Deployment Algorithms to Simulate Large-scale Node Failures in Wireless Sensor Networks

Izzet Fatih Senturk
Dept. Computer Engineering
Bursa Technical University
Bursa, TURKEY
izzet.senturk@btu.edu.tr

***Abstract*—Wireless sensor networks (WSNs) enable monitoring surrounding physical phenomena in inhospitable environments through employing low-power sensor nodes with limited transmission range. A less resource-restricted base station (BS) provides long-range wireless communication to connect the network with the remote user. Within the network, nodes form a multi-hop network to reach the BS. However, some of the nodes may fail arbitrarily and impair the network connectivity. Depending on the network topology and the damage scale, network can be divided into disjoint subsets where some of the nodes are isolated from the rest of the network. Consequently, data collected in remote partitions cannot be delivered to the BS and the coverage drops drastically. Such failures can be tolerated with one of the existing connectivity restoration algorithms. However, despite abundance of self-configuring fault-tolerance schemes, research on the relationship between the deployment scheme and the recovery cost is limited. This paper presents three different node deployment schemes to simulate large-scale node failures which lead to partitioning. We have also investigated the impact of deployment schemes on the cost of recovery.**

***Index Terms*—WSN, fault tolerance, deployment, connectivity, k-means, repulsive force**

I. Introduction

Proliferation of microelectromechanical systems (MEMS) has enabled WSNs featuring tiny devices with sensing, data processing, and wireless communication capabilities [1]. Deployment of WSNs in remote areas where human intervention is dangerous, paved the way for a wide array of applications including crop protection [2], environmental monitoring [3], irrigation facilities management [4], aquatic environmental monitoring [5], etc. In these applications, sensor nodes form an ad-hoc network and operate autonomously. Despite possibility of energy harvesting [6], nodes adapt low-power methods for radio communication in order to extend their lifetime.

Limited transmission range of the nodes entails employment of a BS less restricted in terms of communication range and computational power. Instead of transmitting their data to the BS directly, nodes collaborate to send their data by forming a multi-hop network to reach the BS. Therefore, it is important to maintain network-wide connectivity to ensure sustained coverage and data fidelity at the application layer. Though, some of the nodes may stop functioning arbitrarily due to various reasons including limited on-board batteries and harsh environmental conditions which may inflict widespread damage in the network.

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Grant No. EEEAG-117E050.

k-connectivity may tolerate failure of up to k-1 nodes by providing k independent paths between every pair of nodes. However, ensuring k-connectivity leads to redundancy and increases the hardware cost of deployment. Moreover, proactive measures may not be sufficient to sustain connectivity against large-scale damages. Failure of cut-vertex nodes partitions the network into disjoint sets of nodes as illustrated in Fig. 1. In such cases, data collected in remote partitions cannot be delivered to the BS and the coverage drops drastically.

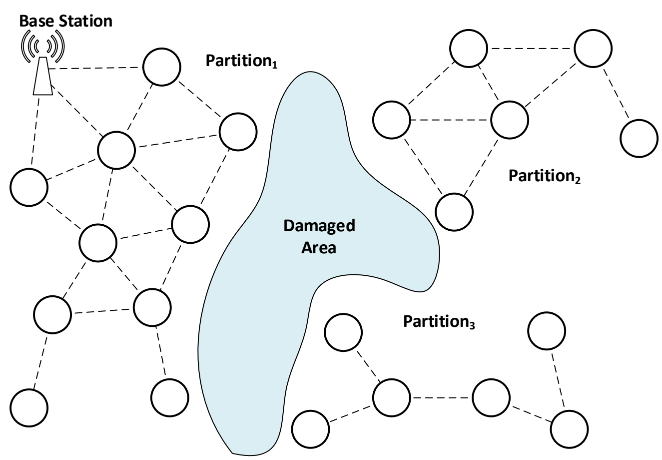

Fig. 1. Damage partitions network into disjoint groups isolated from the rest of the network.

