1. Simulation Parameters. The following simulation parameters are used to perform the experiments:

(i) N: the *number of nodes* that are deployed in the simulation area.

Table 1: Simulation parameters.

Parameters	Values
Number of nodes (N)	100–200
Area	$600 \times 600\ \text{m}^2$
Communication range	225 m
Sensing range	10 m
$E_{\text{threshold}}$	0.5 J
Initial energy of sensor nodes (E_{max})	20 J

(ii) The *area* in which the sensor nodes are deployed.

(iii) The *communication range* of all the sensor nodes in the network which ensures the network's connectivity and coverage.

(iv) The *sensing range* of a node that ensures the node's coverage.

(v) The energy threshold ($E_{\text{threshold}}$) which ensures the minimum energy level of a sensor node.

(vi) The *initial energy of sensor nodes (E_{max})* is the maximum energy level of all the sensor nodes during the initial deployment.

All aforementioned simulation parameters and their associated values are given in Table 1.

4.1.2. Performance Metrics. The performance of IDCRWSN is evaluated using the following metrics:

(i) The *number of nodes moved* during the connectivity restoration process which describes the scope of recovery.

(ii) The *coverage reduction* performance metric evaluates the percentage of reduction in the coverage field

Table 2: Exchanged messages in IDCRWSN algorithm.

Message Types	Description
arrange_soon	Master keeper and slave keeper
arranged_order	Slave keeper and master keeper
energy_alert	Sensor node to neighboring node
recovery_offered	Neighbor node to energy depleting node
energy_defficiency	Recovery neighboring node to slave keeper and neighbor node
activity_time	Slave keeper to master keeper
activity_granted	Master keeper to concerned slave keeper
recovery_time	Slave keeper to its redundant sensor node
arranged_order	Slave keeper and master keeper
energy_alert	Sensor node to neighboring node
recovery_offered	Neighbor node to energy depleting node

Table 3: Compared baseline approaches.

Algorithm	Nodes Deployment	Nodes participation in connectivity restoration	Objective
VCR	Random	Immediate neighbors based on proximity	Connectivity restoration with minimum number of nodes to be moved
C^3R	Random	All neighbors	Coverage-awareness in the connectivity restoration process

relative to its prefailure level during the connectivity restoration process.

(iii) The *total distance moved* by the participating sensor nodes in the connectivity restoration process that ensures the efficiency of IDCRWSN.

(iv) The *number of exchanged packets* during connectivity restoration process evaluates the recovery process overhead. Exchanged messages during the execution of IDCRWSN algorithm are given in Table 2.

4.1.3. Baseline Techniques. The performance of IDCRWSN is compared to C^3R (Coverage Conscious Connectivity Restoration) and VCR (Volunteer-instigated Connectivity Restoration). C^3R strives to restore the connectivity and ensures the coverage through involving one or multiple neighbors of the failed node. All neighbors of the failed node relocate themselves temporarily, one at a time, to substitute the failed node. C^3R does not deal with permanent relocation of sensor nodes. VCR also involves neighbors of the failed actor node to restore the connectivity. The main criterion for restoration is based on proximity and partially utilized transmission range of neighboring nodes. Baseline approaches and their objectives are stated in Table 3.

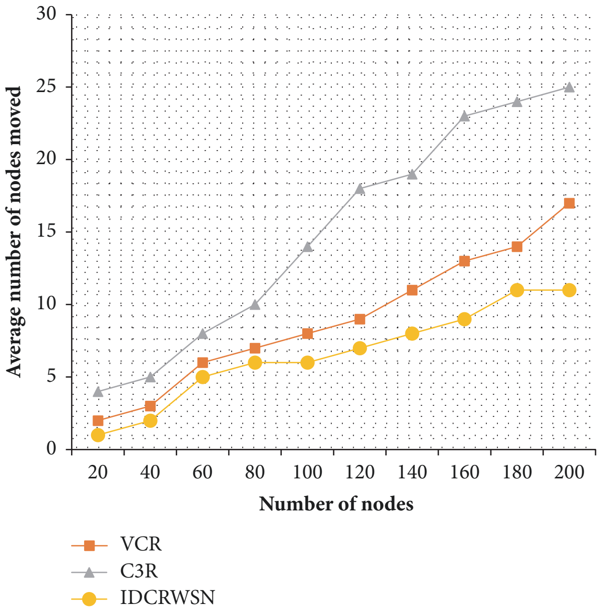


Figure 3: Average number of nodes moved in VCR, C^3R, and IDCRWSN.

