Distributed Approach for Reconnecting Disjoint Segments

Yatish K. Joshi and Mohamed Younis
Dept. of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250
yjoshi1, younis@umbc.edu

***Abstract*— Due to low-risk and cost-effectiveness, Wireless Sensor Networks (WSNs) have become the primary choice for serving in inhospitable environments like battlefields or security surveillance. In these application setups, nodes operate in harsh conditions and become susceptible to failure. In addition, the environment makes it dangerous and sometime impossible to replace a node that depletes its energy or gets damaged. When multiple nodes fail at the same time the network may get partitioned into disjoint segments and its service may significantly degrade or even cease. Therefore, the network must self-heal using existing resources. The major loss of connectivity and the lack of centralized control leave distributed recovery procedures as the most appropriate option for recovery. In this paper we present DarDs, a distributed approach for reconnecting disjoint segments. The basic idea is to determine the position of the fewest relay nodes that enable the network to restore connectivity. Then, nodes are moved from the individual segments to the designated relay positions such that the total travel overhead is minimized. The performance of DarDs is validated through mathematical analysis and simulation.**

***Keywords:* Topology repair, Fault recovery, Connectivity restoration, Fault-tolerance, Wireless sensor networks.**

I. INTRODUCTION

Due to their low cost and versatility, WSNs have become an attractive choice for data gathering and surveillance applications carried out in inhospitable environments like sea beds, vast borders, space and battlefields. These challenging setups make random deployment the only feasible option for placing the network nodes. After deployment nodes are expected to coordinate with each other and form a network to carry out their designated tasks. Limited node battery life and harsh operating conditions can take their toll and make the network susceptible to large scale damage that causes it to partition into disjoint segments. For example, bombing or missile strikes in a battlefield can destroy many nodes in and around the vicinity of the targeted area. Manual human intervention to recover the network would not be feasible or at least would not be timely; therefore a WSN should have the ability to self-heal by restoring network connectivity and resuming network service.

Multiple strategies have been pursued to tolerate node failures in WSNs with controlled node-placement where node positions are determined prior to network deployment. Some solutions leverage this preplanned positioning and provision redundancy to mitigate individual node failures, either by establishing k-connected topology [1][2], by ensuring k-coverage of important spots [3], or by employing redundant nodes to serve as backups [4][5]. However, such a strategy cannot be adopted in random deployment scenarios therefore autonomous recovery procedures have been proposed for dealing with occasional failure of individual nodes in these scenarios [6][7][8]. But, these approaches rely on the direct neighbors of a failed node to spearhead the recovery and thus cannot handle multiple node failures where the nearest surviving node may be many hops away. Therefore, detecting and tolerating simultaneous failures of multiple collocating nodes requires maintaining p-hop information where p is relatively large or using a centralized recovery procedure. This strategy imposes a significant overhead and is also impractical for large networks. Alternatively, the use of long-range, e.g., satellite links, transmissions is not feasible in most randomly deployed WSNs due to cost and line-of-sight constraints.

This paper proposes a <u>D</u>istributed <u>a</u>pproach for <u>r</u>econnecting <u>D</u>isjoint <u>s</u>egments (DarDs) formed as a result of the failure of multiple collocated nodes in a WSN. DarDs utilizes mobile relay nodes (RNs) for recovering the network. The goal is to relink all disjoint segments by using the least number of RNs. This optimized relay placement problem is equivalent to the Steiner Minimum Tree with Minimal Steiner Points and Bounded Edge-Length (SMTMSPBEL), which is shown to be NP-Hard by Lin and Xue [9]. A number of heuristics have been proposed in the literature for solving the SMT-MSPBEL problem [9][10][11][12]. DarDs uses IO-DT [12], one of the superior approaches for solving the SMT-MSPBEL problem. As DarDs is a distributed approach, the number and location of the surviving segments is unknown after a failure takes place. Therefore, each segment populates RNs towards the center to restore the connectivity. DarDs then applies IO-DT to determine the fewest RNs required for sustaining connectivity and returns the extra RNs to their respective segments such that the travel overhead is minimized. DarDs is validated through simulation and results show that DarDs outperforms competing schemes in terms of the number of required RNs and the travel overhead.

The paper is organized as follows. The next section sets DarDs apart from existing related work in the literature. Section III defines the problem model. Section IV describes DarDs in detail. Section V reports the simulation results. Finally, Section VI concludes the paper.

II. RELATED WORK

A number of approaches have recently been proposed for re-establishing network connectivity after the failure of a single node [6][7][8]. The basic premise for these solutions is for the neighbors of the failed node to collaborate together to restore the pre-failure state by using 1-hop or 2-hop neighbor information. However, these approaches are not suitable as they cannot be scaled to handle simultaneous multiple node failures since the nearest live node may be many hops away and not part of the 1-hop or 2-hop neighbor lists maintained by surviving nodes. This involves maintaining a larger local state which imposes a large overhead increasing exponentially with size of the network.

