Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance

Sookyoung Lee a, Mohamed Younis b, Meejeong Lee a,

a Department of Computer Science and Engineering, Ewha Womans University, Seoul, Republic of Korea
b Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, USA

Article info
Article history:
Received 28 November 2013
Received in revised form 9 July 2014
Accepted 28 July 2014
Available online xxxx

Keywords:
Wireless sensor networks
Network partitioning
Topology repair
2-Vertex connectivity
Fault tolerance
Relay node placement

Abstract
Wireless sensor network (WSN) applications, especially those serving in inhospitable environments such as battlefield reconnaissance, are susceptible to large-scale damage which usually causes simultaneous failure of multiple collocated sensors and gets the network divided into disjoint partitions. In order to prevent the WSN from being inoperative, restoring the overall network connectivity is crucial. Furthermore, it may be desirable to make the repaired network topology resilient to future single node failures caused by aftermath. In this paper, we present an effective strategy for achieving such recovery goal by establishing a bi-connected inter-partition topology while minimizing the maximum path length between pairs for partitions and deploying the least count of relay nodes (RNs). Finding the optimal number and position of RNs is NP-hard and we thus pursue heuristics. The proposed Connectivity Restoration with Assured Fault Tolerance (CRAFT) algorithm strives to form the largest inner simple cycle or backbone polygon (BP) around the center of the damaged area where no partition lies inside. RNs are then deployed to connect each outer partition to the BP through two non-overlapping paths. We analyze the properties of CRAFT mathematically and validate its performance through extensive simulation experiments. The validation results show that CRAFT yields highly connected topologies with short inter-partition paths while employing fewer RNs than competing schemes.

1. Introduction
In recent years, numerous applications have fueled interest in WSNs, especially those serving in harsh environments [1]. In a hostile application setup, such as coast and border surveillance, search-and-rescue and battlefield reconnaissance, the unattended operation of miniaturized sensors would decrease the cost of the application and avoid the risk to human life. 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 [2].

Upon deployment, the sensor nodes are expected to stay reachable to each other in order to coordinate their actions while performing a task, and to forward their readings to in situ users. Therefore, the inter-sensor connectivity has a significant impact on the effectiveness of WSNs and should be sustained all the time.

Meanwhile, a sensor is susceptible to failure due to the small form factor and limited onboard energy supply. Moreover, a WSN deployed in a harsh environment is susceptible to partitioning due to: (i) a failure of a single node caused by battery depletion, malfunction of external hazard, if the faulty node is a cut vertex, or (ii) simultaneous failure of multiple collocated nodes. The former can be mitigated by provisioning bi-connectivity so that the network does not get partitioned. The latter is handled by relay node placement to re-establish a connected
inter-partition topology after the failure takes place, as in the case of our proposed CRAFT approach. Mitigating the simultaneous failure of multiple colocated nodes through resource provisioning would require massive resources that will most probably involve nodes outside the area of deployment due to susceptibility to damage. Thus, it is not cost effective to guard the network against such type of failure. Therefore, CRAFT reacts to simultaneous node failure and provisions resources to tolerate only a single node failure in the recovered network topology since it could be more possible due to aftermath or unexploded bombs, etc., in addition to contemporary causes of failure such as exhaustion of the onboard energy supply and electronic breakdown. In addition, the probability of a single node failure is higher than that of multiple colocated nodes failure, bi-connecting the partitions helps to reduce the probability of network re-partitioning after recovery.

Fig. 1(a) shows an articulation, where the area covered by dark circles represent the extent of the damage, after which the surviving sensor nodes are grouped into seven disjoint partitioned WSNs due to the loss of connectivity. Restoring inter-partition connectivity would be crucial so that the WSN becomes functional again. A similar scenario arises when multiple autonomous WSN segments are required to collaborate to achieve a common task as seen in Fig. 1(b). Forming a bi-connected inter-segment topology, i.e., establishing two vertex-disjoint paths among every pair of segments would boost the application robustness and balance the inter-segment traffic in a network.

We shall use the term federation to refer establishing connectivity in both scenarios, i.e., repairing a damaged WSN or linking multiple independent WSN segments. Federating multiple WSN partitions while forming a bi-connected topology is an under-researched problem. When the network nodes are not mobile, the topology cannot be autonomously reconfigured and relays nodes (RNs) need to be deployed to achieve the federation goal. Naturally the RN count opts to be minimized in order to cope with resource scarcity or just to cut the federation overhead. Therefore, the federation problem is fundamentally how to establish a 2-vertex connected inter-partition topology where no cut-vertex exists by deploying the least RN count. Such RN placement optimization problem can be mapped to forming Steiner Tree with minimum Steiner points and Bounded Edge-Length, which is proven to be an NP-hard problem by Lin and Xue [3]. In order to address such complexity this paper presents CRAFT, a polynomial-time Connected Restoration algorithm in a partitioned WSN with Assured Fault Tolerance. In addition to the 2-vertex connectivity, CRAFT also opts to: (i) reduce the maximum path length between pairs of segments in order to help in bounding the inter-partition data latency, and (ii) boost the average node degree in order to further provide route alternatives and enable load balancing.

CRAFT strives to identify an inner ring formed by exploiting the Steiner Point (SPs) which connect outer partitions with the fewest relays based on which non-ring partitions are bi-connected by exploiting the found SPs to the ring. The rationale is that a ring is a bi-connected topology that involves the fewest edges among the partitions. Given that CRAFT opts to employ the fewest relays, finding the least-cost cycle, ring, in a graph is NP hard and thus heuristics are pursued. CRAFT operates in two phases. In the first phase, CRAFT forms the largest inner simple cycle, referred to as backbone polygon (BP), which does not contain any partition. CRAFT opts to find the BP by creating an inward growing topology from the outmost partitions. This is done in rounds. The first round forms the convex hull of all partitions and identifies an SP of every three adjacent partitions on the convex hull. The set of partitions and SPs will be collectively referred to as terminals. In each of the subsequent rounds CRAFT identifies border terminals (BTs) in the set S of unprocessed SPs and Pi’s, i.e., were not considered in any of the previous rounds, by computing the convex hull of all elements in S. Then, the newly found SPs, each of which connects a group of three adjacent BTs

![Fig. 1. Illustration of two scenarios which require inter-partition connectivity in WSNs. (a) the dark circles represent dead sensors in the damaged area; the surviving (light) sensors are split into seven disjoint partitions. (b) Four standalone WSNs are to be federated in order to achieve a common task, e.g., set pressure of water sprinklers.](image-url)
with the least cost are added to the set $S$ and the SPs, and partitions on the convex hull are removed from $S$ in preparation for the next round. The rationale for forming a convex hull during the BP identification is that all identified SPs reside inside of the found convex hull in each round and thus the entire process converges inwards towards the center of the deployment area. In other words, the size of convex hull shrinks in rounds and finally the one where all remaining unprocessed $P_i$’s lie becomes the BP.

The second phase of CRAFT forms a bi-connected inter-partition topology using the identified SPs and the BP. First an initial 1-connected topology is established where the terminals of the BP acts as a core mesh and each partition $P_i$ that is not part of the BP is connected to the BP over the shortest SP-based path. Let $Prim_i$ denoting the primary bi-connection point, be the terminal of the BP to which $P_i$ connects. Based on the initial topology CRAFT establishes an alternative path to the BP for each $P_i$ by picking the secondary bi-connection point $Second_i$ ($Second_i 
eq Prim_i$) and connecting $P_i$ to $Second_i$ in order to form a simple cycle $SC_i$ that is connected or merged with the BP. Finally all links in the topology are steinerized based on the communication range of the RN. The use of BP helps in bounding the maximum path length between pairs of segments since many partitions will be connected to the BP and not on the BP and thus an inter-partition path will not pass through many other partitions.

The rest of the paper is organized as follows. Related work is covered in Section 2. The problem is formally defined and the considered system model is described in Section 3. The details of CRAFT are provided in Section 4. The validation results are presented in Section 5. The paper is finally concluded in Section 6.

2. Related work

Technologies for tolerating node failures that may cause network partitioning can be classified into two methodologies, namely, pre-cautionary and reactive [4,5]. The precautionary methodology proactively provides spare connections in the network at setup or during normal operation so that the network does not suffer partitioning when a failure takes place. Such a proactive fault-migration strategy establishes $k$-node distinct communication paths between every pair of nodes, where $k > 2$, so that the WSN survives $k$ – 1 node failures. Employing the least number of redundant nodes to achieve such fault resilience is usually a design objective for approaches that follow the precautionary methodology [6,7]. On the other hand, the reactive methodology strives to repair the network topology and restore the WSN connectivity to its pre-failure level. Algorithms in this category, either utilize mobility of existing nodes to restore connectivity [8,9], or employ additional nodes during recovery [10,11]. CRAFT is a connectivity restoration algorithm which strives to deploy the least count of additional relays to repair a partitioned network and to establish a 2-connected topology. CRAFT can also be applied proactively if a set of standalone WSN segments are to be federated by forming a robust inter-segment topology that can tolerate an occasional single relay node failure. Given the scope of the contribution, in the balance of this section we focus on published work in the proactive category for which a $k$-connected topology is established. For comprehensive coverage of reactive algorithms and techniques for forming inter-segment topology, the reader is referred to [4,5].

