

Distributed Energy Efficient Node Relocation Algorithm (DEENR)

Mahmood ul Hassan¹, Muhammad Amir Khan², Shahzad Ali³, Khalid Mahmood¹, Ansar Munir Shah¹

¹Department of Computer Science IIC University of Technology, Cambodia

²Department of Information Systems, King Khalid University, Abha, Kingdom of Saudi Arabia

³Department of Computer Science, Al Jouf University, Tabarjal, Kingdom of Saudi Arabia

Abstract—Wireless Sensor Networks (WSNs) due to their inherent features are vulnerable to single or multiple sensor node failure. Node's failure can result in partitioning of the networks resulting in loss of inter-node connectivity and eventually compromising the operation of the sensor network. The recovery from partitioning of network is crucial for inter-node connectivity. In literature, a number of approaches have been proposed for the restoration of inter-node connectivity. There is a need for a distributed approach that has an energy efficient operation as energy is a scarce resource. By keeping this in mind we propose a novel technique to restore the connectivity that is distributed and energy efficient. The effectiveness of the proposed technique is proven by extensive simulations. The simulation results show that the proposed technique is efficient and capable of restoring network connectivity by using the mechanisms for improving the coverage.

Keywords—Wireless sensor network; node failure; network connectivity

I. INTRODUCTION

In recent years, a number of applications have gained interest in wireless sensor networks for the fact that they are applicable in harsh environments predominantly in the setup of hostile applications, such as reconnaissance of the battlefield, surveillance of coast and border, rescue and search, outer space, and deep ocean exploration [1]. Mainly, wireless sensor nodes are used for monitoring in the areas of health, residential, and military purposes since they are self-healing, self-organized, and fault tolerant. In the application for military, WSNs are widely used and apt to different tasks such as command and control, targeting, communication, and surveillance [2].

Wireless sensor nodes are small, having limited computing and processing power. Each node can be equipped with one or multiple sensors. A variety of magnetic, chemical, thermal, optical, biological and mechanical sensors could be combined to the wireless sensor nodes to measure the properties of the environment [2]. These sensors can measure, sense and collect data from the environment and they can also broadcast the sensed data to the manipulator. These nodes comprise a power supply, an actuator, memory, processing unit, and radio transceivers. Typically, wireless sensor nodes are deployed in hard-to-access areas where human intervention is difficult or not possible. As the sensor nodes are generally cheap therefore they have very limited memory, limited power source and a transceiver for sending and receiving data. The central power source of sensor node is the battery. A secondary power

source that gathers energy from the external environment could be added such as solar cell, depending on the sensor nodes and the type of application [3].

The topology of network, scheme of deployment, and the size of the network is affected by monitoring environment. In an indoor environment, less number of nodes can be deployed and a pre-planned network can be initiated. When the sensor nodes need to be deployed in a large area in open then pre-planned network is not ideal. For a large area in open, a large number of sensor nodes should be deployed for making sure that the whole area is covered [4]-[6]. Increasing the number of nodes in an area can also improve the reliability of collected information. WSNs can be deployed in the areas where it is humanly impossible to go therefore they can reduce the risks associated to human life. Once the nodes in a sensor network are deployed, these nodes establish a network to coordinate their actions and share information while performing the assigned task. Normally in all of these activities, sensor nodes need to collaborate with each other for optimizing the performance and increasing the network lifetime. Over time, the battery of the sensor nodes is depleted due to communication and processing. It is of immense importance that the nodes distribute the tasks of communication and processing among themselves in such a way that the total lifetime of the network extends to the maximum. As the nodes in a network start to die, the connectivity of nodes is affected. Inter-node connectivity is very important in sensor networks because dis-connectivity among nodes may lead to loss of important data to be communicated with the user terminal. In case of dis-connectivity among nodes, a sensor node first need to detect a node that failed/died in its vicinity and typically have to notify its neighboring node to reposition itself in such a way that the nodes become connected again [7]. However, nodes in the network may die at any time due to depleted batteries or physical impairments produced by an unfriendly environment in which WSNs operate in.