Despite several connectivity restoration solutions to recover connectivity, deployment algorithms to simulate large-scale node failures attracted limited attention. To the best of our knowledge, this is the first study which investigates the impact of deployment schemes on the cost of recovery. In this paper, we have presented three different algorithms to deploy partitioned networks to simulate node failures. The first algorithm deploys a connected component and applies k-means algorithm to identify clusters. Then the network is partitioned according to the obtained clusters. The second algorithm is similar to the first one but strives to minimize redundancy by applying repulsive force on the nodes. The nodes are regarded as magnets with the same magnetic pole which push each other if they are neighbors. The resulting topologies are more uniformly distributed compared with the first one. The last algorithm deploys multiple connected components step-by-step. We have employed a centralized heuristic to restore connectivity on the generated topologies and considered total travel distance and the number of relocated nodes to investigate the relationship between the deployment scheme and the recovery cost.

The rest of the paper is organized as follows. Related work is summarized in Section II. Deployment schemes are presented in Section III. Approaches are evaluated in Section IV. The paper is concluded in Section V.

II. Related Work

A. Connectivity Restoration Solutions

Self-configuration is regarded as autonomous maintenance of WSNs after deployment. Various issues can be addressed through self-configuration including node scheduling [7], coverage and connectivity configuration [8], fault-tolerance [9], etc. In this paper, we focus on self-configuring fault-tolerance schemes which aims to tolerate node failures. Fault-tolerance solutions can be classified into groups based on the resource provisioning time. While proactive approaches take preventive measures such as providing k-connectivity [10], reactive approaches provide demand-based recovery [11]. Considering the unpredictability of the damage scale and location, proactive measures cannot guarantee a solution at all times.

Reactive solutions can also be classified based on the adopted method. Some solutions assume mobility of the nodes and restructure network topology by relocating nodes [12], [13]. The goal in such approaches is minimizing the movement cost. On the other hand, some approaches assume possible intervention to the application area and deploy additional nodes to re-establish connectivity among partitions [14], [15]. If the number of nodes deployed in the application area is not sufficient to connect partitions, it is possible to employ new nodes as mobile data collectors which visit partitions one-by-one and relay data to the BS [16].

B. Deployment Algorithms

Node deployment is a fundamental issue which needs to be addressed in WSNs. Nodes can be deployed in a random fashion or in a deterministic manner [17]. Random deployment sets the positions of the nodes randomly and independently. Consider a scenario where the nodes are randomly dropped from a drone to a hostile environment. Despite the simple application, random deployment requires more nodes to be deployed to provide the same coverage compared with the deterministic deployment. Deterministic deployment can optimize the deployment layout by considering various goals such as coverage, lifetime, energy consumption, and connectivity [18] and place nodes at precise locations.

In this paper, we apply a hybrid approach as the node deployment strategy. Node positions are determined randomly but not independently. Thus, we ensure that the nodes to be deployed form connected components as many as desired. As explained in Section III, we pick a node in the application area based on some rules and deploy the next node within the transmission range of the selected node according to the employed deployment scheme. We also apply repulsive force between nodes in one of the proposed deployment schemes. Application of virtual forces has been a technique exploited by earlier studies as well [15], [19].

Earlier work on node deployment algorithms which simulate large-scale node failures is limited despite availability of various deployment solutions which address connectivity, coverage, energy consumption, delay, etc. To the best of our knowledge, this is the first study which investigates the relationship between deployment schemes and the recovery cost of the partitioned WSNs.

III. Approach

In this section, we present three different algorithms to form partitioned networks in order to simulate large-scale node failures which create multiple disjoint partitions in the network. Due to page limits, we skip the formal algorithms of the proposed deployment schemes. Instead, the formal algorithm to form a single connected component is given in Algorithm 1 briefly. Algorithm 1 is a major part of the proposed deployment schemes which we will discuss in the following subsections.