Hao et al. [6] strive to place the minimum number of RNs such that each sensor is connected to at least two RNs and the inter-RN network is 2-connected. The authors formulate the placement problem as “2-Connected Relay Node Double Cover”, to determine the locations of the fewest RNs, so that each sensor is covered by the communication range of at least two RNs and the group of RNs becomes bi-connected. All RNs are assumed to have the same communication range $R$ that is at least twice the range of a sensor node $r$. The proposed algorithm computes a possible position $p$ of a relay and a set of sensor nodes $C(p)$ which is covered by a relay placed at $p$. These positions can be found at the intersections of the communication ranges of neighboring sensors. Then, the algorithm identifies the positions at which RNs are placed to cover the maximum number of sensors. The algorithm checks whether the RNs form a 2-connected graph and every sensor can reach at least 2 RNs. If not, more relay positions are populated. The latter step is repeated till the objective is achieved. Tang et al. [7] pursue a similar idea while modeling the area as a grid. The connectivity is first handled within the individual cells in the grid and then at the inter-cell level. However, $R$ in this case is assumed to be at least four times as much as $r$.

Kashyap et al. present a polynomial time heuristic to achieve 2-connectivity among $n$ nodes with the least relay count assuming $R = r$ [12]. The authors focus on establishing both $k$-edge and $k$-vertex connectivity separately by using the algorithm in [13]. In order to form the $k$-vertex connectivity, they first compute $k$-vertex connected spanning sub-graph $g$ and then steinerize each edge of $g$. This is the best known heuristic in the literature for minimizing the relay count for 2-connectivity, i.e., $k = 2$ and effectively forms a large ring that includes all partitions. However, the formed topology has a node degree of 2 and long inter-segment path length. CRAFT strives to overcome these shortcomings. Meanwhile, Zhang et al. [14] study two variants of the problem, namely, single-tiered and two-tiered relay placement for $R > r$. In the single-tier version, a path between two sensors, terminals in the context of CRAFT, may have both RNs and sensors (terminals). In the two-tier version, only RNs can serve on the paths between terminals. The proposed algorithm for the single-tier version is in essence a generalization of the approach of [12] for $R > r$. The two-tier solution steinerizes the edges of minimum 2-connected spanning sub-graph using [13] in a similar way to that of [12]. In addition, the 2C-SpiderWeb approach [15] establishes an inter-segment topology that resembles a spider web, for which the partitions are situated at the perimeter and a mesh is formed at the center of the deployment area. An edge between each terminal and the centroid is formed and RNs are placed at a distance $R$ from the centroid. The edge between each pair of RNs on the mesh is then steinerized. To achieve 2-vertex connectivity each terminal at the perimeter is connected to two
distinct RNs on the inner mesh through 2 RN-disjoint paths.

A number of the other published solutions have focused on the $k$-connectivity problem. For example, a graph-theory based solution is proposed [16]. First, a complete graph, say $G$, for the set of vertices (nodes) is formed and each added edge is assigned a weight based on the number of required RNs 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 additional nodes. A similar idea is pursued in [17] for a heterogeneous WSN in terms of the transmission range of the network nodes. Meanwhile, in [18–20] the authors assume that all RNs have the same communication range and solve the “minimum $k$-vertex connected spanning graph” (MKCSG) problem. Then the algorithm places the least number of RNs to establish $k$-vertex connectivity ($k \geq 1$) by providing a one-way or a two-way steinerized path along each edge of the found MKCSG. Moreover, in [21] $k$-connectivity is provided as a byproduct of determining a connected dominating set to obtain a robust backbone.

Furthermore, $k$-connectivity has been 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 [22] that for a grid structure the network will be connected as long as $r > \sqrt{3}s$. The results have been extended to higher levels of connectivity by proposing to use some deployment patterns. For instance, in [23], the authors prove that a strip pattern of sensors can provide optimal full coverage along with 2-connectivity for all different $r/s$ ratios. The horizontal strips are formed at a distance of $x = \min(r, \sqrt{3}s)$ apart while the distance between vertical strips is $\beta = s + \frac{\sqrt{3}r^2 - \alpha^2}{4}$. In addition, horizontal strips are shifted to the right a distance of $\frac{\beta}{2}$ 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 [24,25].

Obviously these approaches are limited to WSNs for which $r > \sqrt{3}s$. Wang et al. [26] prove that $k$-coverage implies $k$-connectivity of the entire network in WSNs if $r > 2s$.

Meanwhile, the focus of the approach in [27] is on 3D setups with the additional objective of prolonging 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 authors formulate node relocation problem for balancing the traffic load and thus extending the network longevity as a Mixed Integer Linear Program optimization. A variant of the approach is presented in [28] 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. Although the $k$-connectivity approaches ($k \geq 2$) generate more resilient topology than that of the bi-connectivity algorithms, the topology formation may be too costly and impractical with large $k$.

CRAFT strives to restore a partitioned WSN and form a bi-connected topology which is tolerant to a single node failure ($k = 1$). It considers the scenario where the distance among sensors is more than $R$ which necessitates the formation of multi-hop inter-RN paths. Unlike the algorithm of Kashyap et al. [12], CRAFT and 2C-SpiderWeb [15] opts to reduce the maximum path length between segments and yield a topology with a high average node degree. They both bi-connect the segments based on an inner ring. The rationale is that a ring is a bi-connected topology that involves the fewest edges among the partitions. The problem of finding the least-cost cycle, ring, in a graph is NP hard and thus the use of heuristics like the case for CRAFT and 2C-SpiderWeb is popular. While both CRAFT and 2C-SpiderWeb form an inner ring to save more on relay count, the two approaches differ in how to form such a ring and how the other terminals are bi-connected to the ring. 2C-SpiderWeb forms an inner ring based only on the outermost terminals and 2-vertex disjoint paths are established based on the proximity of a non-ring terminal to the ring [15]. Meanwhile CRAFT forms an inner ring by considering the location of the inmost terminals and the identified SPs which connect outer terminals with the least cost and the other terminals are bi-connected by 2-vertex disjoint paths found by using the identified SPs. Finally, the main advantage of the ring formation process in CRAFT over that of 2C-SpiderWeb is due to the use of rigorously identified SPs that factor in the cost of connecting non-ring terminals to the inner ring. The performance of CRAFT is compared to those of 2C-SpiderWeb and Kashyap et al. [12] through simulation, as discussed in Section 5.

3. Problem statement

In this paper we consider a set of WSN segments that can be either standalone networks or partitions of a WSN which was subject to a major damage, e.g., inflicted by explosives in a battlefield or landslides/avalanches. We strive to establish a bi-connected inter-partition topology which provides 2-vertex disjoint paths between every pair of partitions using the least number of relay nodes. For the scenario where individual WSNs are considered to be connected to support collaborative execution of tasks, it is required for any pair of two segments to communicate with a desire to have bounded the inter-partition delay. Moreover, we assume that each partition is represented by a terminal, e.g., a gateway node that serves as an interface for the partition. Such a terminal node can be specified in case the segment is a standalone WSN. When considering partitions of a damaged topology, designating the terminal can be incorporated in the failure detection procedure. An example procedure is as follows.

When a set of collocated nodes fail, their 1-hop neighbors detect the node loss and assess the scope by correlating consecutive node failures in the vicinity, e.g., by witnessing a major and sudden drop in communication traffic and/or by noticing unreachability of a certain set of remote nodes. Upon confirming the multi-node failure and the split of the network into disjoint partitions, these neighbors become the border nodes and broadcast a message on active links to notify all reachable nodes, which will naturally belong to their partition. After some
pre-determined convergence time, a representative terminal (a gateway) for each partition is elected in regard to a variety of terminal selection attributes such as application features, and onboard energy. The effect of the representative selection criteria on the RN placement optimization is not investigated and the focus in the rest of the paper is only on forming a connected topology using the partition representatives. Fig. 2 which serves as a working example in this paper, shows the terminals, $P_0, P_1, \ldots, P_6$ for the partitions in Fig. 1(a).