The failed node could disturb the connectivity of network and disorder the collected sensed information. In the worst case the network could be separated into numerous segments or partitions as well as the information flow from sensor nodes to the user terminal can be completely cut-off. In order to avoid this scenario, the connectivity of network should be restored. Quick recovery of connectivity is necessary in order to maintain the network to observe activity. Deploying redundant nodes instead of dead nodes is a slow process and is

repeatedly impossible in harsh and environmentally challenging areas. Therefore, the healing process should be self-organized comprising of the existing alive nodes. Provided the unsupervised and autonomous supervision of WSN allows the failure that could be recovered in a categorized way. In addition, overhead of sensor nodes should be minimized in order to meet with the resource constraints. A number of approaches have preferred the repositioning of survivor node for the recovery of the partitioned network [8], [9]. However, these works have focused on recovery of inter-node connectivity except for observing the unfavorable effects of repositioning nodes on the network coverage and energy consumption. Primarily, it depends upon sensing and communication ranges, along with the deployment of redundant nodes. Formerly proposed algorithms for connectivity restoration may exclude several segments of the examined area uncovered by any node. Though coverage is a vital design parameter for sensor networks, connectivity together with coverage can be further utilized to evaluate the quality of service of WSN [10]. Certainly, connectivity and coverage have to be considered in an integrated manner. In this work, we present a distributed energy efficient node relocation algorithm that is capable of recovering the network from node failure.

The rest of the paper is organized as follows. Section II explains the related work and Section III explains the proposed algorithm. In Section IV, simulation results are presented and Section V presents the discussion. Subsequently Section VI concludes this paper.

II. RELATED WORK

The relocation of wireless sensor nodes is thoroughly studied in [11]. Some algorithms allow movement on-demand while others allow post-deployment movement. In hostile environments, designers cannot place the node in most effective areas by hand. Nodes are randomly deployed by aerial deployment. Some areas may receive a dense amount of nodes while some areas may remain vacant. As a result, nodes are deployed inefficiently and nodes relocation or deploying more nodes might be considered.

A number of approaches have only focused on the connectivity aspect. Different techniques are proposed to maximize nodes coverage without affecting connectivity. In [16] the authors proposed a distributed algorithm capable of restoring network connectivity in case of node failure. The mobility of nodes is exploited and for the better coverage of area, the repelling forces idea is proposed among. However, the technique does not restore network disjoint issues originated by failed nodes. In robot networks, a similar technique is studied for the maintenance of connectivity [12]. The 2-connected network concept is used which means that there should be minimum of two pathways among each pair of nodes. The objective of this technique is to achieve a 2-degree of connectivity in case of a node failure. In this technique, a pair of sensor nodes is moved in such a way that 2-connectivity is restored in case of a failure of a node. The techniques that resemble our work are explained below. C^2AP is a technique that considers post-deployment coverage and connectivity and increases coverage by spreading inter-

connected nodes [13]. A hierarchical architecture is proposed in COCOLA [14], where coverage is maximized without forwarding data path to 1-tier node by the incremental relocation of higher-tier nodes. Neither C^2AP nor COCOLA deals with the implications related to failed nodes.

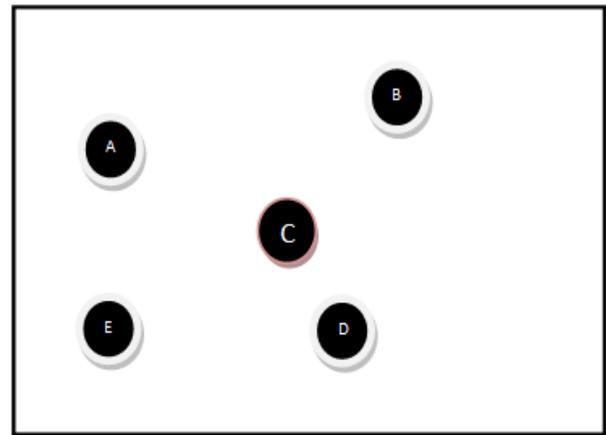
In [15], a cascaded movement technique is proposed for the recovery of node failure. In this technique, a failed node is replaced with the nearby node, which is then replaced by another node and this process continues until reaching to a redundant node. The techniques more related to our work are C^3R [18], RIM [16] and AUR [19]. C^3R assumes single or multiple neighbor nodes failure. The failed node is substituted by each neighbor node temporarily and gets back to its original position after spending limited time at a new location. Another approach named DARA [17] considers a probability scheme for the identification of cut vertices and selects a neighbor node to the failed node for relocation based on the number of communication links. In [16], a multimode repositioning technique called RIM (Recovery by Inward Motion) is considered in which all the neighbor nodes of a failed node temporary relocate it. RIM requires just 1-hop information. It is a scheme based on localized information about the neighbors. The main idea is to move the neighbors of a failed node inward towards the position of the failed node so that they would be able to reach each other. The main idea is that these neighbors are the ones directly impacted by the failure, and when they can reach each other again, the network connectivity would be restored to its pre-failure status. The relocation procedure is recursively applied to handle any node that gets disconnected due to the movement of one of their. RIM is known for its simplicity and effectiveness. However, the main drawback of RIM is that under higher node densities it tends to move many nodes and increases the total travel overhead on the network. Moreover, it is not very effective in case of multi-node failure.