4.2. Simulation Results and Analysis. The performance results are critically analyzed and compared between IDCRWSN and baseline approaches C^3R and VCR on different random topologies by varying the number of nodes and communication range.

4.2.1. Number of Nodes Moved. Figure 3 shows the average number of nodes moved during connectivity restoration process. The simulation results show that, in case of C^3R, connectivity restoration in case of a node failure results in moving more nodes. The major reason behind this is that, in case of C^3R, in order to recover the network, cascaded relocation is needed. While in case of VCR we can also see that failure recovery is done by moving more nodes as compared to our proposed approach. This will result in not only more coverage reduction but also more power

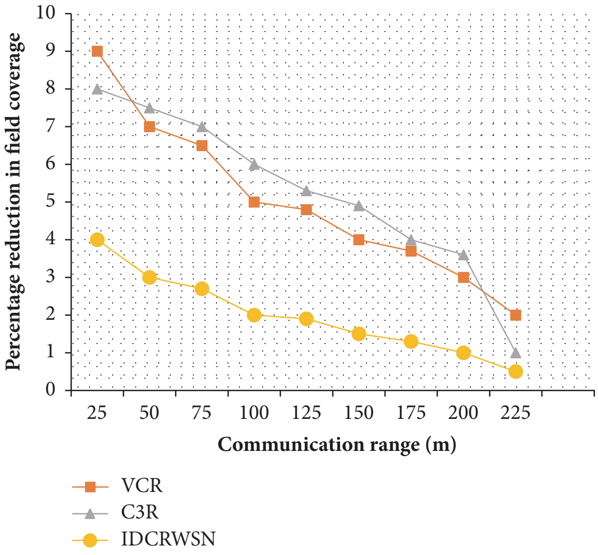


Figure 4: Percentage reduction in field coverage.

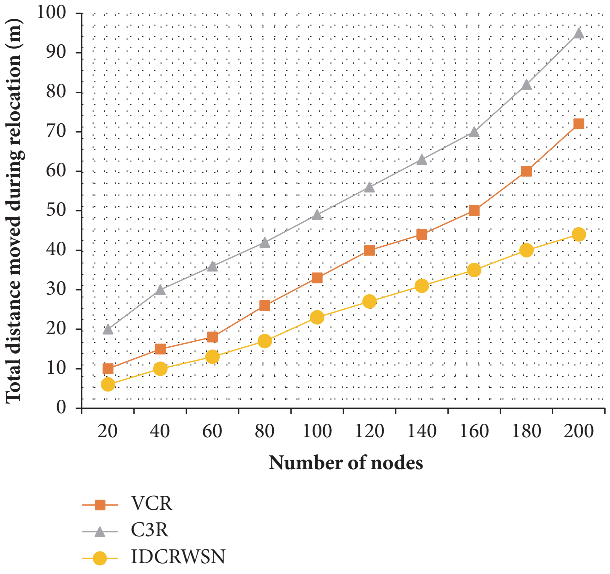


Figure 5: Nodes versus distance moved.

consumption. Our proposed algorithm outperforms in terms of number of nodes moved, as we introduce the stationary nodes while only the caretaker will move only on demand. As a result, the number of nodes moved for our proposed algorithm is better as compared to C^3R and VCR as shown in Figure 3.

4.2.2. Coverage Reduction. Figure 4 shows how the coverage is influenced by analyzing the percentage reduction in field coverage. In case of C^3R the cascaded relocation of nodes causes the holes in coverage. In Figure 4, it can be seen that if the radio range of the nodes increases, the percentage reduction in the coverage also increases. In case of VCR, the recovery through actor nodes also results in the topology change and holes in the coverage. IDCRWSN limits the coverage reduction more as compared to VCR and C^3R.

