Computer Communications xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

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

Exploiting skeletonization to restore connectivity in a wireless sensor network

Q1 Yatish K. Joshi, Mohamed Younis*
Q2
Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD 21250, USA

ARTICLE INFO

Article history:
Received 20 May 2015
Revised 22 July 2015
Accepted 24 July 2015
Available online xxx

Q3 *Keywords:*
Topology repair
Medial axis
Straight skeleton
Fault recovery
Connectivity restoration
Wireless sensor networks

ABSTRACT

Recent technological advances and an increasing thrust toward automation have resulted in a rapid adoption of Wireless Sensor Networks (WSNs) as the de facto choice in monitoring and surveillance applications. Their low cost, versatility and ability to operate around the clock in inhospitable terrains without endangering human life make WSNs ideal for applications like space exploration, environmental monitoring and combat zone surveillance. In these applications WSNs are to operate autonomously for prolonged durations; thus self-healing from failures becomes a requirement to ensure robustness through sustained network connectivity. The paucity of resources makes node repositioning the method of choice to recover from failures that partition the network into numerous disjoint segments. In this paper we present a Geometric Skeleton based Reconnection approach (GSR) that exploits the shape of the deployment area in order to restore connectivity to a partitioned WSN in a distributed manner. GSR decomposes the deployment area into its corresponding two dimensional skeleton outline, along which mobile relays are populated by the surviving disjoint segments to reestablish connectivity. The performance of GSR is validated through mathematical analysis and simulation.

1. Introduction

Decreasing costs and increasing functionality of embedded computation and communication devices have made Wireless Sensor Networks (WSNs) attractive for applications that serve in harsh environments like planetary exploration, border surveillance, environmental monitoring and military surveillance. In these applications, network formation is ad-hoc in nature; for example a swarm of sensor nodes are dropped aerially in the area of interest. After deployment, these nodes are expected to discover and establish communication links with other nodes around them and form a connected network. The limited processing and communication capabilities of the deployed nodes require them to collaborate with one another to carry out application specific tasks. Thus maintaining a connected network topology is of paramount importance for the functioning of a network throughout its lifetime.

The harsh operating environment, the inherent danger in the application area, e.g., bombs in a battlefield, and limited onboard energy supply increase the probability of node failure. Some failures can cause a loss of connectivity and potentially partition the network into disjoint segments. Basically the failure of a single node can cause a network to split into disjoint blocks if such a node serves as a cut-vertex in the network topology [1]. A similar, yet more difficult scenario is when multiple collocated nodes get damaged by an external event, e.g., an explosion, flooding, sand storms, etc. Given the importance of data sharing in achieving the application goals, sustaining connectivity is critical for network operation. Therefore a network must have the ability to tolerate the occasional failure of nodes and restore connectivity without relying on external resources, e.g., remote command center, to coordinate recovery.

Tolerance of failure: Strategies for failure recovery depend on the scope of failure and the node capability [1]. The scope of failure is defined by the multiplicity of affected nodes and their location. The failure of a single node is the easiest to handle. However, the failure of multiple nodes is a major challenge, particularly when the nodes are collocated as a major void is caused and the network becomes fragmented into disjoint segments. In addition, unlike the failure of a single node or even the failure of multiple dispersed nodes, it is difficult to determine the scale of the damage if multi-collocated nodes fail. Basically, a healthy node will not be able to determine whether it lost contact with other parts of the network because a single neighbor failed, i.e., a node that acts as a cut vertex in the topology is lost, or due to the failure of multiple collocated nodes.

Tolerance mechanisms can be classified as proactive or reactive. The former is based on provisioning redundant resources at network setup in order to mitigate failure, e.g., by establishing a k-connected topology [2,3], or providing backups for faulty nodes [4,5]. Obviously

* Corresponding author. Tel.: +1 410 455 3968; fax: +1 410 455 3969.
Q4 *E-mail addresses:* yjoshi1@umbc.edu (Y.K. Joshi), younis@cs.umbc.edu, younis@umbc.edu (M. Younis).

http://dx.doi.org/10.1016/j.comcom.2015.07.022
0140-3664/© 2015 Published by Elsevier B.V.

such methodology is not suitable for ad-hoc networks since the randomness of deployment, e.g., aerial deployment of nodes cannot guarantee node placement with the required precision and would thus require excessive resources, e.g., number of redundant nodes, to achieve k-connectivity.

Reactive strategies are based on reconfiguring the network topology to deal with failure. They restore connectivity by forming an inter-segment topology or by regrouping surviving nodes. Reactive approaches can be broken down into two main classes centralized and distributed. A centralized approach assumes knowledge of the entire network state. This global state information is utilized to optimally allocate the existing resources and coordinate recovery. Meanwhile distributed recovery schemes operate based on local state and trade off optimality in terms of desired topology features for reduced coordination overhead and responsiveness. While quite a few distributed algorithms have been proposed for tolerating single node [6,7] and non-collocated multi-node failures [8], most solutions for tolerating the failure of multiple collocated nodes are centralized and often pursue the placement of additional relays to form a connected inter-segment topology [1]. While ideal for handling simultaneous multi-node failures, a centralized approach needs to know the scope of failure, locations of surviving segments and nodes before it can begin recovery. Thus, the applicability of a centralized approach depends on the availability of external sources, i.e., satellite imagery, aerial support from drones or UAVs to collect and disseminate global network state information on demand. Such external support may not be available at all times in ad-hoc WSNs due to the harsh operating environment, resource constraints or technical difficulties. This makes distributed approaches the most suitable choice for recovery.

Distributed reactive strategies utilize the surviving nodes to recover from node failures. Most published recovery schemes in this category, e.g. [6–8], can deal with only a single node failure or multiple non-collocated failures as they rely on the neighbors of a failed node to restore connectivity by utilizing 1-hop or 2-hop information to either move inwards in the direction of failure until connectivity is restored or by moving a redundant node to the failed node's location. These schemes however do not scale if the failure spans multiple collocated nodes since the surviving nodes do not have enough information to determine the scope of failure and appropriately plan the recovery. Distributed approaches like DarDs [9] and DORMS [10] handle recovery from multiple collocated node failures by provisioning a common meeting point before failure that serves a point of convergence for all disjoint segments and is used to restore initial connectivity.

Contribution: We present a novel distributed Geometric Skeleton based Reconnection (GSR) approach that restores connectivity after the failure of multiple collocated nodes. Our approach is motivated by the fact that the WSN topology affects its operation and in practice it is influenced by the terrain and environmental factors at time of deployment. Unlike other published schemes we aim to factor in the pre-failure network topology in our recovery strategy and provision a plan that can be implemented after failure. We argue that the use of a common meeting point imposes increased overhead and may slow down convergence. Using a geometric skeleton will enable efficient handling of failure in any part of the network by allowing surviving nodes from the individual segments to reach each other faster and facilitate localized tolerance of failures that are far from any common meeting point. It is worth noting that GSR may be applied in some applications, which can be viewed as mobile ad-hoc networks (MANET). An example of that includes a networked set of robots that participate in a combat or reconnaissance mission. We argue, however, that most MANETs involve coordination among the nodes to deal with broken links caused by node mobility and not due to the failure of multiple collocated nodes.

Fig. 1 illustrates our strategy. Given an ad-hoc network Fig. 1(a) we utilize the WSN boundary to decompose the network into its constituent geometric skeleton Fig. 1(b). A geometric skeleton is a descriptor that decomposes a shape into its essential support structure, e.g., bone structure in humans, based on how components are connected. This skeleton is stored by nodes within the network and serves as a backbone, along which mobile nodes can be deployed by disjoint segments after failure, Fig. 1(c), in order to find other survivors and reestablish network connectivity, as shown in Fig. 1(d). Thus by exploiting the pre-failure network topology, GSR provisions a recovery plan that can be independently implemented by the surviving segments to restore connectivity in the network while reducing the recovery overhead. We consider two types of skeletons, namely the straight skeleton [12,13] and the medial axis [14] described in later sections, categorize their impact on recovery, highlight their differences and provide guidelines for choosing the best skeleton for GSR. GSR is validated through extensive simulation experiments and is shown to outperform competing schemes both in terms of the travel distance overhead and the number of involved nodes. The paper is organized as follows. The next section sets GSR apart from existing solutions in the literature. GSR consists of two phases; the first takes place before a failure takes place and is described in Section 3; the second phase is for restoring connectivity in reaction to failure and is detailed in Section 4. The performance of GSR is analyzed in Section 5 and is validated through simulation in Section 6. Finally, the paper is concluded in Section 7.

