Computer Networks 58 (2014) 254–283

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Survey Paper

Topology management techniques for tolerating node failures in wireless sensor networks: A survey

Mohamed Younis [a,*], Izzet F. Senturk [b], Kemal Akkaya [b], Sookyoung Lee [c], Fatih Senel [d]

[a] Dept. of Computer Science and Elec. Eng., University of Maryland Baltimore County, Baltimore, MD 21250, United States
[b] Department of Computer Science, Southern Illinois University, Carbondale, IL 62901, United States
[c] Department of Computer Science and Engineering, Ewha Womans University, Seoul, Republic of Korea
[d] Department of Computer Science, Antalya International University, Antalya 07190, Turkey

A R T I C L E I N F O

Article history:
Received 11 January 2013
Received in revised form 11 July 2013
Accepted 23 August 2013
Available online 2 September 2013

Keywords:
Topology management
Wireless sensor networks
Connectivity restoration
Fault-tolerance

A B S T R A C T

In wireless sensor networks (WSNs) nodes often operate unattended in a collaborative manner to perform some tasks. In many applications, the network is deployed in harsh environments such as battlefield where the nodes are susceptible to damage. In addition, nodes may fail due to energy depletion and breakdown in the onboard electronics. The failure of nodes may leave some areas uncovered and degrade the fidelity of the collected data. However, the most serious consequence is when the network gets partitioned into disjoint segments. Losing network connectivity has a very negative effect on the applications since it prevents data exchange and hinders coordination among some nodes. Therefore, restoring the overall network connectivity is very crucial. Given the resource-constrained setup, the recovery should impose the least overhead and performance impact. This paper focuses on network topology management techniques for tolerating/handling node failures in WSNs. Two broad categories based on reactive and proactive methods have been identified for classifying the existing techniques. Considering these categories, a thorough analysis and comparison of all the recent works have been provided. Finally, the paper is concluded by outlining open issues that warrant additional research.

1. Introduction

The growing interest in applications of wireless sensor networks (WSNs) has motivated a lot of research work in recent years [1–4]. For some of these applications, such as space exploration, coastal and border protection, combat field reconnaissance and search and rescue, it is envisioned that a set of mobile sensor nodes will be employed to collaboratively monitor an area of interest and track certain events or phenomena. By getting these sensors to operate unattended in harsh environments, it would be possible to avoid the risk to human life and decrease the cost of the application.

Since a sensor node is typically constrained in its energy, computation and communication resources, a large set of sensors are involved to ensure area coverage and increase the fidelity of the collected data. Upon their deployment, nodes are expected to stay reachable to each other and form a network. Network connectivity enables nodes to coordinate their action while performing a task, and to forward their readings to in situ users or a base-station (BS) that serves as a gateway to remote command centers [5,6]. In fact, in many setups, such as a disaster management application, nodes need to collaborate with each other in order to effectively search for survivors, assess damage and identify safe escape paths. To enable such interactions, nodes need to stay reachable to each other and route data to the BS). Therefore, the inter-sensor

* Corresponding author. Tel.: +1 410 455 3968; fax: +1 410 455 3969.
E-mail addresses: younis@cs.umbc.edu (M. Younis), isenturk@cs.siu.edu (I.F. Senturk), kemal@cs.siu.edu (K. Akkaya), sookyounglee@ewha.ac.kr (S. Lee), fatih.senel@antalya.edu.tr (F. Senel).

1389-1286/$ - see front matter
http://dx.doi.org/10.1016/j.comnet.2013.08.021

connectivity as well as the sensor-BS connectivity have a significant impact on the effectiveness of WSNs and should be sustained all the time.

However, a sudden failure of a node can cause a disruption to the network operation. A node may fail due to an external damage inflicted by the inhospitable surroundings or simply because of hardware malfunction. The loss of a node can break communication paths in the network and make some of its neighbors unreachable. Moreover, WSNs operating in a harsh environment may suffer from large scale damage which partitions the network into disjoint segments. For example in a battle field, parts of the deployment area may be attacked by explosives, and thus a set of sensor nodes in the vicinity would be destroyed and the surviving nodes are split into disjoint partitions (segments). Restoring inter-segment connectivity would be crucial so that the WSN becomes operational again.

In this paper, we first highlight the challenges that node failures introduce to the operation of WSNs and provide taxonomy of recovery techniques that are geared for restoring the network connectivity. We categorize fault-tolerance techniques proposed in the literature according to the pursued recovery methodology into proactive and reactive techniques. Further classification is done within each category based on the system assumptions, required network state, metrics and objectives for the recovery process, etc. Under each category, we discuss several algorithms and highlight their strengths and weaknesses. Finally, we enumerate open research issues that are yet to be investigated by the research community. To the best of our knowledge, this paper is the first to survey contemporary connectivity-centric fault-tolerance schemes for WSNs, and sheds light on several practical issues for application designers. It will also be a good resource for newcomers to this research area.

Since the process of providing fault-tolerance is in general a form of topology management (i.e., often leads to changes in the network topology parameters), we start in Section 2 with an overview of contemporary techniques and objective of topology management in WSNs. The rest of the paper is organized as follows. In Section 3, we describe our categorization of the existing approaches. The remaining sections follow this categorization. Section 4 discusses techniques for tolerating a single node failure or a sequence of independent and non-simultaneous failures affecting non-collocated nodes. Recovery from simultaneous failure of multiple nodes is covered in Section 5. Section 6 enumerates open issues and outlines possible future research directions. Finally, Section 7 concludes the paper.

2. Topology management techniques in WSNs

Networks require monitoring and maintenance whether they are wired or wireless. The service which provides these tasks is called network management. Network management includes five functional areas as identified by the International Organization of Standardization (ISO): configuration management, fault management, security management, performance management and accounting management [1,5,7]. The unique requirements and constraints of wireless networks such as WSNs have inspired a new functional area, namely topology management. This term is sometimes used interchangeably with topology control and refers to the management of parameters such as degree of connectivity of the network, transmission power, state, or role of the nodes, etc. By modifying these parameters, one can change the topology of the network. Note that this stage naturally follows the creation of an initial topology.

The primary objective of the topology management techniques in WSNs is to achieve sustainable coverage while maintaining network connectivity and conserving energy. For example, these techniques are employed to track the status of communication links among the nodes, to conserve energy by switching off some of the nodes without degrading network coverage and connectivity, to support hierarchical task assignment for data aggregation, to balance the load on existing nodes and links, or to provide scalability by minimizing medium access collision and limiting overhead. Topology management in WSNs can be done through deterministic node placement or performed autonomously after random deployment given the limited human intervention [8]. Existing topology management techniques/algorithms for WSNs can be classified into the following five categories:

- **Node Discovery:** Detecting the nodes and their locations is an essential function in a WSN not only after the initial deployment but also for integrating newly added nodes. The scope of node discovery is subject to certain trade-offs based on the application goals. For instance, for large networks, resource savings in terms of energy and bandwidth can be achieved by not sharing some of the topology details that are deemed unnecessary for certain parts of the network [9].
- **Sleep Cycle Management:** To conserve energy and extend the network lifetime, some of the redundant nodes in a WSN can be turned off. In addition to the energy savings, this technique causes the number of transmitted messages to decline, which lowers signal interference and the failed transmission attempts. Determining the sleep schedule while sustaining full area coverage and strong network connectivity is a popular topology management optimization that has received quite an attention from the research community [10–13].
- **Clustering:** To achieve scalability and energy efficiency, nodes of a WSN may be grouped to form a hierarchical topology. In this way, nodes can send their readings to a cluster-head which in turn aggregates and forwards the data to the sink node after eliminating redundant data [14]. Although the failure of the cluster-head often requires re-clustering, some approaches have provisioned the topology adjustment by associating primary and backup cluster-heads for each sensor node [15–17].
- **Power Control:** The transmission range reflects the maximum distance at which a receiver can be from a sender. The longer the range is, the higher the power consumption would be. Many of the advanced radios allow programmable transmission power so that a node

can avoid consuming excessive energy in reaching nearby receivers. Low power transmission can also reduce interference and boost the network throughput. However, the use of low transmission power limits the network connectivity since nodes would have fewer directly reachable neighbors. Unlike sleep cycle management, power control is purely a link-layer technique which does not affect coverage or the data-processing tasks that a node performs. Many power control optimization techniques have been proposed to exploit such trade-off to appropriately manage the WSN topology [18–20].
- **Movement Control:** Node mobility has been exploited as a means for optimizing the network performance. The objectives achieved by the movement vary. For example, in [21–24], the focus is on prolonging the network lifetime by reducing energy consumed by stationary sensors, whereas in [25,26] other metrics such as asset safety and data delivery latency have been targeted. In addition, mobile relays with more capabilities than sensors are used as data forwarders in order to prolong the lifetime of a network of stationary sensors [27,28] or to link disjoint batches of nodes [29–31].

Due to the harsh environment, limited energy and hardware resources in WSNs, topology management can also be considered together with fault management. For instance, sensor failures can create holes in the coverage area and even disconnect the network into multiple partitions leaving multiple functional nodes inaccessible. In such a case, topology management must function as self-diagnostic and self-healing and serve as a fault handling service. A number of solutions are available to follow such as increasing transmission range (e.g., power control), repositioning existing node/s (e.g., movement control) or adding relay nodes so that topology management can act as a fault management service by discovering/establishing alternative paths. In the rest of the paper, we will focus on topology management in WSNs to provide fault-tolerance. We will survey the fault-tolerant techniques that can be considered under both topology and fault management.

3. Classification of fault tolerance techniques in WSNs

In this section, we classify fault-tolerance techniques in WSNs that are applied in response to the loss of sensor nodes. Depending on the nature of the failure, different approaches may be required. Therefore, before describing the classification of the fault-tolerance techniques, we first explain the different failure models.

3.1. Node failure models

In WSNs, node failures can be classified into two categories; single and multi-node failures. A single node failure model indicates the loss of one node at a time. This type of failure can be simply detected using local heartbeat messages. Unless there is overlap in coverage, the failed node will leave out part of the area unmonitored as shown in Fig. 1. On the other hand, the node position within the network topology determines its criticality to connectivity. Considering the topology as a graph, a leaf node does not serve on the path between any two nodes and thus would not be critical to connectivity. Node M_{15} in Fig. 2 is an example of leaf nodes. Some nodes like M_{13} are also not critical to network connectivity since its neighbors M_{12} and M_{14} have a path between them that does not include M_{13}.

However, some nodes act as cut-vertices and when any of them fails the network gets partitioned into disjoint blocks. A cut-vertex in a graph is a vertex that splits the graph into multiple connected sub-graphs if it is removed. In other words, a cut-vertex node in the network plays the role of a gateway between two sub-networks. In Fig. 2, nodes M_1, M_2, M_6, M_7, M_9 and M_{10} are cut-vertices and are considered critical for connectivity. The failure of a single critical node thus negatively affects the network operation and may deem the network useless. Obviously the effect of the partitioning depends on the size of the network and the amount of traffic exchanged among the nodes.

The second failure model is based on the simultaneous failure of multiple nodes. WSNs operating in a harsh environment may be subject to damage that can be so significant in a part of the covered area that the network gets partitioned into disjoint segments. For example in a combat field, parts of the deployment area may be bombed, destroying the sensor nodes in the vicinity. Fig. 3 shows an articulation, where the dark areas represent the extent of the damage. The simultaneous failure of multiple collocated nodes is very challenging, not only in the recovery process but also in determining the scope of the failure. Techniques for tolerating a single node failure are neither able to analyze the scope of the failure nor recover the network from large scale damages. Consequently, special approaches have been proposed to handle such simultaneous node failures.

A variant of the multi-node failure model is a composition of spatially-independent single-node failures. For instance, multiple nodes may fail at different parts of the network at the same time. In principle, these failures can be handled independently. However, in some cases resource conflicts and race conditions may arise and the recovery procedure has to provision resource sharing and synchronize the handling of the individual failures.

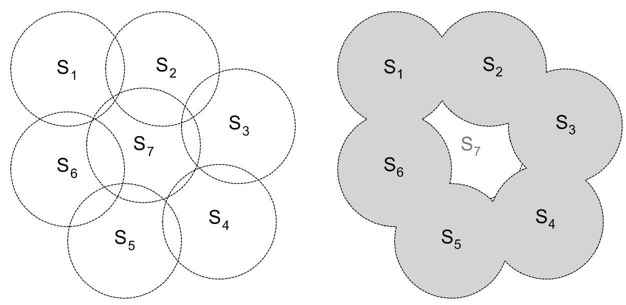

Fig. 1. Assuming a disc coverage model, the failure of S_7 causes a coverage gap in the network.

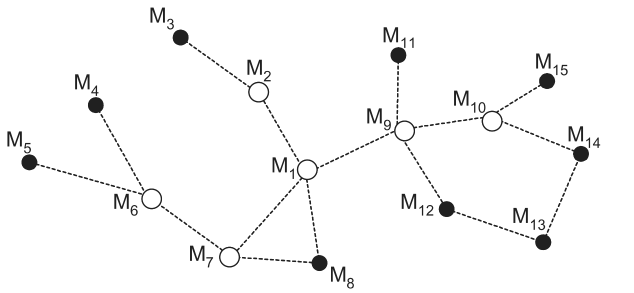

Fig. 2. Example of a single node failure scenario; white nodes are cut vertices and the failure of those nodes splits the network into multiple disjoint partitions, meanwhile black nodes are not essential to node reaching and their failure does not cause network partitioning.

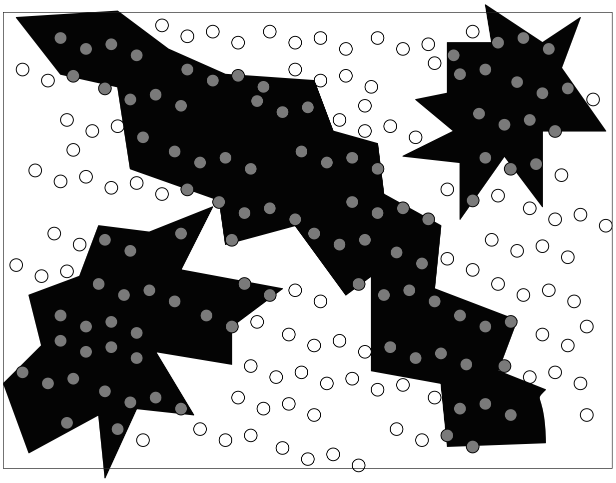

Fig. 3. Illustration of a segmented WSN due to large scale damage; solid circles indicate failed sensors, white ones are operational nodes.

3.2. Taxonomy of fault-tolerance techniques

In order to tolerate node failures which cause network partitioning, two methodologies can be identified: (i) pre-cautionary, where fault-tolerance is provisioned; and (ii) reactive through real-time restoration (repair) of lost connectivity and/or coverage. The pre-cautionary methodology, which is also referred to as *proactive*, strives to provision resources in the network topology both at setup and during normal operation in order to mitigate the effect of node failures. Two variants of this methodology exist. In the first, fault-tolerant topologies are formed at the time of deployment. The second variant is based on augmenting an existing topology with redundant nodes or designating connectivity-unessential node as spares. This category of work is unsuitable for dealing with multiple collocated failures.

For reactive schemes, three major strategies have been pursued in the literature. The first utilizes mobile nodes that are part of the network and repositions them to restore connectivity. Published work distinguishes between the two node failure models explained above and can be further classified based on whether centralized or distributed recovery procedures are employed as well as the node selection criteria and the additional objectives of the recovery process. The second strategy of reactive schemes involves the careful placement of relay nodes to restore connectivity, and is used mostly for dealing with collocated multi-node failures. The employed relays stay stationary at the designated spots. In addition to connectivity, some approaches strive to provide additional features such as quality of service (QoS). Overall, reducing the relay count required for re-establishing connectivity is the popular metric targeted by these approaches. The third strategy pursues recovery using mobile relays and takes the form of establishing intermittent links where the relays tour the disjoint blocks of nodes and carry the data among them. A variant of this strategy exploits the availability of both stationary and mobile relays. In that case, the stationary nodes are used to establish some stable links among subset of segments/sensors or are just placed to shorten the tour that the mobile nodes have to make.

The categorization of the fault-tolerance techniques is summarized in Fig. 4. Each category is considered in detail in the rest of the paper. It is worth mentioning that we focus in this paper on fail-stop scenarios in which the faulty node seizes its operation upon failure. In addition, benign and symmetric failure models are assumed [32], in which a single healthy node can determine whether another node fails and the faulty state of the node is consistently viewed across all healthy nodes within its communication range. In other words, the paper does not cover work that addresses inaccuracies in the sensor measurements.

4. Tolerating single and non-collocated failures

As pointed out in the previous section, published techniques for tolerating a node failure that causes network partitioning either provision fault-tolerance in the network topology both at setup and during normal operation, or pursue a reactive strategy by repositioning healthy nodes. In this section we discuss both tolerance strategies, in the context of a single node failure or a sequence of independent and non-simultaneous failures affecting non-collocated nodes. Recovery from simultaneous failure of multiple nodes will be covered in Section 5.

4.1. Provisioned tolerance schemes

A proactive strategy for preserving the network connectivity in the presence of faulty nodes opts to mitigate the effect of the failure so that a network partitioning will never happen. Two notable methodologies have been pursued in the literature. The first is to carefully place redundant nodes in a WSN. The idea is to provide more than one routing path between every pair of sensors in the network. The route alternatives should also be node disjoint so that the failure of a single node will not break all viable routes. This idea is referred to as k-vertex connectivity or simply k-connectivity ($k \geqslant 2$) where the failure of $k-1$ nodes does not create any partitioning problem [33–37]. Most of the published schemes have focused on 2-connectivity while few proposed generalized solutions to k-connectivity. Forming k-connected topology is often considered a fault mitigation rather than recovery strategy [32]. Meanwhile,

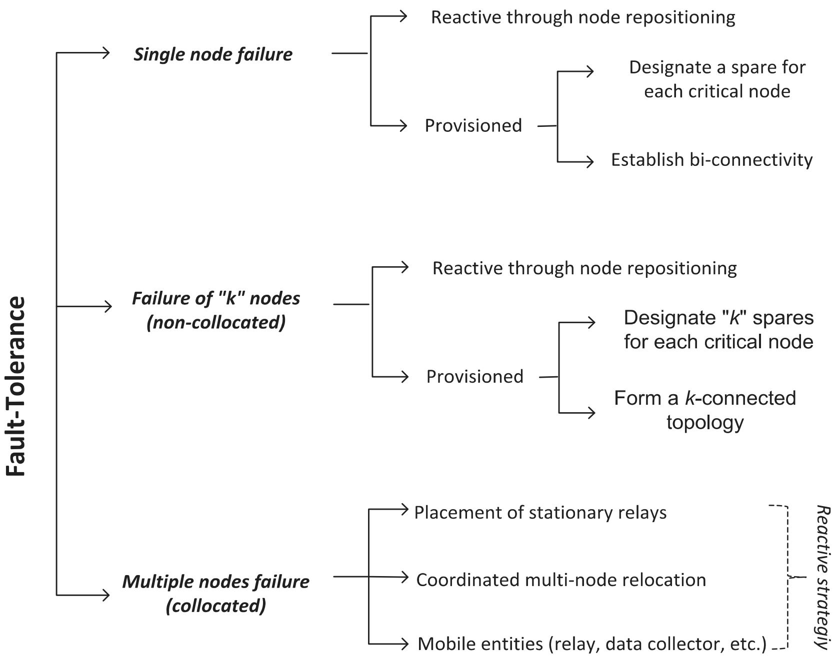

Fig. 4. Classification of the fault-tolerance mechanisms for WSNs.

the second methodology is to designate spares for critical nodes in the network [39,40]. A critical node in this context is a cut-vertex in the network topology. The following summarizes published schemes for these two methodologies.