Then the problem that we tackle in this paper becomes forming a 2-connected topology among the terminals $P_i$’s by deploying the least number of RNs such that 2-vertex disjoint paths between every pair of partitions are provided, which eventually forms a simple cycle and thus any single RN failure will not cause the network to (re)partition. The number of relays $N_{RN}$ is captured mathematically by the following formula:

$$N_{RN} = \arg\min_{N(0)} \sum_{i=0}^{N_p-1} \left( \sum_{j=0}^{N_p-1} \text{Cycle}(P_i, P_j) \right),$$

where

$$N_p = \text{number of partitions},$$

$$\text{Cycle}(P_i, P_j) = \begin{cases} RN_k, k = 0, \ldots, N(i) - 1 \\ RN_i \neq RN_m, \forall i, m \in \{0, \ldots, N(i) - 1\} \\ P_i \neq P_j \in RN_k \end{cases}$$

(1)

In formula (1), $\text{Cycle}(P_i, P_j)$ becomes a list of RNs which forms a simple cycle that includes two partitions $P_i$ and $P_j$ and no intermediate points (i.e., one of the RN) are repeated. Since the formulated problem is a known NP-hard problem [3], we pursue heuristics.

The paper focuses on the algorithmic aspect of the internetworking problem without considering the diversity of the physical, link and network layers. Moreover, the deployed RNs are assumed to be more capable than a sensor node, with significantly richer energy reserve, and stronger computation power. It is also assumed that all relays have the same communication range $R$ which is equal to that of a sensor $r$, i.e., $R = r$. This is a simplifying assumption to ease the presentation; the approach can handle the case when the RN has larger communication range than a sensor node. While RNs may have sensing capabilities that can mitigate coverage loss caused by the damage, coverage is not the focus of CRAFT. Intuitively relays are more expensive than stationary sensors, and the number of deployed relays is thus naturally fewer and should be minimized during the bi-connectivity establishment process.

4. The CRAFT approach

CRAFT opts to employ the fewest RNs such that the resulting topology provides two disjoint inter-terminal paths between every pair of terminals. CRAFT also opt to increase the efficiency of the formed topology by boosting the node degree and reducing the maximum path length between pairs of partitions in order to help in bounding the inter-partition data latency. The main idea of CRAFT is to form an inward ring and establishes two disjoint paths from each terminal to such a ring. The proposed CRAFT approach pursues a greedy heuristic and operates in two phases. The first phase determines the $BP$ and establishes 1-vertex connectivity. CRAFT strives to identify the largest polygon $BP$ that is located around the center of the area that ought to be populated by RNs and within which no $P_i$ resides. This is done iteratively by forming the convex hull and determining SPs. By deploying RNs on the edge between consecutive terminals on the periphery of the $BP$ a ring is formed. The identified $BP$ plays a role of backbone, i.e., core mesh, in the resulting bi-connected network topology.

After that, each non-BP terminal $P_i$ is connected to the $BP$ by selecting a primary bi-connection point, denoted as $\text{Primi}$, for which $P_i$ can be connected over the least-cost
path. The path to Prim’s that consists of the found SPs during the BP identification process are populated with RNs to form the initial 1-connected topology \( G = (V, E) \). In the second phase, CRAFT opts to introduce a simple cycle and merges it into the BP. For each non-BP terminal \( P_i \), CRAFT selects an RN in \( G \) to serve as the gateway for \( P_i \) on such a secondary cycle. The selected RN is referred to as secondary bi-connection point and in short, Secondi. To avoid the involvement of cut-vertices during the selection of Secondi, CRAFT performs coloring of vertices in \( V \) such that adding a new link between two groups of different colors does not include a cut-vertex in \( G \). Finally, the BP gets expanded by steinerizing the links on the secondary cycle and deploying RNs. CRAFT is described in detail in the balance of this section. A summary of acronyms used in the discussion can be found in Appendix A.

4.1. The 1st Phase: Forming the 1-vertex connected topology \( G \) based on BP

During the first phase, CRAFT strives to form the initial 1-vertex connected topology \( G \) by computing BP, to which each partition \( P_i \) is connected over the least cost path. Thus, the 1st phase consists of two main steps: BP identification, and formation of 1-vertex connected initial topology \( G \).

4.1.1. BP Identification

The BP is basically a convex polygon that includes one or multiple terminals on its boundary and no other terminal inside. In order to identify the BP, CRAFT finds the potential bi-connection points in rounds starting from the border partitions inward towards the center of the damaged area. In the first round \( r = 0 \), CRAFT thus identifies border terminals (BT) which lie on the convex hull of all \( P_i \)'s by applying the Graham scan algorithm [29] which identifies the border of \( n \) partitions by starting from the left-most (least \( y \)-coordinate) partition \( P_k \), then sorting partitions in increasing order of an angle with \( x \)-axis relative to \( P_k \), and finally selecting partitions from the sorted list so that right-turns are maintained when navigating from \( P_k \) until ending at \( P_k \). The time complexity of Graham scan algorithm is \( O(n \log(n)) \), where \( n \) is the number of partitions in our case. After that all partitions are divided into two groups; the first includes the partitions (BTs) on the found polygon \( PG_0 \), i.e., convex hull, and hereafter is denoted as onPG0(), and the second group, referred to as inPG0(), includes those terminals which reside inside \( PG_0 \). Upon the identification of the first convex hull, two cases are separately handled due to the number of partitions (\( N_p \)) or a particular formation of partitions: (i) if every \( P_i \) becomes a BT and be included in onPG0() in the first round, then CRAFT terminates and returns the fewest number of required RNs to form a ring for the terminals of onPG0() in order to establish a bi-connected topology; (ii) if all \( P_i \)'s are collinear, they will be part the BT; then CRAFT steinerizes the edge “e” between each pair of adjacent partitions and populates double of the least count of RNs before terminating.

Otherwise, based on the found onPG0(), CRAFT finds SPs which connect sets of adjacent BTs in onPG0() with the least cost. These SPs will be located inside \( PG_0 \) and are referred to hereafter as inward-BT-SPs. The rationale of finding inward-BT-SPs is twofold: (i) the identified BP will be located close to the center of the damaged area when inward-BT-SPs are involved during the identification process, and (ii) an inward-BT-SP would be the potential point where the paths from multiple partitions to the BP can merge and thus connect with the BP at the same Prim. For computing the inward-BT-SP’s, we employ the \( k \)-restricted loss-contracting algorithm (k-LCA) with \( k = 3 \) [30] which is the best-known approximation algorithm for solving a Steiner Tree Problem (STP). k-LCA opts to seek the \( k \)-restricted full component “F”, which is the minimum-cost Steiner tree (MST) that spans \( k \) BTs as leaves by trying all candidate SPs. The rationale of selecting \( k = 3 \) for computing inward-BT-SPs is that for \( k \) values exceeding three it becomes quite complex to find \( F \) that has \( k \) nodes as leaves and the cost of forming an MST becomes excessively high.

![Fig. 3. The shade cells in each triangle \( T \) represent the potential SPs among which k-LCA (k = 3) identifies an SP that connects the three corners A, B and C of \( T \).](image-url)
Fig. 4. BP identification during the 1st phase of CRAFT: (a) In the first round \((r = 0)\), four partitions \(P_0, P_2, P_5\) and \(P_6\) become a border terminal \((BT)\) and four SPs such as \(SP_0^i\)'s, \(i = 0, \ldots, 3\) are computed. Therefore, \(\text{onPG}_0()\) and \(\text{inPG}_0()\) includes \{\(P_0, P_2, P_5\)\} and \{\(P_3, P_4, SP_0^0, SP_0^1, SP_0^2, SP_0^3\)\}, respectively. (b) During the second round \((r = 1)\), two more \(P_i\)'s \(P_1\) and \(P_4\) are identified as a BT and each of \(\text{onPG}_1()\) and \(\text{inPG}_1()\) becomes \{\(P_1, P_3, SP_0^0, SP_0^1, SP_0^2\)\} and \{\(P_0, SP_1^0, SP_1^1, SP_1^2, SP_1^3\)\}, respectively. (c) The found \(\text{onPG}_2()\) in the third round \((r = 2)\) becomes BP since the last unengaged partition \(P_3\) is identified on \(\text{onPG}_2(P_3, SP_1^0, SP_1^1, SP_1^2, SP_1^3)\) and no more \(P_i\) resides inside the BP.
For candidate SPs, we model the area of interest as a grid of equal-sized cells as seen in Fig. 3 and limit the set of potential SPs to the cells in the triangle formed by three BTSs T(A,B,C) while finding the MST that connects the three corners A, B, and C. Since the cost of the MST is determined by the fewest number of RNs to connect an SP to each of three BTSs (corners), we can find the optimal SP by trying all cells inside the triangle T(A,B,C). It is worth noting that the found SPs SP₀’s for all boundary partitions will be inward and thus will be added to inPG₀(). In addition, inPG₀() also includes Pᵢ’s that is not part of onPG₀(); that is, inPG₀() = inPG₀() ∪ (SP₀, vⱼ). Then the next round (r = 1) begins with the updated inPG₁(). In the second round, CRAFT again identifies the polygon PG₁ which matches the convex hull of nodes in inPG₀(). In other words, inPG₀() is divided into two sets onPG₁() and inPG₁() that contains (the SP₀’s of the first round and unprocessed partitions) on and inside PG₁, respectively. k-LCA is applied to find SP₀’s for the BTSs ∈ onPG₁(), and the identified SP₀’s are then added to inPG₁(), namely the inPG₁() is updated as inPG₁() ∪ (Pⱼ, vⱼ). The same procedures are repeated until no more Pᵢ’s are left in the updated inPGᵢ₋₁() in round r, i.e., all Pᵢ’s have been identified as a BT in onPGᵢ(), i = 0, ..., r − 1.