2. Related work

As pointed out earlier, strategies for tolerating node failure can be classified based on the scope of failure into single node and multi-node failures. The latter can be further categorized into collocated and non-collocated failures. To tolerate a single node failure, most distributed approaches in the literature pursue relocation of nodes and rely on local neighborhood information stored by neighbors of the failed node to initiate recovery [6,7]. When multiple non-collated nodes fail, these techniques may cause resource conflicts, e.g., engage a node in more than one failure recovery. Some approaches, e.g. [8], avoid resource conflicts by synchronizing the various recovery actions. However even these approaches cannot be scaled to handle multiple collocated node failures, since a node would have to maintain state information spanning many hops, in fact it should have the entire network state available in order to avoid conflicts. Thus, they are deemed ineffective as the messaging and storage overhead required to maintain a multi-hop network state increases exponentially with network size. Given the focus of the paper, the remaining part of this section covers tolerance of multiple collocated node failures. A survey and detailed analysis for the recovery schemes for single and multiple non-collocated node failures can be found in [1].

Strategies for repairing a network topology after multiple collocated nodes failure can be classified into centralized [16–18] and distributed [9,10,19]. Centralized schemes utilize relays to form a connected inter-segment topology and re-establish communication paths between the disjoint segments. These relays may be new nodes or simply existing nodes whose repositioning does not seriously impact the intra-segment topology. Since the entire network state is factored in, centralized recovery schemes provide the best solution in terms of metrics like the number of relays deployed and the total distance traveled by them during the recovery process, if existing relays relocate as part of the solution. However, centralized schemes cannot be applied in all scenarios since their applicability is dependent on the entire network state being known after failure, which may not be feasible as pointed out earlier.

Distributed schemes are the solutions of choice for autonomously-operated ad-hoc WSNs, e.g., those serving remote or inhospitable areas such as space exploration or military reconnaissance. The general methodology in this case is to utilize mobile nodes that exist in the network. Published schemes can be classified based on the node mix

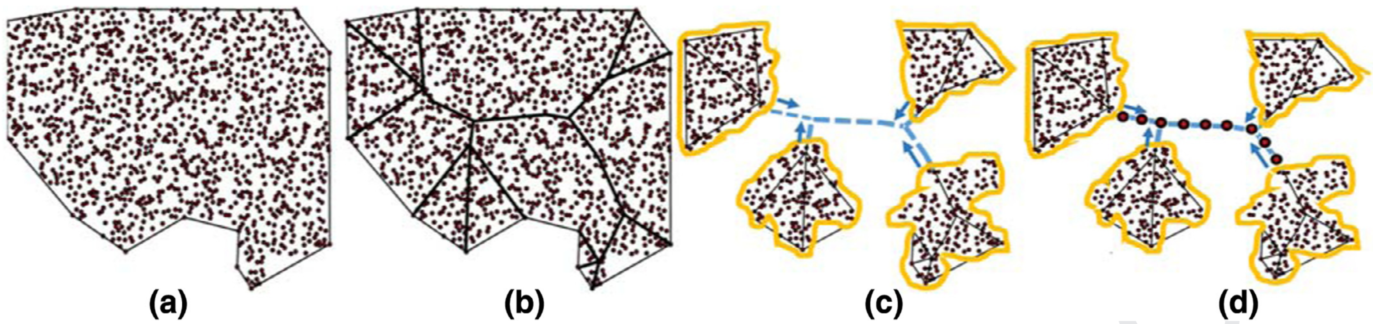


Fig. 1. Overview of GSR. (a) A ad-hoc WSN. (b) The skeleton (black lines) of the WSN. (c) WSN partitioned into disjoint segments after failure. (d) Mobile Nodes populated along the skeleton to reestablish connectivity.

in the network. AuR [19] is specifically tailored for networks composed solely of mobile nodes. Surviving nodes utilize 1-hop neighborhood information to determine the direction of failure and pursue node self-spreading and movement inwards toward the center of the deployment area to reconnect the network. On the other hand networks with a mix of stationary and mobile nodes establish links amongst the disjoint segments. Recovery schemes that establish an inter-segment topology can be categorized based on the role mobile nodes play in the recovery into: (1) stationary relays between disjoint segments to act as a gateway or provide a stable path between segments [9,10], and (2) mobile data mules that provide intermittent connectivity by traveling between segments and transferring data [21,22]. The latter is used by approaches like MiMSI [23] when insufficient mobile nodes are available. Basically, MiMSI solves a constrained version of the federation problem wherein it aims to connect *N* disjoint segments or terminals when fewer nodes are available for recovery than what is required to form a stable inter-segment topology. Unlike our GSR approach, MiMSI assumes that the segment positions are known beforehand and does not solve the segment discovery problem.

Since the location or scope of failure is not known to the surviving nodes, i.e., a segment does not know where the other segments are, forming a star shaped inter-segment topology around the center of the area has been deemed a safe approach in order to ensure convergence. Basically, each segment populates mobile nodes toward the center of the deployment area in order to reconnect with other segments. DarDs [9] and DORMS [10] employ this methodology, but differ in how they optimize the inter-segment topology after connectivity has been restored. Unlike DORMS, DarDs realizes that representative relays from different disjoint segments may come in contact with each other while moving inwards toward the center, and consequently declares the respective segments connected if they come in contact with one another and merges their paths to the center. Once a center-connected topology is established, both DORMS and DarDs run optimization heuristics, namely, k-LCA [20] for DORMS and IODT [18] for DarDs to minimize the number of mobile relays deployed to sustain connectivity.

Both DORMS and DarDs utilize the center as a meeting point to guarantee convergence. This tactic though leads to extra relays being deployed, especially, when the disjoint segments are not evenly spread out in the deployment area. In other words, the shape of the deployment area is not exploited and segments that are physically near to each other may not get connected until their representatives reach the center. Also these approaches do not take into account the unique challenges that come with ad-hoc deployment, they model the deployment area as a simple square geometric region that contains no holes/obstacles. This simplistic model of the network boundary can lead to a poor choice of center resulting in degraded performance in practical scenarios. Moreover, center-based approaches also run an optimization phase to reduce the number of deployed relays once initial connectivity is reestablished. This step requires the location of all deployed relays to be known and is in essence a centralized operation with a relatively high runtime complexity. The relocation of relays to their new positions increases the total travel cost and adversely impacts the network lifetime.

Our proposed approach avoids these shortcoming by using the geometric skeleton as the inter-network connectivity structure. Disjoint segments populate mobile nodes along the stored skeleton to discover one another. The recovery is complete when the skeleton paths are populated with nodes. Our preliminary results [11] have demonstrated the effectiveness of our approach for convex deployment areas. This paper generalizes the approach to address all shapes of the deployment areas.

3. Skeleton construction

GSR consists of two phases, namely, skeleton construction, and mobile node deployment. This section focuses on the first phase, which is applied before a failure takes place. Before describing such a phase, we discuss the system model.

3.1. System model

GSR considers an ad-hoc WSN randomly deployed in an area of interest and is assumed to be composed of a mix of stationary and mobile nodes (MNs), or all MNs. The network is assumed to serve in-situ users who cross-by, i.e., show up from time to time and does not have a stationary base-station that acts as a sink for all network traffic. The network boundary is assumed to be available to the nodes in the WSN through the help of a satellite or an aircraft at the time of deployment or via the implementation of boundary detection techniques [34,35]. The network boundary is utilized to model the WSN as a simple polygon which can be defined as a sequence of '*n*' points (x_0, $x_1,\ldots, x_{n-1}$) such that $x_i\, x_{i+1}$ for $i = 0, 1,\ldots, n-1$ forms an edge and no two nonconsecutive edges intersect. The closed polygon divides the plane into interior and exterior regions. Any holes, e.g., lakes, present within the network boundary are modeled in the same manner.