4.1.1. Sensor placement for forming a k-connected WSN

The most popular proactive strategy for preserving the network connectivity in the presence of a faulty node is to carefully place redundant sensor nodes during or after the initial deployment of a WSN. The objective is to form k-node disjoint communication paths between pairs of nodes in the network and achieve that using the least number of redundant nodes. Such optimization is a very challenging problem that has been proven to be NP-hard for most of the formulations of sensor deployment, even for $k = 1$ [41]. To tackle such complexity, several heuristics have been proposed to find sub-optimal solutions. Published work on sensor placement can be grouped into two categories. The first tries to just establish connectivity between end points, i.e., $k = 1$ [41,43–45]. In the second category, higher degrees of connectivity is to be achieved [33,46–48]. Given the scope of this section, we focus on the second category. The first category will be discussed in Section 5.

Although provisioning k-connectivity enables the network to tolerate the failure of up to $k - 1$ consecutive node failures without suffering partitioning, establishing bi-connectivity has been the most popular goal given the complexity of the node placement problem and the increased node count required for achieving high level of connectivity. A variant of the problem is considered in [33], in which sensor nodes are distinguished from relaying nodes on the route and k-connectivity is only applied to the inter-relay topology. Such a variant is named *partial k-connectivity*. In some publications, k-connectivity is provided as a byproduct of determining a connected dominating set to obtain a robust backbone [48]. In addition, some work assumes that the relays possess more capabilities than the sensor nodes [47].

Some of the earliest solutions for the k-connectivity problem has been presented in [41,49]. The authors have shown that the problem is NP-hard and proposed two heuristics based on graph-theory. First, a complete graph, say G, for the set of vertices (nodes) is formed and each added edge is associated with a weight. This weight basically indicates the number of additional nodes to be placed on the edge $e(u, v)$ to establish connectivity between u and v. The problem is then mapped to finding a minimum-weight k-vertex-connected sub-graph "g". Finally, missing links (edges) in g are established by deploying the least number of additional nodes. The approximation ratio of this problem is reported to be $O(k^4\alpha)$ where α refers to the approximation ratio of any algorithm to compute a minimum-weight k-connected spanning sub-graph of a weighted complete graph. This approximation ratio is further improved in [50] and has been shown to be upper bounded by $O(k^3\alpha)$.

A similar work is reported in [33] where the authors distinguish between sensors and the added nodes for establishing connectivity which are referred to as relays. Unlike [41], they consider a heterogeneous WSN where sensors have different transmission radii and opt to deploy the least number of relays. However, all relay nodes have the same communication range. The authors proposed an approximation algorithm by solving the *minimum k-vertex connected spanning graph* (MKCSG) problem [51–53] and then placing the least number of relay nodes to establish k ($k \geqslant 1$) vertex-disjoint paths between every pair of sensor nodes. Considering a directed or an undirected graph, the algorithm provides a one-way or a two-way steinerized path along each edge of the found MKCSG, as seen in Fig. 5. Since two sensor nodes u and v have different radio ranges, for building an asymmetric communication link between them more relays are required, as demonstrated in Fig. 5(b).

The goal of Tang et al. [46] and Hao et al. [47] is to place the minimum number of relay nodes such that each sensor is connected to at least two relays and the inter-relay network is 2-connected. Tang et al. divide the area into cells and find a position for a relay so that it becomes connected to all sensors in a cell and also to other relays in neighboring cells. The work is further extended in [47] by formulating the same placement problem as *2-Connected Relay Node Double Cover* (2CRNDC), which finds the fewest locations for placing relay nodes, so that each sensor is covered by at least two relays and the group of relay nodes will be bi-connected. This problem can be reduced to the well-known *Minimum Geometric Disc Cover problem* which is NP-complete and they thus present a polynomial time approximation algorithm. The algorithm computes a possible position p of a relay and a set of sensor nodes C(p) which is covered by a relay node locating at p. Then, the algorithm simply identifies positions that cover the maximum number of sensors, at which relays are virtually placed. By analyzing the inter-relay connectivity, the relays with most coverage are switched from virtual to real, in order to form a 2-connected graph.

In [56], Kashyap et al. present $O(1)$-approximation polynomial time heuristic to achieve 2-connectivity among n nodes with the least relay count. The authors focus on both edge and vertex connectivity separately. Using the algorithm in [57], they first form k-vertex connected spanning sub-graph. Then by steinerizing the edges of the spanning sub-graph they establish the k-vertex connectivity. They also prove that the approximation ratio of this algorithm is 10 for $k = 2$. This is the best known heuristic in the literature in terms of relay count for 2-connectivity. Meanwhile, Zhang et al. [58] study both single-tiered and two-tiered relay placement for achieving 2-connectivity under a condition $R \geqslant r$ where R and r are communication ranges of relays and sensor nodes, respectively. A network is said to be single tiered if each sensor node can reach at least one relay and network of relays and sensors is 2-connected, i.e., achieve *partial 2-connectivity*. Similarly, in two tiered networks, a sensor must reach at least two relays and the network of relays and sensor is 2-connected. The optimization objective is again to deploy the fewest relays. They present a 14-approximation algorithm and a $(14 + \varepsilon)$-approximation algorithm for single tiered problem and two-tiered problem respectively. The common idea behind all of algorithms in [58] is to steinerize the edges of minimum 2-connected spanning sub-graph. As in [56], they use the same algorithm presented in [57] in order to find the 2-connected minimum spanning sub-graph.

k-Connectivity is also studied along with the coverage problem under different sensor transmission (r) and sensing (s) ranges. Typically, $r > s$ in many applications and thus once full coverage is achieved the network topology often becomes strongly connected. It is shown in [35] that for a grid structure the network will be connected as long as $r > \sqrt{3}s$. This result has been extended to higher level of connectivity by proposing to use some deployment patterns. For instance, in [59], the authors prove that a strip pattern of sensors can provide optimal full coverage along with 2-connectivity for all different r/s ratios as seen in Fig. 6. In this figure, the horizontal strips are formed by a distance of $\alpha = \min\{r, \sqrt{3}s\}$ while the vertical strips are formed by a distance of $\beta = S + \sqrt{s^2 - \alpha^2/4}$. In addition, horizontal strips are shifted to the right a distance of $\frac{\alpha}{4}$ at alternating rows to guarantee 2-connectivity among the nodes. The work has been extended to 3, 4, 5, and 6 connectivity using different deployment patterns [60,61]. Obviously these approaches are limited to deterministic node placement for which $r > \sqrt{3}s$. Wang et al. [62] proved that k-coverage implies k-connectivity of the entire network in WSNs if $r > 2s$.

Meanwhile, the focus of the approach in [63] is on 3D setups. The objective is to extend the network lifetime while making the network resilient to up to k independent node failures. A grid model is employed in which nodes are allowed to be positioned at the intersection points. The main idea is to regularly relocate nodes in order to balance the traffic load and extend the time until the first node runs out of energy. The repositioning problem is formulated as a Mixed Integer Linear Program optimization. A variant of the approach is presented in [64] where the objective is to maximize the connectivity, in terms of the node-degree of the inter-segment topology, while ensuring a lower bound on the network lifetime

4.1.2. Designating backups for critical nodes

The other popular proactive methodology is to designate backups for critical nodes that act as cut-vertices in the network topology. The backup nodes can be simply passive spares among redundant nodes. These spares will

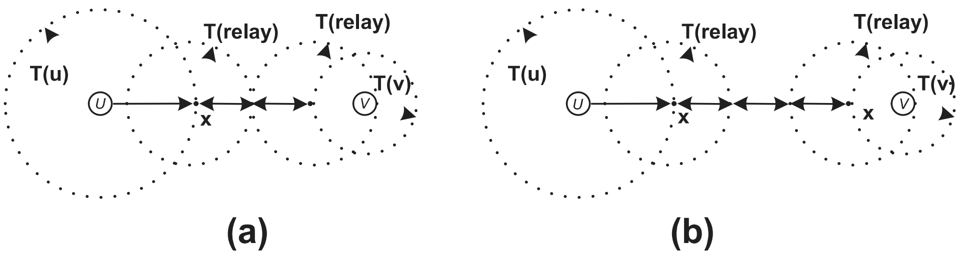

Fig. 5. Between two sensor nodes u and v, the least number of relay nodes (black circles) are placed to establish (a) a one-way or (b) two-way steinerized path. The idea is to have every relay reach 2 consecutive Steiner points on the path so that u and v stay connected even if any relay is lost. The figure is redrawn from [33].

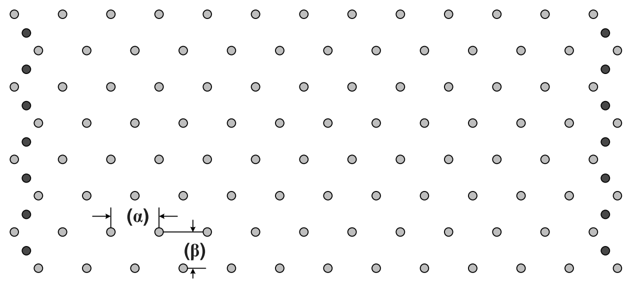

Fig. 6. Sensors are deployed to form horizontal strips to provide full coverage and form two vertical strips to achieve 2-connectivity [59]. The symbols α and β refer to the horizontal and vertical distances between sensors. To reach a sensor in a different level, a sensor has 2 distinct paths by routing to the right or to the left.

be called for duty if a critical node fails. Obviously, this is straightforward if redundant nodes exist. However, if all nodes are serving the application, spares have to be picked among the active nodes. When failure of critical nodes is detected, these active spares will have to quit what they are doing and relocate to substitute failed nodes. To achieve resilience of up to k failures, each critical node is to be assigned k distinct spares. This will enable the recovery to take place even if $k-1$ of these spares die before the primary, critical, node fails. It is worth noting that although the fault-tolerance provisioned by designating the backup before the failure takes place, the recovery is performed in response (after) the failure is detected. Therefore, one may consider this methodology to be based on a hybrid proactive and reactive strategy.

NORAS [39] is one of the approaches that pursue this methodology. To pick a suitable active spare, NORAS factors in both the node criticality to connectivity and coverage. Basically a node that is a cut-vertex or solely covering an important landmark will not be eligible to serve as a backup. Among the legible candidates, a non-critical node that would cause the least coverage degradation is favored. In addition, low node degree would make a node an attractive backup since limited degradation of the network connectivity will be inflicted. NORAS opts to localize the scope of the recovery by picking backups within the 2-hop neighborhood of a failed critical node A_f. If no candidate backup exists in A_f's 2-hop neighborhood, the search widens to include more distant nodes. Upon detecting the failure of A_f, the designated spare will travel to replace A_f or a series of cascaded relocation on the shortest route between A_f and the selected backup will be triggered to split the travel load on multiple nodes.

Another distributed approach which utilizes 2-hop information is proposed in [54]. The idea is do preplanning by designating nodes to lead the recovery when a failure takes place. The proposed *Partition Detection and Recovery Algorithm* (PADRA), identifies a connected dominating set (CDS) for the network. The authors employ the approximate algorithm of Dai and Wu [55] for finding the CDS using 2-hop information. PADRA designates for each cut-vertex A_i a failure handler within the network that would start the recovery process when A_i fails. The ideal handler will be a dominatee neighbor of A_i that can simply replace A_i. If a dominatee is not available, the closest dominator is picked as the failure handler of A_i. Repositioning the failure handler at the position of A_i will then trigger cascaded relocation until a dominatee is encountered. There is no procedure followed for finding the closest dominatee to the failure handler as a means of minimizing the total traveled distance for all involved nodes. Instead, a greedy heuristic is pursued where the closest dominator is picked if a dominatee is not available. Fig. 7 illustrates the operation of PADRA through an example.

Like NORAS, the *Detection and Connectivity Restoration Algorithm* (DCR) [40] proactively identifies nodes that are critical to the network connectivity based on local topological information, and designates appropriate, preferably uncritical, backup nodes. This idea is similar to that of PADRA [54] but no CDS is employed. Only 1-hop information is used, which degrades the accuracy of the node criticality assessment and leads to assigning backups to non-critical nodes. Unlike NORAS, backups are picked within the 1-hop neighbors. If an uncritical neighbor is not available to serve as a backup and a critical one is picked. Upon failure detection, the backup initiates a recovery process by replacing the failed node. The departure of the backup can trigger more relocation if it is also a critical node. The simulation results have demonstrated that by focusing only on connectivity and limiting the backup selection to 1-hop neighbors DCR reduces the travel distance overhead and engages few nodes.

4.2. Reactive connectivity restoration schemes

Real-time connectivity restoration implements a recovery procedure when a node failure is detected. Such a reactive methodology better suits dynamic WSNs, since they are asynchronous in nature and the network topology may change over time. Therefore, adaptive schemes can best scope the recovery process depending on the effect of the failure on the network connectivity. The idea is to utilize existing alive nodes which can move and reposition them to the appropriate locations. Effectively, the network topology is restructured to regain strong connectivity. Various approaches to address these questions have paved the way for a variety of schemes as detailed next.

4.2.1. Categorization of approaches

Published reactive recovery schemes can be categorized based on the following attributes:

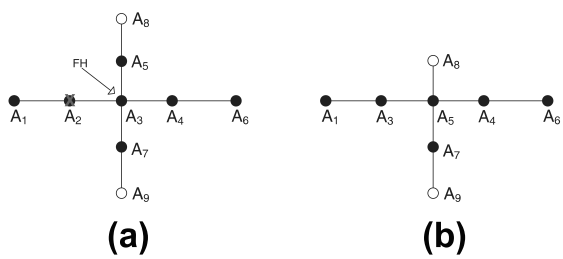

Fig. 7. A sample execution of PADRA [54]; dark nodes are dominators and white hallow are dominatees. (a) When A_2 fails, node A_3 becomes the failure handler and starts the recovery process. (b) A_3 replaces A_2, A_5 replaces A_3 and finally A_8, which is a dominatee, replaces A_5 and the connectivity is restored.

- *Required network state:* As mentioned, identifying cut-vertices requires a network wide analysis. In addition, it is conceivable to generate an optimal recovery plan by considering the state of the entire network. Some of the published work strives to pursue a local analysis instead, in order to ensure scalability to large networks. Basically, the accuracy of determining a cut-vertex and the optimality of the recovery process are traded off for reduced overhead and rapid convergence. While the use of 2-hop information seems popular in the relevant literature [54,65], some allows the quality of the solution to scale when further nodes, that are p-hop away, are known [66].
- *Scope of the recovery:* Another factor that differentiates among published schemes is how many nodes are involved in the recovery. Two main methodologies can be identified; block and sequential node movements. In block movement, a set of connected nodes travel together as unit. The idea is to re-link disjoint partitions or to boost the network connectivity, e.g., eliminate cut-vertices, by moving one partition towards another [67]. The second methodology is to tolerate the degradation in connectivity, caused by the loss of the failed node, or by relocating one or few nodes in a non-coordinated manner [65,66,68,79]. Since the relocated nodes may get detached from their neighbors, a cascaded repositioning is pursued where neighbors follow through in order to sustain connectivity. The process is repeated recursively until reaching nodes whose movement would not violate the connectivity of the WSN.
- *Connectivity goal:* Published schemes also differ in the degree of network connectivity that ought to be sustained. Most efforts have been dedicated to repairing a partitioned network, i.e., to become 1-connected [65,68,79], and to restoring bi-connectivity [66,67]. To the best of our knowledge, there is no published work that achieves general k-connectivity through controlled mobility of nodes. Note that there may be other goals in addition to connectivity. However, in this section we focus on the approaches which have connectivity as a primary goal.
- *Type of algorithm:* Some of the published work employs centralized algorithms where one of the nodes takes charge of generating the recovery plan and coordinating the relocation process [79]. These approaches rely on the availability of an alternate communication path to inform other nodes on what to do. For example in [67], the network is assumed to be bi-connected prior to the failure and thus the healthy nodes can still reach each other. On the other hand, distributed algorithms have been the preferred choice for restoring connectivity of large networks and for repairing partitioned networks [65,68,69–78]. In such a case, the nodes are assumed to have some pre-failure state (e.g., k-hop) information and utilize that information to detect and recover from network partitioning. The rationales are that localized and distributed algorithms scale well and that partitioned networks would not allow communication with a centralized coordinator for the recovery.
- *Secondary Performance Metrics:* All published reactive approaches for tolerating a single node failure pursue node repositioning and differ in the secondary performance objective of the recovery process, other than re-establishing network connectivity. Examples of the considered metrics include the number of nodes that get engaged in the recovery, the relocation overhead in terms of the traveled distance and messaging, the network coverage, etc.

In the balance of this section we summarize the published real-time connectivity restoration schemes. We first start with the distributed approaches focusing on minimizing the total movement distance by requiring local information such as 1 or 2-hops. Later we look at other approaches with additional goals such as limits on the formed topology or on the moving nodes/environment. A summary of the all approaches along with their attributes is shown in Table 1 at the end of this subsection.

4.2.2. Recovery with 2-hop network state

The *Distributed Actor Recovery Algorithm* (DARA) [65] is among the connectivity restoration approaches that require 2-hop information to assess the criticality of the node and orchestrate recovery in a distributed manner. Two variants of the algorithm, namely DARA-1C and DARA-2C, are developed to address 1 and 2-connectivity requirements, respectively. The idea is to identify the least number of nodes that should be repositioned in order to re-establish a particular level of connectivity. DARA strives to localize the scope of the recovery process and minimizes the movement overhead imposed on the involved nodes. In other words, DARA pursues coordinated multi-node relocation in order to re-establish communication links among the impacted nodes. The main idea of DARA-1C is to replace the dead node by a suitable neighbor. The selection of the best candidate (neighbor) is based on the node degree and the physical proximity to the dead node. The relocation procedure is recursively applied to handle any nodes that get disconnected due to the movement of one of their neighbors (e.g., the best candidate that has replaced the faulty node).

DARA-1C is further extended to restore 2-connectivity. In a 2-connected network, there are at least two node-independent paths among each pair of nodes. Similar to DARA-1C, DARA-2C identifies the nodes that are affected, i.e., lost their 2-connectivity property, due to the failed node [69]. Detecting these nodes and restoring their bi-connectivity is a very challenging problem. Nonetheless, through a careful analysis the solution space is proven to be limited to boundary nodes in the network. Such analysis has made DARA-2C a very efficient approach for restoring bi-connectivity. Basically, only a subset of the neighbors of the failed node is relocated in order to restore 2-connectivity.

A variant of the PADRA approach [54], discussed in Section 4.1, has been applied to Wireless Sensor and Actor Networks (WSANs) [70]. However, this approach is restricted to relinking only two partitions and does not handle multi-segment scenarios. The idea is to pick the closest node in the two partitions and move them towards each

Table 1
Reactive (and hybrid) recovery approaches for single node failures.