In [19], the authors proposed a new approach called Autonomous Repair (AuR). AuR is based on repositioning of nodes towards the center of the deployment area. The design principle of AuR is based on modeling connectivity between neighboring nodes as a modified electrostatic interaction based on Coulomb's law between charges. In AuR, the recovery is based on the nodes having only localized information about the immediate neighbors. The neighbors of the failed nodes detect the failed nodes and lead the recovery process by spreading out towards the lost nodes, causing the intra-segment topology to be stretched. If connectivity is not restored, the segment is then moved as a block towards the center of the deployment area. Moving all segments towards the center will increase the node density in the vicinity of the center point and ensures that the connectivity gets reestablished. However, none of the above mentioned works considers connectivity, coverage, and energy efficiency collectively. Our work addresses connectivity restoration, better coverage, and efficient use of energy in an integrated manner.

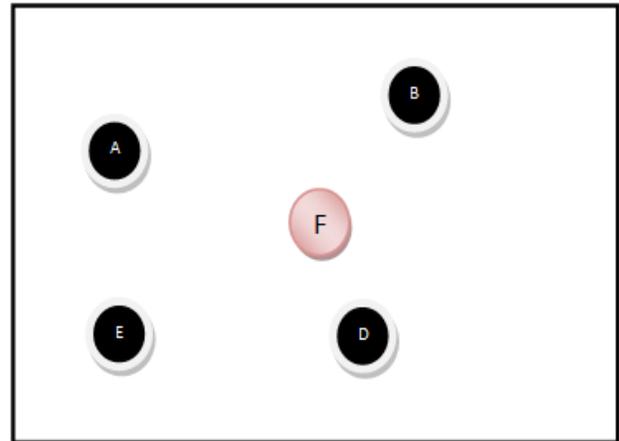
III. DEENR ALGORITHM DESCRIPTION

DEENR is a distributed algorithm with an objective to restore connectivity for sensor networks. For our proposed

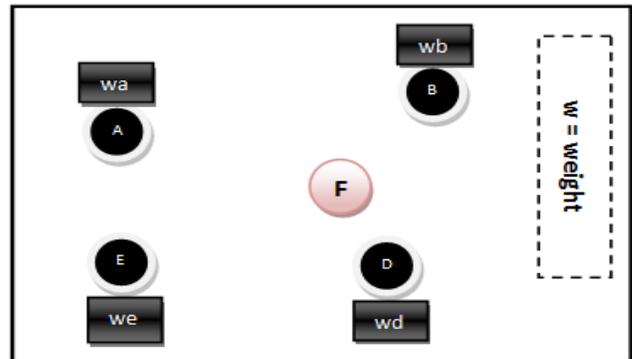
algorithm we assume that all the sensor nodes are densely deployed in an integrated manner and their sensing and communication ranges are equal. The proposed algorithm is presented in Table I. All the nodes are deployed randomly in an area of interest and upon deployment the nodes discover each other. Nodes send periodic hello messages to all their neighbors. In this way each node maintains a neighbor list. When a node dies due to drainage of the battery, it will not send the hello message. In this way the neighboring nodes know about the failure of a certain node. When a node identifies a failed node within its vicinity, it consider itself a potential candidate node to take part in the recovery process and calculates a weight based on the closeness from the failed node and its current residual energy. After calculation of weight, each node sets a timer based on the calculated weight as shown in line 20 of algorithm. The timer is set in a way such that a node having a more weight sends the broadcast message before the nodes having lesser weights. In this way the number of messages transmitted by the nodes can be reduced. Upon receiving a message containing a weight, each node analyzes the weight and compares the received message weight with its current weight. Receiving a weight higher than its own weight means that a more suitable node is available in the vicinity of the failed node. This enables our algorithm to refrain from cascaded relocation which has been proven to be detrimental for the network in terms of network coverage and energy efficiency. After the selection of the appropriate node that will take part in the recovery process, the next step is movement of that node towards the failed node. The node that is selected as the recovery node for the failed node calculates the maximum distance from itself from all of its neighbors. If this distance is greater than $R_c/2$, then this node has the potential to move a maximum of $R_c/2$ towards the failed node without disturbing the network topology or doing a cascaded movement. Else if the distance is less than $R_c/2$, it means that the node can move towards the failed node with a maximum distance of the R_c minus the maximum distance from the neighboring node. Therefore this node moves towards the failed node with this distance. Then it stays at this place until it receives hello messages from the neighbors of the failed node. Upon receiving these messages if the node determines the recovery was successful then it stays at the current position. If not, then it has to rely on cascaded relocation of the neighboring nodes for the recovery. The initial weight calculation for our algorithm is depicted in Fig. 1. Fig. 1(a) shows the initial simple topology having 5 nodes. Now let's suppose that node C fails as shown in Fig. 1(b). The node failure of node C will be found by absence of periodic hello message from node C. Upon detecting the failure of node C, all the nodes that are neighbors of node C will compute a weight on the basis of their current energy level and the closeness to node C. These weights are represented by w_a , w_b , w_d , and w_e respectively as shown in Fig. 1(c). On the basis of these weights, each node calculates a timer for sending a broadcast message containing the computed weight as shown in Fig. 1(d). If the node having the highest weight does not receive any broadcast message having more weight than itself then it means that this node is the suitable candidate node for relocation. Therefore as shown in Fig. 1(e), this node moves towards the failed node according to the proposed algorithm.