This boundary determination needs to be done only once at network setup (before any failure takes place) and can be updated after tolerating a failure to reflect the new network topology. The network boundary is utilized to construct the skeleton of the network and this procedure is explained in detail in the next section. We assume that each MN is aware of its position, e.g., using contemporary localization schemes. GSR also assumes that all disjoint segments in the damaged WSN have sufficient MNs to participate in the recovery process. Node failures are detected through missed heartbeat messages and inability to reach parts in the network [21].

3.2. Determining the geometric skeleton

Being distributed, GSR assumes no prior knowledge about the location of surviving segments in a partitioned network; each segment

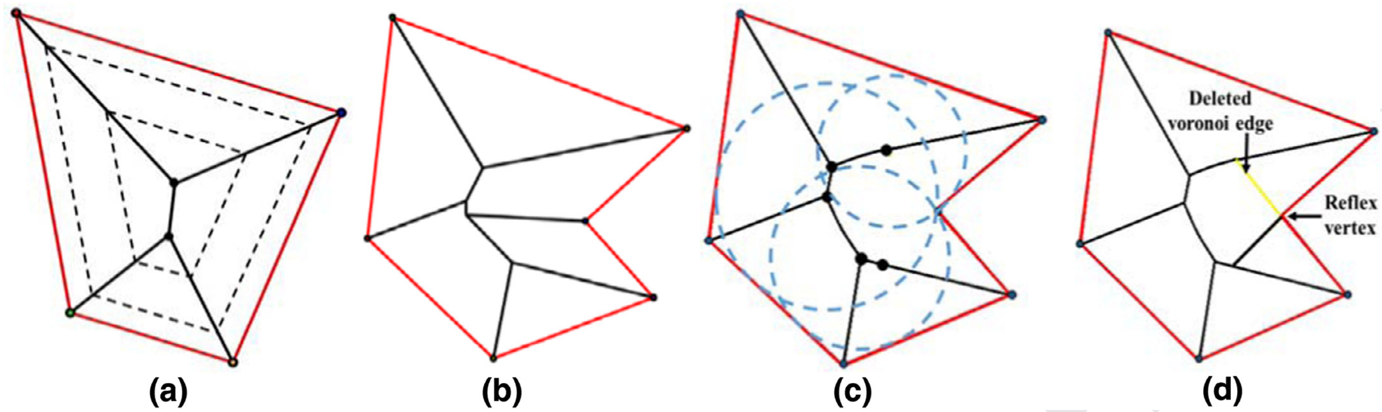


Fig. 2. Geometric Skeleton construction. (a) The dotted lines represents the polygon shrinking inwards. The lines traced by the inwards motion comprise the straight skeleton and medial axis. (b) Straight skeleton of a non-convex polygon. (c) Medial axis (depicted in black) is the locus of the centers of circles that are tangent to two or more polygon edges. A subset of circles (in blue) are depicted above. (d) Medial axis construction from a Voronoi Diagram. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article). Q5

independently runs the GSR algorithm. To restore connectivity a segment must determine the direction in which it should populate its mobile nodes so that it can reconnect with other surviving segments. Competing approaches [9,10] utilize the center of the deployment area as a rendezvous point to guarantee convergence. GSR does not require a fixed rendezvous point; instead it exploits the shape of the deployment area to generate travel paths along which mobile nodes can be deployed to restore connectivity when a major failure partitions the network.

GSR employs two types of geometric skeletons, namely, the straight skeleton and the medial axis, as follows:

(i) The straight skeleton is defined as the set of lines traced by the polygon vertices as the polygon edges are continuously shrunk inwards at a constant speed parallel to themselves. Fig. 2(a) showcases this shrinking process (dotted lines represent the polygon shrinking). The set of lines traced by the vertices during the inward motion form the straight skeleton. Fig. 2(b) shows the straight skeleton of a non-convex polygon.

(ii) The medial axis of a simple polygon P can be defined as a set of points $\{q\}$ that lie within P such that there exists at least two points on the polygon boundary equidistant from $\{q\}$ and that are also the closest to $\{q\}$ as shown in Fig. 2(c). The medial axis can also be obtained from the Voronoi diagram of the vertices and edges of the simple polygon by removing edges that are incident on the reflex vertices of the polygon [15]. A vertex is reflex if the internal angle it subtends is greater than π. Fig. 2(d) shows the Voronoi diagram for a simple polygon and the two edges that need to be deleted incident on the reflex vertex to obtain the medial axis. As seen in Fig. 2(d), the medial axis skeleton (in black) used by GSR consists of line segments and parabolic arcs that are introduced due to the presence of reflex vertices. GSR requires one skeleton line to originate from each polygon vertex to serve as a travel path. A reflex vertex is a special case for which there are two Voronoi edges incident on a vertex. Of the two Voronoi edges shown in Fig. 2(d), the longest edge in yellow is deleted and the shortest becomes a part of the final skeleton.

Next we discuss the formation of the straight skeleton and medial axis for convex and non-convex polygons.

- Convex Polygons: The straight skeleton and medial axis result in identical skeletons for convex polygons, and can be determined in linear time [24,25].
- Non-Convex Polygons: A non-convex simple polygon has reflex vertices. If the shape of the deployment area is non-convex we have two options: (1) construct a convex hull approximation of the network boundary as seen in Fig. 4 and form a straight or medial axis skeleton, or (2) generate the skeleton over the non-convex polygon as shown in Fig. 2(b) and (d). The first option has the advantage of constructing the skeleton in linear time and will thus keep the time complexity low. However, this option is unsuitable since the skeleton derived from the convex hull no longer bears a strong correspondence to the shape of the network. Also as seen in Fig. 4, such an approximation can cause parts of the skeleton (blue lines) to lie outside the original network boundary (highlighted in red) which causes nodes in the repaired topology to traverse and be deployed in an undesired or unrequired area. Therefore, GSR aims to determine the skeletons over the non-convex network boundaries to preserve locality information and ensure that nodes remains within the original deployment region while restoring connectivity.

Fig. 3 showcases the effect that holes within the deployment area have on the medial axis and straight skeleton. The shape of the skeleton is influenced by the network boundary and the interior whole boundary. As seen in Fig. 2, unlike the convex case the medial axis and straight skeleton for non-convex polygons result in different skeletons. This difference is caused by the way reflex vertices are handled and is explained in detail in Section 5. Distributed approaches like [32] can also be used to compute the medial axis. Once computed, the skeleton is stored by all nodes in the network as a set of lines defined by their (x, y) coordinates. The stored skeleton lines serve as paths along which mobile nodes are populated to recover the network after failure.

4. GSR-based recovery

Upon the detection of collocated multi-node failure, the second phase of GSR is applied. Fig. 5 shows an example WSN deployment area and its associated skeleton. Lines that originate from polygon vertices are known as contour lines. The point at which contour lines intersect is known as a skeleton vertex. Lines that connect two skeleton vertices together are known as skeletal lines. These skeleton lines, both contour and skeletal, serve as paths along which mobile nodes are populated.