Fig. 4 shows the identified polygons onPGᵢ()’s and inward-BT-SPᵢ’s during three rounds r = 0, 1, and 2. In the first round P₀, P₂, P₅ and P₆ form PG₀ and the found four SP₀’s SP₀₀, SP₀₁, SP₀₂, SP₀₃ are added to inPG₀() along with P₁, P₃, and P₄ which lie inside PG₀ as seen in Fig. 4(a). During the second round seen in Fig. 4(b), a new polygon onPG₁(P₁, P₄, SP₀₀, SP₀₁, SP₀₂, SP₀₃) is computed and the newly found SP₀’s SP₁’s, j = 0, ..., 3 are added to inPG₁() which also includes P₃. It is noted that no inward-BT-SP for P₁, SP₁₀, and SP₁₁ is identified, since the three BTSs are collinear. In the third round, P₅ is finally identified as a BT by PG₂ and added in onPG₂(P₃, SP₁₀, SP₁₁, SP₁₂, SP₁₃) which becomes the backbone ring BP and the first phase of CRAFT terminates, as seen in Fig. 4(c). It is worth noting that the identified BP is located most inward towards the center of the area of partitioned network among the found onPGᵢ’s in all rounds since the onPGᵢ for round r is covered by all previously found convex hulls, i.e., PGᵢ’s, k = 0, ..., r − 1, which will be proven in the Section 4.3.

4.1.2. Construction of an initial 1-vertex topology

With the identified BP and inward-BT-SPᵢ’s during the previous step, CRAFT then strives to form the 1-vertex initial topology G by finding Prim that lies on the found BP for each non-BP terminal Pᵢ. The goal of picking Primᵢ is to provide the primary connectivity among partitions which is efficient in terms of a rely count. Therefore, for each Pᵢ its Primᵢ is selected to reduce the overall count of required relays to connect a pair of Pᵢ and Primᵢ, ∀i. In order to find the least-cost Primᵢ for a Pᵢ, CRAFT employs a greedy process. First, CRAFT selects the closest node to Pᵢ that has become a BT in onPGᵢ₋₁() in round i, among the ones in inPGᵢ₋₁() which may be another partition or an inward-BT-SP and denotes it as £₀ᵢ. If £₀ᵢ does not lie on BP, then the same process is repeated and the nearest node to £₀ᵢ among the nodes covered by onPGᵢ(), where i is the round in which £₀ᵢ is identified as a BT becomes £ᵢᵢ. As the chosen £ᵢᵢ lies on the BP, £ᵢᵢ becomes Primᵢ and the primary path from Pᵢ to BP, hereafter denoted as Eᵢ = (Pᵢ, £₀ᵢ, £₁ᵢ, ..., £ᵢ₋₁ᵢ, £ᵢᵢ = Primᵢ) is formed.

Fig. 5 shows the primary path Eᵢ as a set of arrows. For instance, for P₀ another partition P₁ is selected as £₀₁ among inPG₀(P₀, P₁, P₄, SP₀₀, SP₀₁, SP₀₂, SP₀₃) since P₁ is closer to P₀ than SP₀₁. After that, SP₁₀ becomes £₁₀ and also Prim₀ since it is on BP, and a set of two links (P₀, P₁) and (P₁, SP₀₀) then forms the path £₀ that connects P₀ to the BP. Similarly, for P₁, P₂, P₄, P₅, and P₆ the paths E₁ = (P₁, SP₁₀), E₂ = (P₂, SP₀₀, SP₁₀), E₃ = (P₃), E₄ = (P₄, SP₁₁), E₅ = (P₅, SP₀₁, SP₁₁), and E₆ = (P₆, SP₀₂, SP₁₂) are found, respectively. It is worth noting that £₀₀ and £₁₀ are merged and P₀ and P₁ are thus connected to BP throughout the same Prim which is SP₀₀.

Fig. 5. Arrows present a £ path, for instance Pᵢ and SPᵢ₀ are selected as £ᵢ₀ and £ᵢ₁ for P₀ respectively. P₀ finally reaches BP via the path £₀ = (P₀, P₁, SP₀₀). Meanwhile, SPᵢ₀ is removed from Fig. 4(c) since it does not support any partition during Prim selection.
i.e., \( \text{Prim}_0 = \text{Prim}_s \). As explained early, this is because for finding \( \mathcal{E}_i \) we employ the set of the inward-BT-SPs, each of which connects three of BTs that may be a partition or a previously identified SP with the least cost. Hence, such merging apparently aids the reduction of a RN count to connect all partitions to the backbone ring \( \mathcal{B}P \) in the resulting bi-connected topology. In addition for \( P_3, \mathcal{E}_3 \) contains only \( P_3 \) itself since \( P_3 \) is already on the \( \mathcal{B}P \). As a result, the initial topology \( G \) that provides 1-vertex connectivity among partitions is formed by populating RNs on each path \( \mathcal{E}_i \) such that the distance between a pair of RNs does not exceed the communication range of a relay \( R \). Moreover, the additional RNs are deployed on each steinerized edge of \( \mathcal{B}P \).

Then, the 2nd phase of CRAFT begins based on \( G = (V,E) \) seen in Fig. 6, where \( V \) is a set of populated RNs and \( E \) includes communication links between vertices in \( V \). It is worth noting that \( G \) contains one simple cycle \( \mathcal{B}P \) and multiple paths \( \mathcal{E}_i \)'s along each of which one or more partitions is connected to \( \mathcal{B}P \) without introducing a cycle as will be proven in the Analysis section. The pseudo-code for the first phase of CRAFT is presented in Algorithm 1.

**Algorithm 1. Pseudo-code for the 1st phase of CRAFT**

```plaintext
Identification_BP()
1: \( G = (V,E); V = \emptyset = \emptyset; \)
2: \( \text{InPGr}[i] = P_i, \forall i; \text{onPGr}[i] = \emptyset; \)
3: \( r = 1; /^* \text{ round } */^* \)
4: \( \text{onPG}_i = \text{NODE}(\text{convex hull of } \text{InPG}_{r-1}[i]); \)
5: \( \text{if } P_i \forall i \in \text{onPG}_i \) then
6: \( \text{N}_R = \#P(\text{node}), \forall \text{ on the polygon } \text{onPG}_i; \)
7: \( \text{Return } \text{N}_R; \)
8: \( \text{end} \)
9: \( \text{while} \{ \)
10: \( \text{InPG}_r[i] = \{\text{SPs for each set of } 3 \text{ adjacent BTs on } \text{onPG}_r[i]; \)
11: \( \text{InPG}_r[i] = \text{InPG}_r[i] - \text{onPG}_r[i]; \)
12: \( r++; \)
13: \( \text{onPG}_r[i] = \text{NODE}(\text{convex hull of } \text{InPG}_{r-1}[i]); \)
14: \( \text{do}(\exists P_i \in \text{InPG}_{r-1}[i] - \text{onPG}_{r-1}[i]) \}
15: \( \text{BP}[i] = \text{onPG}_r[i]; \)
16: \( \text{for each } P_i \)
17: \( j = -1; \)
18: \( \text{while} \{ \)
19: \( j++; \)
20: \( E_{j}[i] = \text{Closest node to } P_i \in \text{InPG}_r[i], r \text{ is the round where } P_i \text{ is identified as a BT on } \text{onPG}_r[i]; \)
21: \( \text{do } ((E_{j}[i] \in \text{BP}[i]) \}
22: \( \text{Prim}_i \rightarrow E_{j}[i]; \)
23: \( \text{end} \)
24: \( V_{BP} = \{\text{RN}(e), \forall e \text{ on BP}[i]; \)
25: \( V = V_{BP} \cup \{\text{RN}(\text{PATH}(P_i, \text{Prim}_i), \forall i); \)
26: \( E = \{\text{LINK}(u,v), \forall u \text{ and } v \in V; \)
```

Fig. 6. The initial 1-vertex connected inter-partition topology \( G \) is formed by the identified \( \mathcal{B}P \) and primary paths \( \mathcal{E}_i \)'s during the first phase of CRAFT. In \( G \), circles and lines represent an RN and a communication link, respectively. In addition, the number written inside a circle is a result of the coloring procedure explained in Section 4.2. Having the same number implies that the corresponding nodes belong to the same color group and adding a link between different numbers does not involve a cut-vertex in \( G \). The black circle means it is already in a backbone simple cycle.