Algorithm 1 deploySingleSegment($numNodes$, G, TR, $Height$, $Width$, $damageScale$)

for $i = \{0, 1, \ldots, numNodes - 1\}$ **do**
    x = y = -1, $A[numNodes]$;
    while (x, y) is within distance of $TR * damageScale$ to any node in G || (x, y) $\notin$ ApplicationArea **do**
        if $i == 0$ **then**
            x = Random.Next($Height$)
            y = Random.Next($Width$)
        else
            g = getNodeWithTheLeastDegree(A)
            $radian$ = Random.Next(360)*Pi/180
            $radius$ = Random.Next(TR)
            x = Cosinus($radian$)*$radius$ + g.getX()
            y = Sinus($radian$)*$radius$ + g.getY()
        end if
    end while
    $A[i]$ = new Gateway(x, y))
end for
add A to G

A. Single Connected Deployment then k-means (SDkM)

$SDkM$ deploys a single connected component in the given application area by employing Algorithm 1 and then partitions the network. In order to designate partitions, $SDkM$ employs k-means clustering algorithm. k is set to the number of partitions desired to reach. Output of the k-means algorithm is k sets of nodes but the network is still connected. To partition the network, we pick two clusters C_1 and C_2 and pick the node $n_1 \in C_1$ such that n_1 is the closest node to C_2. If n_1 is a cut-vertex node, then we employ block movement and move C_1 as whole towards the opposite direction of C_2. Otherwise, we simply relocate n_1 next to the furthest node in C_1 randomly. The idea is increasing the minimum distance between two clusters with every movement. Eventually, the minimum distance between the clusters will be greater than the transmission range and the partitions will be formed. We also define damage scale as a factor of the transmission range to represent different scales of damages. The procedure is applied in an iterative manner until forming the partitions apart from each other according to the given damage scale.

B. Single Connected Deployment with Force then k-means ($SDkM$-F)

$SDkM$-F diversifies from $SDkM$ by the application of the repulsive force before partitioning. $SDkM$-F populates a single connected component as $SDkM$. However, before proceeding with the partitioning, $SDkM$-F applies repulsive force on the nodes. The idea is simulating the magnetic forces in Physics. The nodes are assumed to have the same magnetic pole and push each other if they are neighbors. The force to be applied decreases with the increased distance between the nodes. When multiple neighbors are present, total force is calculated and applied accordingly. Application of the repulsive force before the partitioning phase enables a network of uniformly distributed nodes and avoids node redundancy.

C. Multiple Connected Deployments (MD)

MD also employs Algorithm 1 to deploy a single connected component. But unlike $SDkM$ and $SDkM$-F, MD calls Algorithm 1 multiple times for each partition to be formed. Algorithm 1 ensures that each partition is deployed apart from other partitions with a minimum distance computed based on the supplied damage scale.

IV. Experimental Evaluation

A. Experiment Setup

We have investigated the relationship between the presented deployment algorithms and the recovery cost through simulations. We have considered an application area of 600 meters $\times$ 600 meters for node deployment. Nodes and the BS have the same transmission range of 30 meters. We have exploited three different parameters (i.e. the number of nodes, the number of partitions, damage scale) by varying them to examine their impact on the recovery cost. First, we set the number of partitions to 3 and varied the number of nodes between 50 and 200. Then we set the number of nodes to 100 and varied the number of partitions between 2 and 5. The damage scale, on the other hand, was set to 4 for both cases. Finally, we have varied the damage scale between 1 and 4 while considering different number of nodes and partitions. For each case, 50 different topologies were created and tested for significance and the average is reported.

B. Connectivity Restoration Heuristic

In this paper, we have assumed mobility of the nodes as in mobile sensor networks and employed a mobility-based connectivity restoration algorithm. However, it should be noted that the generated topologies can also be used to evaluate relay placement algorithms. We have considered a distance-based centralized heuristic ($DiCH$) which picks the next partition to be connected based on the total distance to other partitions. Since the BS is typically stationary, $DiCH$ only considers partitions without BS for relocation. Once the partition (i.e. P_{min}) is selected, $DiCH$ determines the closest partition (i.e. P_{target}) as the target partition for connection. Note that, BS partition can also be P_{target}. Then the closest pair of nodes (i.e. $n_1 \in P_{min}, n_2 \in P_{target}$) between P_{min} and P_{target} is determined. Then n_1 moves towards n_2 and stops when the distance is equal to TR. The same procedure is repeated with updated n_1 and n_2 until P_{min} is connected to P_{target}. Recovery proceeds with the next partition until network-wide connectivity is ensured.