4.1. Mobile nodes deployment phase

Once failure is detected, surviving nodes use heartbeat messages to discover others around them to form segments and coordinate recovery; henceforth a segment refers to a connected set of nodes in a

Please cite this article as: Y.K. Joshi, M. Younis, Exploiting skeletonization to restore connectivity in a wireless sensor network, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.07.022

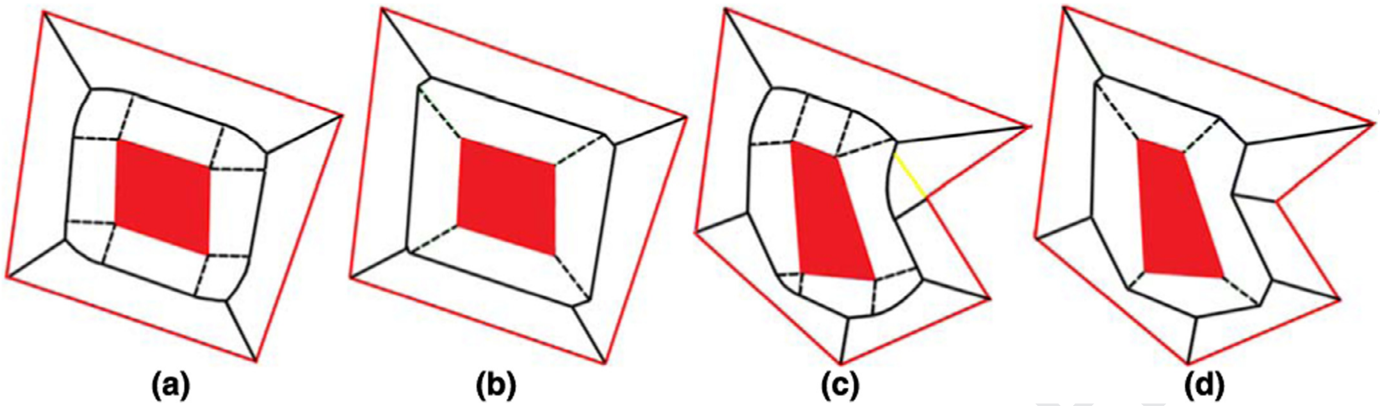


Fig. 3. Effect of obstacles (shaded in red), within the deployment area on the geometric skeleton (depicted by black lines). (e) Medial axis of a convex polygon. (b) Straight skeleton of a convex polygon. (c) Medial axis of a non-convex polygon. (d) Straight skeleton of a non-convex polygon. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

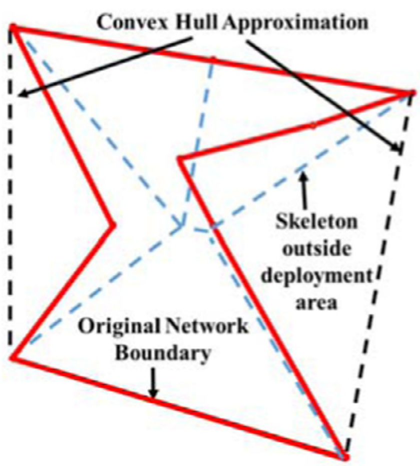


Fig. 4. Approximating the network boundary by a convex hull and the resulting skeleton.

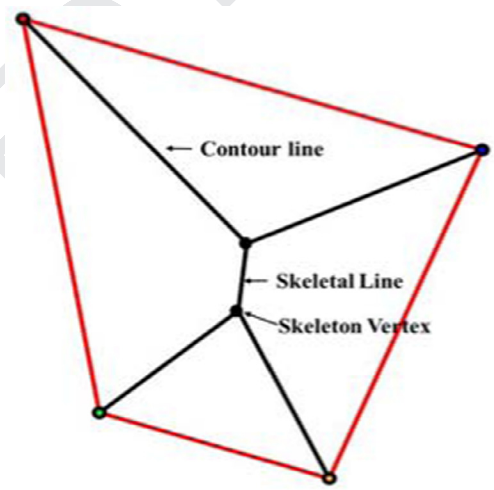


Fig. 5. A deployment area and its associated skeleton.

geographical region within the network boundary. Each segment initiates the recovery process and applies GSR autonomously. A segment first picks a representative MN that spearheads the recovery process; such a representative is followed by other MNs in a cascaded manner to stay connected to the originating segment. It is assumed that the segment stays connected despite the departure of MNs, e.g., by pursuing cascaded relocation within the segment [7,8]. The intra-segment MN selection process can also take into account the residual energy of the MN to ensure balanced power consumption. To determine where to populate MNs, the segment first determines its position relative to the stored skeleton lines. Segments fall into two categories: (1) contour segments: those that lie on or have a contour line nearest to them; (2) skeletal segments: those that lie on a skeletal line or have one closest to them.

Contour segments populate mobile nodes inwards along the contour line toward the skeleton vertex in hope to connect with other disjoint segments located in the interior as shown in Fig. 6(a). If the nearest line is a contour line the segment drops a perpendicular to it and populates mobile nodes along the perpendicular and on reaching it, inward towards the skeleton vertex as shown in Fig. 6(b). This represents the best chance for a segment to find other survivors if they exist in that region as they will also move along the contour line. Contour segments do not populate outwards along their own contour line since any surviving segments upstream will rendezvous with the interior segments eventually as they move inward towards the skeleton vertex. Therefore motion along the contour line is strictly one way, inward toward the skeleton vertex. Contour segments will keep populating MNs inward until they cover their contour lines and their associated skeletal lines or come in contact with nodes from another segment. Once the leading MN of a contour segment reaches a skeleton line its motion planning choices are like that of skeletal segments described below.

For skeletal segments seen in in Fig. 6(c) and (d), we have the same two scenarios where either the segment lies on the skeletal line or if the nearest line is a skeletal line the segment drops a perpendicular to it and populates MNs to reach the skeletal line. In this case unlike the contour line we are faced with a choice, we can proceed to either end points *A* or *B* of a skeletal line since both end points are skeleton vertices. Since GSR operates in a distributed manner, a segment has no idea in which direction it is likely to find a survivor so it pursues a greedy approach and sends out representatives MNs in the direction of the nearest skeleton vertex, which in this case is vertex *A* in Fig. 6(c). Upon reaching vertex *A*, the representative checks whether there are any other skeletal lines that originate from vertex *A* and keeps populating MNs along those lines. Once all the skeletal lines are exhausted, due to either reconnecting with another survivor coming from the opposite direction or if all the lines ending at vertex *A* were contour lines, it is concluded that no further motion is possible in the current direction and the segment will take the option to populate toward vertex *B*. It is worth noting that such MN travel pattern can also enable the discovery of segments that have insufficient MNs and thus integrate them in the repaired network topology. All

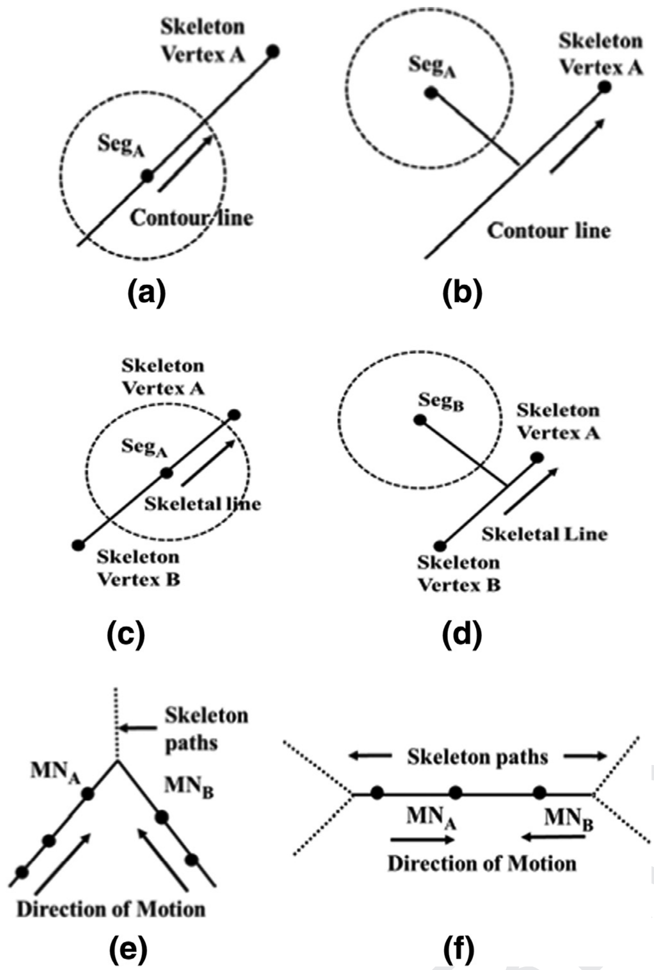


Fig. 6. Motion planning for segments after failure using the stored skeleton.

these decisions on which line to populate along are taken by simply comparing with the set of stored skeleton paths determined before failure. Unlike center-based approaches, by using the stored skeleton lines, the disjoint segments can make intelligent choices for populating MNs and increase their likelihood of merging with neighboring segments along the way.