Please cite this article in press as: S. Lee et al., Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.07.012
In order to reduce the number of required RNs for bi-connectivity, CRAFT opts to create a cost efficient simple cycle for each \( P_i \). It then selects \( \text{Second}_i \) for each \( P_i \) in \( V \) for which \( SC_i \) is formed with the deployment of the fewest additional RNs. To identify \( \text{Second}_i \), \( \text{A} \) CRAFT performs coloring during which each path \( E_i \) is marked by a distinct color. First, CRAFT picks \( N_{BP} \) distinct colors, i.e., \( \text{color}(0), \text{color}(1), \ldots \text{and color}(N_{BP}-1) \), where \( N_{BP} \) is the number of \( BTs \) on \( BP \), which equals to \(|\text{on}_{PG_f-1}(i)|\), where \( f \) is the final round of the \( BP \) identification phase. Then, each \( \text{Prim}_i \) is assigned a distinct color. Recall that \(|\text{on}_{PG_f-1}(i)|\) includes disengaged partitions and \( SPs \) via which more than one partition may be merged before reaching the \( BP \). Thus \( N_{BP} \) is less than or equal to \( N_P \), where \( N_P \) is the number of partitions in the network since during the \( BP \) formation phase the number of the identified \( SPs \) in a round cannot be larger than the number of \( BTs \) in that round.

After that, CRAFT colors the RNs belonging to the path \( E_i \), rooted at the \( \text{Prim}_i \), with \( \text{COLOR(Prim}_i) \). In other words, coloring is performed in the backward direction of finding \( \text{Prims} \) by assigning \( \text{COLOR(Prim}_i) \) to \( E_{\text{Prim}_i-1}, E_{\text{Prim}_i-2} \), and so on until reaching \( P_i \). A merged primary will be assigned the same color. Fig. 6 shows the results of the coloring process for the topology of Fig. 5, where there are six different color groups, \( \text{color}(0), \ldots, \text{color}(5) \), generated for seven partitions since \( P_1 \) has been merged into the path for \( P_0 \) whose \( E \) path arrives at \( \text{Prim}_0 \). In addition, the group \( \text{color}(5) \) has only one element which is \( P_3 \) since \( P_3 \) has been identified on \( BP \) itself. Moreover, the RNs populated at the steinerized points on \( BP \) becomes \( \text{black} \), which represents a backbone ring as seen in Fig. 6. Thus all vertices (RNs) in \( V \) are categorized into total of seven distinct color groups, \( \text{color}(c), c=0, \ldots, 5 \), and \( \text{black} \), based on which the \text{Second selection} is performed. With the help of \textcolor{black}{} it can be noticed that adding a path between certain pair of vertices in the same color group generates a cut-vertex. Therefore, CRAFT selects \( \text{Second}_i \) by simply finding the closest RN to \( P_i \) whose color is not the same as \( \text{COLOR}(P_i) \). It is worth to note that due to coloring the vertex which does not lie on the backbone ring and is located closer to \( P_i \) can be easily identified and selected as \( \text{Second}_i \) and thus the additional RN count to form the alternative path to link \( P_i \) to \( BP \) can be reduced as seen in the example of Fig. 7(a).

In addition, the \text{Second} selection is performed only for the \( P_i \)’s located at periphery or leaf in \( G \), for instance, \( P_0, P_5, P_2, P_3 \), and \( P_6 \). This is because a non-leaf \( P_i \) will share the same \( \text{Prim} \) with a leaf partition and their primary paths merge; thus \( P_i \) will be automatically included in the simple cycle \( SC_i \) formed by the selection of \( \text{Second}_i \) for \( P_i \), whose \( E_i \) includes \( P_i \). For example, in Fig. 8(a) \( P_i \) automatically lies in \( SC_0 \) created by \( \text{Second}_0 \) selection. After finding each \( \text{Second}_i \), the edges on the corresponding cycle are steinerized and RNs are populated. Before processing the next \( P_i \), the RNs involved in \( SC_i \) are colored \text{black} except the one selected as \( \text{Second}_i \). It is worth noting that the original color of the selected \( \text{Second}_i \), should remain intact since it may be also \( \text{Prim} \) of another partition which has not been bi-connected yet. Coloring the selected \( \text{Second}_i, \text{black} \) may generate an unexpected cut-vertex in the resulting topology as illustrated in the example in Fig. 7(b).

Finally, the found \( SC_0 \) is naturally combined into the existing \( BP \) as explained earlier, which then becomes a larger backbone ring \( BP^{\#} \) that equals to \( \{BP \cup SC_0 \} \), shown as a bold circle in Fig. 8(a). Fig. 8(b) and (c) also show the updates of \( BP^{\#} \) by each selection of \( \text{Second}_2 \) and \( \text{Second}_3 \) for \( P_2 \) to \( P_6 \). As a result, CRAFT bi-connects the seven partitions.

![Fig. 7](https://example.com/fig7.png)

**Fig. 7.** (a) When applying path coloring the selected \( \text{Second}_4 \) and \( \text{Second}_6 \) do not lie on \( BP \) which reduces the RN count for forming a secondary path from \( P_4 \) and \( P_6 \) to the \( BP \). (b) Coloring the \( \text{Second} \) with black may introduce a cut-vertex in the resulting topology. If \( \text{Second}_3 \) would become \text{black} in the left side of (b), then the node 3 may be selected as \( \text{Second}_3 \) based on proximity as illustrated in the right side. In such a case, the failure of \( \text{Second}_3 \) would get two \( D \) nodes disconnected from the topology. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
in Fig. 2 by populating total 52 of RNs in the resulting topology $G$ seen in Fig. 8(c). Overall, the resulting final topology contains multiple simple cycles (rings) which are inter-connected based on the inner ring, BP, since the two-vertex disjoint paths from each partition to the BP form a distinct simple cycle. In addition to being tolerant to a single RN failure, $G$ also provides the efficiency in terms of connectivity, as will be elaborated later. The pseudo-code for the second phase of CRAFT is shown in Algorithm 2.

Algorithm 2. Pseudo-code for the 2nd phase of CRAFT

```
Color(v, color)
1: COLOR(v) = color;
2: for u, $\forall u$ whose Prim, is v
3: COLOR(u, color);
4: end

Identify_Second_Connection_Points(G, V_BW)
5: COLOR(v in V_BW) = black;
6: for i = 0, c = 1 to SIZE(BP[ ]); i++, c++;
7: COLOR(Prim, color(c));
8: end
9: for each leaf $P_i$, $\forall i$
10: Second, $\forall v \in V$; MIN(EuclideanDist(v, $P_i$)) and COLOR($v$) = COLOR($P_i$);
11: $V = \{RN\/PATH(P_i, Second, )\}$;
12: COLOR($\forall v \in V$ & Prim, black );
13: Keep color of Second, as it was;
14: $V = V \cup V$;
15: $E = E \cup \{New$ available links$\}$
16: end
17: Return ($N_{RN} = |V|$);
```

4.3. Algorithm analysis

The convergence and runtime complexity of CRAFT are analyzed in this subsection. We mainly focus on proving that CRAFT converges to form a bi-connected inter-partition topology within an execution time bound and thus the restored topology can tolerate a single node failure. We introduce the following theorems and lemmas:

Lemma 1. inPG$^r_G$ is a subset of inPG$^r_G$ for $r' > r$.

Proof. This lemma is proven by induction. First, by definition the initial polygon or convex hull PG$^0_G$ computed in round 0 covers all $P_i$'s and there are no SPs identified yet. In addition, SP$^0_i$, $\forall j$ which connects a pair of three adjacent BTs identified on PG$^0_G$ should be inside of a triangle formed by those three terminals, as shown in Fig. 3. Since the vertices of such a triangle are in PG$^0_G$, SP$^0_i$, $\forall j$ are also inside PG$^0_G$, i.e., SP$^0_i \in$ PG$^0_G$, $\forall j$. Then, in the next round (i.e., $r = 1$) the identified SP$^1_i$'s and all partitions that are not in PG$^0_G$ are considered to compute the next polygon PG$^1_G$ and may become a BT. Since all considered nodes for determining onPG$^1_G$ are in PG$^0_G$, the nodes on onPG$^1_G$ also lie completely inside PG$^0_G$. The same can be proven for onPG$^r_G$ and onPG$^r_G$ for $r' = r + 1$ and by induction for any $r', r' > r$.

Lemma 2. The path $E_i (E_i^0, E_i^1, \ldots, E_i^{m-1}, E_i^m)$ from a partition $P_i$, where $E_i^m \in$ BP, grows from $P_i$ to the center of the area and includes no cycle.