C. Performance Metrics

- *Total travel distance*: This metric measures total distance traveled by the nodes which are relocated as part of the recovery. Considering the excessive energy cost of mechanical motion, total travel distance should be minimized to extend the lifetime of the network.
- *The number of relocated nodes*: This metric indicates the scope of recovery. When the movement of a single node is not sufficient to ensure recovery, multiple nodes will be relocated in a cascaded manner. However, relocation of each node may create further partitions and it is desired to limit the number of nodes involved in recovery to enhance the network lifetime.

D. Performance Results

1) Total Travel Distance: Figs. 2, 3, 4, 5 present recovery cost in terms of total travel distance with respect to number of nodes, number of partitions, and damage scale. According to Fig. 2, the least recovery cost can be attained on topologies created by $SDkM$. Despite similar behavior, $SDkM$-F performs worse than $SDkM$. Unlike $SDkM$ and $SDkM$-F, recovery cost declines considerably on denser topologies generated by MD. This behavior can be attributed to the fixed damage scale and individual deployment of multiple partitions by MD. Recall that, the damage scale is also applied to MD. However, damage scale sets the minimum distance between partitions. In denser networks, on the other hand, maximum distance between partitions is also expected to decline for MD.

Fig. 3 suggests that the total travel distance increases for all deployment schemes when the number of partitions is increased. This is due to the increased likelihood of encountering partitions with further distances when the number of considered partition count is higher. Again, $SDkM$ generates topologies with the least cost. For MD, cost increase slows down after 4 partitions. If the number of partitions is increased further, considering the fixed size of the application area, we expect the cost to increase up to a certain point and then decline as low as the distance set by the damage scale.

Considering the page limit and to improve clarity of presentation, reported results are limited with $SDkM$ and MD in Figs. 4 and 5. Varying damage scale is denoted with DS following the deployment scheme's name in the figures. As can be observed from Fig. 4, recovery cost increases with the extended damage scale as expected. Despite some fluctuation for MD, cost increase gains momentum with the extended

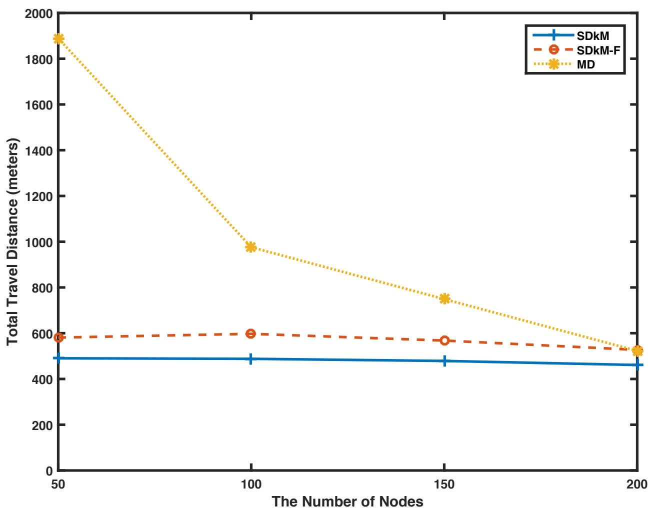

Fig. 2. Total travel distance with respect to network size. The number of partitions is set to 3. The damage scale is set to 4.