During recovery, if the representatives MNs from different surviving segments come into contact with each other, the segments will merge and combine their resources. In case they are moving in the same direction then the segment whose MN is further along the designated path will become the leader of the merged segment in that direction and lead the recovery. As seen in Fig. 6(e) MN_A and MN_B representatives of two differing segments come into contact with one another while proceeding in the same direction. MN_A becomes the leader since it's further along the path than MN_B. In case if they are moving towards one another like in Fig. 6(f), then each will disregard the skeleton lines in the opposing direction since there is a segment already present in that direction.

Thus exploiting the locality information, provided by the skeletal structures and intelligent motion planning enable GSR to efficiently restore connectivity with reduced overhead, as will be shown through simulation in Section 6.

4.2. Illustrative example

Fig. 7 illustrates the application of GSR when the WSN shown in Fig. 5 gets partitioned into 6 disjoint segments. After failure, the only information each segment has is the skeleton outline of the deployment area determined at network setup. Each segment finds its nearest skeleton line. As seen in Fig. 7(a), there are 4 segments Seg_0, Seg_1, Seg_2 and Seg_3 lying along the contour lines. Each is represented in the figure by a distinct color and their deployed MNs will be of the associated segment color. They each deploy MNs along their respective contour lines inward toward their respective skeleton vertices. Contour lines are only populated inward since if there are any survivors outward their representatives will come in contact with these interior segments during their inward motion. We have two segments Seg_4 and Seg_5 that do not lie on contour lines but are contour segments on account of being near to a contour rather than skeletal line. Seg_4 and Seg_5 each drop a perpendicular toward their respective contour line and starts populating MNs in that direction. As seen in Fig. 7(a) the nearest skeleton line for Seg_4 is the contour line containing Seg_0 and for Seg_5 is the contour line containing Seg_3.

The representative MNs sent out by both Seg_0 and Seg_4 come within transmission range of each other during their motion, causing both segments to merge, Seg_0 remains in charge of the merged segments on account of having a MN closer to the skeleton vertex. The representative MN from Seg_0 continues toward the skeleton vertex while keeping the Seg_4 representative within its transmission range. Seg_5 moves along the perpendicular toward the nearest contour line and along it. Fig. 7(b) shows the snapshot of the network after time 't', required by the MNs to move a distance R. In Fig. 7(c), Seg_1, Seg_2, Seg_3 and the merged Seg_0-Seg_4 all populate MNs as they have unexplored contour paths and they need to maintain contact with their leading MN representative which are R unit away after the first iteration. During this motion the representative MNs from Seg_2 and Seg_3 both come in range of Seg_5 and stop; they do not proceed any more as the representative from Seg_5 is further along the path than them.

Meanwhile the representative of Seg_5 is still within the transmission range after traveling along the perpendicular to reach the contour line. It moves and stops at the skeleton vertex which is 'R' away from the segment. The skeletal line is the only path left to explore but its length is less than the MN communication range and there are no more skeletal paths left in that direction. Therefore, Seg_5 does not deploy any more MNs. If there are any surviving segments in that direction once their MNs reach the opposite skeleton vertex they will come into contact with Seg_5 which is now a merged segment consisting of Seg_3-Seg_5-Seg_2. On the other side MNs from the merged Seg_0-Seg_4 segments and Seg_1 come in contact, their tie is broken by checking which has a MN nearer to the skeleton vertex where their respective contour lines intersect; accordingly the MN from Seg_0-Seg_4 continues as it is nearer to the skeleton vertex. The representative from the merged Seg_0-Seg_4 continues onward until it comes within range of a MN from Seg_5. At this point the recovery process ends as there are no more skeleton lines left for the MNs to explore.

Overall, the GSR recovery process utilizes 10 mobile nodes in this example. Fig. 7(d) shows the connected topology generated by the DarDs approach during its initial deployment phase. DarDs populates MNs along the line connecting a segment to the center of the deployment area, with MNs being deployed until one reaches the center or merges with a segment while doing so. In this example, DarDs utilizes 11 MNs to reconnect the damaged topology. Seg_3 and Seg_5 are geographically close to each other but in DarDs do not meet until becoming 1-hop away from the center. Thus it is clear that moving along skeleton paths gets the locally near segments connected much quicker with each other wherein they can pool resources while exploring the remaining skeleton paths.

5. Analysis

This section analyzes and contrasts the performance of the medial axis with the straight skeleton based GSR and looks at the time complexity in skeleton construction and storage overhead imposed by

Please cite this article as: Y.K. Joshi, M. Younis, Exploiting skeletonization to restore connectivity in a wireless sensor network, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.07.022

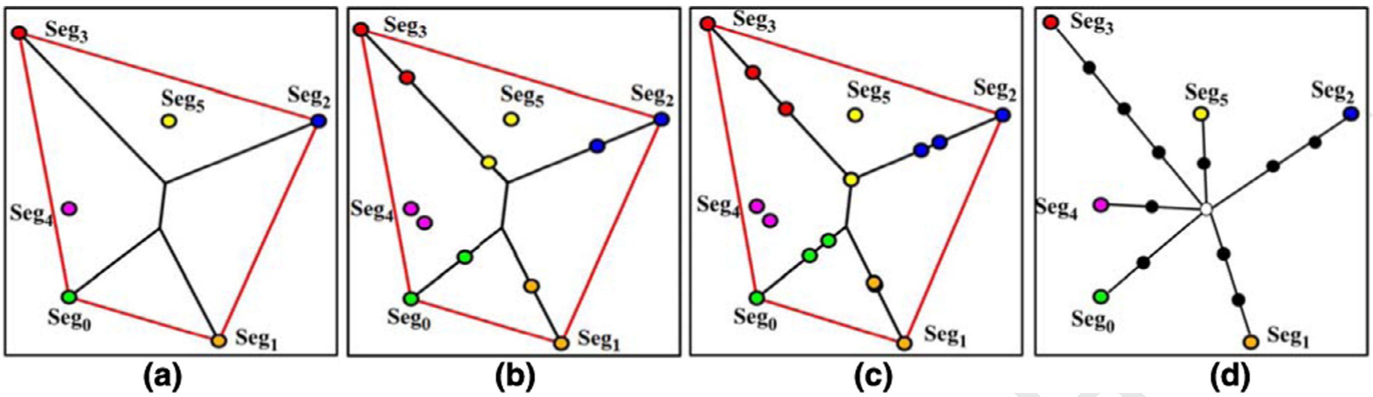


Fig. 7. Application of GSR. (a) Six disjoint segments of a damaged WSN. (b) Network after first MN Deployment, MNs share same color as their respective segments. (c) Final Recovered topology. (d) Center connected topology constructed by competing DarDs approach.

```
V ← set of  nodes in the network.
B ← set of edges comprising the network boundary.
L ← set of Lines forming the geometric skeleton
n ← No of disjoint segments after failure
R ← Relay Transmission Range

DetermineNetworkBoundary(V);
If UseStraightSkeleton
  L= ConstructStraightSkeleton(B);
Else
  L= ConstructMedialAxis(B);

DistributeSkeletonLines(L);
If Failure for each n
DetermineneaarestSkeletonLine(L)
WHILE (L exists)
    Populate Mobile Nodes Along L;
    Decrement Length(L) by R;
END WHILE

ConstructMedialAxis(B)
 - Form Simple Polygon using network boundary B;
 - Construct Voronoi Diagram
 - Extract Medial Axis from Voronoi;
RETURN

ConstructStraightSkeleton (B)
 - Form Simple Polygon using network boundary B;
 - Determine Straight skeleton;
RETURN

DistributeSkeletonLines (L)
- Distribute skeleton as (x,y) coordinate pairs;
```

Fig. 8. Pseudo code for GSR Algorithm.

the GSR approach. The pseudo-code of the GSR algorithm is shown in Fig. 8.