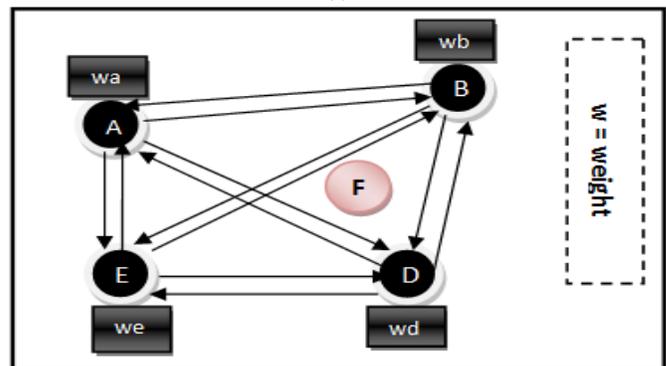
(a)



(b)



(c)



(d)

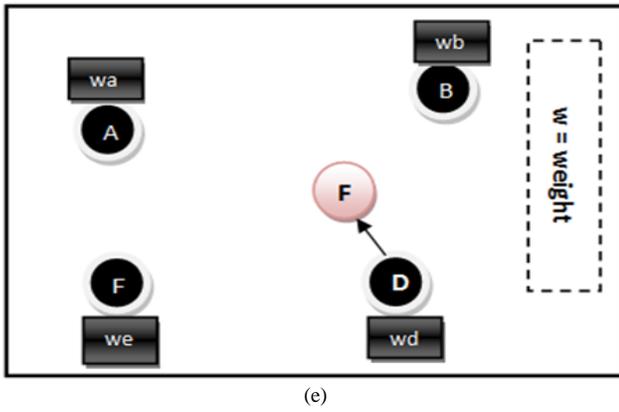


Fig. 1. (a) Initial topology, (b) Node failure, (c) Calculation of weights, (d) Broadcast message, (e) Node relocation.

A. Energy Model

We have assumed the energy model depicted in [20] 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:

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

where 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:

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

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:

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

TABLE I. DISTRIBUTED ENERGY EFFICIENT NODE RELOCATION ALGORITHM (DEENR)

Input: Area, Rc, n	
1	begin // Randomly deploy all nodes in the network
2	For each node N_i ,
3	Send Hello message()
4	Maintain neighbor list L_i
5	For each end
6	For each node N_i do
7	For each neighbor
8	If $E_{n_j} = 0$ // Means n_j is dead
9	Calculate $weight(N_i, n_j)$
10	SendDeadNodeBroadcastMessage(n_i, N_i)
11	end
12	SendDeadNodeBroadcastMessage(n_j, N_i)
13	begin
14	For each node N_j receiving this message