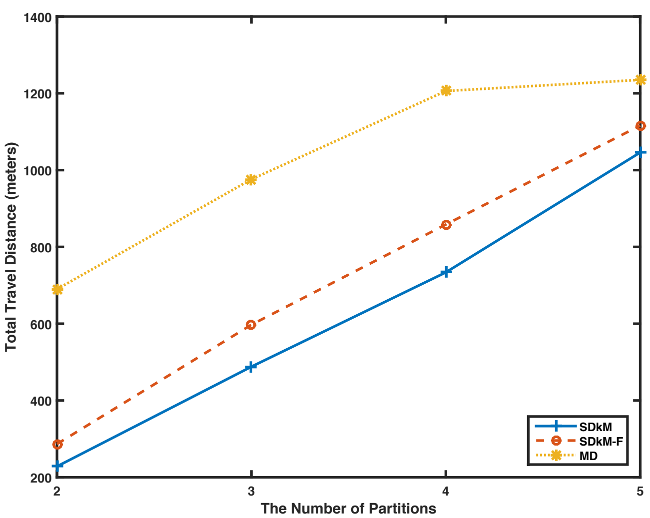

Fig. 3. Total travel distance with respect to the partition count. The number of nodes is set to 100. The damage scale is set to 4.

damage scale when $SDkM$ is employed. Recall that the damage scale provides the minimum distance to be considered between the partitions. Since the initially connected network is partitioned with $SDkM$, one may expect the nearby nodes to be within a certain distance which limits the maximum distance. On the other hand, MD deploys multiple partitions individually in a random fashion and the maximum distance between them is expected to be greater than in $SDkM$ due to random deployment.

Fig. 5 indicates that the total travel distance increases with the increased damage scale and the partition count. Note that both damage scale and partition count have an adverse effect on the recovery cost. Again, results fluctuates for MD and the cost increase accelerates when the damage scale is extended for $SDkM$. Fluctuations in MD is due to random deployment of individual partitions as justified earlier.

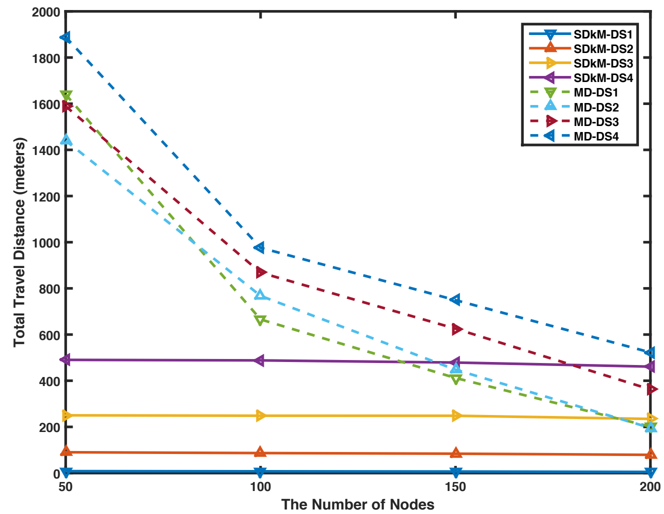

Fig. 4. Total travel distance with respect to network size and the damage scale. The number of partitions is set to 3.

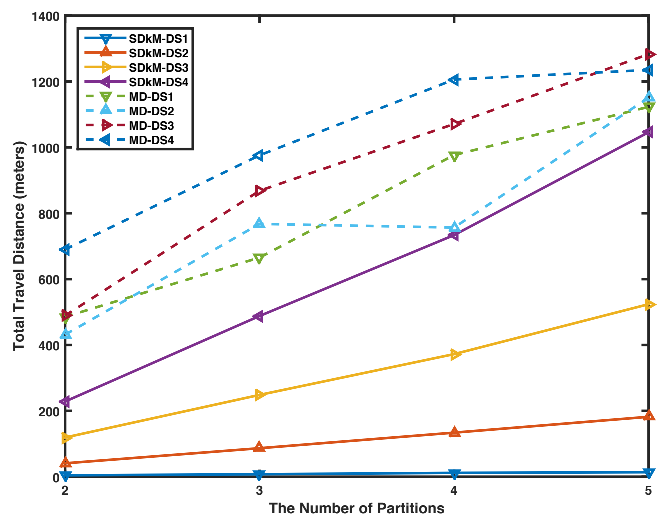

Fig. 5. Total travel distance with respect to the partition count and the damage scale. The number of nodes is set to 100.