Theorem 1. *A straight skeleton for a simple polygonal deployment area with n vertices will have (n-2) internal skeleton vertices and comprise of (2n-3) lines.*

Proof. The skeleton consist of contour and skeletal lines. There is a contour line emanating from each polygon vertex resulting in a total of *n* contour lines. Based on [12], there are *(n-2)* skeleton vertices in a straight skeleton. The fewest number of lines to interconnect these vertices is *(n-3)*, by forming a minimum spanning tree. Thus, a straight skeleton is comprised of *(n+n-3=2n-3)* lines. This is the number of lines that are to be stored by the network nodes for recovery. □

Theorem 2. *A straight skeleton for n vertices constitutes a connected topology with (2n-3) edges.*

Proof. In [12], a straight skeleton is proven to be a tree that divides a polygon into *n* partitions, where *n* is the number of vertices. From Theorem 1, the straight skeleton consists (*2n-3*) lines. Thus, the straight skeleton is a connected topology with (*2n-3*) edges. □

Theorem 3. *A straight skeleton of a simple polygonal deployment area having n vertices and r reflex vertices can be constructed in* $O(n\log^2 n + r\sqrt{r}\log r)$ [30].

Theorem 4. *In GSR the medial axis for a simple polygonal deployment area with n vertices and r reflex vertices will have at most (n+r-2) internal skeleton vertices and comprise of (2n+r-3) lines.*

Proof. We know that the medial axis is defined as the set of Voronoi edges minus the set of edges incident on the reflex vertices. The Voronoi diagram of a simple polygon with *n* vertices and *r* reflex vertices has at most *2(n+r)-3* Voronoi edges and *(n+r-2)* Voronoi vertices [15]. In GSR, since we require a contour line to originate from each polygon vertex to serve as a recovery path, we delete only the longer of the two edges incident on a reflex vertex giving us *r* less edges than the Voronoi diagram resulting in *(2n+r-3)* skeleton lines and *(n+r-2)* internal skeleton vertices. Alternatively we can also say that contour lines will originate from all vertices, resulting in *n* contour lines. From [15], there will be at most *(n+r-2)* internal skeleton vertices; these vertices can be interconnected by a minimum spanning tree of (n+r-3) edges. So the total number of lines generated are (n+n+r-3=2n+r-3) lines. □

Theorem 5. *GSR construct the medial axis skeleton of a simple polygonal deployment area in O(n), where n is the number of vertices* [31].

From Theorems 2 and 5 we see that for simple non-convex polygons, the straight skeleton construction has a higher time complexity than the medial axis. This is due to fact that unlike the medial axis it cannot be obtained from an abstract Voronoi diagram [29] hence standard computational geometry techniques cannot be applied to speed up construction.

Theorem 6. *The medial axis formed by GSR for a simple polygonal area of n vertices constitutes a connected topology with (2n+r-3) edges.*

Proof. The medial axis is derived from the Voronoi diagram which has the property of being path connected [15], i.e., for any two points *u* and *v* in a Voronoi polygon $V(e_i)$, there exists a path connecting *u* and v that is totally contained in $V(e_i)$ where e_i is a polygon edge. The medial axis contains all the Voronoi vertices joined together by Voronoi edges, it is a tree having n leaves originating from the polygon vertices, and $(n+r-2)$ internal skeleton vertices, i.e., a total of $2n+r-2$ vertices that are connected through $(2n+r-3)$ skeleton edges. □

Theorem 7. *The storage overhead of GSR is linear in the number of polygon vertices.*

Proof. From Theorems 1 and 4 it is clear that the number of skeleton lines to be stored in GSR grows linearly with the number of vertices that lie on the network boundary for both geometric skeletons. □

It is worth noting that in case of storage constraints, the number of vertices comprising the deployment boundary can be reduced by removing collinear points and carrying out localized reductions [33] or contour approximations.

Theorem 8. *In GSR the mobile node deployment phase is linear in the number of vertices of the deployment area.*

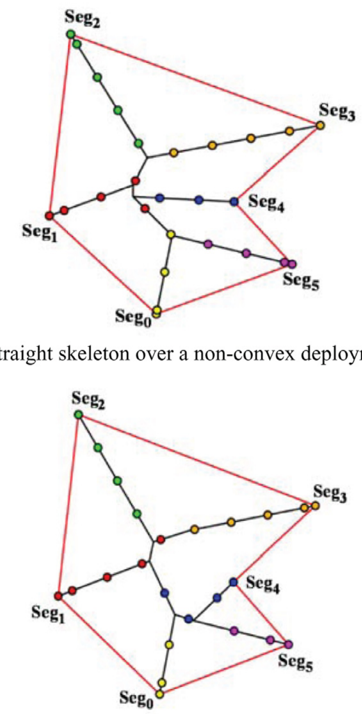
Proof. As discussed in the previous section, the GSR based recovery is deemed complete when all skeletal lines (lines joining the skeleton vertices) are occupied by MNs. In the worst case scenario, if there is only one surviving segment say on a contour line then in that case it will deploy mobile nodes along its contour line and all skeletal lines. This means that for a straight skeleton based GSR n-2 skeleton lines (1 contour line and n-3 skeletal lines) will be explored; in case of the medial axis based GSR $n+r$-2 skeleton lines (1 contour line and $n+r$-3 skeletal lines) will be explored. Hence, the number of lines along which nodes are to be deployed is linear in n. □

Theorem 9. *GSR successfully restores network connectivity if there are sufficient MNs.*

Proof. To prove this theorem, it is sufficient to show that GSR forms a tree whose non-leaf edges are fully populated with MNs that are R units or less apart, where R is the communication range of a mobile node. Based on Theorems 2 and 6, the skeletons formed by GSR are trees. In addition, as pointed out in the proof of Theorem 8, all non-leaf edges are populated with MNs even under the worst case scenario where there is only one surviving segment. Thus, by placing MNs on the skeletal lines and connecting all segments to these skeletal lines, a connected topology will be formed. □

Effect of reflex vertices: As we have seen the straight and medial axis skeletons differ for a non-convex shaped deployment area due to the way they are being formed. During the medial axis construction the interaction between a boundary line and a reflex vertex introduces a parabolic arc that is a locus of points equidistant from the vertex which acts as the focus and the directrix line. Whereas in the case of the straight skeleton no parabolic arcs are introduced since all edges shrink inwards at the same speed, with each vertex following its angle bisector. The skeleton lines are straight because they are angle bisectors. The presence of sharp reflex vertices though can result in a skewed skeleton as seen in Fig. 9, since a reflex vertex moves further inward the sharper the angle before intersecting with another angle bisector. This skewed skeleton results in longer skeleton paths since the intersection of angle bisectors happens further in the interior of the polygon as compared to the symmetrical skeleton obtained for the medial axis. For example in the WSN shown in Fig. 9, the straight skeleton based GSR utilizes 20 mobile nodes to restore connectivity while the medial axis based solution requires only 18 mobile nodes. This improvement can be attributed to the symmetrical skeletal structure generated by the medial axis.

So with a higher run time complexity it would seem that the straight skeleton is inferior to the medial axis and GSR should stick



(a): Straight skeleton over a non-convex deployment area

(b): Medial axis over a non-convex deployment area

Fig. 9. A comparison of the skeletons generated by the straight skeleton and the medial axis for a non-convex deployment area.

with the medial axis skeleton. The attractiveness of the straight skeleton based GSR lies in the fact that the skeleton is composed of only straight lines and does not comprise of curved arcs if there are reflex vertices present as seen in Fig. 2. The straight skeleton may be favored if it is used for other application-related purposes or if having straight lines is a requirement. Note that for the sake of simplicity, in the medial axis based GSR we can approximate the parabolic arcs introduced by reflex vertices with straight line segments or alternatively use the linear axis introduced by Tanase and Veltkamp [28] which gives a linearized approximation of the medial axis in linear time. The accuracy of the approximation is controlled by a factor k; the higher the value of k, the finer the approximation will be.