15	If n_i is present in neighbor list // means that n_i is a common neighbor
16	Add N_i as candidate node for recovery
17	$weight_{i,j} = (\alpha \times energy_i) + (\beta \times distance_from_n_i)$
18	Where $\alpha + \beta = 1$
19	Before Moving \rightarrow send Broadcast msg containing weight.
20	Broadcast Timer = $1/weight \times seconds$
21	end
22	Calculate $weight(N_i, n_j)$
23	begin
24	$W_{i,j} = \alpha * energy_i + \beta * distance_from_n_j$
25	end
26	MoveTowardsFailedNode()
27	begin
28	Set $max=0$;
29	/* Calculate the maximum distance of the current node from all of its neighbors */
30	For each node N_i do
31	For each neighbor
32	If $distance_{i,c} > max$
33	$max = distance_{i,c}$
34	//end if
35	//endforeach
36	//endforeach
37	If $max < Rc/2$
38	Move the node with distance $Rc/2$ towards failed node
39	Else
40	Move a distance max towards the failed node
41	Wait for broadcast hello messages from neighbors
42	If recovery successful
43	Stay at the current position
44	Else
45	Cascaded relocation until recovery

IV. RESULTS AND DISCUSSION

Randomly dense deployed WSN topologies are involved in the experiment with varying communication ranges and number of nodes. The number of nodes has been set to 100, 150, 200 and 250 in the field with dimensions of $900 \times 900 m^2$. However, for our algorithm DEENR, experiments were conducted by varying the sensing and communication ranges. The initial energy of every node has been set to 100 joules avoiding the energy consumption in initial relocation. Table II summarizes the simulation parameters used during simulation.

Consumption of energy occurs due to communication, sensing, and movement. Each experiment is done 20 times and then result is averaged. All the results are subjected to 88% confidence analysis interval and stay within 12% of simple mean. The results are compared with baseline algorithms RIM [6], C^3R [8], and AUR [9]. The major difference between our algorithm and the baseline algorithms is that the former describes nodes permanent relocation with less traveled distance.

TABLE II. SIMULATION PARAMETERS

Simulation parameters	Value
Simulation Area	$900 \times 900 m^2$
Number of nodes	100 - 250
R_c	25 - 150 m
Simulation tool	OMNeT++

A. Total Distance Moved during Relocation

Fig. 2 presents the total distance moved by all nodes until the restoration of connectivity is done. Our algorithm outperforms all the other algorithms because it moves only non-critical nodes in order to avoid cascaded relocation. The improvement in performance by our algorithm remains consistent even by increasing node communication ranges and densities. This is because our algorithm avoids the movement of critical nodes which causes further partitioning of the network. Furthermore, our algorithm implements cascaded relocation only when non-critical neighbor nodes fail.

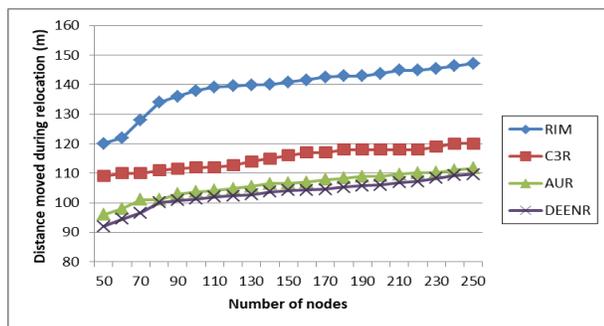


Fig. 2. Node relocation.

B. Number of Nodes Moved

Fig. 3 presents the average number of nodes moved while connectivity restoration. The simulation results confirm the advantage of DEENR which moves fewer nodes as compared to other algorithms. The main reason behind this is that DEENR limits the recovery scope and avoids continuous and cascaded relocation. Moreover, the average number of nodes moved for the case of DEENR increases less significantly as compared to RIM, C3R, and AUR which shows the great scalability that our algorithm can achieve.

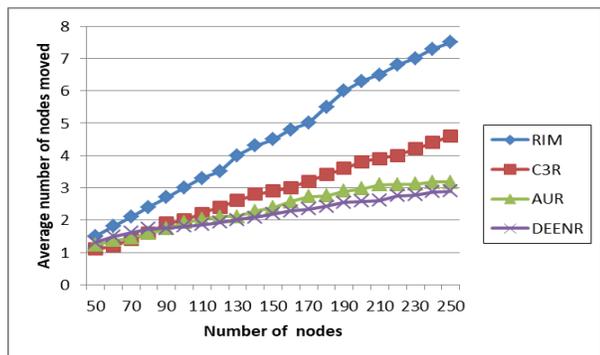


Fig. 3. Average number of nodes moved during recovery.