2) The Number of Relocated Nodes: Figs. 6, 7, 8, 9 present recovery cost in terms of the number of relocated nodes with respect to number of nodes, number of partitions, and damage scale. According to Fig. 6, $SDkM$ requires the least nodes to be relocated up to 150 nodes. However, despite initial poor performance, MD slightly outperforms $SDkM$ in the densest topologies. The patterns of the performance behaviors for $SDkM$, $SDkM$-F and MD also vary. While $SDkM$ and $SDkM$-F provide slightly fluctuating results, the number of relocated nodes declines with the increased node count when MD is employed. Almost constant cost provided by $SDkM$ and $SDkM$-F can be attributed to the fixed damage scale. On the other hand, maximum distance between the partitions is expected to decrease in denser networks which alleviates the cost for MD.

It can be observed from Fig. 7 that the number of relocated nodes increases with the increased number of partitions. This

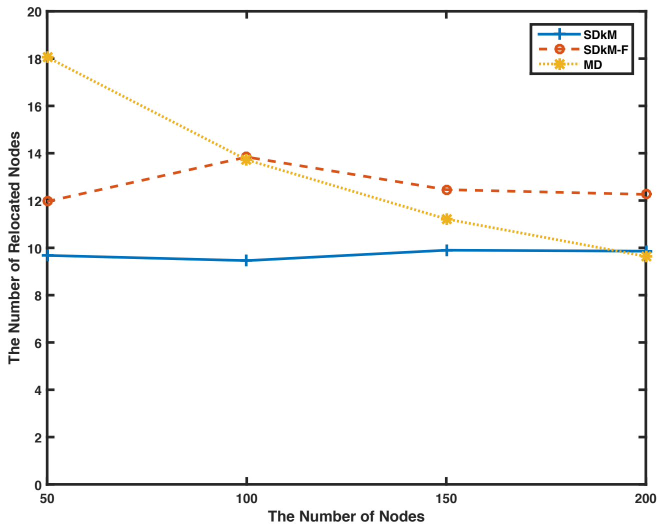

Fig. 6. The number of relocated nodes with respect to network size. The number of partitions is set to 3. The damage scale is set to 4.

is expected due to elevated demand for recovery from additional partitions. Topologies generated by $SDkM$ requires the least recovery involvement while $SDkM$-F and MD provide almost similar results. Recall the total travel distance results where MD incurs higher cost compared with $SDkM$-F. Therefore, it can be concluded that despite involving similar number of nodes, MD leads to longer travel distances compared with $SDkM$-F.

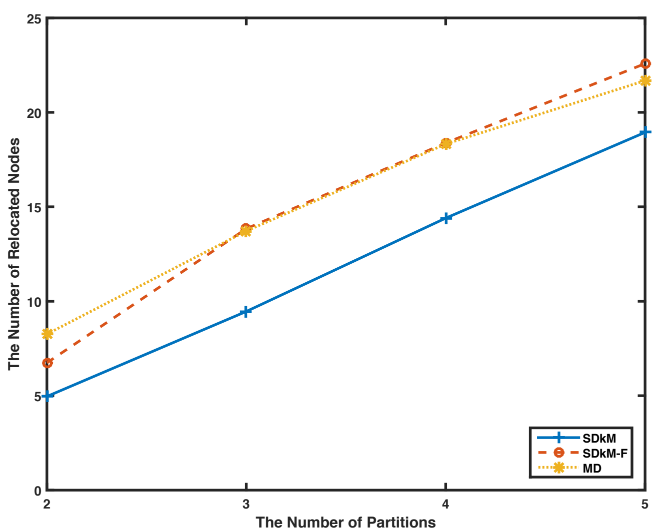

Fig. 7. The number of relocated nodes with respect to the partition count. The number of nodes is set to 100. The damage scale is set to 4.