6. Performance evaluation

The effectiveness of GSR is validated through simulation. This section discusses the simulation setup, performance metrics and results. The experiments are conducted in a 1500 m × 1500 m square area where random topologies are generated for varying number of segments (5 to 20) and communication range (50 to 100 m) for a mobile node. In the experiments all mobile nodes have the same communication range R. To ensure correct and precise mathematical computation, the CGAL computation library [26] was used to construct the straight skeleton, medial axis and to carry out other geometric computations. The results obtained show the 90% confidence interval bars. We consider the following metrics to assess performance:

- *Number of deployed RNs*: GSR aims to utilize the minimum number of MNs to populate the skeleton paths to restore inter segment connectivity in a WSN.
- *Total Travelled Distance*: This reflects the sum of distances travelled by the individual mobile nodes that were engaged in recovery. This metric assesses the resource overhead of the recovery process.

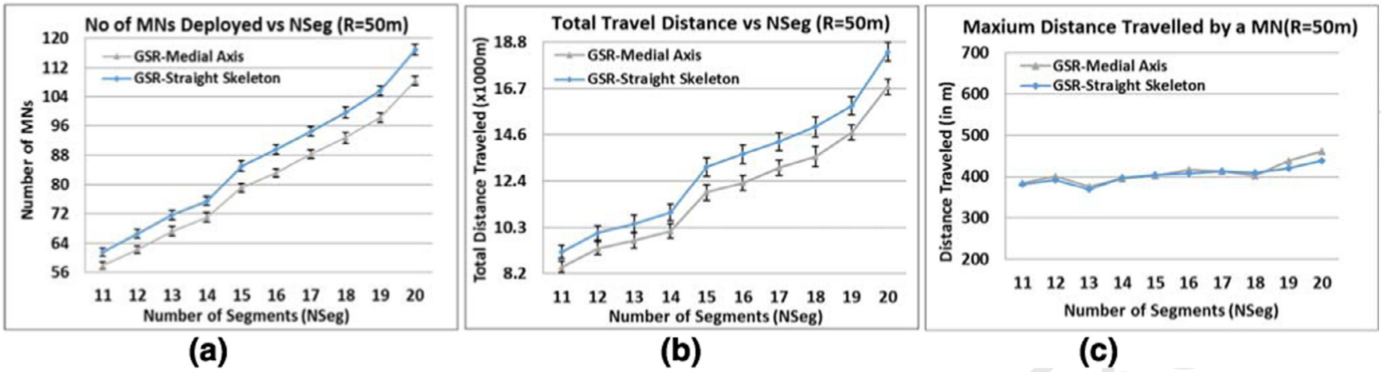


Fig. 10. A comparison of the straight skeleton and medial axis skeletons on recovery for a fixed transmission range w.r.t. (a) the number of MNs deployed to restore connectivity, (b) the total distance travelled by MNs to establish a connected topology, and (c) The maximum distance travelled by a MN during recovery.

- *Maximum Distance Traveled*: This metric looks at the maximum overhead that is placed on a node during the recovery process.
- We study the performance while varying the following parameters:
- *Number of Segments (N_{seg})*: A higher number of disjoint segments in a WSN extends the connectivity requirements and may engage a greater number of mobile nodes in the connectivity restoration process. On the other hand, increasing N_{seg} increases the likelihood of finding survivors in the local neighborhood during recovery and could potentially have a positive impact on the distance travel for recovery overhead.
- *Communication Range (R)*: It impacts the maximum distance between nodes on the formed inter-segment topology and consequently the number of MNs required for restoring the network connectivity.

For all experiments the considered ad-hoc topologies are devoid of interior holes to keep the comparisons fair with competing approaches. Since GSR and all baseline approaches assume that segments have sufficient number of MNs to undertake the recovery, the position of segments are marked when generating a topology without defining the intra-segment node count and connectivity. The first set of experiments compares the performance of the medial axis and straight skeleton for non-convex WSN topologies. Since in convex-shaped topologies, the medial axis and straight skeleton result in identical skeletons, only non-convex WSN topologies with no interior holes were considered in this experiment. In the second set of experiments, the performance GSR is compared to DarDs [9] and DORMS [10]; both of these distributed algorithms reconnect the disjoint segments by populating MNs toward the center of the deployment area. GSR fundamentally differs from these center based approaches, since it does not require a fixed convergence point, e.g., the center, and generates paths for MNs based on skeletal-shape of the deployment area. This also enables GSR to avoid the computationally heavy optimization phase and final relocation that both DarDs and DORMS perform in order to reduce the deployed MN count.

6.1. Simulation results

This section discusses the obtained results. In the first experiment we compare the performance of the medial axis to that of the straight skeleton for non-convex topologies. Each configuration is averaged over 50 different random topologies with a 90% confidence interval. Fig. 10 shows the effect of varying N_{seg} on performance. The number of segments N_{seg} is varied from 11 to 20 while the communication range R is fixed to 50 m. As is evident from Fig. 10(a), the medial axis skeleton outperforms the straight skeleton by utilizing fewer MNs to reconnect the network. This is due to the impact that reflex vertices have in a straight skeleton causing edges incident to them to grow in length and consequently increasing the size of the skeleton. This effect is also reflected in the total distance the MNs have to travel to reconnect the network as seen in Fig. 10(b). The medial axis based GSR is not impacted by the reflex vertices like the straight skeleton and it remains symmetric around its shape. The maximum distance traveled by a mobile node during recovery is nearly constant for both skeletons, as shown in Fig. 10(c). This is because when segments merge the leading MN is the one farthest along the path and it will keep exploring any remaining skeleton paths along its direction of motion.

We also studied the effect of varying the transmission range R between 50–100 m while fixing the number of segments to 11. Fig. 11(a) shows that as transmission range increases fewer MNs are required to restore connectivity, since MNs with larger ranges cover a greater

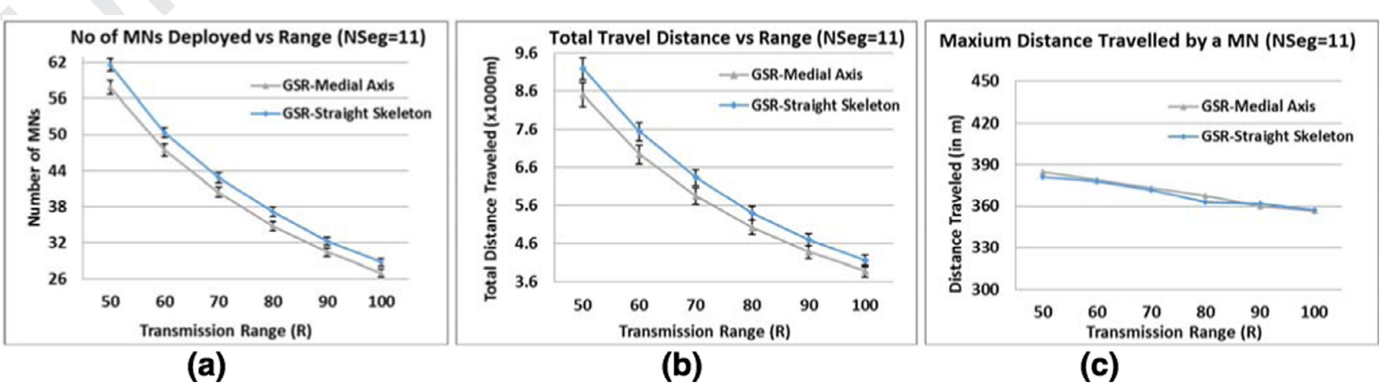


Fig. 11. The effect of varying a node's transmission range on the performance of GSR using Medial Axis vs Straight Skeleton w.r.t. (a) the number of MNs deployed to restore connectivity, (b) the total distance travelled by MNs to establish a connected topology, and (c) maximum distance travelled by a MN during recovery.