Paper	Level of connect.	Additional goal	Centralized/ Distributed	Network state	Node selection criteria	Movement
[39]	1	Minimizing the total distance and coverage loss	Distributed	2-hop	Overlapped coverage and node degree	Cascaded
[40]	1	Minimizing the total distance, messaging cost and coverage loss while considering recovery time	Distributed	1-hop	To be a non-critical node, degree and proximity	Cascaded
[54]	1	Minimizing message overhead, total and maximum distance	Distributed	2-hop	The closest dominatee to the failed node	Cascaded
[65]	1	Minimizing total distance	Distributed	2-hop	Node degree and proximity to the failed node	Cascaded
[66]	2	Minimizing total distance	Distributed	p-hop	Node ID	Iterative
[67]	2	Minimizing total distance	Centralized	Global	Based on network-wide optimization	Block
[68]	1	Minimizing message overhead and total distance	Distributed	1-hop	Rank (based on neighborhood)	Cascaded
[69]	1, 2	Minimizing total distance and messaging overhead	Distributed	2-hop	Node degree and proximity to the failed node	Cascaded
[70]	1	Minimizing total and maximum distance	Distributed	2-hop	Proximity to other partition and dominatee nodes are favored for relocation	Cascaded
[71]	1	Minimizing total distance and messaging overhead	Distributed	1-hop	Proximity to the failed neighbor and the utilization of the transmission range	Cascaded
[72]	1	Minimizing total distance	Distributed	2-hop	Proximity to the failed neighbor and serving as non-cut-vertex	Controlled
[73]	1	Maintaining the same length for the data paths	Distributed	No	Block size and proximity to the failed node	Block
[74]	1	Minimizing total distance and number of relocated nodes	Distributed	No	To be the part of the smallest disjointed block	Iterative
[75]	1	Minimizing the total distance and coverage loss	Distributed	1-hop	Overlapped coverage, proximity to the failed node, remaining energy	Temporary
[76]	1	Minimizing the impact of the recovery at the application	Distributed	2-hop	Availability to move (MRI), number of neighbors which can move (MP)	Cascaded
[77]	1	Minimizing the total travel distance, messaging cost and coverage loss while considering application level interests	Distributed	1-hop	Node criticality, availability to move, priority of the current task being executed, degree of connectivity, overlapped coverage	Cascaded
[78]	1	Minimizing the total distance and avoiding failure locations	Distributed	3-hop	Node degree and proximity to the failed node	Cascaded

other until a communication link can be established. Actors are assumed to be initially disconnected and thus there is no node selection or network state maintenance. The closest nodes from each partition are identified by sending messages from sensors and actors move towards each other. However, since such movement may disconnect the repositioned nodes from their partitions, additional nodes are then picked from each partition for performing cascaded movements in order to maintain the intra-partition connectivity. The collective effect is like stretching the topology of the participating sub-networks towards each other. Similar to [54], the selection of appropriate nodes for the cascaded motion is made by considering a CDS of the individual partitions. The approach opts to reposition non-CDS nodes in order to limit the effect of relocating a node on the connectivity of the other nodes in the partition. Fig. 8 shows an illustrative example.

4.2.3. Recovery without explicit state update

The approaches discussed above assume that every node knows its 2-hop neighbors and can assess the seriousness of the impact inflicted by the failure of one of the neighboring nodes. Unlike these approaches, the RIM algorithm, denoting *Recovery by Inward Motion*, requires just 1-hop information [68]. RIM is a localized scheme that limits the scope of the recovery process and operates in a distributed manner. The main idea is to move the neighbors of a failed node A_f inward towards the position of A_f so that they would be able to reach each other. Fig. 9 illustrates the idea. The rationale is that these neighbors are the ones directly impacted by the failure, and when they can reach each other again, the network connectivity would be restored to its pre-failure status. The relocation procedure is recursively applied to handle any node that gets disconnected due to the movement of one of their neighbors (e.g., those which have moved towards the faulty node). The main advantages of RIM are its simplicity and effectiveness. RIM employs a simple procedure that recovers from both serious and non-serious breaks in connectivity without checking whether the failed node is a cut-vertex or not. The simulation validation of RIM has shown that RIM does well in sparse networks and outperforms approaches, such as DARA [65], that use 2-hop neighboring information. However, with higher node densities RIM tends to move many nodes and increases the total travel overhead on the network. Nonetheless, the overhead per node would still make RIM a favorite approach since it balances the load among nodes.

To limit the motion overhead imposed by RIM, a *Volunteer-instigated Connectivity Restoration* (VCR) algorithm has been proposed [71]. In VCR the neighbors of the failed node volunteer to restore connectivity by exploiting their partially utilized transmission range and by repositioning closer to the failed node. These neighbors volunteer by

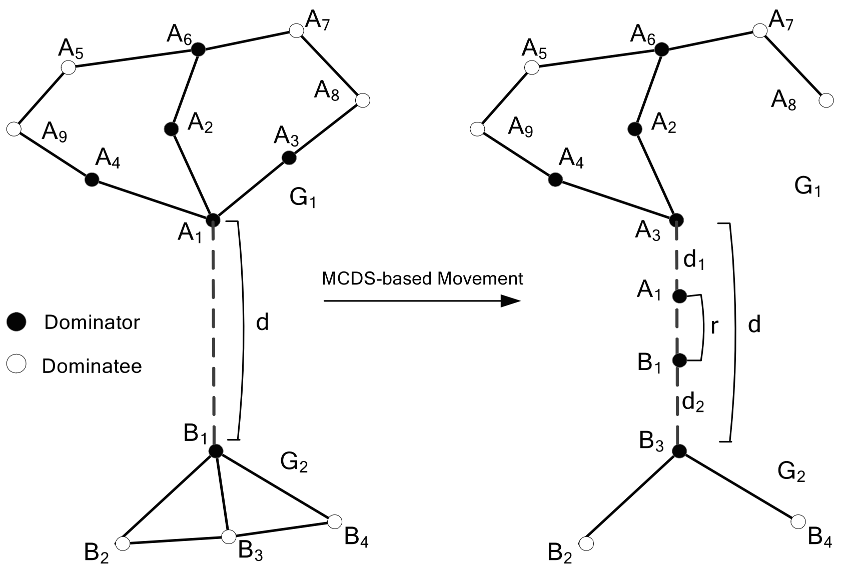

Fig. 8. Illustraing how to connect two disjoint partitions, G_1 and G_2, using the approach of [70]. The nodes in the two partitions that are closest to each other are A_1 and B_1. Let the distance between these two nodes be d, which is assumed to be greater than the transmission range of nodes, r. In order to establish connectivity some nodes from both G_1 and G_2 need to be positioned along the line A_1B_1 such that they remain connected to either G_1 or G_2 and have a distance less than r. Since B_1 is a dominator, when it moves, it has to be replaced by the closest dominatee if available, which is B_3 in the example. Moving the dominator A_1 will requires the relocation of the closest dominators, namely, A_3, since there is no dominatee connected to A_1. Because node A_3 is not connected to any other dominator, no further relocation is necessary.

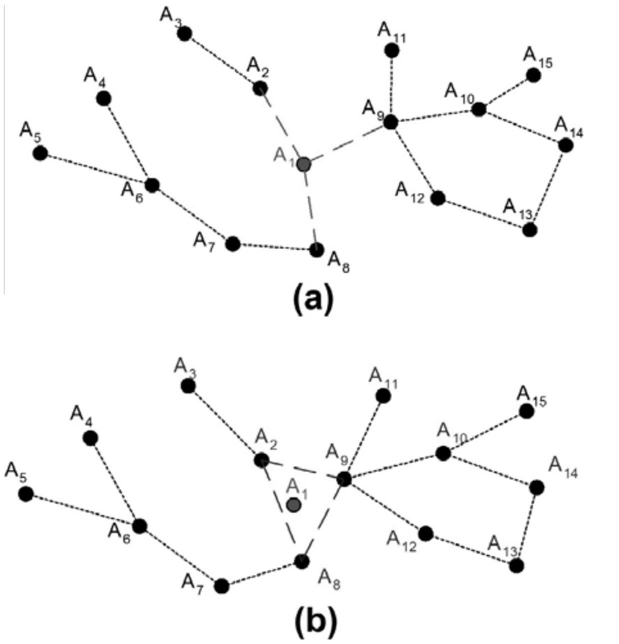

Fig. 9. Illustrating the basic idea of RIM; (a) When A_1 fails, its neighbors A_2, A_8, and A_9, applies RIM; (b) Nodes A_2, A_8, and A_9, move inward towards A_1. Such a motion triggers relocation of A_3 to stay in contact with A_2, and then relocation of A_{10}, A_{11}, and A_{12}, to sustain their links with A_9.

increasing their transmission power and moving towards the failed node A_f. In order to avoid increased medium access collision in the vicinity of A_f, VCR applies a diffusion force among volunteer nodes based on their transmission range so that they spread while staying connected.

Another variant of RIM has been published recently [72]. The approach is called *Least Distance Movement Recovery* (LDMR) and operates in a distributed manner. The idea is for a set of direct neighbors of the failed node A_f to move toward A_f, very much like RIM [68], while their original positions are replaced with the nearest uncritical nodes. The recovery process starts with a search phase where each neighbor broadcasts a message containing the failed node ID, neighbor node ID and, Time-To-Live to limit the scope of the recovery. A candidate node responds to all requests. When a neighbor of A_f receives responses, it chooses the best candidate based on a certain criteria (e.g., distance). To avoid overbooking a candidate, a confirmation message is sent to ensure that no two neighbors of A_f will rely on the same candidate. If uncritical nodes are not available, the cascaded relocation of RIM is applied.

4.2.4. Considering secondary performance objectives

There have been variations of the above approaches which consider some constraints or additional objectives. For instance, LeDiR [73], denoting *Least-Disruptive topology Repair*, considers the connectivity restoration problem subject to path length constraints. Basically, in some applications such as combat robotic networks and search-and-rescue operation, timely coordination among the nodes is required and extending the shortest path between two nodes as a side effect of the recovery process would not be acceptable. LeDiR relies on the local view of a node about the network to relocate the least number of nodes and ensures that the shortest path between any pair of nodes is not extended relative to its pre-failure status. When a node A_f fails, its neighbors will individually consult their possibly-incomplete routing table to decide on the appropriate course of action and define their role in the recovery if any. If the failed node is a critical node, i.e., cut-vertex, the neighbor A_i that belongs to the smallest partition reacts. LeDiR limits the relocation to nodes in the smallest disjoint partition in order to reduce the recovery overhead. The smallest block is the one with the least number of nodes and would be identified by finding the reachable set of nodes for every 1-hop neighbor of A_f and then picking the set with the fewest nodes. Again, the routing table will be used for that. Intra-partition connectivity is sustained through cascaded relocation as in DARA [65] and RIM [68], where a node A_j that loses contact with a neighbor A_i travels toward the new position of A_i.

A variant of LeDiR, called a *Least-Movement Topology Repair* (Le-MoToR) algorithm, has been recently published [74]. Compared to LeDiR, Le-MoToR relocates a node A_j in a partition on a different travel path by making moving parallel to the line A_iA_f. This modification reduces the number of moved nodes and sustains the coverage achieved by the node in the smallest partition by not shrinking the topology towards A_i after it replaces A_f. A sample execution of Le-MoToR is shown in Fig. 10 with the comparison to RIM.

A combined coverage and connectivity metric has also been considered in the recovery process. An example is the NORAS approach [39] discussed earlier which factors in the node's coverage when designating a replacement for connectivity-critical nodes. The coverage conscious connectivity restoration (C^3R) approach [75] opts to handle the case when no healthy node can be spared to replace the failed node. The main idea is to exploit both temporal and spatial domain. Basically, the neighbors of A_f will collaborate in the recovery by taking turns. Each participating node will reposition to the vicinity of A_f, serve the network for some time and then go back to its original location. A heuristic solution has been proposed to identify the nodes that should be involved and a schedule is devised for them to serve the area covered by A_f. This leads to intermittent connectivity and monitoring of all the originally covered spots. A sample execution is shown in Fig. 11. An optimized version of C^3R, called ECR, is also proposed. ECR is geared for energy efficiency at the expense of coverage and connectivity and is suitable for applications where network longevity is a prime objective. ECR devises a recovery schedule that minimizes the ratio of travel-imposed energy to that being consumed while a node is stationary. Such a scheduling problem has been formulated as a linear program.

4.2.5. Constrained motion

Repositioning may itself be constrained due to various reasons such as the environment and application level constraints. For example, the node may be assigned an important task that is spatially tied to a certain spot and thus cannot move. To handle such a scenario, the DARA approach is extended in [76] to factor in the importance of on-going tasks in the selection of a candidate node for replacing A_f. In addition, a hybrid Application-centric Connectivity Restoration (ACR) algorithm is proposed in [77]. ACR factors in application level interests besides efficient resource utilization while recovering from critical node failures. Like DCR [40], ACR determines critical nodes and designates for them backups as part of pre-failure planning. Each node discovers in a distributed manner whether it is critical or not based on local topology information. Each critical node (primary) picks a suitable backup that can satisfy application-level constraints. While choosing a backup, a primary node strives to find a nearby uncritical node in order to limit the scope of the recovery and reduce the overhead. Moreover, ACR strives to minimize the effect of node failure on coverage and connectivity by engaging strongly connected nodes with overlapped coverage. ACR can be viewed as a hybrid proactive and reactive strategy.

Finally, the location of the failed nodes may impose some motion restrictions. With the exception of RIM and its variants, the other reactive approaches for restoring connectivity are based on a single underlying principle of replacing the failed node without considering the possible fact that the location of the failed node could have been the reason for its failure. These approaches also tend to trigger a cascaded relocation of many nodes resulting in increased overhead. For example, DARA [65] and RIM [68] often terminate after engaging the nodes at the network periphery. To counter such shortcoming, an algorithm for Connectivity Restoration through node Rearrangement (CRR) is proposed in [78]. CRR pursues rearrangement of nodes while limiting the scope of the recovery to the vicinity of the failed node. The main idea is to reposition the 1-hop neighbors of the failed node so that the topology becomes strongly connected. The node rearrangement is modeled as a variant of the Steiner minimal tree formation problem to connect the 2-hop neighbors of A_f. The 1-hop neighbors A_f are then placed at the identified Steiner points (SPs). If the 1-hop neighbors are fewer than the SPs, the 2-hop nodes are relocated.

4.2.6. Comparative summary

Table 1 provides a comparison of the reactive schemes discussed in this subsection. Although designating backups is categorized as provisioned recovery, the restoration process is performed in response to the failure. Therefore, we have considered these approaches as hybrid schemes and included them in the comparison. The table highlights the following attributes:

- *Node Selection Criteria:* The nodes are picked based on several criteria such as distance to the failed node, node degree, and criticality, before or after the failure occurs.
- *Network State Needed:* To decide which nodes to move and their final locations, some network state need to be kept at each node. This ranges from knowing 1-hop neighbors to the whole topology.
- *Centralized/Distributed Operation:* The approaches can be either orchestrated by the sensors or controlled by a single base-station. In the latter case, the base-station would need global network information.
- *Movement Type:* The movement of the nodes may also differ. For instance, some approaches may move the nodes directly while others can distribute the load and follow a cascaded motion. In addition, the movement can also be constrained based on the environment or application level needs.
- *Performance metrics:* In addition to storing network connectivity, the recovery approaches also opt to achieve a secondary objective, e.g. increased cover-

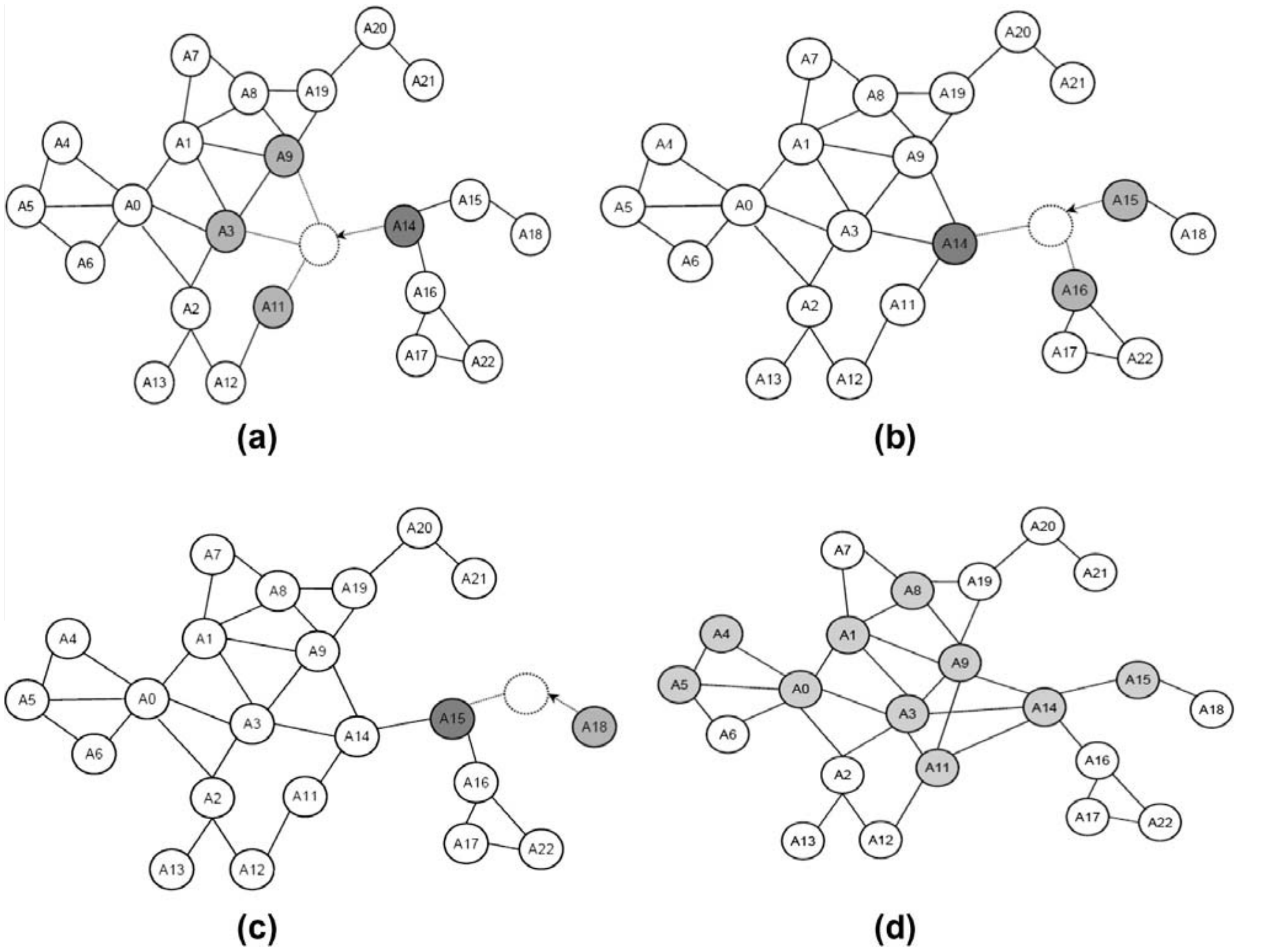

Fig. 10. Illustration of how Le-MoToR [74] involves nodes while restoring connectivity. Only 3 nodes are involved in the recovery process. (a) Node A_{10} fails. (b) Node A_{14} replaces node A_{10}. (c) Node A_{15} replaces node A_{14} and node A_{18} initiates replacement of A_{15}. (d) For the same scenario, more nodes (gray nodes) are involved in the recovery process when RIM [68] is applied.

age, and/or optimize some performance metrics such as energy efficiency and QoS. Energy efficiency is achieved via the reduction of movement and messaging overhead.