Again, the results are limited with $SDkM$ and MD in Figs. 8 and 9 to improve clarity of presentation. Fig. 4 indicates that the extended damage scale always leads to higher involvement in recovery independent from the network size. On the other hand, network density alleviates the cost for topologies generated by MD. This is unlike the case of $SDkM$ where the number of relocated nodes is almost constant despite varying network size.

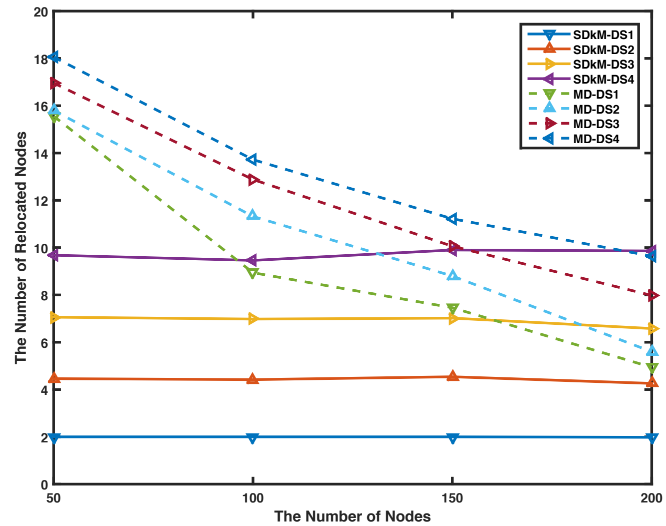

Fig. 8. The number of relocated nodes with respect to network size and the damage scale. The number of partitions is set to 3.

It can be observed from Fig. 9 that the number of relocated nodes increases with the increased number of partitions and the damage scale. This is expected since both partition count and damage scale affect recovery cost adversely. Also, analogous to earlier results, MD requires more nodes to be involved in recovery compared with the $SDkM$.

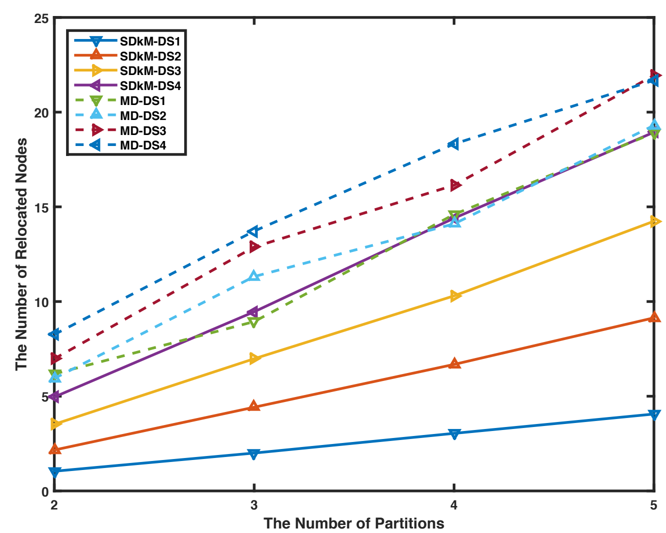

Fig. 9. The number of relocated nodes with respect to the partition count and the damage scale. The number of nodes is set to 100.

V. Conclusion

In this paper, we consider partitioned MSNs and investigate the relationship between the node deployment schemes and the connectivity restoration cost. To simulate large-scale node failures in MSNs, we have presented three different node deployment schemes, namely $SDkM$, $SDkM$-F, and MD. $SDkM$ populates a single connected component and then applies k-means clustering algorithm to designate partitions. Unlike $SDkM$, $SDkM\text{-}F$ simulates repulsive force on nodes like magnets of opposite orientation. Application of the repulsive force enables forming topologies with nodes distributed uniformly. MD, on the other hand, deploys multiple connected components in an iterative manner. MD ensures that each connected component is deployed apart from each other with a minimum distance based on the defined damage scale. To restore connectivity, we have employed a centralized heuristic and evaluated the cost of recovery in terms of total travel distance and the number of relocated nodes. We have observed that the recovery cost pattern is closely related with the employed deployment algorithm. Therefore, novel recovery solutions must be evaluated according to the considered deployment scheme.