Proof. This lemma is proven by induction using Lemma 1. In round $r$ where each $P_i$ is identified as a BT and included in onPG$^r_G$, $E_i^0$ is selected among the members of inPG$^r_G$ which equals to $\{(a$ of all non-BT $P_j$'s and SPs up to a round $r) \cup (a$ set of found SP$^{-1}_k$s in round $r$) as seen in line 19 of Algorithm 1. Since inPG$^r_G$ is a subset of onPG$^r_G$ as proven in Lemma 1, $E_i^0$ is a distinct node from $P_i$ and also is located closer towards the core of the area than $P_i$. In other words, $P_i$ is located closer to the network periphery than $E_i^0$. After that if $E_i^0$ is not on BP, $E_i^1$ is again selected among the elements of inPG$^r_G$, which is a subset of inPG$^r_G$. Thus, the same insight applies to the relationship between $E_i^1$ and $E_i^2$. Since the $E_i$ path ends at $E_i^m$ which lies on BP, $E_i^m \neq E_i^l$, $0 < q < s < m$ and $E_i^q$ is situated towards the border of the area than $E_i^{m-1}$ in the path $([P_i, E_i^q], \ldots, (E_i^q, E_i^{m-1}), \ldots, (E_i^{m-1}, E_i^m))$. Therefore, it is proven that each $E_i$ path which connects $P_i$ to BP at $E_i^m$ includes distinct elements, i.e., no points are revisited in $E_i$. □

Lemma 3. Using coloring to select Second, for $P_i$ that is not on BP leads to forming a simple cycle which connects $P_i$ to BP through at least two distinct points.

Proof. After applying the coloring process during the 2nd phase of CRAFT all partitions are categorized into three different groups; (i) partitions which directly lie on BP, for example $P_n$ seen in Fig. 9 belongs to this group; (ii) partitions like $P_i$ in Fig. 9, whose $E_i$ path is completely included in the path of another partition $P_k$ which is colored prior to $P_i$; (iii) partitions like $P_i$ and $P_k$ in Fig. 9, which are leaves of their Primpaths that may get merged. Since the partitions in the first two groups (i) and (ii) are directly or indirectly absorbed into BP after the 2nd phase, we try to prove that a simple cycle exists for the last group (iii).

For the partition $P_i$ in the group (iii), its Primpath $E_i$ includes $\{(P_i, E_i^0, \ldots, E_i^{m-1}, E_i^m = Prim_i)\}$ where no point is repeated as proven in Lemma 2. The secondary path $E_i$ is basically a steinerized edge between Second, and $P_i$. Let $E_i = (P_i, x_0, \ldots, x_l, Second_i)$ where $x_0, \ldots, x_l$ denotes the least number of deployed RNs while steinerizing the edge between Second, and $P_i$. The nodes $x_0, \ldots, x_l$ are newly deployed RNs and are thus distinct from $E_i^0, \ldots, E_i^{m-1}$. Since CRAFT selects Second, as the RN whose color is different from $P_i$, it is natural that $Second_i \neq E_i^m$, $\forall m$. In addition, CRAFT populates additional RNs on the secondary path $E_i$, it is evident that $Second_i \neq x_r$, $\forall i$, and thus two paths $\{(P_i, E_i^0, \ldots, E_i^{m-1}, E_i^m = Prim_i)\}$ and $\{(P_i, x_0, \ldots, x_l, Second_i)\}$ are distinct.

Moreover, by construction BP forms a simple cycle where no points are revisited and thus there sure is a vertex-distinct path $\gamma$ on BP starting at Prim, and ending at Second. Therefore, for $P_i$ that is not on BP, there exists a simple cycle $SC_i = \{E_i, \gamma, E_i\}$ which includes at least two rendezvous points Prim, and Second, with BP. In addition, like $P_i$ the selected Second, may join at another $SC_i$ as seen
in Fig. 9(c). For this case, the same proof for \(P_0\) is applied to \(P_j\) since the previous \(BP\) is expanded by combining \(SC_i\). \(P_j\) also has thus two vertex-disjoint paths to reach \(BP\) at two points \(Prim_j\) and \(Second_i\).

**Theorem 1.** The resulting topology \(G\) formed by CRAFT guarantees 2-vertex connectivity.

**Proof.** In order to prove Theorem 1, it is sufficient to show that all partitions lie in one vertex-disjoint cycle. Using Lemma 3, it is proven that every \(P_i\) lies on \(BP\) or on a simple cycle \(SC_i\) which meets \(BP\) at two distinct rendezvous points. Let \(SC_i\) and \(BP\) be \(\{E_i, Y_i, E'_i\}\), where \(E_i\) is the primary path for \(P_i\) and \(E'_i\) is secondary path for \(P_i\) and \(\{Y_0, Y'_i\}\), where \(Y'_i = BP - Y_i, (|Y'_i| > |Y_i|)\) \(Y_i\) is the vertex-disjoint path starting at \(Prim\) and ending at \(Second_i\) in \(BP\), respectively. Then each

---

Please cite this article in press as: S. Lee et al., Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.07.012
time $P_t$, $i = 0, \ldots, Np - 1$ is processed during the 2nd phase of CRAFT. $BP$ may be intact for a partition on $BP$ or extended into larger vertex-disjoint backbone ring $BP = \{e_i, y_i, e'_i\}$. For example, $BP = \{e_0, y_0, e'_0\}$. $BP = \{e_1, y_1, e'_1\}$, where $y_i = \{e_0, y_0, e'_0\}$. $y_i$ is the shortest vertex-disjoint path starting at $Prim_k$ and ending at $Second_i$ in $BP$. $BP = \{e_k, y_k, e'_k\}$, where $y_k = \{e_{k-1}, y_{k-1}, e'_{k-1}\}$. $y_k$ is the shortest vertex-disjoint path starting at $Prim_k$ and ending at $Second_i$ in $BP$. $BP$ and so on. Since it is proven that $E_k$ and $E'_k$ are vertex-disjoint in Lemma 3 for $\forall k$, and $y_k$ and $y_k$ are also vertex-disjoint by construction of $BP$, it is evident that the extended backbone ring $BP$, $\forall k$, i.e., $k = 0, \ldots, Np - 1$, becomes 2-vertex disjoint. Therefore, it is proven that the resulting topology $G$ formed after the second phase of CRAFT contains all partitions $P$, $\forall i$ in one simple cycle $BP$ and thus $G$ is 2-vertex connected.

**Theorem 2.** The time complexity of CRAFT is proportional to the number of partitions $Np$ and inversely proportional to $R$ as $O(N_{end} \cdot Np \cdot (logNp + \lceil 2 \cdot Size(LP_k)/R^2 \rceil + |V|)$, where $N_{end}$ is the number of rounds until forming the BP.

**Proof.** The time complexity for identifying the BP during the first phase of CRAFT is bound to $O(N_{end} \cdot Np \cdot (logNp + \lceil 2 \cdot Size(LP_k)/R^2 \rceil + |V|)$, which has been proven in [31]. In addition, finding $Prim$ path $E_i$ for each $P_i$ is $O(Np)$. After the first phase is done, $G = \{V, E\}$ is formed where $V$ is a set of all RNs populated in $BP$ and $E_i$ paths, $\forall i$. Moreover, the execution time for the second phase during which every RN in $V$ is colored and $Second$, for each $P_i$ is selected is bounded to $O(|V| + Np)$. Since more RNs are required with smaller $R$ to connect $G$, $|V|$ is primarily determined by $R$. Overall, the time complexity of CRAFT is thus bounded by $O(N_{end} \cdot Np \cdot (logNp + \lceil 2 \cdot Size(LP_k)/R^2 \rceil + |V| + Np)$ which equals to $O(N_{end} \cdot Np \cdot (logNp + \lceil 2 \cdot Size(LP_k)/R^2 \rceil + |V|)$, where $N_{end}$ is the number of rounds taken until no more $P_i$’s reside inside the identified polygon and $BP$ is thus identified, $Np$ is the number of partitions, $R$ is a communication range of a RN, $LP_k$ is the smallest rectangle which includes the largest triangle formed by the three $BT$s identified in the first polygon $onPG()$ and $|V|$ equals to the least count of required RNs to form the initial topology $G$ to be mainly affected by $R$. In addition, a value of $N_{end}$ may be one if all partitions or no partitions are identified as a $BT$ in the first round, or $[Np/3]$ in the worst case where the partition distribution in the damaged area is such that exactly three partitions are found as a $BT$ in each round, and thus $1 \leq N_{end} \leq |Np/3|$. The more spread the partitions get, the more relays are required to bi-connect them and also the more the time complexity becomes. Therefore, the runtime complexity of CRAFT is bounded to between $O(Np \cdot logNp)$ and $O(Np^2 \cdot logNp)$, where $Np$ is the number of partitions.

5. Performance evaluation

In this section, the effectiveness of CRAFT is validated. We have implemented a simulation environment in C. The simulation experiments study the performance with respect to the number of relays populated by CRAFT in comparison to the best known and recently published algorithms in the literature. The quality of the resulting topology of CRAFT is also compared and discussed.