5. Tolerating multi-node failures

Due to the harsh surroundings, more than one sensor node may simultaneously fail. In addition, the network may suffer a large scale damage that involves many nodes and would thus create multiple disjoint segments. Restoring connectivity in this case is more challenging than the single node failure scenarios. In cases, where the simultaneously-failed nodes are not spatially adjacent, the problem is tackled as a multiple version of single node failures with special handling of potential resource conflicts. However, tolerance of simultaneous failure of collocated nodes is a significantly more difficult problem. Three strategies have been pursued in the literature to recover from collocated node failures. In the first strategy, the network topology is restructured by repositioning nodes from the various segments in order to re-establish connectivity. This methodology would support network self-healing and enable distributed implementation. In the second strategy, additional relay nodes are deployed to interconnect the disjoint segments. Finally, the third strategy involves mobile data mules that tour the area and carry data from one segment to another. In the balance of this section we first cover techniques for tolerating simultaneous non-collocated node failures. Then we direct our attention to spatially adjacent failures, highlighting the popular objectives of the recovery process and summarizing published schemes for each recovery strategy.

5.1. Handling simultaneous failure of non-adjacent nodes

A variant of the multi-node failure problem involves spatially non-adjacent nodes that fail at the same time. This may happen when nodes fail in various parts of the network one after the other such that the time between two consecutive failures is less than the time for tolerating a single failure. Independently handling the failure of the individual nodes in this scenario is not guaranteed to converge due to the potential resource conflict. For example, if node relocation is pursued and two failures are simultaneously handled in two parts of the network, a healthy node may be confronted with two conflicting requests and the recovery process will be disrupted. Consider the

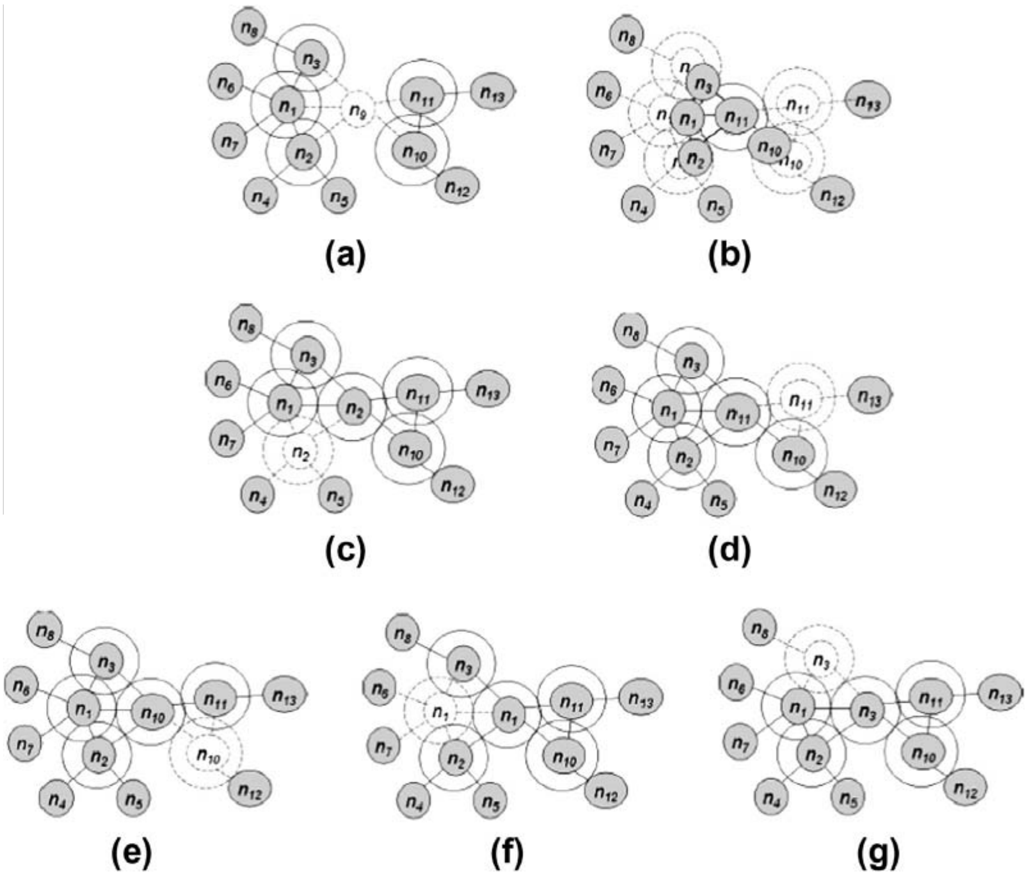

Fig. 11. Showing sample operation of C^{3}R [75]. When n_9 fails (a), node n_{11} acts as the recovery coordinator (b). Nodes collect the schedule from n_{11} and move back to their positions except n_2 which is the first node scheduled (c). The other neighbors move back and forth successively to tolerate the failure of node n_9 (d–g).

scenario in Fig. 12. The failure of A_1 would require the engagement of its neighbors, including A_7, in the recovery process. Now if before or during the recovery node A_6 fails, A_7 will have conflicting tasks since it is a neighbor of A_6 as well.

Provisioned recovery solution will be able to tolerate this type of failure up to k nodes by establishing k-connectivity or assigning k distinct backups, as discussed in Section 4. Meanwhile, most of the published reactive schemes can only tolerate a single node failure. Nonetheless, few have factored in the potential of having multiple non-collocated failed nodes in the network and proposed a procedure that ensures convergence and successful recovery. We categorize these schemes based on whether centralized or distributed processes are pursued. It should be cautioned that these techniques cannot handle collocated failures or major topology damage, which will be covered later in this Section 5.

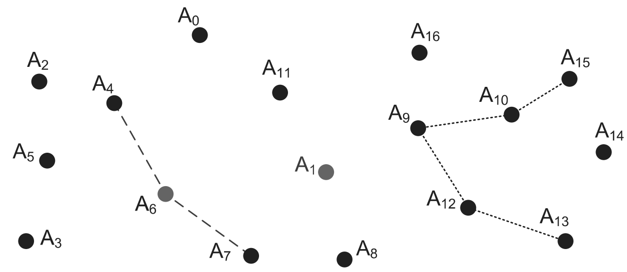

Fig. 12. The failure of A_1 and A_6 introduces a conflict at node A_7 since in many single node failure recovery schemes it has to be engaged in the recovery process for both of them.

Centralized schemes: This category of work handles network architectures in which a base-station can be notified of the failure of multiple nodes and develop a recovery plan that involves relocating nodes in the network. It is assumed that the position of the failed nodes can be collected by some other means such as via a flying unattended air vehicle (UAV). The base-station uses the position of the failed and healthy nodes to determine an optimal solution in terms of node movement. In [79], the recovery problem is formulated as an Integer Linear Program (ILP). The objective of the ILP based optimization model is to form a connected topology while minimizing the distance that the individual nodes have to travel and minimizing the loss in coverage caused by the failed nodes. In order to limit the search space, the solution is limited to the position of the nodes in the network prior to failure.

While the scheme of [79] factors in the positions of all nodes in the optimization model, such an assumption is not made in [80] and all possible locations are thus considered. The idea is based on transportation network flow model where eventually every node in the network should be able to go to all destinations, i.e., reach all other nodes, when the network is connected. Given the infinite solution space, the problem is modeled as a mixed-integer formulation and polygon approximation method is used so that the nonlinear terms in the model (i.e., due to distance computation) are linearized. Although such simplification leads to a sub-optimal solution, it has been shown it outperforms the solution of [79] in terms of the total travel distance. Nevertheless, this approach does not scale well and becomes computationally intractable after approximately 30 nodes.

Another centralized approach is proposed in [81] to improve scalability by reducing the number of candidate locations. A relay placement algorithm is used to determine the set of positions which can guarantee connectivity if relay nodes are to be deployed at those positions. However, instead of relay node deployment, the determined positions are occupied by existing nodes in the network. The relocated nodes are identified by forming a CDS of the individual partitions and selecting dominatee nodes whenever possible, as done in PADRA [54]. For minimizing the total movement distance of the nodes, a greedy heuristic is followed. In this approach, the closest target-dominatee pair is picked and the dominatee node is relocated to the corresponding target location. After populating each target location, the CDS of partitions are updated until all target locations are filled and connectivity is restored. The simulation results have indicated that the proposed heuristic scales well and outperforms the solutions of [79,80].

Distributed Schemes: Published schemes in this category apply a single node recovery solution and employ some synchronization mechanism in order to avoid resource conflicts during the recovery. For example, the DCR algorithm [40], discussed earlier, has been extended in [82] to handle multiple non-collocated node failures. The extended algorithm, which is named RAM, identifies critical nodes and designates distinct backups for them. The key feature that enables RAM to handle the simultaneous failure of multiple nodes lies in the backup selection process. The idea is to elevate the criticality of a backup node in order to deal with the possibility of having both primary and backup failing at the same time. The increased criticality warrants designating backups in a recursive manner. To limit the scope and avoid making the entire node population critical, a node is allowed to serve as a backup for multiple primary nodes. A mutex is used to avoid race conditions while relocating nodes in order to prevent a new failure from causing an on-going recovery process to diverge. The extended approach is shown to successfully tolerate the failure of multiple non-collocated nodes and two adjacent nodes.

Similarly failure handlers (i.e., backups) have been used in [83] to handle multiple node failures that happen simultaneously at different locations. Basically, the idea is to run the PADRA [54] approach at multiple locations using the failure handlers as the initiators of the processes. Two types of failure handlers, namely primary and secondary, are designated for a node. In case, the primary handler cannot complete the recovery within a pre-specified duration, the secondary handler steps forward. However, when the nodes are relocated in a cascaded manner, two recovery processes may need to relocate the same nodes to two different locations which creates race conditions. This problem is addressed via lock procedures meaning that the node does not respond to a request until it is done with another request. Several cases were considered and resolved as shown in Fig. 13. In case the nodes are locked, then a secondary failure handler will need to back off and wait for a certain amount of time to retry in the future hoping that some of the locked nodes will be released. In this way, any number of failures can be handled as opposed to the RAM approach.

5.2. Tolerating collocated failures through node relocation

Restoring connectivity after the simultaneous failure of multiple collocated nodes through the repositioning of some of the healthy nodes is very challenging. The main difficulty is determining the scope of the damage and coordinating the response of the individual nodes. Unlike the single node failure where the neighbors of the failed node have a consistent and deterministic view of the problem, a node relocation-based recovery has to be based on partial or uncertain information about how many nodes have failed and where the other healthy nodes are. As a result, it is necessary to make some assumptions such as knowledge of the routes, knowledge of the center of the region, availability of cameras, etc.

For instance, the main idea of the approach of autonomous repair (AuR) is based on repositioning towards the center of the area which is assumed to be known in advance. AuR regroups the healthy nodes by moving towards one another and towards the center of the deployment area [84]. The design principle of AuR is based on modeling connectivity between neighboring nodes as a modified electrostatic interaction based on Coulomb's law between charges. In AuR, the recovery is localized with nodes only interacting with their immediate neighbors. The neighbors of the failed nodes are to lead the recovery process by spreading out towards the lost nodes, causing the intra-segment topology to be stretched. If connectivity is not restored, the segment is then moved as a block towards the

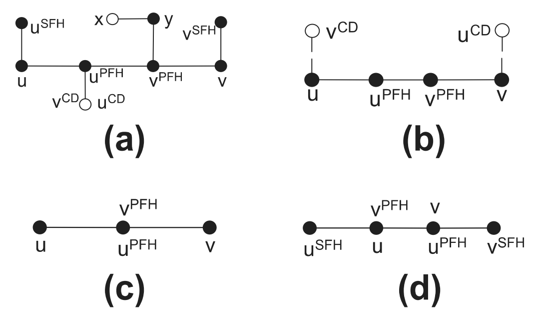

Fig. 13. Situations where paths for recovery have common nodes/links for the nodes u and v. PFH and SFH stand for primary and secondary failure handler of the corresponding node, respectively, while CD stands for the closest dominatee node. In cases where u and v fails around similar times, their closest dominatee nodes maybe the same and thus a lock mechanism will be needed. Once locked, a node cannot be used for relocation. The figure is taken from [83].

center of the deployment area. Moving all segments towards the center will increase the node density in the vicinity of the center point and ensures that the connectivity gets reestablished. Fig. 14 illustrates the idea.

Similarly the Distributed algorithm for Optimized Relay node placement using Minimum Steiner tree (DORMS) assumes that the center of the network is known in advance [85]. The problem is modeled as a relay placement problem but the relays are picked among the existing nodes in the respective partitions. Since in autonomously operating network it is infeasible to perform a network-wide analysis to diagnose where segments are located, DORMS moves relay nodes from each segment toward the center of the deployment area. As soon as those relays become in range of each other, the partitioned network resume operation. The goal of DORMS is to design an efficient topology, in terms of the path length among segments, while minimizing the number of required additional nodes. Therefore, DORMS further models such initial inter-segment topology as a Steiner Minimum Tree (SMT) in order to reduce the count of required relays. In order to find a topology which reduces the node count, DORMS employs k-LCA [86], which is the best known approximation algorithm for finding an SMT. The identified SPs are populated and the other initially-employed relays return to their respective segments to resume their pre-failure duties.

Another distributed approach to handle multiple failures is presented in [87] based on the nodes' knowledge of their full path to the sink. The pre-failure route information is utilized in order to determine the location of the damaged nodes. The location of nodes along the path to the sink is collected when the paths are established. Thus, upon partitioning, nodes can attempt to reconnect to the sink node by moving to the next hop towards the sink. When a node "*i*" discovers the failure of its next hop "*j*" on the path to the sink, node "*i*" will relocate to the position of "*j*". This step may continue over the known pre-failure data route until reaching the sink node or finding an alternative path to the sink node. Since the former path was once operational, this approach is shown to guarantee connectivity. However, because many nodes may do so in a partition, the recovery cost can be high. To limit the overhead, the recovery process elects only one node to lead the motion towards the sink. The leader node is selected in a distributed manner based on the distance to the failed node. When the leader node moves, cascaded node relocation within the partition is pursued in order to sustain intra-partition connectivity. For that DARA [65] is employed. The approach is further extended in [88] by factoring in proximity to the failed node and using PADRA [54] to optimize the intra-partition cascaded relocation process.

Meanwhile, the approach of [89] is based on incomplete node failure information. This work assumes that the nodes are equipped with primitive cameras to collect partial topology information, mainly the number of boundary nodes. The connectivity restoration problem is then modeled using Game Theory. The pay-off function is based on the node degree, where a partition opts to grow the node degree of its nodes, in particular its boundary nodes, by joining other partitions. Due to the lack of centralized control, the nodes to be moved and the direction of the movement are determined with limited knowledge about the partitioned topology and about the location of the failed nodes. Specifically, the nodes collaborate to reach Nash equilibrium. When playing, the number of nodes in neighboring partitions, and the number of boundary nodes that face neighboring partition are estimated using cameras. While this requires some image processing and may yield inaccurate information, the probabilistic nature of the approach can handle such uncertainties. Basically, a probabilistic model is used to determine the boundary nodes distributions for the partitions. Each partition then tries to merge with the partition that maximizes the pay-off and node relocation takes place to complete such merge until network wide connectivity is achieved. The simulation results indicate that the performance of the proposed approach is superior to the other distributed approaches such as [88] and comes very close to centralized solutions.

Another Game Theory based approach is proposed in [90]. By assuming complete knowledge of the location and count of the partitions and failed nodes, each partition (i.e., a representative from the partition) is used as a player in a game. Again the pay-off function is based on the node degree. The representative opts to maximize the pay-off for its partition, which motivates the partition to move towards other partitions. The approach is centralized and every representative will know the pay-off function of other partitions and the network eventually reaches Nash equilibrium when all partitions are connected. The motion towards other partitions takes the form of relocating nodes from each partition towards each other. This relocation is performed by following a similar approach to that of [70]. Basically, the closest nodes from each partition are picked and move towards each other on the line connecting them.

None of the proposed approaches consider the terrain conditions and obstacles in the environment when the nodes are moving. Therefore, a direct path is always assumed between two locations. To counter this issue, the approach in [91] does not assume that a direct path is always available during movement and presents a solution by considering terrain features. The approach factors in the terrain conditions to determine the right nodes to move and find an obstacle-free and energy-efficient path. Application area is assumed to be composed of regions with different risk and elevation values. Energy consumption per meter is not constant and it is minimized by determining optimal trajectories through applying A^* search algorithm to reach the destination via best route. The terrain is divided into cells with different characteristics such as flat, grass and water. Therefore, each cell is represented by a number indicating the risk and elevation factor of the cell as shown in Fig. 15. The algorithm finds the most energy-efficient path by considering the numbers in the cells and the distance from source to destination. The A^* is a path finding algorithm used in robotics. The algorithm is implemented for PADRA [54] and DARA [65] under realistic terrain conditions. The results have revealed that the energy consumption due to movement is significantly higher and in some topologies DARA or PADRA's direct movement even may not converge.

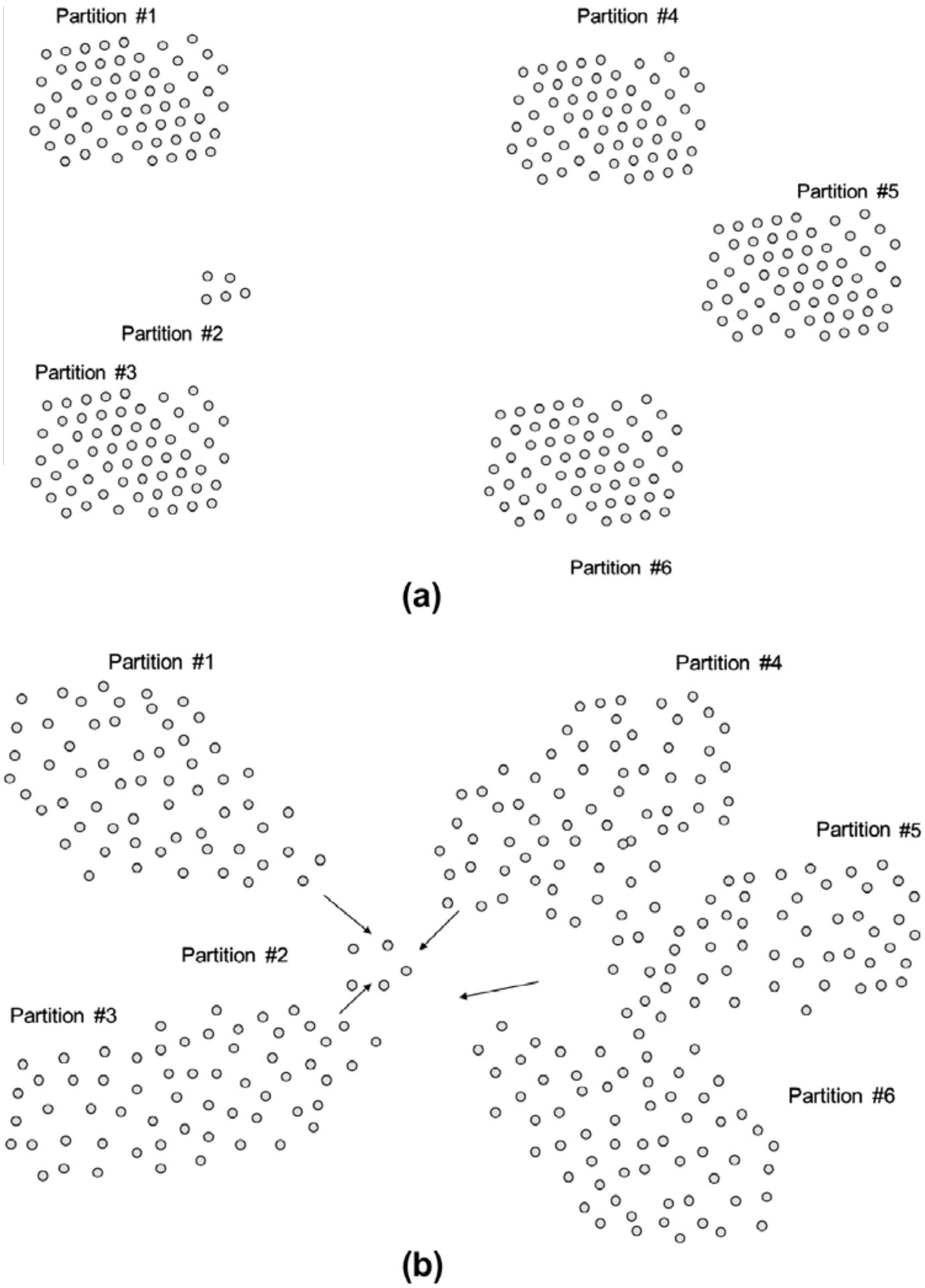

Fig. 14. AuR employs self-spreading and motion towards the center if the deployment to reconnect the disjoint segments of the network, (a) the damaged topology, (b) AuR in action.

Table 2 provides a comparative summary for the node-relocation based multi-node failure recovery schemes discussed in this subsection.

5.3. Recovery through deployment of stationary relays

In setups in which the nodes are not mobile or the number of healthy nodes is not sufficient to re-establish connectivity while meeting the application requirements, the deployment of additional nodes is inevitable. The deployed nodes would act as relays between the partitions/segments. The terms "partition" and "segment" are used interchangeably in published work in this category and also in the balance of this section. The connectivity restoration problem then becomes determining the fewest number of relays and their locations so that data routes are formed between every pair of partitions. The relay nodes are typically assumed to be stationary nodes. However, mobile relays are also possible as will be discussed in the next subsection.

Since the main goal of placing relays is to restore connectivity, most of the proposed approaches in the literature solely focused on achieving this objective with the least number of relays as a way to limit the recovery cost. There are, however, some approaches which have considered other objectives, in addition to connectivity, which

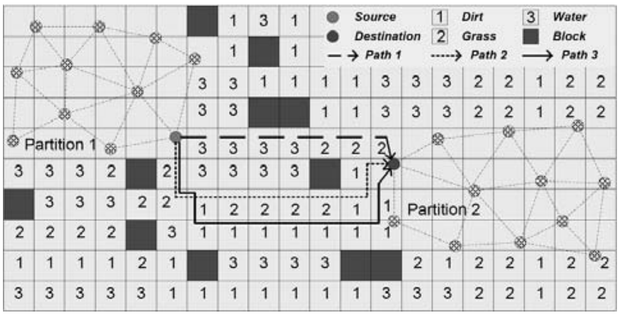

Fig. 15. Determining the most energy-efficient and obstacle-free path between two locations. Source picks path 3 to minimize energy consumption even though path 1 and path 2 have shorter lengths.

makes the problem becomes more challenging. This subsection surveys published work and structures the presentation based on the objective of the relay placement optimization. It is worth noting that the placement approaches are typically centralized; however distributed implementations have also been discussed in some work. A comparative summary of all the approaches for stationary relay placement is provided in Table 3.

Connectivity with the least number of relays: Restoring connectivity using stationary relays requires determining the fewest number of relays and their locations so that data routes are formed between every pair of partitions. Thus, the recovery problem is mapped to finding the Steiner Minimum Tree with Minimum number of Steiner Points (SMT-MSP), which is shown to be an NP-hard problem by Lin and Xue [41]. The SMT-MSP is a well-studied problem in the literature and a number of heuristics have been proposed. A summary and comparison of some of the published techniques can be found in [8]. To avoid repetition we summarize sample of early and fundamental work and focus on the recently-published solutions in the context failure recovery.

Lin and Xue [41] have proposed an algorithm which populates relays on the edges of the minimum spanning tree (*mst*) of the terminals which represents the partitions. First, the algorithm constructs the complete graph $G = (V, E)$ of terminals where V is the set of terminals and E is the set of all edges (u, v) where u, v are in V. Then using Kruskal's *mst* algorithm, the tree edges are computed. Relays are populated along each edge in the tree at a distance of at most R apart, where R is the communication range of a node. In [49], Chen et al. have shown that the approximation ratio of the algorithm presented in [41] is equal to 4.

In [42], Cheng et al. have presented a three-steps approach to tackle the problem. It first connects the nodes which can directly reach one another. In the second step the algorithm considers each subset of three terminals (u, v, z) and forms a 3-star if there exists a point s such that s is at most R units away from u, v and z. At the end of this step, the algorithm forms a set of islands and individual terminals. In the third step, the algorithm establishes connectivity by filling the gap between all u and v's, such that $(u, v) \in E_{mst}$, where E_{mst} is the set of edges on the *mst*, and u and v are in different connected components.

To counter the complexity of the problem, Lee and Younis [92] have modeled the deployment area as a grid of equal-sized cells, and each network segment is assumed

Table 2
Reactive approaches for multiple node failures.

Paper	Level of connect.	Additional goal	Centralized distributed	Adjacent failures	Network state	Node selection criteria	Movement
[79]	1	Minimizing total distance and meeting a certain coverage	Centralized	No	Global	Based on network-wide optimization	Controlled
[80]	1	Minimizing the total and maximum travel distance	Centralized	Yes	Global	Based on network-wide optimization	Controlled
[81]	1	Minimizing the total and maximum travel distance	Centralized	Yes	Global	Based on network-wide optimization	Controlled
[82]	1	Minimizing the total distance and the scope of the recovery	Distributed	Yes	1-hop	To be a non-critical node, degree and proximity	Cascaded
[83]	1	Minimizing the total travel distance and messaging cost	Distributed	No	2-hop	To be a dominatee node and proximity to the failed node	Cascaded
[84]	1	Minimizing the total travel distance	Distributed	Yes	1-hop	All nodes are engaged	Block
[85]	1	Minimizing the number of populated relays and generating more efficient topology in terms of the average degree of connectivity and balanced traffic load	Distributed	Yes	None	The number of lost connections with the neighbors	Cascaded
[87]	1	Minimizing the total and maximum travel distance	Distributed	Yes	2-hop	Proximity to sink (leader), node degree and node id (cascaded movement)	Cascaded
[88]	1	Minimizing total distance	Distributed	Yes	2-hop	Proximity to the failed node (leader selection); node degree and to be a dominate node (cascaded movement)	Cascaded
[89]	1	Minimizing the total travel distance and coverage loss	Distributed	Yes	1-hop	Proximity to the destination partition	Cascaded
[90]	1	Minimizing the total travel distance	Centralized	Yes	Global	To be a dominatee node and proximity to the void area (where nodes have failed)	Cascaded
[91]	1	Minimizing energy consumption	Distributed	Yes	2-hop	Minimum overhead in terms of energy consumption	Cascaded

Table 3
Comparison of heuristics for recovery through stationary relay placement.

Name	Objectives (Other than connectivity)	Approach	Connectivity degree	Model	Time complexity
SMST [41]	Minimizing relay count	Centralized	1	Each partition is represented by a single node	$O(n \log n)$
SMT-MSP [49]					$O(n^3)$
FeSTA [93]					$O(n^4)$
IO-DT [94]					$O(n^2)$
SpiderWeb [98]	Providing high-quality topologies	Centralized/ Distributed	1 and 2		$O(n \log n \lceil \frac{d}{R} \rceil)$, where n is the number of partitions, d is the length of longest line, and R is the relay node transmission range
CIST [95]	Minimizing relay count	Centralized	1	Each partition is represented by its members (multiple)	Not available
ORP [35][37]	Maximizing inter-relay node degree while maintaining cost constraints	Centralized	Opts is to maximize		
CBP-D [38]	Maximizing inter-relay node degree while maintaining cost constraints and meeting a delivery delay bound	Centralized	Opts is to maximize		
CORP [92]	Providing high-quality topologies	Centralized/ Distributed	1	Each partition is represented by a cell	$O(n,r)$ such that $r = o\left(\lceil \frac{f(X,Y)}{2} \rceil\right)$, where n is the number of partitions, and X and Y are the two furthest partitions with $f(X,Y)$ being the Manhattan distance between X and Y
EQAR [96]	Guaranteeing link bandwidth	Centralized	N/A		$O(n^2 \lvert V_Q \rvert \log(\lvert V_Q \rvert + \lvert E_Q \rvert))$, where n is the number of partitions, $\lvert V_Q \rvert$ is the number of cells, and $\lvert E_Q \rvert$ is the number of directed edges in the grid
QRP [97]	Guaranteeing link bandwidth		N/A		Same like CORP

to be located in the middle of the cell. The optimization problem is then mapped to selecting the fewest number of cells for populating relays such that all segments are connected. A Cell-based Optimized Relay node Placement (CORP) algorithm is proposed. CORP is a polynomial-time algorithm that pursues greedy heuristics. It defines the best neighboring cell of a segment Seg_i as the one that lies on the shortest paths connecting Seg_i to the other segments. CORP operates in rounds. In each round, the best cells are selected and populated with relays based on where the most recently populated nodes are located. The overall placement process converges by populating relays inwards until all relays become reachable to one another. After all segments are connected, the algorithm prunes redundant relay nodes. Fig. 16 illustrates the operation of CORP. A distributed implementation of CORP has also been proposed.

Another recent approach is FeSTA [93], which denotes Federating Network Segments via Triangular Steiner Tree Approximation. FeSTA deploys relay nodes and forms connected components of segments by finding local sub-optimal solutions for groups of three segments. A segment is represented by a terminal and all possible 3 distinct subsets of segments are listed. For every triangle, i.e., subset of three terminals, FeSTA decides either to form a new connected component or to incorporate the terminals of that triangle to an existing connected component based on the required number of relay nodes. Forming a new connected component, in essence, is equivalent to finding SMT-MSP of the subset of three terminals. The terminals (segments) join an existing connected component, if the cost for connecting these terminals individually is less than forming a distinct component. After all segments are covered, i.e., become part of a connected component, the scale of the problem is reduced to linking the individual connected components by steinerizing an edge between two nodes in distinct components.

FeSTA considers all possible triangles. Since for n terminals there are $\binom{n}{3} = O(n^3)$ possible triangles, the runtime complexity grows significantly for large networks. To reduce such complexity, the Incremental Optimization based on Delaunay Triangulation (IO-DT) algorithm is proposed [94]. First, IO-DT proves the feasibility of finding the optimal solution for the SMT-MSP problem for the case of three terminals. Unlike FeSTA, IO-DT tries to minimize the total relay count for larger networks by using selected subsets of three terminals. IO-DT observes that most of the triangles considered by FeSTA, are not needed since big triangles having multiple terminals inside may require many redundant relays. This observation is illustrated in Fig. 17. The IO-DT algorithm is designed to avoid those redundancies and thus to improve time complexity significantly. In computational geometry Delaunay Triangulation guarantees that no terminal is located inside a triangle (i.e., subset of three terminals) which is the desired case in this problem. Therefore the algorithm finds the subset of three terminals generated via Delaunay Triangulation. The IO-DT algorithm first sorts the triangles according to their weights (i.e., the minimum number of relays to connect the corners of the triangle). In each iteration the algorithm steinerizes the triangle as part of the final topology if selecting, such a triangle provides a reduction in the total number of required relays as compared to an *mst*-based solution. A worked example of IO-DT algorithm is illustrated in Fig. 18.

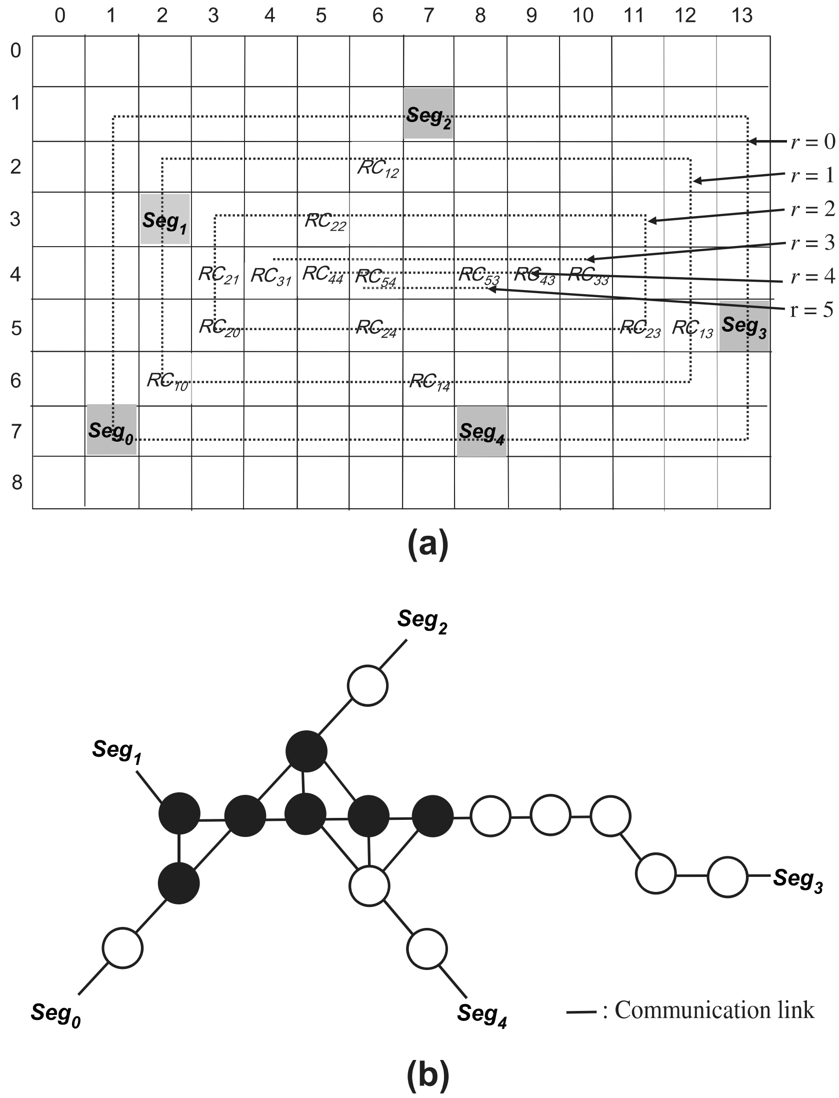

Fig. 16. Illustrating how CORP operates in rounds, shown in (a), to form the topology, shown in (b). The area is modeled as a grid and in each round a relay is populated in the best neighboring cell for each segment. In the 6th round all segments become connected.

The approaches mentioned assume that a partition is represented by a single terminal. However, in reality, a partition may consist of several sensors and thus the connection interface point with other partitions may not be the same node all the time. Based on the assumption of partitions with sensors, a two-step heuristic called CIST has been proposed in [95]. CIST also strives to minimize the required number of relays. Unlike other schemes, each segment is not represented by a single terminal, instead, all nodes located on the boundary of the individual segments are considered. CIST leverages the ideas promoted by FeSTA. First, CIST determines the *mst* edges between segments and estimates the corresponding number of relays required to establish connectivity on these edges. CIST then considers all combinations of three segments that include two segments with an *mst* edge. For each of these combinations, CIST determines the fewest relays needed to form SMT-MSP for a triangle whose vertices are located in distinct segments, and reports the reduction in the relay count relative to the *mst*-edge based connection of the corresponding three segments. In the next step, CIST connects groups of three segments with positive gain, i.e., reduction in relay count, by steinerizing the corresponding triangle. Finally, the remaining segments are connected via steinerizing *mst* edges. Fig. 19 illustrates the basic idea. Instead of deploying additional relays, the approach of [81] employs PADRA [54] to determine dominatee nodes within the segments that can be moved at the spots identified by CIST. The criterion of selecting which dominatee nodes to be

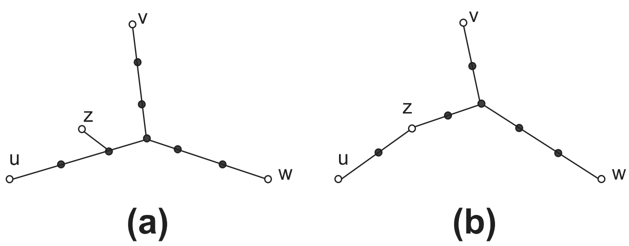

Fig. 17. Illustration of redundant node deployment as a result of steinerizing a triangle having another terminal inside. (a) 7 Relays are required (b) 6 relays are required for connecting 4 terminals.

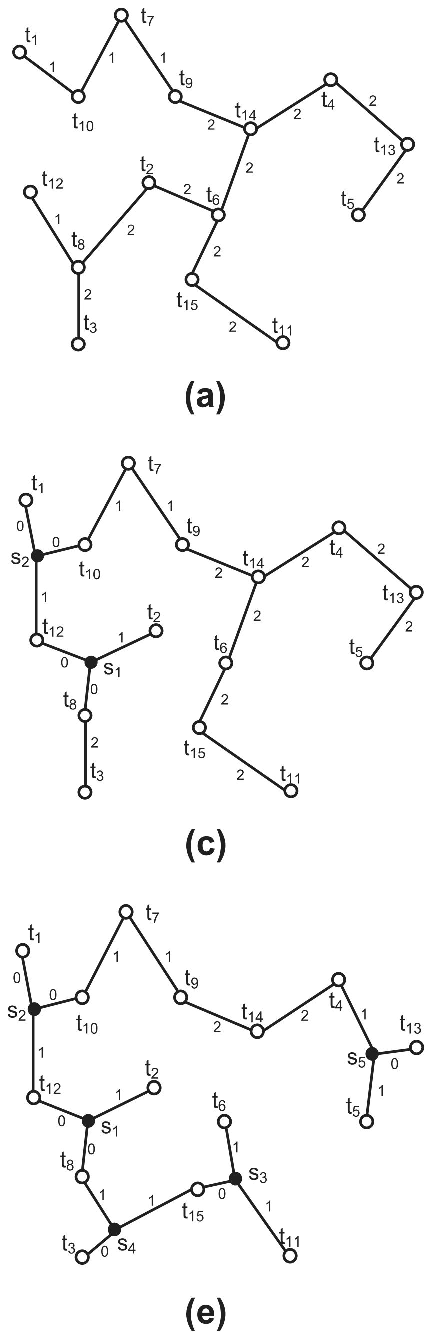

Fig. 18. A Step-by-step illustration of how IO-DT algorithm works.

repositioned is based on minimizing the total travel distance for these nodes.