4.2.3. Total Distance Moved during Relocation. Figure 5 shows the total distance moved by the nodes which are participating in restoration of connectivity. Minimum distance is moved by the slave keeper as on time of request from the sensor nodes and master keeper. So IDCRWSN definitely outperforms both C^3R and VCR. In VCR the actor nodes are power consuming as they are not confined to repair node failure within certain range as shown in Figure 5, while in C^3R all neighboring nodes take part in the recovery process and perform cascaded movement towards failed node. Therefore, the total distance moved by the nodes during relocation is more in C^3R as shown in Figure 5. So, the performance of IDCRWSN is better than C^3R in terms of total distance moved during relocation.

4.2.4. Number of Exchanged Packets. Figure 6 presents the number of packets exchanged during connectivity restoration in proposed and baseline techniques. After performing extensive simulations, it is observed that the proposed IDCRWSN technique has minimum messaging overhead, while the VCR and C^3R exchange maximum number of messages. The proposed technique exchanges minimum number of packets due to the fact that only master keeper, slave keeper, and neighboring node having sufficient energy take part in the connectivity restoration process, while in VCR and C^3R all neighboring nodes take part in connectivity restoration process. Moreover, all neighboring nodes in VCR and C^3R perform cascaded movement towards the failed node which results in more messaging overhead during connectivity restoration.

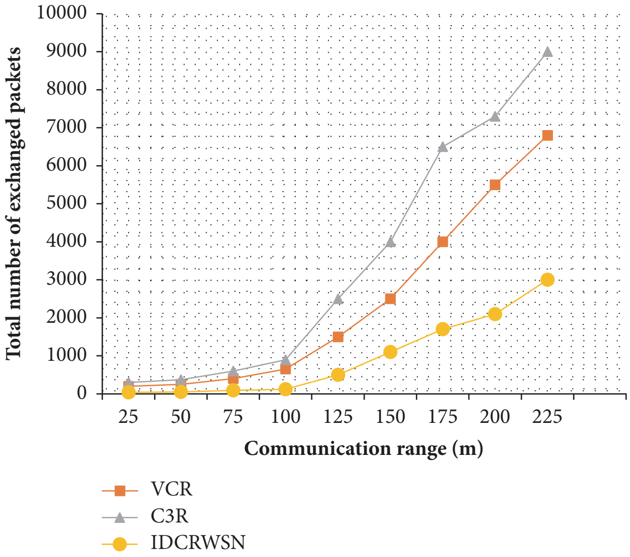


Figure 6: Total number of exchanged packets.

5. Conclusion

The connectivity restoration has significant effect on the performance of WSNs. In this paper we have presented IDCRWSN, a novel connectivity restoration approach to solve the problem of connectivity restoration in an efficient manner. Most of the previous works require cascaded movement of large number of nodes without taking into account the energy level of the participant nodes to restore the connectivity. However, this way of restoring the connectivity is not realistic. We have proposed a novel connectivity restoration technique IDCRWSN, which restores the connectivity with efficient utilization of energy and minimum number of nodes to be moved during relocation. We have compared IDCRWSN with VCR and C^3R and find out that IDCRWSN outperforms them in terms of nodes moved, percentage of coverage reduction, and message overhead.

Although IDCRWSN is an efficient technique, it is designed only for the scenarios where area of interest has stationary sensor nodes surrounded by mobile slave keepers. For the future work, we will consider mobile nodes present in the network. Moreover, we also intend to evaluate the performance using more realistic mobility traces and communication models.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors would like to express their gratitude to IIC University of Technology, Kingdom of Cambodia, for providing good research environment.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] M. F. 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)*, pp. 1–5, IEEE, December 2008.

[3] N. Tamboli and M. Younis, "Coverage-Aware Connectivity Restoration in Mobile Sensor Networks," in *Proceedings of the ICC 2009 - 2009 IEEE International Conference on Communications*, pp. 1–5, Dresden, Germany, June 2009.

[4] 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.

[5] 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 Wireless Communications and Networking Conference (WCNC '08)*, pp. 2480–2485, Las Vegas, Nev, USA, April 2008.

[6] M. Sookhak, A. Akhundzada, A. Sookhak et al., "Geographic wormhole detection in wireless sensor networks," *PLoS ONE*, vol. 10, no. 1, Article ID e0115324, 2015.

[7] 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.