Centralized approaches for tolerating simultaneous multi-node failures such as IO-DT [12] provide the best results when used to recover WSNs deployed in controlled environments where careful monitoring is performed and full network state is available. However, these centralized schemes may be impractical in remotely-deployed autonomous WSNs where it is expensive to have satellite links available for all RNs, and the links themselves may not be available due to line of sight or equipment malfunction issues and transmission delays and glitches make coordination difficult.

Distributed approaches for recovery from multi-node failures, like AuR [13], reconnect the network by using only 1-hop information via a combination of self-spreading and movement inwards towards the center of the deployed area. AuR though requires all nodes to be mobile and to move as part of the recovery process which may not be applicable to all deployment and application scenarios, where the WSN is composed of a mixture of stationary sensor nodes and mobile relays. Like DarDs, DORMS [14] uses mobile RNs. The idea is to place RNs towards the center of the deployment area along the shortest path to restore connectivity by forming an inter-segment topology at the center.

Once connected DORMS executes an optimization phase which aims to reduce the number of deployed RNs. It employs k-LCA [15] to solve the Steiner tree problem of finding the minimum number of RNs required to sustain the recovered inter-segment topology. DarDs is similar to DORMS in terms of populating RNs towards the center; however it differs henceforth as it allows 2 segments to merge if they meet before reaching the center. Once all disjoints segments in the WSN get center-connected DarDs runs IO-DT to determine the minimum number of RNs required and their positions. IO-DT is shown to outperform other published heuristics for solving the SMTMSPBEL problem. Furthermore, DarDs opts to minimize distances RNs travel during recovery and thus avoids unnecessary energy expenditure.

In summary, unlike DORMS and other published SMT-MSPBEL approaches whose primary aim is to only reduce the number of RNs, DarDs provides a complete distributed recovery solution where the network is initially reconnected, the minimum number of RNs is found and an optimal polynomial time relocation and relay assignment solution is also provided to minimize RN travel distance.

III. System Model

The proposed DarDs approach considers a WSN that has been partitioned into disjoint segments due to multiple node failures. The WSN under consideration is assumed to have a mix of stationary and mobile relay nodes (RNs) or all RNs. We assume that each RN is aware of the size and center of the deployment area and its position, e.g., using contemporary localization schemes. All RNs are assumed to have the same communication range R. Since RNs consume energy at higher rate compared to stationary nodes, DarDs aims to minimize the number of relocated RNs. DarDs assumes that all disjoint segments have sufficient RNs to reach the center of the deployment area and their deployment does not affect the intra-segment topology. The surviving nodes in the segments detect single or multiple node failures by pursuing heuristics to determine how many neighbors have simultaneously lost contact, to assess drop in network traffic or to detect inability to contact some other part of the network [16]. This detection procedure is conducted by nodes at the border of individual segments since they are in the vicinity of failure.

Each segment picks a RN that leads the recovery process and is followed by other RNs. The selection of the leader can simply be based on the proximity of the RN to the center of the deployment area. It is assumed that the segment stays strongly connected despite the departure of RNs, e.g., by picking dominate nodes in the topology or pursuing cascaded relocation within the segment [7]. Figure 1(a) shows a WSN that got partitioned into seven disjoint segments Seg_0, Seg_1, …, Seg_6. The center of the deployment area *C* is represented by *O*. DarDs aims to connect these disjoint segments using the minimum number of RNs.

IV. DARDs Approach

DarDs consists of three phases; initial relay deployment, relay optimization phase and a final relocation phase. These phases are described in detail in the balance of this section.

A. Initial Relay Deployment

Being distributed DarDs does not assume any knowledge about the number or location of disjoint segments in the WSN suffering from failure. A segment does not know the location of other surviving segments; it initially populates relays in a straight line towards the center of the deployment area *'C'*, separated by '*R*' the communication range in order to rendezvous with counterparts from the other segments. The relocation is led by the leading RN of each segment as shown in Figure 1(b). To ensure coordination during redeployment and to associate RNs to their respective segments, the relays are uniquely labeled as RN_{ij}, where '*i*' is the segment ID which identifies the association of relay and '*j*' is the index which distinguishes the order of relays deployed in the path from the segment to '*C*'. The leading relay is identified as RN_{i0}. The number of RNs required to reach '*C*' from a segment '*i*' is:

$$N_{Relays}=\frac{Dist(Seg_i\,,C)}{R}-1 \quad (1)$$

Where $Dist(Seg_i,C)$ is the Euclidean distance between the segment and '*C*'. The segment closest to the center will supply the center RN on account of having the shortest path to the center and all other remaining RNs will establish links with the center RN once they are within its communication range. It is worth noting that deploying RNs to form a communication path between a segment and the center RN has numerous advantages over the intuitive approach of only sending the lead RN from the segment to the center. First the lead RN will not have to travel back and forth which will limit the overhead on that individual RN. Secondly recovery time will be shorter since RNs will be close to their ultimate positions in the final inter-segment topology, i.e., after applying IO-DT.

With the recovery process being applied simultaneously by all segments, a RN of one segment may come in range of others from different segments before reaching '*C*'. DarDs aims to minimize the number of deployed RNs; so in case multiple leading RNs come in range of one another, they exchange their respective segment information and the closest RN to the center among them continues until reaching '*C*' while the others stop. For example in Figure 1(b), RN_{50} of Seg_5 comes in contact with RN_{60} of Seg_6, after exchanging segment information RN_{50} discovers that RN_{60} is closer to '*C*' (in this case already in contact) and thus RN_{50} ceases its advance. RN_{60} now represents both Seg_6 and Seg_5, it passes this information to RN_{30}, the center RN supplied by Seg_3.

As shown in Figure 1(b), a leading RN can come in contact with another segment like RN_{40} comes in contact with

Seg_3 while moving towards 'C'. In this case, the two segments are merged and Seg_4 information is passed down through Seg_3 to RN_{31}, RN_{30} and finally to the center RN. This initial RN deployment restores connectivity in the partitioned WSN and reconnects all disjoint segments as shown in Figure 1(b). The number of RNs populated during initial relay deployment can be given by:

$$Max_{Relays} \leq \sum_{i=1}^{\#segments} N_{Relays} + 1 \quad (2)$$

The maximum RN count required to establish a connected inter-segment topology will be less than or equal to the individual sum of all RNs on the path from the individual segments to the center, in addition to the center RN. To reconnect the partitioned WSN in Figure 1(a), 15 RNs are deployed; the resulting inter-segment topology after the end of RN deployment is shown in Figure 1(b). The center RN_{30} of Seg_3 will lead the optimization phase in DarDs. Note that in practice the actual RN count is often less than Max_{Relays} since some segments get merged during the inward relay placement.

B. Relay Optimization Phase

After the initial relay positioning, DarDs conducts an optimization phase to find the minimum number of RNs and their locations necessary to maintain network connectivity. The optimization phase is led by the center RN, which waits for a certain amount of time to ensure that the leading RN of the farthest segment can reach a position at least 'R' units away from 'C'. The maximum distance between a segment and the center will be half the diagonal length of the deployment area. The maximum wait time is given by:

$$T_{Wait} = \frac{\frac{L_{Diagonal}}{2} - Distance_{CenterRN}}{v} \quad (3)$$

Where $L_{Diagonal}$ is the diagonal of the deployment area and 'v'is the average speed of a RN. The center RN takes into account the time it takes to reach 'C' and subtracts it from the maximum time to compute T_{Wait}. Once T_{Wait} has elapsed the center collects the segment information and deployed relay locations from all leading RNs in its communication range.

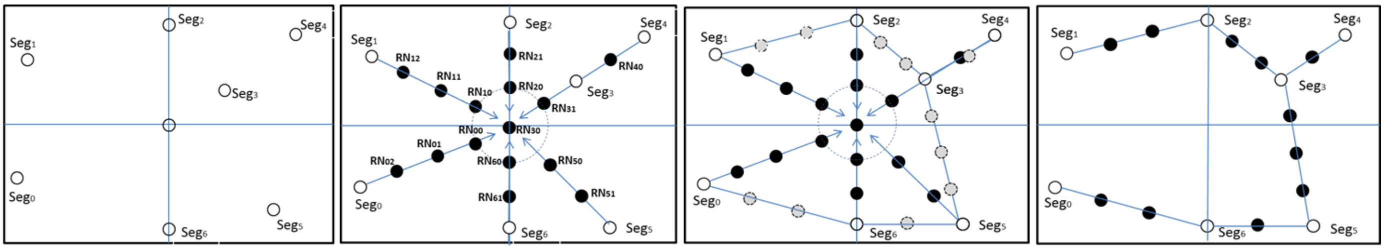
With the center RN knowing the positions of all disjoint segments, the problem boils down to connecting 'n' disjoint segments using the minimum number of RNs and determining their positions. This can be solved by forming a Steiner Minimum Tree with Steiner Points and Bounded Edge length (SMT-MSPBEL) but since this problem is NP-hard we pursue a heuristic, namely, Incremental Optimization based on Delaunay Triangulation (IO-DT) [12]. IO-DT optimally solves the SMT-MSPBEL problem for the case of three terminals and leverages that in handling the general case of $n>3$. Basically, it identifies subsets of 3 terminals to form triangles such that there is no terminal located inside any of the formed triangles. It then applies the 3-terminal optimal solution for each of these triangles. Finally a spanning tree of all terminals and RNs are formed. As shown in [12], and also confirmed by our simulation results, IO-DT outperforms contemporary SMT-MSPBEL solutions and has a runtime complexity of $O(n^2)$. Figure 1(c) shows the topology formed by IO-DT for the partitioned network in Figure 1(a). The grey circles represent RN locations, i.e., Steiner Points, determined by IO-DT. As indicated in the figure, IO-DT requires only 11 RNs to connect the disjoint segments as compared to 15 used in the initial phase (Figure 1(b)). Note that the final topology generated by DarDs is not center connected whereas DORMS is constrained to always have a center RN as 'C' is considered to be one of the fixed terminals during the k-LCA optimization process. The requirement of always optimizing with respect to 'C' limits RN reduction. DarDs imposes no such restraint and aims to always find the best solution through IO-DT.

C. Relocation Phase

The final phase of DarDs aims at relocating the RNs deployed in the initial phase to the Steiner Points (SPs), identified in the optimization phase. Unnecessary RNs are returned back to their respective segments where they can either resume pre-failure duties or boost intra-segment connectivity. Many approaches either skip this step or like [8], provide simplistic solutions like moving the nearest RNs to the desired relay positions without factoring in the distances RNs collectively have to travel during relocation and the peak travel overhead that some of these RNs incur. This shortcoming results in large energy consumption at the level of individual RNs and thereby adversely affects network lifetime. DarDs aims to provide a comprehensive recovery solution that also optimizes total travel cost and balances the overhead on individual RNs.

The relay-SPs assignment problem boils down to choosing an optimal allocation for 'l' deployed RNs to the 'm' SPs determined by IO-DT and returning the extra 'l-m' RNs to their respective originating segments such that their total travel distance is minimized. A brute force approach for solving this problem would have to consider $l!$ assignments, which makes it computationally infeasible for anything but small values of 'l'. DarDs solves the optimal assignment problem by mapping it to the minimum matching problem in bipartite graph. Basically, given a weighted complete bipartite graph G= ((X, Y), E), where X and Y are sets of vertices in each partition of the graph with |X|= |Y| and E is the set of edges $\overline{xy}$ of weight $w(x,y)$ for all $x \in X$ and $y \in Y$, the goal is to find a matching from X to Y with minimum total weight. DarDs employs the Munkres algorithm [17] to solve this problem in polynomial time.

The Munkres Algorithm finds an optimal solution to the assignment problem by matrix operations; given a '$l \times m$' matrix, find a permutation 'k' of {1, 2, 3,…., m}for which the relocation cost is minimum, i.e.,



(a) (b) (c) (d)

Figure 1: Application of DarDs (a) Seven disjoint segments of a damaged WSN, (b) Initial RN deployment, black circles represent deployed RNs (c) : Optimal relay placement topology by IODT.(d) Final topology after excess RNs are relocated back to their respective segments.

$$Relocation_{cost} = \min\left(\sum_{i=1}^{m} w(x_i\, y_{k(i)})\right) \tag{4}$$

In our case X={x_1, x_2, x_3…, x_l} is a set of RNs deployed in the initial deployment phase and Y= {y_1, y_2, y_3…, y_m} is the set of optimal relay locations determined in the optimization phase. With $l > m$ we need the mapping done such that exactly '*m*' out of the '*l*' RNs from set X are mapped to the '*m*' locations in set Y and the remaining '*l-m*' RNs in X returning to their corresponding segments. Directly applying Munkres gives us the optimal assignments of '*m*' RNs in X to those in Y; however it does not consider the cost of returning '*l-m*' RNs back to their respective segments. Some can argue that Munkres can be applied again to assign the '*l-m*' RNs to their segments but this does not yield the best overall solution, since it might be better to assign a higher cost RN to one SP than a lower cost one in terms the overall travel overhead.

DarDs factors in the cost for both the assignment and the return to home segment. In other words, DarDs models the problem using an '$l \times l$' matrix. Let us consider X={x_1, x_2, x_3, x_4, x_5, x_6} be a set of 6 RNs deployed in the initial phase and they are to be mapped to the 3 optimal relay locations represented by Y= {y_1, y_2, y_3}. This can be represented by a '6×3' matrix as shown in (5). The matrix entries represent the travel cost to move a RN '*x*' to the corresponding '*y*' position.

$$\begin{bmatrix} \boldsymbol{x\backslash y} & \boldsymbol{y_1} & \boldsymbol{y_2} & \boldsymbol{y_3} \\ \boldsymbol{x_1} & xy_{11} & xy_{12} & xy_{13} \\ \boldsymbol{x_2} & xy_{21} & xy_{22} & xy_{23} \\ \boldsymbol{x_3} & xy_{31} & xy_{32} & xy_{33} \\ \boldsymbol{x_4} & xy_{41} & xy_{42} & xy_{43} \\ \boldsymbol{x_5} & xy_{51} & xy_{52} & xy_{53} \\ \boldsymbol{x_6} & xy_{61} & xy_{62} & xy_{63} \end{bmatrix} \tag{5}$$

From (5) it is clear that of out of the 6 RNs in X only 3 with the lowest overall travel cost can be mapped to the 3 available Y locations. The remaining RNs are to return back to their respective segments as they are no longer required to maintain the intersegment connectivity. To include the relocation cost into account, we add 3 additional columns to the matrix as shown (6) to represent the return cost (*RC*) incurred. For each relocation entry each RN inserts the travel distance to its respective segment. So 'x_1' enters RC_{x1} for all 3 entries and so on. Now the 6 RNs in X are to be assigned to the 3 positions in Y and the excess 3 RNs will be returned to their home segments such that the total travel cost is minimized.

$$\begin{bmatrix} \boldsymbol{x\backslash y} & \boldsymbol{y_1} & \boldsymbol{y_2} & \boldsymbol{y_3} & \boldsymbol{RC_1} & \boldsymbol{RC_2} & \boldsymbol{RC_3} \\ \boldsymbol{x_1} & xy_{11} & xy_{12} & xy_{13} & RC_{x1} & RC_{x1} & RC_{x1} \\ \boldsymbol{x_2} & xy_{21} & xy_{22} & xy_{23} & RC_{x2} & RC_{x2} & RC_{x2} \\ \boldsymbol{x_3} & xy_{31} & xy_{32} & xy_{33} & RC_{x3} & RC_{x3} & RC_{x3} \\ \boldsymbol{x_4} & xy_{41} & xy_{42} & xy_{43} & RC_{x4} & RC_{x4} & RC_{x4} \\ \boldsymbol{x_5} & xy_{51} & xy_{52} & xy_{53} & RC_{x5} & RC_{x5} & RC_{x5} \\ \boldsymbol{x_6} & xy_{61} & xy_{62} & xy_{63} & RC_{x6} & RC_{x6} & RC_{x6} \end{bmatrix} \tag{6}$$

For any general assignment of '*l*' deployed RNs to '*m*' SPs, a matrix is constructed with *l* rows and *m* columns and buffered by '*l-m*' return columns resulting in a square cost matrix of size '$l \times l$'. Munkres Assignment algorithm guarantees optimality and provides the solution in polynomial time [17].

D. Algorithm Analysis

DarDs consists of 3 phases each having its own runtime complexity. Each phase runs its own algorithm. Therefore, we will analyze each phase individually before arriving at the overall time complexity. Let '*n*' be the number of segments and '*R*' represent the communication range of the relay nodes.

Theorem 1: DarDs converges to the initial connected topology in constant time namely, $\left(\frac{L_{Diagonal}/2}{v}\right)$

<u>Proof:</u> In the initial phase, RNs are populated by all surviving segments until a leading RN reaches the center, becomes *R* units away from the center, or merges with a segment en route to the center. In the worst case scenario the surviving segments can be located at the corners of a rectangular deployment area, so the maximum distance between a segment and the center will be half the diagonal length. Assuming all segments are located at the corners of the deployment area and travel inwards at a speed '*v*', the time taken will be given by distance divided by speed, i.e., $\left(\frac{L_{Diagonal}/2}{v}\right)$.

Lemma 1: There can only be a maximum of six segments that meet at a distance '*R*' away from the center without merging with one another.

<u>Proof:</u> For the segments to meet only at a distance '*R*' from the center, they need to be separated by '*R*' when they come in contact with the center RN. As shown in Figure 2, only six such positions can exist since if a pair of adjacent segments are separated by '*R*' and equidistant from the center form an equilateral triangle which has a center angle of 60 degrees. The center subtends an angle of 360 degrees so there exist only 6 segments that will meet while being '*R*' away from the center. Thus, for n>6 some segments will merge before reaching the center and the number of employed RNs in the initial phase of DarDs will be reduced significantly.

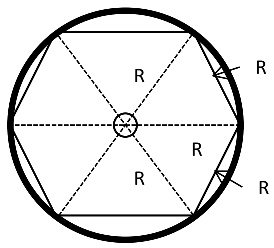


Figure 2: The topology at the center with segments separated from each other and the center by R.

Theorem 2: The maximum number of RNs deployed in the initial phase is bounded by $\sum_{i=1}^{n}\left(\left\lceil\frac{L_{Diagonal}}{2R}\right\rceil-2\right)+7$.

<u>Proof:</u> From equation (2), the maximum number of RNs is given by $(\sum_{i=1}^{n} N_{Relays}) + 1)$ where $N_{Relays} = \frac{Dist(Seg_i, C)}{R} - 1$. So the maximum number of RNs is directly proportional to the distance between the segment and center *C*. In the worst case scenario all segments are at the edge of the deployment area and assuming that none merge before reaching 'C' they all travel a worst case maximum distance $L_{Diagonal}/2$. Substituting $N_{Relays} = \left\lceil\frac{L_{Diagonal}}{2R}\right\rceil$ -1, we get the maximum number of RNs deployed is bounded by $(\sum_{i=1}^{n}(\left\lceil\frac{L_{Diagonal}}{2R}\right\rceil-1)+1)$ or from Lemma 1, we know that for n>6, segments will definitely merge with each other so the actual number of RNs deployed will always be less than the worst case scenario above. Specifically, the center RN will have only 6 neighbors. Thus, the maximum number of RNs would be $\sum_{i=1}^{n}\left(\left\lceil\frac{L_{Diagonal}}{2R}\right\rceil-2\right)+1+6$ or $O\left(n\left(\left\lceil\frac{L_{Diagonal}}{2R}\right\rceil-2\right)\right)$, i.e., linear in the number of segments. It is important to note that this holds for n<6 as well, since the center RN will have less than 6 neighbors which is below the above worst-case bound.

So from the above theorems we see that the number of RNs is heavily influenced by their communication range.

Lemma 2: Time complexity of the IO-DT algorithm run in the optimization phase is $O(n^2)$ as shown in [12].

Lemma 3: The runtime complexity of the relocation phase is $O(l^3)$, where l is bounded by Theorem 2.

Proof: From [17] the number of operations performed to find the optimal solution is $O(l^3)$, or $O\left(\left(n\left(\left\lceil\frac{L_{Diagonal}}{2R}\right\rceil-2\right)\right)^3\right)$ based on Theorem 2.

Lemma 4: DarDs converges to the final connected topology in constant time, namely in $\frac{L_{Diagonal/2}}{v}$.

Proof: During the final relocation phase, in the worst case scenario all 'l' RNs need to relocate and assuming all travel a worst case maximum distance of $L_{Diagonal}/2$. The time taken for RNs to move and establish the final topology will be given by distance divided by speed, i.e., $(\frac{L_{Diagonal/2}}{v})$. This is same as the time taken to establish the initial connected topology as shown in Theorem 1. Therefore relocation takes the same amount of time as initial deployment and DarDs converges to a final connected topology in $\left(\frac{L_{Diagonal}}{v}\right)$ time units.

Theorem 3: The runtime complexity of DarDs is $O(n^3)$.

Proof: The runtime complexity of DarDs can be determined by the complexity of its three phases. Based on Lemma 2, 3 and 4, the runtime complexity of the three phases are $O(n^2)$, $O\left(\left(n\left(\left\lceil\frac{L_{Diagonal}}{2R}\right\rceil-2\right)\right)^3\right)$ and $O(1)$, respectively. Thus, the overall runtime complexity of $O(n^3)$.

V. PERFORMANCE VALIDATION

The effectiveness of DarDs is validated through simulation. This section discusses the simulation setup, performance metrics and results.

A. Performance Metrics and Experiment Setup

The experiments are based on a 1500m × 1500m square area where random topologies are generated for varying number of segments (3 to 10) and communication ranges (50 to 100m). We consider the following metrics:

- *Number of deployed RNs*: DarDs opts to use the minimum number of relays to restore network connectivity.
- *Total Travelled Distance*: It reflects the distance travelled collectively by all RNs during the recovery process. It assesses the resource overhead that recovery imposes.

We study the performance while varying the following parameters:

- *Communication Range(R)*: It significantly affects the number of RNs required for restoring the inter-segment connectivity.
- *Number of Segments (N_{seg})*: A higher number of disjoint segments in a WSN introduces more connectivity requirements and may increase the required RN count.

The performance of DarDs is compared to DORMS [14], a distributed algorithm that reconnects a network using Minimum Steiner tree approximation. Each segment launches a leading RN towards the center which is followed by other RNs in a cascaded movement. All leading RNs do not stop before they are R away from the center. This criterion makes all the segments connected to one another at the center. Upon reaching the center, the leading RNs of each segment begin an optimization phase to check if fewer RNs can be used. Unnecessary RNs are sent back to their respective segments. DarDs fundamentally differs from DORM in three aspects: (1) During the initial placement phase, RNs from distinct segments are allowed to merge on their path to the center; (2) DarDs employs a superior heuristic for determining the least RN count for sustaining inter-segment connectivity and thus engages fewer RNs; (3) DarDs strive to minimize distances RNs travel during recovery.

Two sets of experiments have been conducted. In the first, the number of segments N_{seg} is varied from 3 to 10 using a uniform random distribution while setting the communication range R to 100m. In the second experiment the effect of communication range on performance is captured. Here we fix N_{seg} to 7 and vary R from 50 to 100m.

B. Simulation Results

This section discusses the obtained results. Each configuration is averaged over 50 different random topologies. We observed that with a 90% confidence interval, the results stayed within 6%-12% of the sample mean. Figure 3 shows the effect of varying R on performance for a fixed number of segments (N_{seg} =7). As seen in Figures 3(a) and 3(b) the number of RNs required to restore connectivity decreases as R increases. This is expected as a greater range allows RNs to cover a wider area. As seen in 3(a), DarDs employs 20-25% fewer RNs than DORMS to establish the initial connected topology; this performance improvement is a result of allowing RNs from different segments to merge and proceed as one entity towards the center. Figure 3(b) shows that the final RN count is also better due to RN placement optimization.

DarDs has a significant performance advantage over DORMS with RNs travelling significantly less distance during recovery as shown in Figures 3(c) and 3(d). This improvement is attributed to the optimized population of RNs during initial deployment which reduces the number of engaged RNs as shown in 3(a); this benefit shows up as a reduction in travel distance. In DOMRS, leading RNs from all segments are required to reach the center and merging of segments is not allowed. The savings in DarDs are also magnified by optimal assignment and relocation of RNs during the final phase where we try to find the best RN-to-SP association in order to minimize travel distance, whereas DORMS just relocates RNs that are closest to the SP positions. Since WSNs have a limited energy supply and the cost of travel is high, this saving in travel distance has major implications on network lifetime which favors DarDs. It is worth noting that the total travel distance decreased with the increase in R as fewer RNs were engaged as pointed out by Figure 3(a).

Figure 4 shows the results of the second set of experiments where R is fixed to 70m and N_{seg} is varied from 3 to 10. As seen in Figures 4(a) and 4(b), as the number of disjoint segments in the deployment area increases the RNs required to establish both the initial and final connected topology grows. This is due to increased connectivity requirements. DarDs consistently outperforms DORMS since it uses a better optimization heuristic and allows RNs to coordinate with one another to achieve the connectivity goal. Meanwhile, Figure 4(c) shows that DarDs outperforms DORMS in terms of travel distance with the performance gap becoming wider as the network grows. This can be attributed to the fact that leading RNs from all segments need to reach the central RN in DORMS, therefore as N_{seg} increase so does the distance travelled in the initial topology setup whereas in DarDs the increase in segment count, boosts the likelihood that leading RNs of different segments will come in contact before reaching the center, allowing RN-paths from the segments to the center to merge and reduce the travel cost. Figure 4(d) showcases the DarDs optimization phase where RNs are assigned to SPs to reduce the travel overhead.

Overall, the simulation results shown in Figures 3 and 4 confirm the performance advantage of DarDs as it imposes significantly less overhead in terms of RNs needed and saves energy by minimizing the distance travelled. The performance advantage significantly grows for large networks, which demonstrates the scalability of DarDs.

VI. CONCLUSION

An autonomous WSN should work without human intervention and be able to recover from node failures. In this paper we have presented DarDs, a novel distributed algorithm that enables a network to restore connectivity after the failure of multiple collocated nodes partitions it into disjoint segments. DarDs achieves such a goal by relocating the fewest RNs from individual segments. The relocation-related energy cost is also minimized by using an optimized RN placement strategy. The simulation results have demonstrated that the proposed algorithm scales well and outperforms competing approaches. Future extensions include dealing with cases in which segments do not possess enough RNs to reach the center.

ACKNOWLEDGEMENTS: This work was supported by the National Science Foundation (NSF) award # CNS 1018171.

REFERENCES

[1] X. Han, X. Cao, E. L. Lloyd and C.-C. Shen, "Fault-tolerant Relay Nodes Placement in Heterogeneous Wireless Sensor Networks," *Proc. of INFOCOM'07*, Anchorage AK, May 2007.

[2] N. Li and J. C. Hou, "Flss: a fault-tolerant topology control algorithm for wireless networks," *Proc. of MobiCom '04*, Philadelphia, PA, Sept 2004.

[3] X Wang, G Xing, Y Zhang, C Lu, R Pless and C Gill, "Integrated coverage and connectivity configuration in wireless sensor networks", *Proc. of ACM Sensys'03*, Los Angeles, CA, November 2003.

[4] B. Chen, et al., "Span an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Proc. of ACM MobiCom'01*, Rome, Italy, July 2001.

[5] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks," *Proc. INFOCOM'05*, Miami, FL, Mar. 2005.

[6] M. F. Younis, S. Lee, A. A. Abbasi, "A Localized Algorithm for Restoring Internode Connectivity in Networks of Moveable Sensors," *IEEE Trans. on Computers*, vol. 59, no. 12, pp. 1669-1682, Aug. 2010.

[7] K. Akkaya, F. Senel, A. Thimmapuram, S. Uludag, "Distributed Recovery from Network Partitioning in Movable Sensor/Actor Networks via Controlled Mobility," *IEEE Trans. on Comp.*, 59(2), pp.258-271, 2010.

[8] F. Senel, K. Akkaya and M. Younis, "An Efficient Mechanism for Establishing Connectivity in Wireless Sensor and Actor Networks," *Proc. of Globecom'07*, Washington, DC, Nov. 2007.

[9] G. Lin, G. Xue, "Steiner Tree Problem with Minimum Number of Steiner Points and Bounded Edge-length," *Information Processing Letters*, 69(2), pp. 53-57, 1999.

[10] X. Cheng, D.-z. Du and L. Wang and B. Xu, "Relay Sensor Placement in Wireless Sensor Networks," Wireless Networks, 14(3), pp. 347-355, 2008.

[11] E. L. Lloyd, G. Xue, "Relay Node Placement in Wireless Sensor Networks," *IEEE Trans. on Comp.*, 56(1), pp. 134-138, Jan 2007.

[12] F. Senel and M. Younis," Optimized Relay Node Placement for Establishing Connectivity in Sensor Networks" *Proc. of Globecom'12*, Anaheim, CA, Dec. 2012.

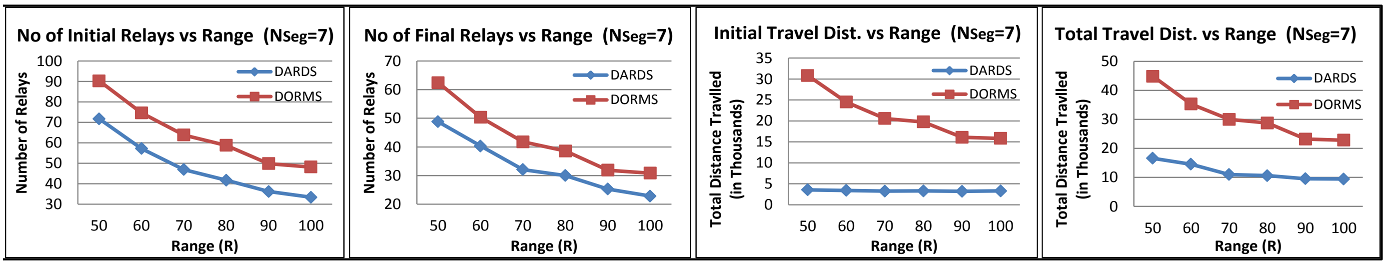
[13] Y. K. Joshi and M. Younis, "Autonomous Recovery from Multi-node Failure in Wireless Sensor Network" *Proc. of Globecom'12*, Anaheim, CA, Dec. 2012.

[14] S. Lee, and M. Younis, "Recovery from Multiple Simultaneous Failures in Wireless Sensor Networks using Minimum Steiner Tree," *Journal of Parallel and Distributed Computing*, Vol. 70, pp. 525-536, April 2010.

[15]G. Robins, A. Zelikovsky, "Tighter bounds for graph steiner tree approximation," *SIAM J. on Discrete Math.*, 19(1), pp.122-134, 2005.

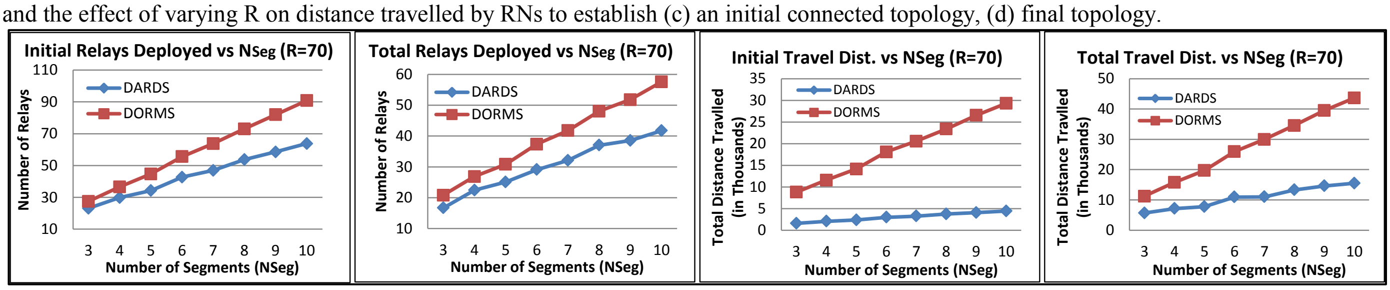
[16]S. Vemulapalli and K. Akkaya, "Mobility-based Self Route Recovery from Multiple Node Failures in Mobile Sensor Networks", *Proc. of WLN '10*, Denver, CO, Oct. 2010.

[17] J. Munkres "Algorithms for the Assignment and Transportation Problems" J. Society for Industrial and Applied Math. **5**(1), Mar. 1957.



(a) (b) (c) (d)

Figure 3: Comparison of DarDs and DORMS w.r.t. the number of RN deployed with a varying R (a) in the initial phase, (b) in Final Topology; and the effect of varying R on distance travelled by RNs to establish (c) an initial connected topology, (d) final topology.



(a) (b) (c) (d)

Figure 4: (a) Initial phase RNs deployed vs. varying N_{seg}. (b) RN count in the final recovered topology vs. N_{seg}. (c) Distance Travelled by RNs in the initial phase vs. varying N_{seg}. (d) Total Distance Travelled to establish final topology vs. varying N_{seg}.