C. Reduction in Field Coverage

Fig. 4 show how coverage is influenced by analyzing the percentage reduction in field coverage. It can be seen from the figure that increasing communication range decreases the percentage reduction in the field coverage for all the protocols. However, our protocol yields the least percentage reduction in the field coverage as compared to the other protocols. In general, DEENR efficiently limits the coverage loss; the nodes having less coverage overlap area in sparse networks. The field coverage under our algorithm is much better as compared

to the baseline algorithms. Field coverage is highly reduced in the case of RIM. In the case of proposed technique, the overlapped coverage is higher due to which substitute nodes have to move a smaller distance. When a node is relocated, its neighbor nodes mostly cover the home area. In sparse networks, less number of nodes are eligible for substitution of failed node and the larger region remains uncovered during this relocation.

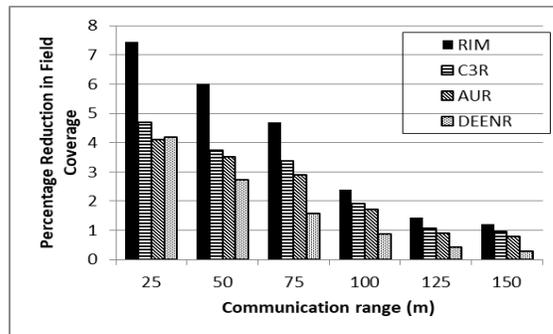


Fig. 4. Percentage reduction in field coverage.

D. Number of Exchanged Packets

Fig. 5 presents the average number of packets exchanged during restoring connectivity both under proposed technique and baseline techniques. Every broadcast is considered as a single message. In case of the proposed technique, the messaging overhead is the minimum, while RIM exchanges maximum number of packets. This is because in proposed algorithm only neighbors are engaged in relocation. As it is proven that reducing the number of messages during the operation of the protocol leads to achieving energy efficiency [5], therefore DEENR algorithm proves to be energy efficient as well as more scalable as compared to the baseline protocols. In comparison with the baseline protocols, there is a substantial difference in terms of total number of packets exchanged.

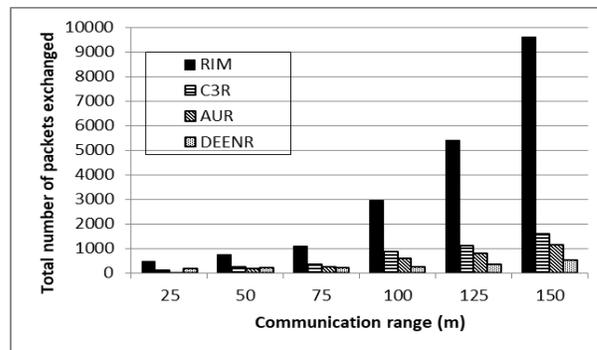


Fig. 5. Total number of exchanged packets.

Connectivity restoration is critical for the operation of sensor networks. A technique is desired that is capable of not only restoring the connectivity but also be coverage-aware and energy efficient. Many techniques proposed in the literature focuses on one of the above features but not all at once. Our goal was to design a technique capable of achieving all the above goals. Achieving connectivity restoration by making the nodes to move the minimum possible distance, making less

number of nodes to move, reducing the decrease in field coverage, and achieving energy efficiency is the primary goal of every connectivity restoration technique. We developed our solution by keeping all these goals in mind. With the help of extensive simulations, we compared the performance of our proposed technique with the baseline techniques and concluded that our technique outperforms the baseline techniques in terms of all the above mentioned performance metrics.

V. DISCUSSION

It can be observed from the presented results that the proposed protocol outperforms some of the well-known protocols proposed in the literature. A realistic simulation evaluation was performed during this work and the comparison with the existing protocols is also presented. However, there are a few aspects that can be studied in future and are out of scope of the current work. First of all, as connectivity restoration is one of the most studied topics in sensor networks therefore recently a lot of new approaches have been presented in literature. There is a need to compare the performance of the proposed protocol with the more recent approaches. Secondly, a more realistic communication model incorporating the effect of propagation and other factors can also be considered for the performance evaluation. Last but not least, a more realistic mobility model can be considered for the performance evaluation. One option can be to use the real world traces for mobility models because they will make the performance evaluation more realistic. All of these questions are open for research and can be considered by the researchers as a possible extension of the current work.