References

[1] M. Kocakulak and I. Butun, "An overview of wireless sensor networks towards internet of things," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2017, pp. 1–6.

[2] V. Bapat, P. Kale, V. Shinde, N. Deshpande, and A. Shaligram, "Wsn application for crop protection to divert animal intrusions in the agricultural land," *Computers and Electronics in Agriculture*, vol. 133, pp. 88 – 96, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169916303441

[3] M. Srbinovska, C. Gavrovski, V. Dimcev, A. Krkoleva, and V. Borozan, "Environmental parameters monitoring in precision agriculture using wireless sensor networks," *Journal of Cleaner Production*, vol. 88, pp. 297 – 307, 2015, sustainable Development of Energy, Water and Environment Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959652614003916

[4] W.-H. Nam, T. Kim, E.-M. Hong, J.-Y. Choi, and J.-T. Kim, "A wireless sensor network (wsn) application for irrigation facilities management based on information and communication technologies (icts)," *Computers and Electronics in Agriculture*, vol. 143, pp. 185 – 192, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169917312255

[5] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, "A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring," *IEEE Sensors Journal*, vol. 11, no. 1, pp. 45–55, Jan 2011.

[6] F. K. Shaikh and S. Zeadally, "Energy harvesting in wireless sensor networks: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041 – 1054, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032115012629

[7] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA '02. New York, NY, USA: ACM, 2002, pp. 32–41. [Online]. Available: http://doi.acm.org/10.1145/570738.570744

[8] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 28–39. [Online]. Available: http://doi.acm.org/10.1145/958491.958496

[9] I. F. Senturk, "A prescient recovery approach for disjoint msns," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[10] Z. Yun, X. Bai, D. Xuan, T. H. Lai, and W. Jia, "Optimal deployment patterns for full coverage and k- connectivity (k $\leq$ 6) wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 934–947, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1109/TNET.2010.2040191

[11] I. F. Senturk and K. Akkaya, "Connectivity restoration in disjoint wireless sensor networks using centrality measures," in *39th Annual IEEE Conference on Local Computer Networks Workshops*, Sept 2014, pp. 616–622.

[12] A. A. Abbasi, K. Akkaya, and M. Younis, "A distributed connectivity restoration algorithm in wireless sensor and actor networks," in *Proceedings of the 32Nd IEEE Conference on Local Computer Networks*, ser. LCN '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 496–503.

[13] K. Akkaya, A. Thimmapuram, F. Senel, and S. Uludag, "Distributed recovery of actor failures in wireless sensor and actor networks," in *2008 IEEE Wireless Communications and Networking Conference*, March 2008, pp. 2480–2485.

[14] F. Senel and M. Younis, "Optimized connectivity restoration in a partitioned wireless sensor network," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, Dec 2011, pp. 1–5.

[15] I. F. Senturk, K. Akkaya, and S. Yilmaz, "Relay placement for restoring connectivity in partitioned wireless sensor networks under limited information," *Ad Hoc Networks*, vol. 13, Part B, pp. 487 – 503, 2014.

[16] I. F. Senturk, K. Akkaya, F. Senel, and M. Younis, "Connectivity restoration in disjoint wireless sensor networks using limited number of mobile relays," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 1630–1634.

[17] W. Y. Poe and J. B. Schmitt, "Node deployment in large wireless sensor networks: Coverage, energy consumption, and worst-case delay," in *Asian Internet Engineering Conference*, ser. AINTEC '09. New York, NY, USA: ACM, 2009, pp. 77–84. [Online]. Available: http://doi.acm.org/10.1145/1711113.1711127

[18] S. Sengupta, S. Das, M. Nasir, and B. Panigrahi, "Multi-objective node deployment in wsns: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 405 – 416, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197612001248

[19] X. Yu, W. Huang, J. Lan, and X. Qian, "A novel virtual force approach for node deployment in wireless sensor network," in *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*, May 2012, pp. 359–363.