Recovery with QoS objectives: In some node failure scenarios the recovery process needs to re-establish connected inter-segment topologies with desired features, e.g., high node degree, bi-connectivity, etc. In other words, unlike the work discussed above, support for some intra-network QoS requirements is desired. Multiple heuristic solutions have been developed to tackle various variants of this problem.

The first approach, which is named EQAR, opts to form a connected topology using the least number of relays while meeting inter-segment capacity constraints [96]. The inter-segment QoS requirements may be just a byproduct of the damage since the segments may be of different sizes and in turn, the volume of the generated traffic may widely vary. In addition, each segment may have its own QoS requirement depending on the application and the number of video and imaging sensors that the segment has. The deployment area is modeled as a grid with equal-sized cells. The problem becomes identifying the cells that ought to be populated with relays so that the total number of deployed relays is reduced and the QoS goals are met. EQAR introduces a cell-based cost function based on the residual capacity of the relays which have been deployed in the cell. The optimization problem is then mapped to finding the

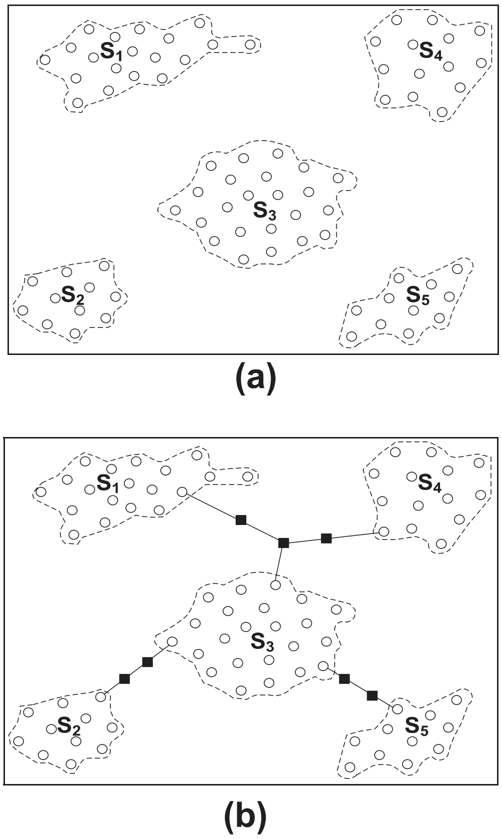

Fig. 19. An illustration of how CIST works. (a) The initial topologies of segments. Dashed lines represent *mst* edges. (b) CIST first processes the triangular subset $\{S_1,S_3,S_4\}$. The dark rectangles represent relays. CIST then processes the triangular subset $\{S_2,S_3,S_5\}$. Since the gain of the subset is turned to be zero, the *mst* edges that connect these segments are steinerized.

cell-based least cost paths that collectively meet the QoS requirements. In other words, the objective of the optimization is to maximize the utilization of the residual relaying capacity. Increasing the utilization of relays also increases the connectivity and allows the resultant inter-segment topology to be more resilient to local damage.

An extended version of the CORP approach [92], discussed earlier, is developed to support inter-segment capacity requirements. The new approach, which is called QRP [97], pursues greedy heuristics to populate the least number of relays such that the disjoint segments are connected and the QoS requirements between every pair of two segments are met. Again, QRP models the area as a grid of equal-sized cells and defines the best neighboring cell of a segment Seg_i as the one that requires the least relaying capacity to connect Seg_i to the other segments with QoS values being met. Like CORP, QRP operates in rounds. In each round, the best cells are selected and populated with relays based on where the most recently populated relays are located. This process concludes when all segments are connected using the newly deployed relays.

Meanwhile the SpiderWeb approach [98] opts to re-establish connectivity using the least number of relays while achieving high node degree in the formed topology. Published schemes often form an *mst* among the isolated segments. An *mst*-based topology usually makes some nodes a hot spot in terms of the traffic load and limits the achievable network throughput, and may thus deem the inter-node collaboration insufficient for specific application tasks. Unlike these schemes, this approach establishes a topology that resembles a spider web, for which the segments are situated at the perimeter. Such a topology not only exhibits stronger connectivity than an *mst* but also achieves better sensor coverage and enables balanced distribution of traffic load among the employed relays. The simulation results have shown that these distinct features are provided with a comparable relay count to that of an *mst*-based solution would involve. To further increase robustness, the approach is extended so that the final topology is guaranteed to be 2-vertex connected. Fig. 20 shows an example of the formed topology.

The problem considered by Al-Turjman et al. [35] is also considering an additional objective which is to improve the inter-relay topology. A certain number of relays were assumed to be available. The area is modeled as a grid and relays are to be placed on the intersection points. An *mst* is formed using the edges of the grid and assuming all intersections have relays. Only the selected intersection points are populated with relays. In the second phase, the unused relays, out of the allowed relay count, are populated so that the node degree of the inter-relay topology is maximized. To achieve that, the connectivity is modeled using the Laplacian matrix of a graph [36]. The connectivity is then measured by computing the second smallest eigenvalue λ_2, where λ_2 indicates the minimum number of links which if omitted the graph loses its strong connectivity, i.e., becomes partitioned. The approach is extended in [37] to handle the general case where the area is not modeled as a grid. To overcome the infinite search space, a set of edges between the segments are established by forming a Steiner tree and identifying points on the tree edges that are R units apart, where R is a the communication range of a relay. These points serve as the set of possible locations and the optimization formulation is then applied in a similar way to the case of a grid. The problem is also solved in

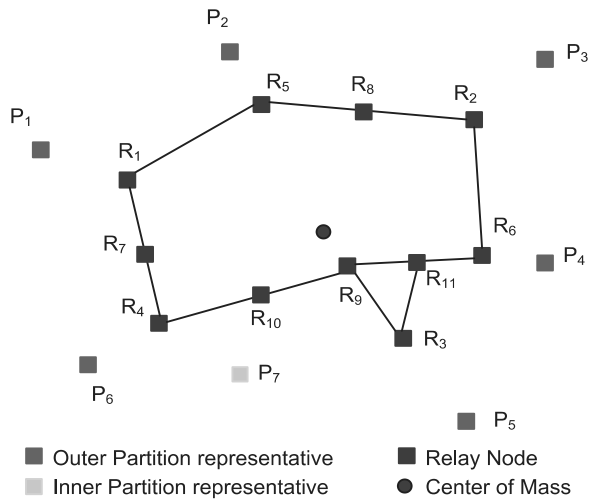

Fig. 20. An illustrative example of topology established by employing Spider-web relay node deployment strategy.

[38] while factoring in a data delivery delay bound as a constraint.

5.4. Re-establishing connectivity with mobile relay nodes

In this subsection, we focus on the strategies for connecting the segments of a partitioned network through the population of additional *mobile entities* (MEs). In the mobility-assisted algorithms, MEs play one of three roles; a *mobile relay node* (MRN) which relay data between segments, a *mobile data collector* (MDC) which visits the individual segments and carries data to the sink or a *mobile base-station* (MBS). The common objective of the ME-based algorithms is to find the shortest tour path along which the MEs visit segments. Furthermore, some research work considers constrained availability of MEs [99,100,29]. In the balance of this subsection we summarize the published ME-based connectivity restoration schemes.

Connectivity with a limited number of MEs: When the number of MEs is limited and not enough to restore connectivity among the segments, the MEs need to travel around to provide intermittent communication links. However, finding the shortest tour of mobile nodes among segments is a classical traveling salesman problem, which is known to be NP-hard even without considering a restriction on the node count [101]. Senel and Younis have investigated this problem and proposed a polynomial time heuristic for interconnecting Disjoint Segments with k MDCs (IDM-kMDC) with the shortest tour path of the MDC [99]. IDM-kMDC is a greedy approach that runs in rounds. First, it picks a node in each segment as a representative, computes an *mst* of representatives, and assigns an MDC on each edge of the *mst*. Then iteratively IDM-kMDC strives to reduce MDC count by one in each round by merging two collocated MDCs such that the merged tour length is minimized. For finding the merged tour, they compute a convex hull of segment representatives of the selected two tours and determine *collection points* for the representatives on or inside of the computed convex hull in a separate way. Fig. 21 illustrates the merging process using an example. The algorithm terminates as the number of MDCs reaches the given k.

Almasaeid and Kamal also tackle the same problem of bridging isolated segments by involving k resource-rich MRNs, where k is less than k^* which is the minimum number of required stationary relays to connect the node segments [100]. In their solution, the employed MRNs keep moving in the network to provide the intermittent connectivity over a time period instead of continuous connectivity. The main contribution of the scheme is to mathematically model the movement of the MRNs as a closed queuing network and achieve steady state results for distributing the MRNs in the network. They also study the effect of the number of the included MRNs and movement pattern on the *segments-to-segments* and *segments-to-sink (end-to-end)* data delivery latency. They have concluded that enhancing the movement pattern may lead to a better result than adding more MRNs.

Meanwhile, the problem studied in [29] is slightly different, where the movement of a limited number of MEs is to be optimized while considering two more roles of the ME, namely, serving as an MDC and an MBS [29]. The proposed algorithm first clusters the sensor nodes around pre-determined *stop points* using any clustering techniques [102]. At these stops, MDCs load or unload data and MBSs load data. Fig. 22 shows an illustration. Based on the clustered network topology, the authors model the end-to-end delay which is dominated by the movement time of MEs. The end-to-end data latency consists of a *loading time* and *unloading time* for an MDC and only *loading time* is considered for an MBS. The *loading time* is defined as the duration after a stop becomes unoccupied until an MDC or MBS arrives. In addition, the *unloading time* is defined as the time it takes an MDC to deliver data collected from sensors in another cluster at a certain *stop point*. Finally, the distribution of *loading time*, *unloading time* and the *end-to-end delay* is analytically derived for both MBSs and MDCs. It is concluded that changing the pre-assumed movement policy or increasing the speed of MEs might produce better performance than just adding more MEs in terms of end-to-end delay reduction.

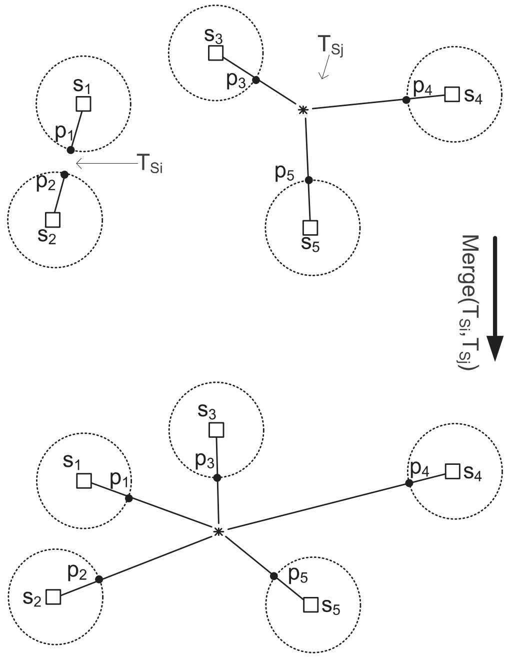

Fig. 21. An example of how IDM-kMDC merges two disjoint tours into one [99].

Connectivity with unconstrained ME count: When the availability of MEs is not constrained and the damaged spots are known, these MEs can simply serve as stationary relays and connectivity is re-established in a centralized manner by optimized relay node placement as discussed in earlier in this section. However, if connectivity is to be restored autonomously by the MEs without knowing where the segments are, the problem becomes different from its centralized counterpart. This recovery scenario arises when the region of interest is not accessible due to environmental conditions or security risks. In that case, the MEs are either randomly dropped to the damaged region or get placed to an accessible part of the region, and are then expected to self-organize to establish links between all segments. Obviously, if the initial batch of MEs

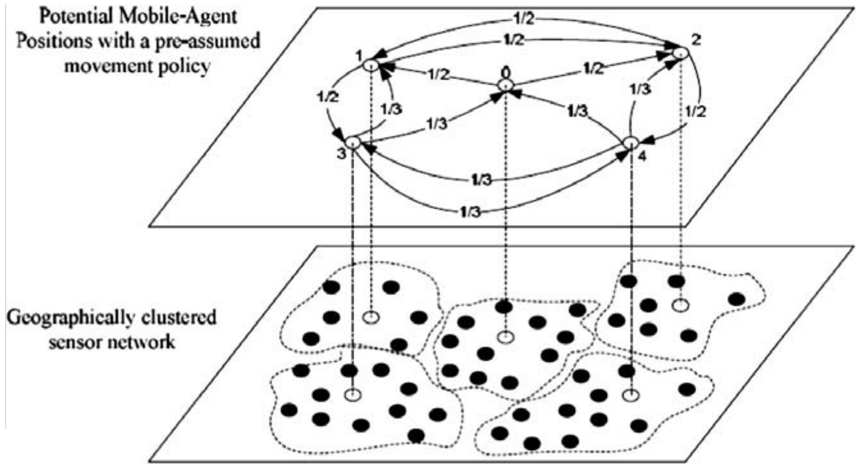

Fig. 22. A WSN clustered around predetermined stop points which are presented as white circles [29].

is not sufficient for forming a strongly connected inter-segment topology, additional batches need to be deployed. The MEs are often assumed to have onboard cameras that that can be used to recognize the damaged area.

Game Theoretic has been exploited in [103] to orchestrate the self-spreading process. Initially the MEs group in the center of the area to form a connected inter-ME topology. Two approaches are proposed to reposition the MEs in order to reach the network segments and establish a connected inter-segment topology. The first approach is based on repelling forces and does not guarantee connectivity. The inter-ME topology is stretched while maintaining its connectivity until it reaches the isolated segments. In return, each segment applies repelling forces to the MEs so they eventually stop. Obviously this approach is not efficient in terms of travel distance since all MEs move. However, it provides a good coverage since the nodes are spreading through the damaged area uniformly. The second approach overcomes these shortcomings and guarantees connectivity. In this approach, each ME computes its pay-off based on the number of outgoing links. The onboard cameras are used to estimate the node degree of other MEs and calculate their pay-off. In this way, the ME topology merges with the rest of the segments starting with the one having the highest degree of connectivity. Since only a limited number of MEs move, the total travel distance can be reduced significantly.

A variant of the ME-based connectivity solution is considered in [22], where MEs are placed to act as mobile access points in order to connect nodes in isolated network segments through airborne units or satellites. The deployed nodes usually do not have expensive radios for long-haul communication and usually serve limited geographical areas. The limited communication range and the occasional failure of nodes may result in partitioning the network, leaving some nodes unreachable to some others. To overcome such structural weaknesses in the network, MEs are employed to interconnect isolated subnetworks through an airborne relay, such as UAV or satellite. An ME acts as an access point for the nodes in its neighborhood. A similar idea is proposed in [105] to tolerate intermittent loss of connectivity.

MEs have also been used for maintaining and restoring connectivity among blocks of nodes in underwater environments [106]. Basically, in an underwater sensor network link may be broken due to the drift or the failure of nodes. To mitigate the negative effect on network connectivity, multiple underwater unmanned vehicles (UUVs), e.g., seabed crawlers, are utilized. While an UUV patrols, it listens for transmissions from sensors and identifies void regions in which no transmission is intercepted. When the loss of a critical inter-sensor link is detected, the UUV stays bridging the gap, as seen in Fig. 23. However, the proposed algorithm does not handle large void areas which cannot be covered by a single UUV and does not factor other objectives such as responsiveness or energy efficiency in the recovery process.

On the other hand, MEs have been used for collecting data in an energy-efficient and reliable manner in cases routes between different nodes of a network are not available. Although the proposed schemes in the literature are not dealing with partitioning conditions caused by node failures [104, 107–111], they can be easily extended to reconnect disjoint segments. For example, in [107,108] the authors have proposed energy efficient data collection protocols in single-hop WSNs by employing *data MULEs* which are capable of short-range wireless communication

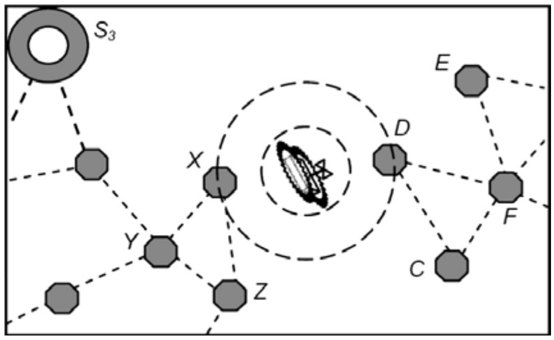

Fig. 23. An UUV patrols to identify a critical region in which an inter-sensor link is not available. The UUV stays in the area and bridges the disconnected sensors during a predefined duration then it deploys an additional sensor in the gap. The figure is from [106].

and move in an uncoordinated manner to provide connectivity in sparse WSNs. In addition [109–111] also propose the data collection schemes which place MBSs such as autonomous unmanned vehicles (AUVs) in order to prolong the network lifetime. Meanwhile, Ekici at al. [104] have presented an algorithm for defining a tour that strives to minimize the data loss rates due to buffer overflow for a specific data generation frequency. However, the tour length is not considered as an optimization metric in most of these schemes.

5.5. Recovery using a mix of mobile and stationary relays

Some of the published connectivity restoration approaches exploit a hybrid solution that employs a mix of stationary and mobile relays. The proposed algorithms in this category consider the situation in which the number of additionally populated relays is not sufficient to form stable data paths among segments. Thus, some of the relays take advantage of their mobility to establish intermittent links in order to overcome the resource shortage. The most popular optimization objective in this case is to minimize the tour length the relays that serve as MRNs. Some approaches have considered reducing the data delivery latency as well [112,113]. In addition, the number of relays that can move has been considered as a constraint in the recovery solution [114].

The design goal in [112] is to inter-connect multiple segments in a damaged WSN with a limited number of relay nodes while minimizing the average end-to-end delay between every pair of segments. The proposed delay-conscious Federation of multiple wireless sensor network Segments using Mobile Relays (FeSMoR) algorithm opts to achieve the minimal delay objective by exploiting relay mobility at the periphery of the federated inter-segment topology. FeSMoR operates in two phases. The first phase forms the Euclidian Steiner Minimal Tree (ESMT) of segments by employing DORMS [85] which generates an inter-segment topology that balances data traffic among the segments. In order to meet the relay availability constraints, in the second phase FeSMoR identifies an edge, part of an edge, or two intersecting edges in the formed ESMT topology that require multiple relays and do serve on the least number of inter-segment paths. On each of these identified edges, multiple relays are replaced by a single MRN until the number of employed relays matches the available count.