[8] X. Wang, L. Xu, and S. Zhou, "Restoration Strategy Based on Optimal Relay Node Placement in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 409085, 2015.

[9] X. Wang, L. Xu, and S. Zhou, "A straight skeleton based connectivity restoration strategy in the presence of obstacles for WSNs," *Sensors*, vol. 17, no. 10, article no. 2299, 2017.

[10] W. Lalouani, M. Younis, and N. Badache, "Optimized repair of a partitioned network topology," *Computer Networks*, vol. 128, pp. 63–77, 2017.

[11] 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.

[12] 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, 2017.

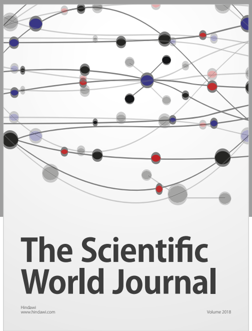
[13] X. Liu, "Survivability-Aware Connectivity Restoration for Partitioned Wireless Sensor Networks," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2444–2447, 2017.

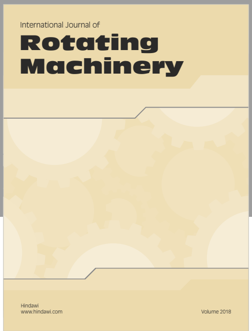
[14] A. Wichmann, T. Korkmaz, and A. S. Tosun, "Robot Control Strategies for Task Allocation with Connectivity Constraints in Wireless Sensor and Robot Networks," *IEEE Transactions on Mobile Computing*, 2017.

[15] 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.

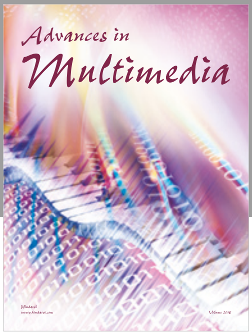
[16] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

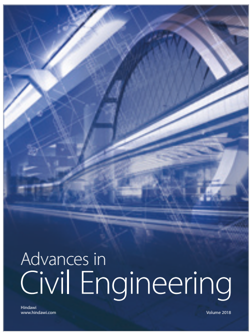
Journal of
Engineering

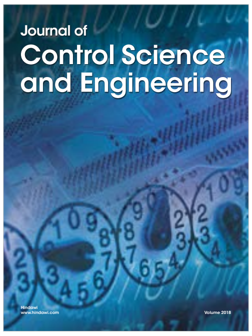

The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

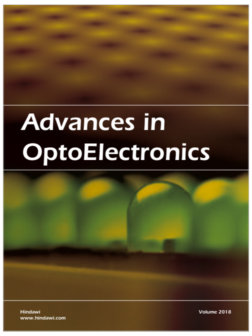

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

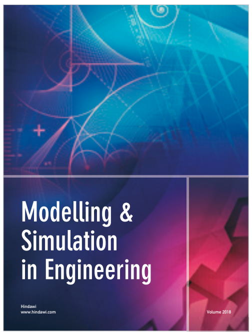
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

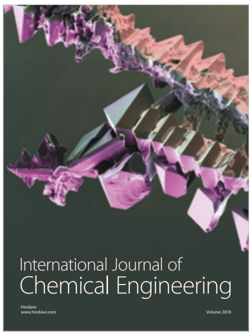

Advances in
OptoElectronics

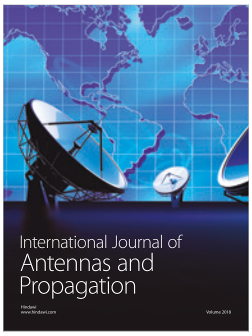
VLSI Design

International Journal of
Navigation and
Observation

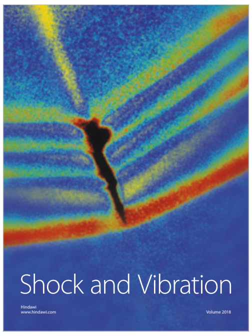

Modelling &
Simulation
in Engineering

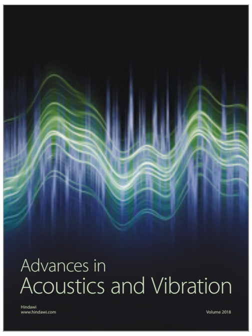
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration