Delay-conscious Federation of Multiple Wireless Sensor Network Segments using Mobile Relays

Jerome L.V.M. Stanislaus, Mohamed Younis
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, Maryland, USA
sjerome1, younis@umbc.edu

***Abstract*—Nodes in wireless sensor networks continuously monitor their surroundings and report to the base station if there is any anomaly. WSN in extreme environmental conditions can suffer from a large scale failure where multiple nodes fail simultaneously and the network gets partitioned into disjoint segments. In addition, in some application multiple standalone WSNs need to be federated to collectively handle an emerging event that requires data sharing among these networks. Numerous approaches have been proposed in the literature to establish connectivity among these network segments by deploying stationary relay nodes (RN) to form data path. In scenarios where the available resources are limited, one or very few mobile data collectors (MDCs) can be used to create intermittent links among the disjoint segments. In this paper, we investigate a federation problem in scenarios where the number of available MDCs is less than the number of RNs required and more than the number of network segments. We present an algorithm to find an optimized travel routes for the MDCs so that the average delay of the network is minimized. The performance of the algorithm is validated through simulation.**

Keywords: Wireless Sensor Networks, Connecting disjoint network, Mobile Data Collectors, Recovery for multiple node failure.

I. INTRODUCTION

In recent years, Wireless Sensor Networks (WSNs) have attracted the attention of researchers for a variety of applications [1][2]. In a typical WSN architecture a group of sensors are deployed and interconnected with each other in order to collectively serve a mission. The sensors forward their collected data to a base station over multi-hop routes. Most notable among the WSN applications are those operating in harsh environments. Examples include outer planetary exploration, combat field reconnaissance, border protection, caves and mines, etc. By use of WSN in these setups, monitoring is achieved without running major risks.

The inhospitable environment and harsh operational conditions in many WSN applications make nodes susceptible to failure that may be wide in scope and involve multiple nodes. For example multiple nodes may get damaged due to detonation of explosives in a battle field, natural calamities in forests, etc. Such node failure may lead to the loss of critical communication links and therefore cause the WSN to be divided into disjoint segments. Federating these segments to restore connectivity would be essential for the network to resume full operation. A similar scenario is when multiple standalone WSNs are to be federated in order to aggregate their services for serving a certain application mission. Example scenarios include search and rescue, military situational awareness, criminal hunting, etc.

Establishing connectivity among WSN segments has received increased attention in recent years. Some of the proposed approaches include repositioning some nodes from the individual segments to form inter-segment paths [3][4][5][6]. However, this strategy may not be feasible if the WSN is composed of stationary nodes. The other popular strategy is to deploy relay nodes (RNs). The employed relays would then form a connected inter-RN topology and re-link the segments using multi-hop paths [7][8]. A variant of the RN-based recovery involves mobile data collectors (MDCs) [9][10][11][12]. The use of MDCs suits constrained setups in which insufficient RN count is available for establishing a stable inter-segment topology. MDCs form intermittent links between the individual segments by touring to carry data from nodes in one segment to recipients in another. If sufficient MDCs are available, a minimum spanning tree of all segments may be formed and MDCs can be designated to serve on the individual links [13].

This paper considers a scenario in which a set of mobile nodes can be employed as relays. However, the relay count is insufficient to establish a stable inter-segment topology; yet it is more than the number of segments and enables optimizing the data delivery in the federated network. We propose an algorithm for determining which inter-segment link will be served by an MDC and defining the tours for MDCs so that the overall data delivery delay in the federated network is reduced. The main objective is to minimize the average delay between any two segments and also minimizing the average distance that an MDC travels. The algorithm is divided into two phases. In the first phase a stable topology is formed assuming the availability of sufficient number of RNs. The RN placement problem is modeled as Steiner Minimum Tree with Steiner Points and Bounded Edge Length (SMT-MSPBEL). Since SMT-MSPBEL is a known NP-hard problem [14], heuristics are used to identify the Steiner points (SP) in polynomial time. In the second phase subsets of RNs are replaced by individual MDCs so that the average tour length of a MDC is minimized. The subsets of RNs to be replaced are picked close to the network periphery in order to limit the effect of link unavailability while an MDC is in motion on the data delivery delay. The overall solution is if the network is federated using a set of stationary and mobile relays. The simulation results confirm the effectiveness of the proposed algorithm compared to recent work in the literature.

This paper is organized as follows. The next section

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

discusses the related work and sets our approach apart from others in the literature. Section III describes the system model and assumptions and formally defines the problem. In section IV, the approach is described in detail. Section V presents the simulation results and finally Section VI concludes the paper.

II. Related Work

Restoring connectivity in a partitioned network has received increased attention in recent years. The published work can be classified based on the scope of the failure into single [3][5] and multiple nodes [4][7] and based on the solution strategy into node relocation [3][5] and relay node deployment, both stationary [4] and mobile [6][10][11][12]. Node relocation is pursued only for single node failure scenarios and the travel distance is often to be minimized. For example, in [3] and [5] a cascaded movement is pursued to share the travel load on multiple nodes. The proposed approach does not assume that segments have mobile nodes and deals with failure scenarios that are wider in scope and greater in impact on the network connectivity.

DORMS [7] tackles the problem of federating disjoint segments. However, stationary relays are used. In addition, quite a few techniques, e.g., [8], have been proposed to establish connectivity via the placement of stationary relay node without exploiting the mobility of these relays to cope with availability constraints. Meanwhile, the focus of [10][11][12] is on collecting data from isolated node blocks using MDCs. However, the use of MDCs as stationary relays has not been explored and an implicit assumption is made that the MDC count is less than the number of segments. IDM-kMDC [13] tackles a similar problem. Here the number of MDCs is assumed to be less than the number of edges of the minimum spanning tree connecting the segments. The main goal of IDM-kMDC is to find least touring path with the number of available MDCs without much attention to data delivery delay.

III. Federation using Multiple MDCs

In this section we describe our algorithm for Federating segments via Stationary and Mobile Relays (FeSMoR). FeSMoR operates in two phases as explained below. It is important to note that we assume that N relays are available for federating the networks. Each of these relays has a communication range of R and is also able to move as needed.

A. Forming Inter-segment Topology

The first phase involves the computation of a Euclidian Steiner Minimal Tree (ESMT) for the segments in order to form a connected topology. Figure 1 shows an example. This phase significantly affects the optimality of the solution for the underlying federation problem since it will determine the number of Steiner points that has to be populated with relays and also the overall length of the interconnections among the segments. The latter not only impacts the number of relays to be deployed, given their limited range, but also the distance that MDCs have to travel if insufficient relays are available. To elaborate, each edge in Figure 1 may be longer than the communication range R of a relay node and thus the individual edges would have to be populated with multiple relays that are R units apart in order to establish connectivity. FeSMoR assumes that insufficient number of RNs is available to form a stable topology using stationary nodes and opts to determine a subset of the edges to be served totally or partially by MDCs instead of stationary RNs. Specifically, if Q relays are needed for forming a stable topology and only N are available, FeSMoR employs some of these N nodes as MDCs in order to eliminate Q-N relays from the formed ESMT topology.

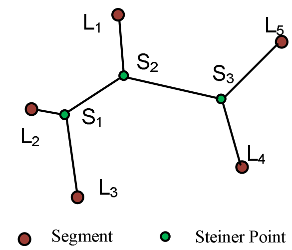

Fig. 1: Sample Steiner tree formed for 5 segments

As pointed out earlier, the relay placement problem is NP-hard, and therefore heuristics are to be pursued. There are numerous polynomial time approximate solutions for Steiner tree problems in the literature; most notably the k-LCA algorithm by Robin and Zelikovsky algorithm [15]. k-LCA achieves the best-known approximation ratio of 1.55. The problem with k-LCA is that it uses centralized approach. On the other hand, DORMS is a distributed approach to create inter-segment topology and also performs better in balancing the data traffic among RNs [7]. We use the DORMS in validating FeSMoR. It should be noted that FeSMoR needs a Steiner tree for connecting the segment and can thus work with other heuristics and not necessarily DORMS and k-LCA.

B. Establishing Stable or Intermittent Connectivity

FeSMoR opts to meet the RN availability constraints by substituting groups of collocated RNs by an MDC. In other words, FeSMoR tries to serve an edge, part of an edge or two intersecting edges in the formed ESMT by a single MDC. Since an edge may need multiple RNs to connect the two ends, using an MDC would eliminate the need for some RNs. FeSMoR iteratively does that until the RN availability constraints are met. The question is which edge(s) are to be considered first so that the average data delivery delay and the average distance an MDC travels are minimized. Reducing the data latency is obviously a plausible goal particularly when MDCs are involved in carrying the data rather than using multi-hop routing over a path of stationary nodes. FeSMoR reduces the average delay for data delivery among segments by using MDCs at the periphery and avoiding the core of the network. The rationale is that an edge serving only ingoing and outgoing traffic of a single segment will have the least impact on the average data latency. For example, in Figure 1, the edge S_3L_4 only forwards traffic to and out of segment L_4, while the edge S_2S_3 serves multiple segments, and thus touring over S_3L_4 will have less impact on the average data latency. Moreover, motion imposes overhead on the MDCs and increases the latency for delivering the data that an MDC carries and thus minimizing the travel distance is highly desired.

For an ESMT, each Steiner points S_i has a degree of three, the edges incident to S_i forms three 120^o angles and the maximum number of Steiner points is $M-2$, where M is the segments [16]. Let S_k denotes the k^{th} Steiner point, where k = 1, 2, ..., (M-2). Let the set T_k denote the set of segments and Steiner points connected to S_k. Before replacing the RNs by the MDCs, all the edges are assigned a rank denoting how far an edge is located from the periphery. The rank of an edge in the constructed ESMT determines which edge is to be replaced first by an MDC. All edges which connect to leaf segments are given a rank of 1. Ranks for the remaining edges are calculated as described below. Initially all the remaining edges are assigned a rank of ∞. Let (u,v) be the vertices of the edge. The rank of the edge (u,v) is identified by the following equation:

$$Rank(u,v) = MIN(MAX(Ranks(u)),MAX(Ranks(v)))+1 \quad (1)$$

where Ranks(w) gives the ranks of all the edges connected to vertex 'w'. The following four rules explain how MDCs are allocated in the federated network so that only N nodes are used. It has to be noted that the rules are applied successively.

i) Edges to segments on the network periphery: FeSMoR first identifies all Steiner points close to the network periphery; that is all points S_i for which the set T_i contains leaf segments, in other words the edges with rank 1. A Steiner point will have at most 2 links connecting to leaf segments. Serving edges at the periphery with MDCs is the preferred choice for FeSMoR in order to minimize the average data latency. However, FeSMoR distinguishes between Steiner points that connect with one and two leaf segments. The case for 2 leaf segments will be discussed below. After identifying the set of edges, FeSMoR finds the shortest edge that connects to a leaf segment and assigns an MDC to tour along this edge. Choosing a touring path at the network periphery enables the establishment of uninterruptable links at the core of the network. The tour length depends on the number of RNs *"q"* on the edge. If (Q - q) is still more than N, this process is repeated for the next candidate edge and so on. The algorithm proceeds to next case when the network connectivity could not yet be achieved, i.e., $Q - \Sigma q > N$.

ii) Steiner points connecting to two leaf segments: When all the edges connecting to leaf segments are covered by MDCs, FeSMoR then identifies all Steiner points which are toured by 2 MDCs; that is all points S_i for which the set T_i contains two leaf segments L_1 and L_2. These two edges are covered by 2 MDCs. We refer to (L_1, S_i, L_2) as a Steiner triangle. Among all candidates of these triangles, FeSMoR picks the ones that are requiring the shortest tour length for the MDCs. For each considered triangle, one MDC which is already touring one of the edges is assigned the tour and the other MDC is retrieved. The saved MDC can be used in the core of the network as stationary or mobile node depending on the needs. The main reason for this step is to minimize the delay on the inner part of the network. This process is repeated for all candidate Steiner triangles to save one MDC per iteration. Replacing RNs from leaf edges helps to retain the communication delay of the other segments in the network and choosing an optimal triangular touring path ensures that MDCs' travel distance is minimized. If the required RNs are not yet reduced to N, FeSMoR considers the choices below.

iii) Unconstrained edges: At this stage, the edges that are uncovered by the first two stages are replaced by MDCs. Here, the edges with lower rank are processed first, starting from rank 2. If multiple edges have the same rank then shorter edges are chosen first similar to case (i). This is repeated until all the relay nodes are covered by MDCs, else FeSMoR pursues the last resolute as explained next.

iv) Unconstrained Steiner triangles: This is similar to the earlier case, except that FeSMoR considers Steiner triangles that do not have leaf segments as vertices. Similar to case (iii) above, Steiner triangle that is farther from the core of the network is processed first; that is the triangle whose edges have lower rank. This is again repeated until the total number of used nodes equals to N.

C. Illustrative Example

Figure 2 illustrates the operation of FeSMoR while federating 6 segments L_1, L_2, ..., L_6 with 12 relays. The numbers on the edges of the ESMT represents the rank of the edge. Due to space constraint, the illustration of rank calculation is not shown. During the first phase of the algorithm, an ESMT is formed using three Steiner points S_1, S_2, and S_3 as shown in Fig 2(a). Based on the communication range, the required RN count is 18. Three stationary RNs are placed at the Steiner points S_1, S_2, and S_3 to avoid time synchronization problem if there are multiple MDCs touring towards a Steiner point. Now the remaining 15 RNs are to be covered by 9 MDCs. The first step of phase 2 finds a Steiner point that connects to a leaf

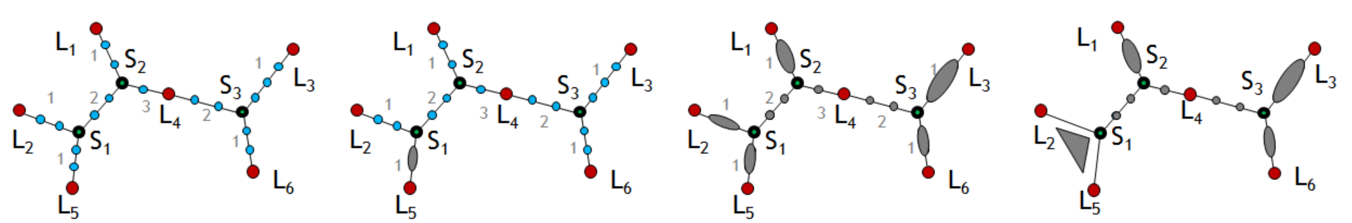

Fig 2: An illustration of how the FeSMoR operates. (a) For the disjoint segments L_1,L_2, .., L_6, a Steiner tree is formed where S_1,S_2 and S_3 are the Steiner points. This estimates a total of 18 RNs required and we have only 12 relays. Three nodes are assigned to S_1, S_2 and S_3. (b) First iteration identifies shortest edge connecting from a Steiner point (S_1) to a segment on the periphery (L_5) and assigns one MDC i.e. the shortest edge with rank 1. (c) All edges with rank 1 are considered and assigned an MDC. (d) Choosing a Steiner triangle to minimize the travelled distance and not affecting the inner edges. FeSMoR terminates after establishing of a connected topology using 4 mobile MDCs and 8 stationary nodes.

segment with the shortest edge. The edge S_1L_5 is found to be a match and is thus assigned an MDC as shown in Fig 2(b). Further iterations resulted in 5 MDCs touring 5 edges in the network as shown in Fig 2(c). By the end of first step of the 2nd phase, there are 5 RNs left to be covered by 4 MDCs. It is to be noted that, all edges connecting to leaf segments are being covered by different MDCs. Instead of covering 2 additional RNs using another MDC, FeSMoR minimizes the average data delivery delay by finding a Steiner triangle. Hence the tour (L_5, S_1, L_2) is assigned to an MDC, as shown in Fig 2(d). This saves us one MDC which will now be used as stationary nodes in the core of the network. Thus, the 18 RNs are covered by 4 MDCs and 8 stationary nodes.

IV. Performance Evaluation

The performance of the proposed algorithm is validated through simulation. This section discusses the simulation environment, performance metrics and simulation results.

A. Experimental Setup and Performance Metrics

In the simulation, segments are randomly placed to a 1500m × 1500m area. The number of segments is varied from 3 to 15. The transmission range of relays, *R*, is fixed at 100m. We also have studied the effect of changing *R* on performance. However, the results could not be included due to space constraints. All MDCs are assumed to have a speed of 6 meters/min. For simulation, the number of available MDCs *"N"* is set to the number of RNs *"Q"* required to form ESMT, multiplied by a fraction "φ". However, the MDC count is kept greater than or equal the number of segments *M*.

The performance of the algorithm under varying N is studied by repeating the experiment for different φ values. The reason is that, for different topologies the segments can be scattered too far or too close to each other and this affects the number of RNs required. Hence, varying M will not lead us to a fair conclusion. Instead, the ratio φ is varied for different topologies to handle such a condition. The following performance metrics are considered for the simulation:

- *Average Communication Delay:* This captures the average delay in the network assuming one byte transmission on the shortest path between every pair of segments. We have assumed that stationary RNs have a latency of 0.1 ms/meter and the latency for MDC is calculated from its speed (6 meters per 60 sec), which would be 10 sec/meter.
- *Maximum Communication Delay:* This reports the maximum delay experienced between any two segments.
- *Total Tour Length:* This is the sum of distances traveled by all MDCs. This metric gauges the inflicted overhead due to mobility.
- *Maximum Tour Length:* This metric shows the longest distance that an MDC has to travel in a network. This is important in analyzing the maximum delay incurred in the network. In addition, this metric indicates the load that one of the MDCs will experience, which would affect the rate of energy depletion and the node lifetime

B. Performance Results

We compare the performance of FeSMoR to that of IDM-kMDC [13]. IDM-kMDC first forms a minimum spanning tree (MST) and assumes that MDCs are deployed on all edges. Tours are merged if the number of available MDCs is less than the number of edges on the MST, i.e., M-1. The main goal of the algorithm is to find the shortest tour lengths without any consideration to the data delay in the federated network. Given the assumptions of IDM-kMDC and the fact that FeSMoR is geared for relatively high MDC count, IDM-kMDC is simulated only for large values of φ with the stationary relay nodes are placed along the longest edges of the MST. Each simulation experiment is repeated for 50 different topologies and the average is reported. It is observed that with a 95% confidence interval, our results stayed within %6-%12 of the sample mean.

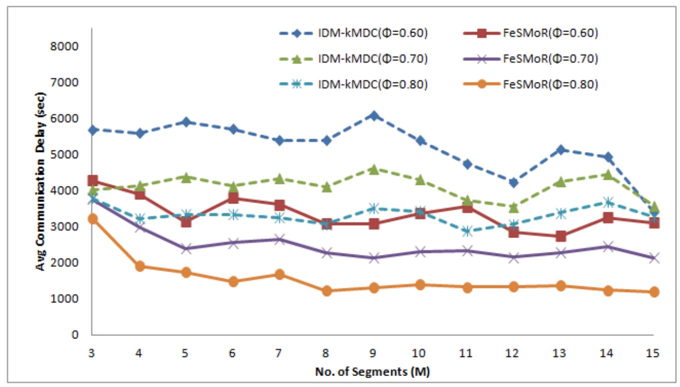

Fig. 3: The average delay as a function of the number of segments

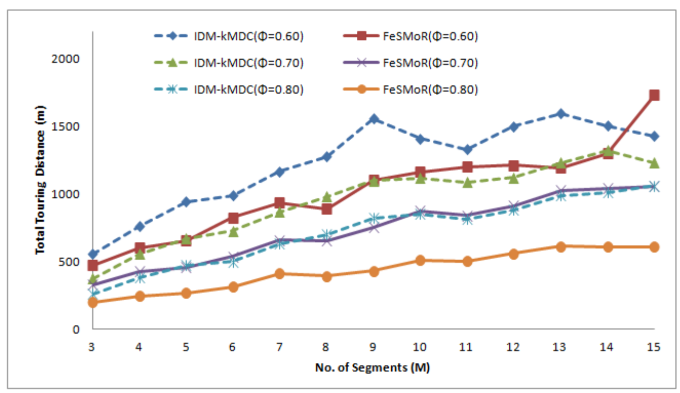

Fig. 4: The average distance that the MDCs collectively travel for varying number of segments

Figure 3 compares the results of FeSMoR and IDM-kMDC for the average communication delay. As expected, the average communication delay of the network is highly improved as we are trying to replace the RNs close to the network periphery first as opposed to IDM-kMDC. The performance is higher when the ratio φ is larger, which is expected due to the increased availability of relays. Figure 4 shows the total tour length and it can be seen that the total distance covered by mobile relays is less for FeSMoR than IDM-kMDC. This is again expected since a Steiner tree offers reduced length to connect segments than an MST. This is further improved by novel heuristics in FeSMoR. The gap between IDM-kMDC and FeSMoR is almost constant for all values of φ. The results of Figures 3 and 4 confirm that FeSMoR effectively meets its design goal.

Figure 5 shows the maximum communication delay between any two segments in the network. It is clear that FeSMoR outperforms IDM-kMDC in terms of maximum communication delay. The maximum distance travelled by an MDC is slightly lower for FeSMoR mainly due to the formation of ESMT. This is also confirmed by the results in Figure 6. While one would argue that IDM-kMDC balances the load on the MDCs better than FeSMoR, it is not really a major issue since the employed relays can take turns in serving as MDC and split the load over time. These results, in conjunction with those of Figure 4 show that FeSMoR indeed balances the load on the nodes that have to move, while IDM-kMDC engages more nodes in touring. Again, experiments under varying communication range have been conducted and the results, which are not shown due to space constraints, stay consistent with those of Figures 3-6.

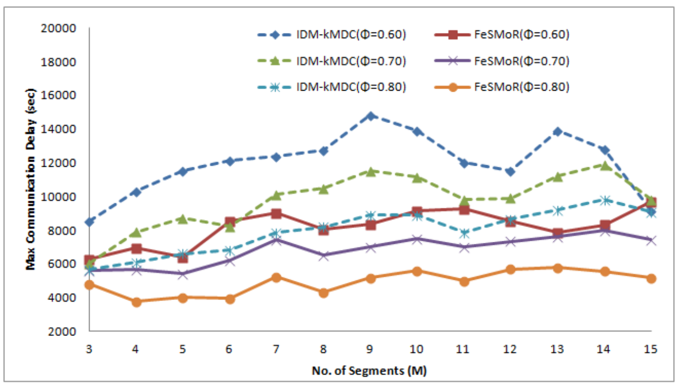

Fig. 5: The effect of the number of segments on maximum delay.

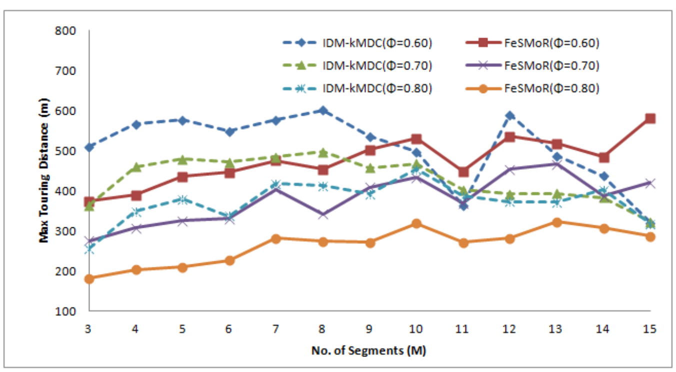

Fig. 6: The maximum tour length as a function of the segments count.

V. CONCLUSION

In many applications of wireless sensor networks (WSNs) nodes operates in harsh environment and become highly susceptible to failure that can cause the network to get partitioned into multiple disjoint segments. Restoring connectivity would then be very essential for sustaining the level of service that the WSN ought to deliver. Similar scenarios arise when multiple standalone WSNs are to be federated to perform some special tasks or handle an emerging event. In this paper, we have proposed FeSMoR, a novel algorithm for connecting multiple WSN segments using a mix of stationary and mobile relays. FeSMoR finds a Steiner tree for which the segments are terminals and estimates the number of relays "Q" required for establishing a connected inter-segment topology. FeSMoR deals with cases in which the available relays N are fewer than Q, yet exceeds the number of edges in the Steiner tree. To meet the resource availability constraints, FeSMoR incrementally replaces groups of relays on the Steiner tree with mobile nodes while minimizing the average data latency and the distance that the mobile nodes collectively have to travel. The simulation results have confirmed the effectiveness of FeSMoR.

Acknowledgement: This work is supported by the National Science Foundation, award # CNS 1018171.

REFERENCES

[1] C-Y. Chong, S.P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. of the IEEE*, **91**(8), pp. 1247- 1256, Aug 03.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E Cayirci, "Wireless sensor networks: a survey," *Comp. Nets*, **38**(4), pp. 393-422, Mar. 02.

[3] P. Basu and J. Redi, "Movement Control Algorithms for Realization of Fault-Tolerant Ad Hoc Robot Networks," *IEEE Networks*, 18(4), pp. 36-44, August 2004.

[4] 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.

[5] A. Abbasi, M. Younis and K. Akkaya, "Movement-Assisted Connectivity Restoration in Wireless Sensor and Actor Networks," *IEEE Trans. on Parallel and Dist. Systems*, **20**(9), pp. 1366-1379, Sept. 2009.

[6] S. Das, et al., "Localized Movement Control for Fault Tolerance of Mobile Robot Networks," *Proc. of the 1st IFIP Int'l Conf. on Wireless Sensor and Actor Networks (WSAN 2007)*, Albacete, Spain, Sept. 2007.

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

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

[9] W. Alsalih, Selim Akl, and H. Hassanein, "Placement of multiple mobile base stations in wireless sensor networks," *Proc. of the IEEE Symp. on Signal Processing and Info. Tech. (ISSPIT)*, Cairo, Egypt. Dec. 2007.

[10] H. Almasaeid, and A. E. Kamal, "Data Delivery in Fragmeneted Wireless Sensor Netowrks Using Mobile Agents," *Proc. the 10th ACM/IEEE Int'l Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Chania, Greece, Oct. 2007.

[11] H. Almasaeid, and A. E. Kamal, "Modeling Mobility-Assisted Data Collection in Wireless Sensor Networks," *Proc. of the IEEE Global Comm. Conf. (GLOBECOM'08)*, New Orleans, LA, Dec. 2008.

[12] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," *Proc. the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'04)*, Tokyo, Japan, May 2004.

[13] F. Senel M. Younis, "Optimized Interconnection of Disjoint Wireless Sensor Network Segments Using K Mobile Data Collectors," *Proc. of Int'l Conf. on Comm. (ICC'12)*, Ottawa, Canada, Jun 2012 (to appear).

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

[15] G. Robins and A. Zelikovsky, "Tighter Bounds for Graph Steiner Tree Approximation," *SIAM Journal on Discrete Mathematics*, 19(1), 2005, pp. 122-134.

[16] D. Chen, et al., "Approximations for Steiner Trees with Minimum Number of Steiner Points," *J. of Global Opt.*. **18**(1), pp. 17-33, 2000.