Unlike FeSMoR, the federation through touring of clustered segment (ToCS) algorithm [113] aims at minimizing both the maximum and average data delivery latency between any two segments. Minimizing the maximum latency is achieved by balancing the travel load among the populated MRNs. The approach is to form a simple star topology using the available MRNs, where each MRN serves a subset of the WSN segments. The algorithm is divided into two phases. The first phase groups segments into clusters so that the tour length of the individual MRNs is minimized. Rendezvous points for the MRNs are determined for exchanging the carried data to be relayed to the target segments. In the second phase the formed clusters are further optimized so that the travel overhead is balanced over all MRNs. The idea is to relocate the rendezvous points of the MRNs in order to equalize the tours. The optimized tour length enables the MRNs to be synchronized with each other with little waiting time, which, along with the pursued star topology, helps in reducing the overall data delivery latency. Fig. 24 illustrates the second phase of ToCS.

Both FeSMoR and ToCS assume a fixed number of MEs and decide to keep some of them stationary and use the rest as MRNs. A variant of the problem is considered in [114] where the recovery is to be performed using l_S stationary relays and l_M MRNs. In other words, the mobility of the available relays is constrained. The objective of the proposed solution is to shorten the tour of the engaged MRNs. An algorithm that uses a Mix of Mobile and Stationary nodes for Inter-connecting a set of partitions (MiMSI) is employed to achieve such an objective. MiMSI first forms a highly connected inter-segment topology in terms of the average node degree by applying FeSTA [93]. The SPs identified by FeSTA are grouped into l_M clusters based on proximity. Each of these clusters will be served by an MRN. MiMSI then determines gateway SPs between every pair of neighboring clusters. A gateway interfaces the relays in its cluster to those in a neighboring cluster. Employing FeSTA serves MiMSI well since the formed inter-segment topology will be highly-connected and many of these gateways can be found. After that, MiMSI opts to populate l_S

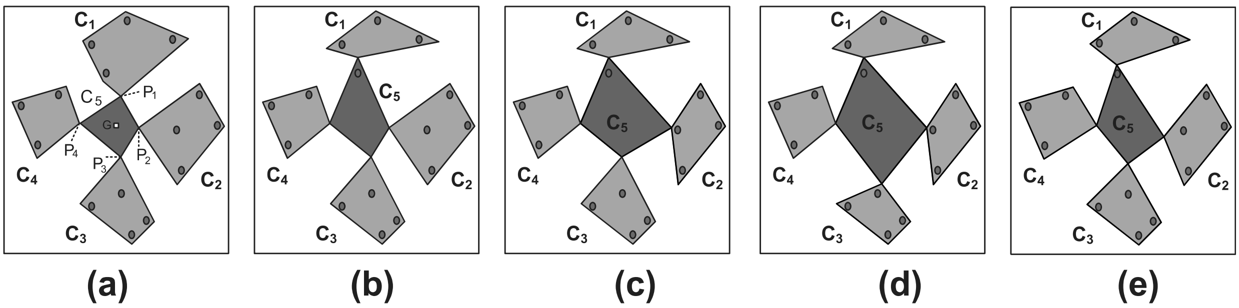

Fig. 24. Illustrating the second phase of ToCS for the segment clusters generated in the first phase. (a) Rendezvous points P_k is chosen for clusters C_k, k = 1, 2, 3, 4. C_5 is the central cluster. (b) During the first iteration, the rendezvous point P_1 is moved towards C_1. A segment from C_1 is transferred to C_5. (c) P_2 is moved towards C_2. (d) At the end of first iteration, the tour length of the MRN serving C_5 exceeds the overall average for all clusters. (e) After multiple iterations the tours of all MRNs are balanced.

stationary relays at the positions where a gateway will be located for connecting two adjacent clusters. If the number of identified gateways (g) is less than l_S, the remaining relays are used to reduce the travel paths for MRNs. On the other hand, when g exceeds l_S, MiMSI drops $(g - l_S)$ gateways and extends the boundary of corresponding clusters to overlap. Finally, MiMSI strives to find a tour path along which an MRN visits the segments and populated relays by using a convex hull of the nodes in the cluster and its center of mass. The optimization of the found path is performed differently depending on the relative positions of convex-nodes and non-convex nodes as seen in Fig. 25, which is redrawn from [114].

Table 4 provides a comparative summary of the ME-based recovery algorithms.

6. Future research issues

While a significant research has been done on topology management techniques to restore connectivity in partitioned WSNs, there are several directions that need further exploration. The following are some open research problems that warrant additional investigation, grouped based on the recovery methodology.

6.1. Recovery through node repositioning

- There is no work which can restore k-connectivity of a k-connected WSN in a distributed and efficient manner. Restoring k-connectivity through a generic algorithm that will work for any given k is certainly an interesting research direction.
- In Underwater Wireless Sensor Networks (UWSN), the nodes are prone to failures more than terrestrial WSNs due to corrosion and fouling. Therefore, UWSN may get partitioned and some of the nodes may not be able to communicate with one another and with the surface station. Exploiting controlled mobility to restore the connectivity in such 3-D networks is very challenging.
- The robustness of the failure detection and tolerance can be enhanced when cross-layer techniques are leveraged. For example, distinguishing node and link failures is often difficult and may trigger many false alarms for node movement. A combined link and network layers methodology can significantly reduce the frequency of false positives. Exploiting cross-layer techniques for fault tolerance is not a sufficiently explored area of research.
- In addition to link/node failure detection, partition detection is also part of connectivity recovery schemes that may affect the overhead. Most of the published studies focus on the recovery by assuming the impact of the failure is determined. In fact, there are not algorithms which can determine the scope of the failure in a distributed manner. In addition, existing node failure detection algorithms need to be improved or redesigned to boost their accuracy.
- The handling of security concerns when dealing with node failures is a tough and unexplored area of research. Security association, trust, and node vulnerability can constrain node placement/repositioning and thus complicate the recovery significantly.

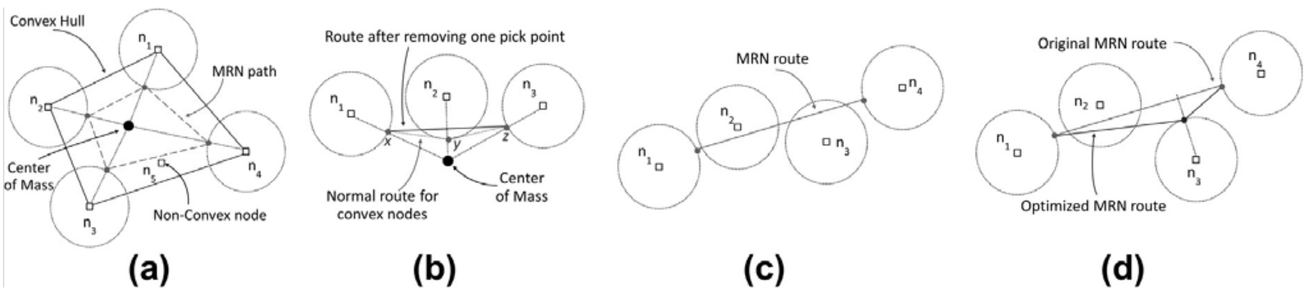

Fig. 25. Illustrating how MiMSI optimizes a tour path; (a) shows how to identify the data pickup points for more than two MRN stops, (b) and {(c),(d)} shows the optimization of the MRN route for convex nodes and non-convex nodes respectively. The figure is from [114].

Table 4
Comparison among topology repair algorithms employing mobile entities.

Paper	Employed node type	Secondary performance objectives (In addition to providing connectivity)	Constraint
[99]	MDC	Minimizing a tour path	Number of MDCs (k)
[100]	MRN	Modeling data delivery in disconnected network, Studying the effect of the end-to-end delay	Number of MRNs (K)
[29]	MRN, MDC, and MBS	Modeling data delivery in disconnected network, Studying the effect of the end-to-end delay	Number of mobile entities
[106]	MRN (Underwater Unmanned Vehicles)	Maintaining or enhancing connectivity	Underwater
[114]	A mix of stationary and mobile relays	Minimizing a tour length, Minimizing the maximum and average data delivery delay	Number of stationary and mobile relays
[112] [113]	A mix of stationary and mobile relays	Minimizing a tour length, Minimizing the maximum and average data delivery delay	Number of deployed relays

- In most of the published approaches, nodes are assumed to be able to move freely in a terrain-friendly environment and no error is assumed in determining the node locations. The effect of navigation and localization errors on the performance of recovery approaches is yet to be assessed.
- Recovery approaches that exploit node repositioning have been mostly evaluated via simulation. Testbeds need to be developed to validate the practicality and capture the performance under more realistic conditions and scenarios. Use of robots or other mobile nodes will help point out subtle issues that can be simulators cannot unveil.

6.2. Relay placement

- In general, the relay placement problem for UWSN poses different challenges due to 3-D environment. Minimizing relay count and determining their locations for establishing network connectivity will still be the main issues for tolerating node failure. However, factoring in the line-of-sight and none-line-of-sight links [115] would expand the design space and complicate the optimization. In other words, instead of designing the placement approaches to work with individual 3-D locations, a range of locations need to be considered so that the effect of movement due to currents and waves can be handled.
- While a number of published studies have employed MEs for connecting disjoint network segments, many research questions are left unanswered, in particular, how to determine the optimal ME count and how to coordinate their motion for optimal performance.
- The use of MEs to tolerate node failure has been mostly limited to mitigating the loss of connectivity. More research is needed to consider the possibility of having heterogeneous set of network parameters such as various sensing types, different inter-segment and intra-segment QoS requirements, etc. Factoring in these parameters in the recovery process make the topology restoration problem more challenging.
- Distributed unsupervised placement of ME is another interesting open area of research. For instance, in case a certain region is susceptible to threats or major damage, a set of MEs can be placed or dropped at the boundary. When a failure is detected, these MEs are to collaborate on restoring connectivity with the least movement. They may have limited or no information about the partitions into which the network is split.

7. Conclusion

In many applications, WSNs operate in inhospitable setups, e.g., battlefield, and the nodes becomes subject to increased risk of getting damaged. Furthermore, nodes are equipped with small batteries and their operation ceases upon depleting their onboard energy supply. The failure of nodes may not only impact coverage and data fidelity but also can cause the network to be divided into disjoint blocks of nodes. The latter can lead to major degradation of the WSN functionality since the failure may prevent data exchange and hinder coordination among subset of nodes. In this paper, we have analyzed the impact of node failures on the connectivity of WSNs and categorized popular network topology management techniques for fault recovery. The scope of the problem has been classified based on the multiplicity of failed nodes and the simultaneity of the failure. Published recovery solutions have been categorized into precautionary, that are performed before a failure takes place, and reactive where the network deals with a failure only when it is detected. A summary of published schemes under these categories has been provided. Within each category, further analysis has been made to highlight the different features. Based on these features, the advantages and disadvantages of the recovery approaches have been discussed. The paper also enlists a number of open issues, unexplored ideas and variants of the recovery problem that warrant further investigation.

Acknowledgments

This work was supported by the National Science Foundation (NSF) awards # CNS 1018171 and CNS 1018404.

References

[1] I.F. Akyildiz et al., Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.
[2] C-Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges, Proceedings of the IEEE 91 (8) (2003) 1247–1256.
[3] D. Estrin, L. Girod, G. Pottie, M. Srivastava, Instrumenting the world with wireless sensor networks, in: Proc. of the Int'l Conf. on Acoustics, Speech, & Signal Processing, vol. 4, 2001, pp. 2033–2036.
[4] D. Ganesan et al., Networking issues in wireless sensor networks, Journal of Parallel and Distributed Computing (JPDC), Special Issue on Frontiers in Distributed Sensor Networks (2003).
[5] H. Karl, A. Willig, Protocols and Architectures for Wireless Sensor Networks, Wiley, United Kingdom, 2005.
[6] K. Romer, F. Mattern, The design space of wireless sensor networks, IEEE Wireless Communications 11 (6) (2004).
[7] L. Bao, J.J. Garcia-Luna-Aceves, Topology management in ad hoc networks, in: Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), Annapolis, MD, June 2003.
[8] M. Younis, K. Akkaya, Strategies and techniques for node placement in wireless sensor networks: a survey, Journal of Ad-Hoc Networks 6 (4) (2008) 621–655.
[9] B. Deb, S. Bhatnagar, B. Nath, STREAM: sensor topology retrieval at multiple resolutions, Journal of Telecommunications, Special Issue on Wireless Sensor Networks 26 (2–4) (2004) 285–320.
[10] A. Cerpa, D. Estrin, ASCENT: adaptive self-configuring sensor networks topologies, in: Proc. of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), New York, NY, June 2002.
[11] P.B. Godfrey, D. Ratajczak, Naps: scalable, robust topology management in wireless ad hoc networks, in: Proc. of the IEEE 3rd International Symposium on Information Processing in Sensor Networks (IPSN 2004), Berkeley, CA, April 2004.
[12] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, Optimizing sensor networks in the energy-latency-density design space, IEEE Transactions on Mobile Computing 1 (1) (2002) 70–80.
[13] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: Proc. of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom'01), Rome, Italy, June 2001.
[14] A.A. Abbasi, M. Younis, A survey on clustering algorithms for wireless sensor networks, Computer Communications 30 (14–15) (2007) 2826–2841.
[15] Y. Lai, H. Chen, Energy-efficient fault-tolerant mechanism for clustered wireless sensor networks, in: Proc. of the 16th International Conference on Computer Communications and Networks (ICCCN'07), Honolulu, Hawaii, August 2007.

[16] G. Gupta, M. Younis, Fault-tolerant clustering of wireless sensor networks, in: Proc. of the Wireless Communications and Networking Conference (WCNC 2003), New Orleans, LA, March 2003.
[17] T. Bagheri, DFMC: decentralized fault management mechanism for cluster based wireless sensor networks, in: Proc. the 2nd International Conference on Digital Information and Communication Technology and its Applications (DICTAP), Bangkok, Thailand, May 2012.
[18] L.H.A. Correiaa, D.F. Macedoa, A.L. dos Santos, A.A.F. Loureiroa, J.M.S. Nogueiraa, Transmission power control techniques for wireless sensor networks, Computer Networks 51 (17) (2007) 4765–4779.
[19] S. Lin, J. Zhang, G. Zhou, L. Gu, J.A. Stankovic, T. He, ATPC: adaptive transmission power control for wireless sensor networks, in: Proc. of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), Boulder, Colorado, October 2006.
[20] J. Jeong, D. Culler, J.H. Oh, Empirical analysis of transmission power control algorithms for wireless sensor networks, in: Proc. of the 4th International Conference on Networked Sensing Systems (INSS '07), Braunschweig, Germany, June 2007.
[21] J. Luo, J.-P. Hubaux, Joint mobility and routing for lifetime elongation in wireless sensor networks, in: Proc. of IEEE INFOCOM 2005, Miami, FL, March 2005.
[22] I. Chatzigiannakis, A. Kinalis, S. Nikoletseas, Sink mobility protocols for data collection in wireless sensor networks, in: Proc. of the 4th ACM International Workshop on Mobility Management and Wireless Access (MOBIWAC), Terromolinos, Spain, October 2006.
[23] Z.M. Wang, S. Basagni, E. Melachrinoudis, C. Petrioli, Exploiting sink mobility for maximizing sensor networks lifetime, in: Proc. of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), Hawaii, January 2005.
[24] W. Alsalih, S. Akl, H. Hassanein, Placement of multiple mobile base stations in wireless sensor networks, in: Proc. of the IEEE Symposium on Signal Processing and Information Technology (ISSPIT), Cairo, Egypt, December 2007.
[25] W. Youssef, M. Younis, K. Akkaya, An intelligent safety-aware gateway relocation scheme for wireless sensor networks, in: Proc. of the IEEE Int'l Conf. on Comm. (ICC 2006), Istanbul, Turkey, June 2006.
[26] K. Akkaya, M. Younis, Sink repositioning for enhanced performance in wireless sensor networks, Computer Networks 49 (4) (2005) 512–534.
[27] W. Wang, V. Srinivasan, K.-C. Chua, Using mobile relays to prolong the lifetime of wireless sensor networks, in: Proc of the 11th International Conference on Mobile Computing and Networking, Cologne, Germany, 2005.
[28] H. Jun et al., Trading latency for energy in densely deployed wireless ad hoc networks using message ferrying, Journal of Ad Hoc Networks 5 (4) (May 2007) 444–461.
[29] H. Almasaeid, A.E. Kamal, Modeling mobility-assisted data collection in wireless sensor networks, in: Proc. of the IEEE Global Telecommunications Conf. (GLOBECOM'08), New Orleans, LA, December 2008.
[30] W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: Proc. of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'04), Tokyo, Japan, May 2004.
[31] H. Almasaeid, A.E. Kamal, Data delivery in fragmented wireless sensor netowrks using mobile agents, in: Proc. the 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Chania, Greece, October 2007.
[32] B.W. Johnson, Design and Analysis of Fault-Tolerant Digital Systems, Addison-Wesley, Reading, Massachusetts, 1989.
[33] X. Han, X. Cao, E.L. Lloyd, C.-C. Shen, Fault-tolerant relay nodes placement in heterogeneous wireless sensor networks, in: Proc. of the 26th IEEE/ACM Joint Conf. on Computers and Comm. (INFOCOM'07), Anchorage AK, May 2007.
[34] N. Li, J. C. Hou, Flss: a fault-tolerant topology control algorithm for wireless networks, in: Proc. 10th ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, PA, September 2004.
[35] F. Al-Turjman, H. Hassanein, M. Ibnkahla, Optimized relay placement to federate wireless sensor networks in environmental applications, in: Proc. of the IEEE International Workshop on Federated Sensor Systems (FedSenS11), Istanbul, Turkey, July 2011.
[36] A. Ghosh, S. Boyd, Growing well-connected graphs, in: Proc. of IEEE Conf. on Decision and Control (CDC), San Diego, CA, December 2006.
[37] F. Al-Turjman, H. Hassanein, W. Alsalih, M. Ibnkahla, Optimized relay placement for wireless sensor networks federation in environmental applications, Wireless Communication and Mobile Computing 11 (12) (2011) 1677–1688.
[38] F. Al-Turjman, H. Hassanein, Towards augmented connectivity with delay constraints in WSN federation, International Journal of Ad Hoc and Ubiquitous Computing 11 (2/3) (2012) 97–108.
[39] S. Vaidya, M. Younis, Efficient failure recovery in wireless sensor networks through active spare designation, in: Proc. of the 1st Int'l Workshop on Interconnections of Wireless Sensor Networks (IWSN'10), Santa Barbara, California, June 2010.
[40] M. Imran, M. Younis, A.M. Said, H. Hasbullah, Partitioning detection and connectivity restoration algorithm for wireless sensor actor networks, in: Proc. of the IEEE/IFIP Int'l Conf. on Embedded and Ubiquitous Computing (EUC 2010), Hong Kong, China, December 2010.
[41] G. Lin, G. Xue, Steiner tree problem with minimum number of steiner points and bounded edge-length, Information Processing Letters 69 (1999) 53–57.
[42] X. Cheng, D.-Z. Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks, Wireless Networks 14 (3) (2008) 347–355.
[43] E.L. Lloyd, G. Xue, Relay node placement in wireless sensor networks, IEEE Transactions on Computers 56 (1) (Jan 2007) 134–138.
[44] A. Efrat et al., Improved approximation algorithms for relay placement, in: Proc. of the 16th European Symposium on Algorithms, Karlsruhe, Germany, September 2008.
[45] S. Poduri, S. Pattem, B. Krishnamachari, G.S. Sukhatme, Sensor network configuration and the curse of dimensionality, in: Proc. of the 3rd IEEE Workshop on Embedded Networked Sensors, Cambridge, MA, May 2006.
[46] J. Tang, B. Hao, A. Sen, Relay node placement in large scale wireless sensor networks, Computer Communications, Special Issue on Wireless Sensor Networks 29 (2006) 490–501.
[47] B. Hao, H. Tang, G. Xue, Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation, in: Proc. of the Workshop on High Performance Switching and Routing, Phoenix, AZ, April 2004.
[48] F. Dai, J. Wu, On constructing k-connected k-dominating sets in wireless ad-hoc and sensor networks, Journal of Parallel and Distributed Computing 66 (7) (2006) 947–958.
[49] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, G. Xue, Approximations for Steiner trees with minimum number of Steiner points, Theoretical Computer Science 262 (1) (2001) 83–99.
[50] J. Li, S.M. Shatz, A.D. Kshemkalyani, Mobile sampling of sensor field data using controlled broadcast, IEEE Transactions on Mobile Computing 10 (6) (2011) 881–896.
[51] J.L. Bredin, E.D. Demaine, M. Hajiaghayi, D. Rus, Deploying sensor networks with guaranteed capacity and fault tolerance, in: Proc. the 5th ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2005), Urbana-Champaign, IL, May 2005.
[52] G. Kortsarz1, Z. Nutov, Approximating node connectivity problems via set covers, Algorithmica 37 (2003) 75–92.
[53] J. Cheriyan, S. Vempala, A. Vetta, Approximation algorithms for minimum-cost k-vertex connected subgraphs, in: Proc. of the 34th Annual ACM Symposium on the Theory of Computing, Montreal, Canada, May 2002.
[54] K. Akkaya, A. Thimmapuram, F. Senel, S. Uludag, Distributed recovery of actor failures in wireless sensor and actor networks, in: Proc. of the IEEE Wireless Comm. and Networking Conf. (WCNC'08), Las Vegas, NV, March 2008.
[55] F. Dai, J. Wu, An extended localized algorithms for connected dominating set formation in ad hoc wireless networks, IEEE Transactions on Parallel and Distributed Systems 15 (10) (2004).
[56] A. Kashyap, S. Khuller, M. Shayman, Relay placement for higher order connectivity in wireless sensor networks, in: Proc. of IEEE INFOCOM, Barcelona Spain, April 2006.
[57] S. Khuller, B. Raghavachari, Improved approximation algorithms for uniform connectivity problems, Journal of Algorithms 21 (2) (1996) 434–450.
[58] W. Zhang, G. Xue, S. Misra, Fault-tolerant relay node placement in wireless sensor networks: problems and algorithms, Proc. of IEEE INFOCOM, Anchorage, Alaska, 2007.
[59] X. Bai, S. Kumer, D. Xuan, Z. Yun, T.H. Lai, Deploying wireless sensors to achieve both coverage and connectivity, in: Proc. of ACM

Annual Int'l Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc), Florence, Italy, 2006.
[60] Z. Yun, X. Bai, D. Xuan, T.H. Lai, W. Jia, Optimal deployment patterns for full coverage and connectivity wireless sensor networks, IEEE/ACM Transactions on Networking 18 (3) (2010) 934–947.
[61] X. Bai, Z. Yun, D. Xuan, T.H. Lai, W. Jia, Deploying four-connectivity and full-coverage wireless sensor networks, in: Proc. of INFOCOM, Phoeniz, AZ, April 2008.
[62] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proc. of ACM Sensys'03, Los Angeles, CA, November 2003.
[63] F. Al-Turjman, H. Hassanein, M. Ibnkahla, Optimized relay repositioning for wireless sensor networks applied in environmental applications, in: Proc. of the IEEE International Wireless Communications and Mobile Computing conf. (IWCMC'11), Istanbul, Turkey, July 2011.
[64] F. Al-Turjman, H. Hassanein, M. Ibnkahla, Efficient deployment of wireless sensor networks targeting environment monitoring applications, Computer Communications 36 (2) (2013) 135–148.
[65] A. Abbasi, K. Akkaya, M. Younis, A distributed connectivity restoration algorithm in wireless sensor and actor networks, in: Proc. of the 32nd IEEE Conf. on Local Computer Networks (LCN 2007), Dublin, Ireland, October 2007.
[66] S. Das et al., Localized movement control for fault tolerance of mobile robot networks, in: Proc. of the 1st IFIP Int'l Conference on Wireless Sensor and Actor Networks (WSAN 2007), Albacete, Spain, September 2007.
[67] P. Basu, J. Redi, Movement control algorithms for realization of fault-tolerant ad hoc robot networks, IEEE Networks 18 (4) (2004) 36–44.
[68] M. Younis, S. Lee, A.A. Abbasi, A localized algorithm for restoring inter-node connectivity in networks of moveable sensors, IEEE Transaction on Computer 59 (12) (2010) 1669–1682.
[69] A. Abbasi, M. Younis, K. Akkaya, Movement-assisted connectivity restoration in wireless sensor and actor networks, IEEE Transactions on Parallel and Distributed Systems 20 (9) (2009) 1366–1379.
[70] F. Senel, K. Akkaya, M. Younis, An efficient mechanism for establishing connectivity in wireless sensor and actor networks, in: Proc. of the 50th IEEE Global Telecommunications Conf. (Globecom'07), Washington, DC, November 2007.
[71] M. Imran, M. Younis, A.M. Said, H. Hasbullah, Volunteer-instigated connectivity restoration algorithm for wireless sensor and actor networks, in: Proc. of the IEEE Int'l Conf. on Wireless Comm., Networking and Information Security (WCNIS2010), Beijing, China, June 2010.
[72] A. Alfadhly, U. Baroudi, M. Younis, Least distance movement recovery approach for large scale wireless sensor-actor networks, in: Proc. of the Workshop on Federated Wireless Sensor Networks (FedSenS 2011), Istanbul, Turkey, July 2011.
[73] A. Abbasi, M. Younis, U. Baroudi, Restoring connectivity in wireless sensor-actor networks with minimal topology changes, in: Proc. of the IEEE Int'l Conf. on Comm. (ICC 2010), Cape Town, South Africa, May 2010.
[74] A. Abbasi, M. Younis, U. Baroudi, A least-movement topology repair algorithm for partitioned wireless sensor-actor networks, Int'l Journal of Sensor Networks 11 (4) (2012) 250–262.
[75] N. Tamboli, M. Younis, Coverage-aware connectivity restoration in mobile sensor networks, Elsevier Journal of Network and Computer Applications 33 (2010) 363–374.
[76] A. Abbasi, U. Baroudi, M. Younis, K. Akkaya, C2AM: an algorithm for application aware movement-assisted recovery in wireless sensor and actor networks, in: Proc. of the 5th Int'l Wireless Comm. and Mobile Computing Conf. (IWCMC 2009), Leipzig, Germany, June 2009.
[77] M. Imran, M. Younis, A.M. Said, H. Hasbullah, Application-centric connectivity restoration algorithm for wireless sensor actor networks, in: Proc. of the 6th Int'l Conf. on Grid and Ubiquitous Pervasive Computing (GPC 2011), Oulu, Finland, May 2011.
[78] M. Younis, R. Waknis, Connectivity restoration in wireless sensor networks using steiner tree approximations, in: Proc. of the IEEE Global Telecommunications Conference (GLOBECOM'10), Miami, FL, December 2010.
[79] A. Alfadhly, U. Baroudi, M. Younis, Optimal node repositioning for tolerating node failure in wireless sensor actor network, in: Proc. the 25th Biennial Symposium on Comm., Kingston, Canada, May 2010.
[80] M. Sir, I. Senturk, E. Sisikoglu, K. Akkaya, An optimization-based approach for connecting partitioned mobile sensor/actuator networks, in: Proc. of International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN), Shanghai, China, April 2011.
[81] I.F. Senturk, K. Akkaya, F. Senel, An effective and scalable connectivity restoration heuristic for mobile sensor/actor networks, in: Proc. of the IEEE Global Communications Conference (GLOBECOM'12), Anaheim, CA, December 2012.
[82] M. Imran, M. Younis, A.M. Said, H. Hasbullah, Localized motion-based connectivity restoration algorithms for wireless sensor actor networks, Journal of Network and Computer Applications 35 (2012) 844–856.
[83] K. Akkaya, F. Senel, A. Thimmapuram, S. Uludag, Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility, IEEE Transactions on Computers 59 (2) (2010) 258–271.
[84] Y. Joshi, Autonomous Recovery from Multi-Node Failures in Wireless Sensor Networks, MS Thesis, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, December 2011.
[85] S. Lee, M. Younis, Recovery from multiple simultaneous failures in wireless sensor networks using minimum steiner tree, Journal of Parallel and Distributed Systems 70 (2010) 525–536.
[86] G. Robins, A. Zelikovsky, Tighter bounds for graph steiner tree approximation, SIAM Journal on Discrete Mathematics 19 (1) (2005) 122–134.
[87] S. Vemulapalli, K. Akkaya, Mobility-based self route recovery from multiple node failures in mobile sensor networks, in: Proc. of IEEE International Workshop on Wireless Local Networks (WLN), Denver CO, October 2010.
[88] K. Akkaya, S. Vemulapalli, I.F. Senturk, Handling large-scale node failures in mobile sensor/robot networks, Journal of Network and Computer Applications 36 (1) (2013) 195–210.
[89] I.F. Senturk, S. Yilmaz, K. Akkaya, Connectivity restoration in delay tolerant sensor networks using game theory, International Journal of Ad Hoc and Ubiquitous Computing 11 (2/3) (2012) 109–210.
[90] I. Senturk, K. Akkaya, S. Yilmaz, A game-theoretic approach to connectivity restoration in wireless sensor and actor networks, in: Proc. of Wireless Sensor Actor and Actuator Networks Workshop, Ottawa, Canada, June 2012.
[91] I.F. Senturk, K. Akkaya, Energy and terrain aware connectivity restoration in disjoint mobile sensor networks, in: Proc. of the 12th IEEE International Workshop on Wireless Local Networks (WLN'12), Clearwater, FL, October 2012.
[92] S. Lee, M. Younis, Optimized relay placement to federate segments in wireless sensor networks, IEEE Journal on Selected Area in Communications, Special Issue on Mission Critical Networking 28 (5) (2010) 742–752.
[93] F. Senel, M. Younis, Relay node placement in structurally damaged wireless sensor networks via triangular steiner tree approximation, Computer Communications 34 (16) (2011) 1932–1941.
[94] F. Senel, M. Younis, Optimized relay node placement for establishing connectivity in sensor networks, in: Proc. of the IEEE Global Telecommunications Conference (Globecom'12), Anaheim, CA, December 2012.
[95] F. Senel, M. Younis, Optimized connectivity restoration in a partitioned wireless sensor networks, in: Proc. of the IEEE Global Telecommunications Conference (GLOBECOM'11), Houston, Texas, December 2011.
[96] S. Lee, M. Younis, EQAR: effective QoS-aware relay node placement algorithm for connecting disjoint wireless sensor sub-networks, IEEE Transactions on Computers 60 (12) (2011) 1772–1778.
[97] S. Lee, M. Younis, QoS-aware relay node placement for connecting disjoint segments in wireless sensor networks, in: Proc. of Int'l Workshop on Interconnections of Wireless Sensor Networks (IWSN'10), Santa Barbara, California, June 2010.
[98] F. Senel, M. Younis, K. Akkaya, Bio-inspired relay node placement heuristics for repairing damaged wireless sensor networks, IEEE Transactions on Vehicular Technology 60 (4) (2011) 1835–1848.
[99] F. Senel, M. Younis, Optimized interconnection of disjoint wireless sensor network segments using K mobile data collectors, in: Proc. of the IEEE Int'l Conf. on Communications (ICC'12), Ottawa, Canada, June 2012.
[100] H. Almasaeid, A.E. Kamal, Data delivery in fragmented wireless sensor netowrks using mobile agents, in: Proc. the 10th ACM/IEEE Int'l Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Chania, Greece, October 2007.

[101] D.L. Applegate, R.E. Bixby, V. Chvátal, W. J Cook, The Traveling Salesman Problem: A Computational Study, Princeton University Press, New Jersey, USA, 2006.
[102] A. Abbasi, M. Younis, A survey on clustering algorithms for wireless sensor networks, Computer Communications, Special Issue on Network Coverage and Routing Schemes for Wireless Sensor Networks 30 (2007) 2826–2841.
[103] I. Senturk, K. Akkaya, S. Yilmaz, Distributed relay node positioning for connectivity restoration in partitioned wireless sensor networks, in: The Proceedings of IEEE International Symposium and Computers and Communications (ISCC'12), Cappadocia, Turkey, July 2012.
[104] C.-C. Shen, O.Koc, C. Jaikaeo, Z. Huang, Trajectory control of mobile access points in MANET, in: Proc. of the 48th IEEE Global Telecommunications Conf. (GLOBECOM '05), St. Louis, MO, November 2005.
[105] M. Ahmed et al., Positioning range extension gateways in mobile ad hoc wireless networks to improve connectivity and throughput, in: Proc. of IEEE Military Communications Conference (MILCOM'01), Washington, DC, October 2001.
[106] W.K.G. Seah, H. Tan, Z. Liu, M.H. Ang, Multiple-UUV approach for enhancing connectivity in underwater ad-hoc sensor networks, Proceedings of MTS/IEEE OCEANS 3 (2005) 2263–2268.
[107] R. Shah, S. Roy, S. Jain, W. Brunette, Data MULEs: modeling a three-tier architecture for sparse sensor networks, in: Proc. of the1st IEEE Int' Workshop on Sensor Network Protocols and Applications (SNPA'03), May 2003.
[108] S. Jain, R.C. Shah, W. Brunette, G. Borriello, S. Roy, Exploiting mobility for energy efficient data collection in wireless sensor networks, Journal of Mobile Networks and Applications 11 (2006) 327–339.
[109] W. Alsalih, S.G. Akl, H.S. Hassanein, Placement of multiple mobile base stations in wireless sensor networks, in: Proc. of the IEEE Symp. On Signal Processing and Info. Tech. (ISSPIT), Cairo, Egypt, December 2007.
[110] W. Alsalih, H.S. Hassanein, S.G. Akl, Placement of multiple mobile data collectors in wireless sensor networks, Journal of Ad Hoc Networks 8 (4) (2010) 378–390.
[111] W. Alsalih, S.G. Akl, H.S. Hassanein, Placement of multiple mobile data collectors in underwater acoustic sensor networks, in: Proc. of the IEEE Int'l Conf. on Comm. (ICC 2006), Beijing, China, May 2008.
[112] J.L.V.M. Stanislaus, M. Younis, Delay-conscious federation of Multiple wireless sensor network segments using mobile relays, in: Proc. of the 76th IEEE Vehicular Technology Conference (VTC2012-Fall), Québec City, Canada, September.
[113] J.L.V.M. Stanislaus, Mobile Relays Based Federation of Multiple Wireless Sensor Network Segments with Reduced-latency, MS Thesis, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Summer 2012.
[114] A. Abbas, M. Younis, Interconnecting disjoint network segments using a mix of stationary and mobile nodes, in: Proc. of the IEEE Conf. on Local Comp. Net. (LCN 2012), Clearwater, FL, October, 2012.
[115] L. Emokpae, M. Younis, Surface based underwater communications, in: Proc. of the IEEE Global Telecommunications Conference (GLOBECOM 2010), Miami, FL, December 2010.

Mohamed Younis is currently an associate professor in the department of computer science and electrical engineering at the university of Maryland Baltimore County (UMBC). He received his Ph.D. degree in computer science from New Jersey Institute of Technology, USA. Before joining UMBC, he was with the Advanced Systems Technology Group, an Aerospace Electronic Systems R&D organization of Honeywell International Inc. While at Honeywell he led multiple projects for building integrated fault tolerant avionics and dependable computing infrastructure. He also participated in the development of the Redundancy Management System, which is a key component of the Vehicle and Mission Computer for NASA's X-33 space launch vehicle. His technical interest includes network architectures and protocols, wireless sensor networks, embedded systems, fault tolerant computing, secure communication and distributed real-time systems. He has published over 180 technical papers in refereed conferences and journals. He has five granted and two pending patents. In addition, he serves/served on the editorial board of multiple journals and the organizing and technical program committees of numerous conferences. He is a senior member of the IEEE.

Izzet F. Senturk received BS and MEng in Computer Science from Ege University, Turkey in 2006 and Cornell University in 2008 respectively. Currently he is a PhD candidate in the department of Computer Science at Southern Illinois University. His areas of interest are mobility and fault-tolerance in wireless sensor networks and energy-aware protocol design for mobile sensor networks. He is a member of IEEE.

Kemal Akkaya is an associate professor in the Department of Computer Science at Southern Illinois University, Carbondale, IL. He received his BS and MS in Computer Engineering from Bilkent University, Turkey and Middle-East Technical University, Turkey in 1997 and 1999 respectively. After working as a software developer in Ankara, Turkey, he moved to US in 2000 for pursuing a PhD degree in Computer Science. He received his PhD in Computer Science from University of Maryland Baltimore County in 2005 and joined the department of Computer Science at Southern Illinois University in 2005. He was a visiting professor at The George Washington University in Fall 2013. His current research interests include energy aware routing, topology control, security and quality of service issues in a variety of wireless networks such as sensor networks, sensor-actor networks, multimedia sensor networks, smart-grid communication networks and vehicular networks. He is a member of IEEE. He is the area editor of Elsevier Ad Hoc Network Journal. He has served as the guest editor for Journal of High Speed Networks, Computer Communications Journal, Elsevier Ad Hoc Networks Journal and in the TPC of many leading wireless networking conferences including IEEE ICC, Globecom, LCN and WCNC. He has published over 80 papers in peer reviewed journal and conferences. He has received "Top Cited" article award from Elsevier in 2010.

Sookyoung Lee received the M.S. and Ph.D. degrees in Computer Science from the Ewha Womans University, Korea, and University of Maryland, Baltimore County, USA in 1997 and 2010 respectively. She has been with LG ELECTRONICS Inc., Electronics and Telecommunications Research Institute, Korea Electrics Technology Institute, and Samsung Electronics Co. LTD., Korea from 1998 to 2004. While at LG, she has developed the IP data server over ATM switch and implemented virtual private network service for multi-protocol label switching system. She was a volunteer of IPv6 forum Korea while at ETRI and has developed the network address and protocol translation system between IPv4 and IPv6. At Samsung, she was a broadband convergence newtork designer especially focusing on requirements for QoS and IPv6. She is currently a research professor in the department of computer science and engineering at the Ewha Womans University, Korea. Her primary research interest includes network architectures and protocols, topology restoration and fault tolerance in wireless sensor networks and network modeling and performance analysis for dynamic and sparse ad-hoc networks.

Fatih Senel received his BS and MS degrees in Computer Science from Bilkent University, Ankara, Turkey in 2005 and Southern Illinois University Carbondale, IL in 2008, respectively. In 2012, he received his PhD in Computer Science from University of Maryland Baltimore County. Currently, he is an assistant professor in the Department of Computer Engineering at Antalya International University, Turkey. His research interests include clustering, relocation and fault-tolerance in wireless sensor networks, and self-deployment of nodes in underwater acoustic sensor networks.