VI. CONCLUSION

In this work, we presented a novel technique capable of providing a solution to the connectivity restoration. The proposed technique is distributed and energy efficient and with the help of extensive simulations, the effectiveness of the proposed technique is proven. Simulation results revealed that the proposed technique requires less number of nodes to be moved for connectivity restoration. The distance that nodes have to move is also less as compared to existing techniques. Moreover, the proposed technique also does not significantly affect the percentage of reduction of the coverage of the field and it is also more energy efficient as compared to other existing techniques. As a future work, a more realistic communication and mobility model can be considered for evaluation of the proposed protocol.

REFERENCES

- [1] Lee, Sookyoung, Mohamed Younis, and Meejeong Lee. "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance." *Ad Hoc Networks* 24 (2015): 1-19.
- [2] Bhuiyan, MdZakirulAlam, et al. "Deploying wireless sensor networks with fault-tolerance for structural health monitoring." *IEEE Transactions on Computers* 64.2 (2015): 382-395.
- [3] Yick, Jennifer, Biswanath Mukherjee, and DipakGhosal. "Wireless sensor network survey." *Computer networks* 52.12 (2008): 2292-2330.
- [4] Ranga, Virender, Mayank Dave, and Anil Kumar Verma. "Node Stability Aware Energy Efficient Single Node Failure Recovery Approach for WSANs." *Malaysian Journal of Computer Science* 29.2 (2016).
- [5] Sharma, Abhinav, and Sandeep Sharma. "A Comparative Review on Reliability and Fault Tolerance Enhancement Protocols in Wireless Sensor Networks." (2016).
- [6] Senouci, Mustapha Reda, and AbdelhamidMellouk. *Deploying Wireless Sensor Networks: Theory and Practice*. Elsevier, 2016.
- [7] Bao, Lichun, and Jose Juaquin Garcia-Luna-Aceves. "Topology management in ad hoc networks." *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2003.
- [8] Abbasi, Ameer Ahmed, and Mohamed Younis. "A survey on clustering algorithms for wireless sensor networks." *Computer communications* 30.14 (2007): 2826-2841.
- [9] Akkaya, Kemal, and Mohamed Younis. "Coverage and latency aware actor placement mechanisms in WSANs." *International Journal of Sensor Networks* 3.3 (2008): 152-164.
- [10] Akkaya, Kemal, and Mohamed Younis. "C2AP: Coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks." *2007 IEEE International Performance, Computing, and Communications Conference*. IEEE, 2007.
- [11] Younis, Mohamed, and Kemal Akkaya. "Strategies and techniques for node placement in wireless sensor networks: A survey." *Ad Hoc Networks* 6.4 (2008): pp. 621-655.
- [12] P. Basu and J. Redi, "Movement Control Algorithms for Realization of Fault-Tolerant Ad Hoc Robot Networks." *IEEE Networks*, vol. 18, no. 4, pp. 36-44, Aug. 2004.
- [13] Akkaya K, Younis M, "Coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks," In: *Proceedings of the 26th IEEE international performance computing and communications conference (IPCCC 2007)*, New Orleans, pp. 281-288
- [14] Akkaya K, Younis M. "Coverage and latency aware actor placement mechanisms in wireless sensor and actor networks." *International Journal of Sensor Networks special vol.3, Issue.3*, pp. 152-164
- [15] Wang G, Cao G La Porta T, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, Issue. 6, pp. 640-652, 2006
- [16] Younis M, 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, and November 2008.
- [17] Abbasi A, Akkaya K Younis M, A distributed connectivity restoration algorithm. *Wireless sensor and actor networks*. In: *Proceedings of the 32nd IEEE conference on local computer networks (LCN 2007)*, Dublin, Ireland, pp. 1-5, October 2007.
- [18] Tamboli, Neelofer, and Mohamed Younis. "Coverage-aware connectivity restoration in mobile sensor networks." *Journal of Network and Computer Applications* 33.4 (2010): pp. 363-374.
- [19] Y. K. Joshi and M. Younis, "Autonomous recovery from multi-node failure in Wireless Sensor Network," *2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, 2012, pp. 652-657.
- [20] Chanak, Prasenjit, Indrajit Banerjee, and R. Simon Sherratt. "Energy-aware distributed routing algorithm to tolerate network failure in wireless sensor networks." *Ad Hoc Networks* 2017: 158-172.