5.1. Validation experiments

The simulation environment, performance metrics, and experimental results are discussed in this subsection.

5.1.1. Experiment setup and performance metrics

The following parameters are used to vary the network characteristics:

- **Communication range of relays ($R$):** $R$ has the most influence on the performance of CRAFT since the minimum number of RNs to provide connectivity among partitions primarily depends on the inter-partition proximity.
- **Number of partitions ($Np$):** Having high count of partitions may increase the connectivity requirement and thus larger number of RNs would be needed.

The performance of CRAFT is assessed using the following three metrics:

- **Number of RNs ($N_{end}$):** Obviously, CRAFT strives to minimize the required number of RNs to form a bi-connected topology. $N_{end}$ is affected by both $R$ and $Np$.
- **Average node degree:** This measures the connectivity of the resulting topology. It equals the total number of direct wireless links among neighboring RNs divided by $N_{end}$. All links are assumed to be bidirectional and each edge in the formed topology is counted twice for.
- **Maximum hop count:** It equals the number of links on the shortest path between two segments that are the furthest apart with respect to an inter-partition hop count in the resulting bi-connected topology. This metric hints the largest data delivery latency among partitions which is determined by a network topology.
- **Average node degree per relay:** It equals average node degree divided by $N_{end}$ and gauges the efficiency of the topology formed by CRAFT; a larger value means higher connectivity is achieved by the same number of relays. Basically, CRAFT strives to satisfy two conflicting objectives, namely, forming an efficient topology with high connectivity and populating the least number of relays. In other words, this metric reflects how the effectiveness of CRAFT in balancing the two objectives.

5.1.2. Baseline approaches

Based on the metrics and environmental parameters stated above, the performance of CRAFT is compared to two approaches. The first approach is the best approximation algorithm for providing 2-vertex connectivity based on minimum spanning sub-graph, and is referred to as 2C-MSS [12]. The second baseline approach is 2C-SpiderWeb [15], which forms a topology that resembles a spider web with partitions located at its perimeter and adds extra edges to make it 2-vertex connected. These baseline approaches address the same problem tackled by CRAFT; however, they use different solution strategies.
• **2C-MSS:** This approach opts to achieve 2-vertex connectivity by populating the smallest number of additional relays among \( n \) terminals. It first models the problem as a weighted complete graph \( G_c = (V_c, E_c) \), where \( V_c \) includes \( n \) nodes which correspond to a set of \( P_i \)'s in this simulation and \( E_c \) is a set of all available links between every pair of vertices in \( V_c \). Then the weight of each edge \( e \) is set to the number of required relays to connect the two end points of \( e \); that is \( \left( \frac{n}{2} - 1 \right) \), where \( |e| \) is the length of \( e \) [12]. Based on \( G_c \), an approximate minimum cost spanning 2-vertex connected sub-graph is computed by using the algorithm of [13]. Then, 2C-MSS populates the least count of RNs \( n' \) in the resulting graph \( G_c \) carries out the pruning procedure during which the unnecessarily RNs are eliminated without which \( G_c \) maintains 2-connectivity. The time complexity of the final step is bounded to \( O(n'm(m+n')) \), where \( m \) is the number of edges in the steinerized graph \( G_c \). Overall, the time complexity of 2C-MSS is bounded to \( O(4n^2+n'm(n+n')) \) where \( n \) is the number of partitions, i.e., \( N_p \). \( m \) is the number of edges in the resulting bi-connected topology and \( n' \) is the number of deployed RNs before its pruning procedure of 2C-MSS [12]. It is known that the number of deployed RNs \( (n') \) to connect partitions is apparently larger than the number of the partitions which is \( N_p \) \((n)\), and the number of edges \((m)\) which bi-connect segments is also larger than or equal to \( N_p \) \((n)\). Therefore, \( (4n^2+n'm(n+n')) > O(4n^2+2n^3) \), where \( n \) is the number of partitions \( N_p \). The execution time of 2C-MSS is significantly higher than that of CRAFT which is \( O(N_{\text{rand}} \cdot N_p \cdot (\log N_p + [2 \cdot \text{Size}(LP_k)/R^2] + |V|) \), where \( N_{\text{rand}} \) is the number of rounds until forming the BP as proven in Theorem 2 in Section 4.3.

• **2C-SpiderWeb:** This algorithm strives to bi-connect partitions inward towards the center of mass (CoM) of a polygon which matches the convex hull of all partitions \( CV_0 \). It first populates RNs along the paths from each partition \( P_i \) which lies on \( CV_0 \) towards CoM. During the deployment of RNs, two RNs that come from different \( P_i \)'s may get within a distance \( R \) and become reachable to each other. Then the two \( P_i \)'s become partially (right or left) connected to each other and the RN deployment is repeated until all partitions on \( CV_0 \) are fully (right and left) connected, and form a ring around CoM. For the other RNs which reside inside \( CV_0 \) and are still disconnected, 2C-SpiderWeb populates additional RNs along the line from \( P_i \) to its closest RN. Then, for every \( P_i \), the additional path from \( P_i \) to the second closest RN in the ring is added. Each time one extra path is added, the ring is expanded based on which 2C-SpiderWeb strives to 2-connect the next \( P_i \).

In summary, 2C-SpiderWeb and CRAFT in essence try to construct a 2-connected network topology inward towards the core of the damaged area where the partitions are located at the periphery as seen in Fig. 10(a) and (b). However, they differ in building the backbone ring. Basically, the ring in 2C-SpiderWeb is formed by considering only the partitions situated at border, while the backbone polygon \( BP \) of CRAFT is formed by considering all partitions. Meanwhile, 2C-MSS does not consider the location of multiple partitions together. As a result, for the working example of Fig. 2 CRAFT, 2C-SpiderWeb and 2C-MSS populate total 52, 70 and 52 relays to bi-connect the \( P_i \)'s \( i = 0, \ldots, 6 \), respectively, as seen in Fig. 10. Albeit 2C-MSS is the best known algorithm for bi-connectivity, its execution time complexity is very high as \( O(N^2_p) \) in comparison to that of CRAFT. The next subsection reports the simulation results.

5.2. Simulation results

We have simulated multiple configurations, each of which has different combinations of values of \( R \) and \( N_p \). The value of \( R \) is varied from 40 m to 200 m with increment of 20 m. \( N_p \) takes the values of 3 to 10. Partitions are randomly located at least \( R \) units away from each other in an area of 1000 m × 1000 m. This placement articulates the effect of damage on a WSN that originally covers the area. The results of the individual experiments are averaged over 30 runs. All results are subjected to 90% confidence interval analysis and stay within 10% of the sample mean.

**Number of RNs \((N_{\text{RN}})\):** Fig. 11 compares the performance of CRAFT in terms of the least count of required RNs to form the bi-connected topology among the partitions to that of the baseline approaches, while varying \( R \) and \( N_p \). In general, \( N_{\text{RN}} \) diminishes with larger \( R \) as seen in Fig. 11(a). This is very much expected because the inter-RN proximity is determined by \( R \) and fewer relays will be needed to steinerize edges in the formed topology as \( R \) grows.

Fig. 11(a) shows that CRAFT outperforms 2C-SpiderWeb regardless of \( R \). This is because in CRAFT the bi-connected topology is formed based on the backbone ring \( BP \) found by considering all partitions. In details, the \( BP \) is identified by the computed inward-BT-SPs in rounds, each of which connects three adjacent BTs with the least cost, and each partition is then primarily connected inward to the \( BP \) along the shortest path which consists of the found inward-BT-SPs. Moreover, the coloring done during the 2nd phase of CRAFT further provides opportunity for reducing the relay count. Therefore, CRAFT saves more RNs to bi-connect the partitions in comparison to 2C-SpiderWeb where a backbone ring is constructed by proximity among only the border partitions and also the extra connection for bi-connecting each \( P_i \) is added without the investigation of the existing RNs. While 2C-MSS populates the fewest RNs among all approaches regardless of \( R \), it is noted that the performance of CRAFT improves and gets closer to that of 2C-MSS when \( R \) grows. This is because with large \( R \), forming a bi-connected topology towards the center of the area is more efficient in terms of the required relay count than forming a bi-connected topology based on a minimum spanning graph of partitions.

Meanwhile, Fig. 11(b) presents the simulation results in terms of the number of WSN segments. It is natural that for all approaches to populate more RNs as \( N_p \) grows. CRAFT...
consistently shows higher efficiency than 2C-SpiderWeb for various $N_p$, since again a backbone ring generated by CRAFT factors the positions of all $P_i$'s while the ring formed by 2C-SpiderWeb considers only the partitions on $CV_0$ which corresponds to $onPG_0()$ in CRAFT. While 2C-MSS shows the best results among all approaches for all $N_p$, CRAFT’s performance stays close with smaller $N_p$. It is worth noting that CRAFT yields almost the same performance of 2C-MSS with much lower time complexity when $N_p$ is three. This is because with three partitions, the resulting topology formed by CRAFT is the identified BP itself during the first phase which is a polygon of the three partitions that resembles the resulting topology generated by 2C-MSS.

**Average node degree:** Fig. 12(a) and (b) demonstrate the efficiency of the resulting topology of CRAFT with respect to connectivity for various $R$ and $N_p$ in comparison to the baseline approaches. The resulting topology generated by CRAFT has higher average node degree than the graph formed by 2C-MSS regardless of $R$ and $N_p$. This is...
because CRAFT bi-connects the partitions in the graph formed inward towards the core of the border polygon which covers all the partitions unlike 2C-MSS in which a mesh-like topology is generated based on the inter-partition edges as seen in Fig. 10(c). As shown in Fig. 12(a), all approaches yield higher node degree in their resulting topologies as R grows. This is expected because larger R yields more communication links of the populated RNs. 2C-SpiderWeb is in the lead since it places about 15% more RNs than CRAFT as seen in Fig. 11(a). Such an increased RN count in turn creates more wireless links among the populated RNs, especially when the communication range R grows. It is worth noting that the average node degree of CRAFT gets closer to that of 2C-SpiderWeb with smaller R despite the larger number of RN that 2C-SpiderWeb deploys.

In addition, CRAFT forms more efficient topology as \( N_p \) grows and its connectivity gets higher than that of 2C-SpiderWeb as \( N_p \) gets larger than or equals to 11 as seen in Fig. 12(b). This is because CRAFT considers the locations of all partitions while identifying the backbone ring and its efficiency thus improves with more partitions. In addition, in CRAFT the primary path for forwarding the inter-partition traffic is formed by computing inward-BT-SPs each of which connects three neighboring border terminals and some partitions will have paths among them without going to the BP. We conclude thus that the created wireless links in the bi-connected topology formed by CRAFT has higher utilization than that of 2C-SpiderWeb which also forms

These results show that CRAFT scales well for highly partitioned networks.

**Maximum hop count:** As seen in Fig. 12(c), for all approaches maximum hop count is reduced with larger R. This is because a large communication range not only reduces the number of required RNs to form a bi-connected inter-partition topology as seen in Fig. 11(a) but also increases connectivity of a populated RN as seen in Fig. 12(a). In Fig. 12(c), more delay is required to deliver inter-partition traffic in 2C-MSS than both CRAFT and 2C-SpiderWeb since 2C-MSS forms mainly a polygon at each corner of which a partition is positioned as seen in Fig. 10(c) and where the average node degree is thus low as seen in Fig. 12(a).

It is worth noting that the maximum hop count of CRAFT gets very close and matches to that of 2C-SpiderWeb as R grows (\( \geq 160 \) m) albeit CRAFT populates fewer RNs, as shown Fig. 11(a), and yields a lower average node degree, as seen in Fig. 12(a) regardless of R. This is because in CRAFT the primary path for forwarding the inter-partition traffic is formed by computing inward-BT-SPs each of which connects three neighboring border terminals and some partitions will have paths among them without going to the BP. We conclude thus that the created wireless links in the bi-connected topology formed by CRAFT has higher utilization than that of 2C-SpiderWeb which also forms
the bi-connected topology inward towards the center of the area.

In addition, as shown in Fig. 12(d) the growth of \( N_p \) boosts the maximum hop count among partitions for all approaches. This is because the formed inter-partition path by each algorithm in the resulting topology may include another partition which results in involving more hops between partitions as \( N_p \) grows. This analysis is more applicable to 2C-MSS as seen in Fig. 12(d) since the resulting topology generated by 2C-MSS is a polygon formed by partitions. In Fig. 12(d), CRAFT requires fewer links to connect the most distant two partitions in the resulting topology than 2C-MSS regardless of \( N_p \), which is due to the advantage of forming the topology inward toward the center.

**Average node degree per relay**: Fig. 13 shows the effectiveness of CRAFT considering its two conflicting design objectives, i.e., reducing the number of the required relays to bi-connect the partitions while forming an efficient topology in terms of connectivity. Fig. 13(a) shows the average node degree per relay as \( R \) is between 40 m and 120 m with \( N_p = 15 \). Regardless of \( R \), CRAFT forms the most efficient topology per populated relay among all approaches. Recall that a larger value of **average node degree per relay** means higher connectivity is achieved by the same number of relays. In addition, Fig. 13(b) also shows that CRAFT generates a higher node degree per deployed relay in the resulting topology than 2C-SpiderWeb and 2C-MSS when \( N_p \) is larger than or equal to 7. It is worth noting that the **average node degree per relay** in the generated topology by 2C-SpiderWeb is reduced as \( N_p \) grows and becomes less than that of 2C-MSS with \( N_p > 11 \). This is because 2C-SpiderWeb populates more relays among which less links are formed when the number of partitions increases in the area.

### 6. Conclusion

WSNs have been receiving increased attention in recent years due to their versatility in serving applications in inhospitable environments. The nodes operating in these harsh surroundings are thus susceptible to damage, which can be so large in scope that causes the network to get partitioned into disjoint segments. Not only the connectivity in the partitioned network should be restored but also a fault tolerant topology should be established in order to prevent future partitioning due to a single node failure. In this paper we have presented CRAFT, a novel approach that forms a bi-connected network topology by populating relay nodes. CRAFT strives to reduce the required relay count and make the resulting topology efficient with respect to connectivity. It first identifies a backbone ring \( BP \) constructed around at the core of the area of interest where no partitions lies inside and then bi-connect each partition to the \( BP \) based on proximity. The simulation results have demonstrated the effectiveness of CRAFT and highlighted its performance advantages over competing techniques in the literature. Overall CRAFT populates fewer relays than 2C-SpiderWeb to bi-connect a given set of partitions and produces a more efficient topology than 2C-MSS with lower time complexity. In addition, CRAFT generates the bi-connected topology with higher node degree per populated relay than 2C-SpiderWeb and 2C-MSS. The convergence and runtime complexity of CRAFT has also been analyzed. It is proven that 2-vertex disjoint paths exist between every pair of nodes in the resulting topology generated by CRAFT and the runtime complexity is proportional to the number of partitions and inversely proportional to the communication range of a relay.

**Acknowledgements**

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2013R1A1A3004374). This work is also supported by the National Science Foundation, award # CNS 1018171.
Appendix A. Acronyms and terminologies

<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RN</td>
<td>Relay node</td>
</tr>
<tr>
<td>CRAFT</td>
<td>Connectivity Restoration in a Partitioned Wireless Sensor Network with Assured Fault Tolerance</td>
</tr>
<tr>
<td>Pi</td>
<td>Gateway node for a partition i</td>
</tr>
<tr>
<td>R</td>
<td>Communication range of a RN</td>
</tr>
<tr>
<td>Np</td>
<td>Number of segments in a partitioned WSN</td>
</tr>
<tr>
<td>Nken</td>
<td>Number of the required RNs to restore a partitioned WSN and construct a bi-connected topology</td>
</tr>
<tr>
<td>onPGi−1()</td>
<td>A set of Pi's or SPs that lie on the polygon identified in round r which covers all unengaged Pi's or SPs after the round r</td>
</tr>
<tr>
<td>BT</td>
<td>Border Terminal which is Pi or SP that lies on onPGi()</td>
</tr>
<tr>
<td>InPGi−1()</td>
<td>A set of Pi's or SPs that reside inside onPGi−1()</td>
</tr>
<tr>
<td>BP</td>
<td>Backbone Polygon formed inward towards the core of the area of interest inside which no Pi resides, which equals PGf−1, where f is a final round of BP identification phase</td>
</tr>
<tr>
<td>MST</td>
<td>Minimum Steiner tree</td>
</tr>
<tr>
<td>STP-MSP</td>
<td>Steiner tree problem with minimum number of SPs algorithm</td>
</tr>
<tr>
<td>Inward-BT-SP</td>
<td>Steiner point which connects three adjacent BTs with the least cost</td>
</tr>
<tr>
<td>k-LCA</td>
<td>k-restricted loss-contracting algorithm</td>
</tr>
</tbody>
</table>

References

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 Electronics 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 network 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.

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. Younis' 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. Younis 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. Younis is a senior member of the IEEE and the IEEE communications society.

Meejeong Lee received her B.S. degree in Computer Science from Ewha Womans University, Seoul, Korea in 1987, M.S. degree in Computer Science from University of North Carolina, Chapel Hill in 1989, and Ph.D. degree in Computer Engineering from North Carolina State University, Raleigh, in 1994. In 1994, she joined the Department of Computer Science and Engineering at Ewha Womans University, Seoul, Korea, where she is currently a professor. She has been engaged in research in the field of computer communications and networks, and performance modeling and evaluation. Her current research interests focus on protocols and architectures of mobile and wireless networks and future Internet.