Please cite this article as: Y.K. Joshi, M. Younis, Exploiting skeletonization to restore connectivity in a wireless sensor network, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.07.022

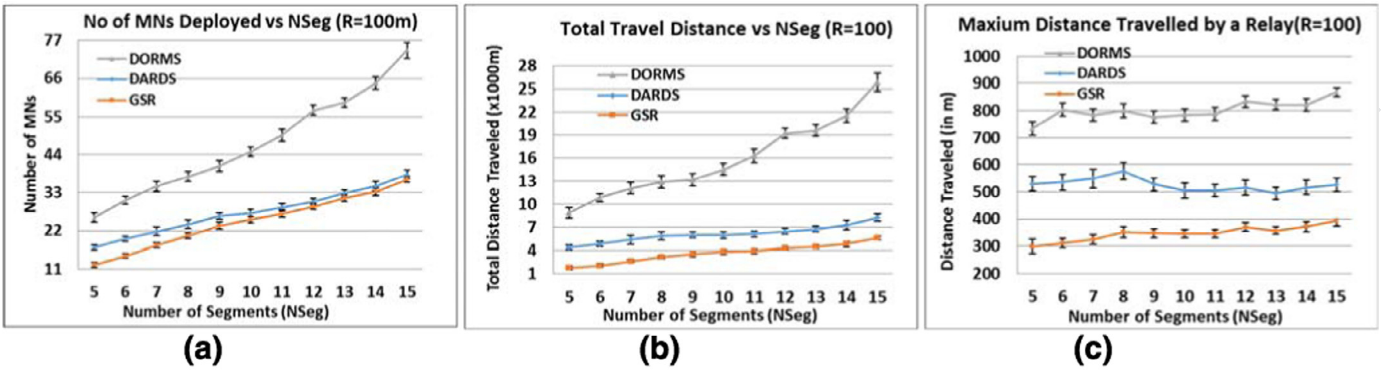


Fig. 12. Comparison of GSR with competing center based approaches DORMS and DarDs w.r.t. (a) the number of MNs deployed to restore connectivity, (b) total distance travelled by MNs to establish a connected topology, and (c) Maximum distance travelled by a MN during the recovery.

area and the paths from the individual segments merge quicker. The total travel distance, shown in Fig. 11(b), also decreases with longer ranges as fewer MNs are deployed. However, the maximum distance travelled by a mobile node during recovery, plotted in Fig. 11(c), is not much impacted by the range setting as the leading MN (representative) from each segment will still have to move approximately the same distance for both type of skeletons. The simulation results and the example in Fig. 9 confirms the superiority of the medial axis over the straight skeleton and combined with the lower runtime complexity makes medial axis based recovery our default approach to reconnect a disjoint network in the second set of experiments.

In the second set of experiments we compare the medial axis based GSR to DarDs [9] and DORMS [10], the competing center based approaches. In Fig. 12, the number of segments *Nseg* is varied from 5 to 15 while the communication range *R* is fixed to 100 m. Each configuration is averaged over 50 different topologies with a 90% confidence interval. From Fig. 12(a), it is clear that DORMS is the worst performing. It requires the most number of MNs to reconnect the network, the performance degrades as the number of segments increase. This is because DORMS requires MNs from all segments to reach the center of the deployment area it does not allow merging until the leading MNs from all segments congregate at the center. DarDs performs better than DORMS because it allows segments that come in range of one another before reaching the center to merge. This merging reduces the MNs required as the segments can pool their resources together from the merge point onward. GSR populates the least MNs and outperforms both DarDs and DORMS. The performance advantage is due to GSR's ability to exploit the deployment area shape and due to intelligent motion planning. Moving along the medial axis skeleton paths is equivalent to moving along angle bisectors for non-reflex vertices, which allows segments nearer to each other the quickly establish a communication path and discover their immediate neighbor since they could be neighboring Voronoi cells. In other words, GSR outperforms the center based approaches since depending on the location of segments, the chances for them merging is highly dependent on the choice of center.

Boosting the number of segments results in an increase in the total travel distance as seen in Fig. 12(b); DORMS has the worst performance due to all segments deploying MNs toward the center regardless of whether they meet earlier. Merging opportunities afforded by the increased number of segments partially offsets the increase in the total travel distance for both DarDs and GSR. Meanwhile, Fig. 12(c) highlights the maximum distance a MN has to travel during the recovery process, the increase in the number of segments has almost no effect on the maximum distance travelled by a mobile node in DORMS since all MNs are expected to reach the center. The location of the farthest surviving segment from the center determines the maximum distance traveled by a mobile node in DORMS. The growth in merging opportunities, due to the increased segments count, offsets the maximum travel for both DarDs and GSR.

Fig. 13 shows the effect of varying the transmission range *R* between 50–100 m, while fixing the number of segments, *Nseg* to 10. Fig. 13(a) shows that as *R* increases, fewer MNs are required to restore connectivity. This is very much expected since MNs with larger range can cover a greater area hence fewer MNs are needed to cover each contour and skeletal lines. The total travel distance, shown in Fig. 13(b), also decreases for longer ranges as fewer MNs are employed in the recovery. However, the maximum distance travelled is not much impacted with range, as indicated in Fig. 13(c), since the first MN (representative) from each segment will still have to move approximately the same distance both in case of center and skeleton based approaches.

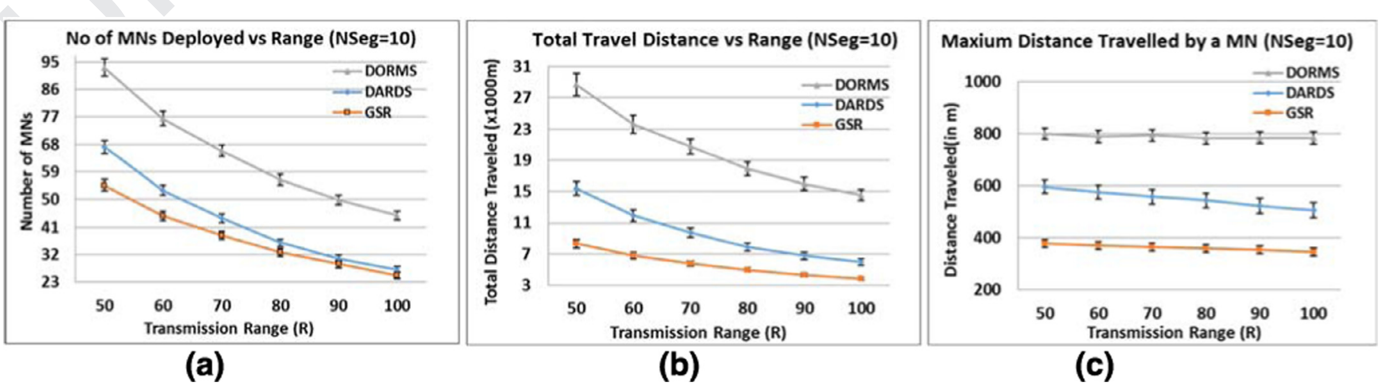


Fig. 13. The effect of varying a node's transmission range on the performance of GSR and baseline approaches in terms of (a) the number of MNs deployed to restore connectivity, (b) the total distance travelled by MNs to establish a connected topology, and (c) maximum distance travelled by a MN during recovery.

Please cite this article as: Y.K. Joshi, M. Younis, Exploiting skeletonization to restore connectivity in a wireless sensor network, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.07.022

Overall, the simulation results shown in Figs. 12 and 13 confirm the performance advantage of GSR over both the competing DORMS and DarDs approaches since it provides disjoint segments the earliest chance at reconnecting by populating MNs along the skeleton lines that comprise boundaries of Voronoi cells and is not dependent on a relative point in the deployment area for recovery. By utilizing fewer MNs in the initial recovery process itself, GSR avoids having to carry out computationally intensive calculations to solve the Steiner Tree Problem with Bounded Edges problem as is the case for DORMS and DarDs to reduce the deployed mobile node count. This optimization also results in further increases in travel distance due to the additional relocation of MNs to the calculated optimal locations. The performance advantage of GSR significantly grows for large networks, as the GSR-based recovery is mostly localized and facilitate the inter-connection of segments early in the recovery process.

7. Conclusion

In this paper we have presented GSR, a distributed algorithm that enables a WSN to restore connectivity after the failure of multiple collocated nodes that partitions the network into disjoint segments. The main idea is to exploit the pre-failure network topology to determine the skeleton of the WSN which can be utilized as a template along which mobile nodes can be deployed in case of failure in order to form a connected inter-segment topology. Unlike previously published approaches GSR forms a resource-efficient inter-segment topology and does not require further optimization once initial connectivity has been established amongst the segments. The simulation results have demonstrated that GSR scales well and outperforms competing approaches in terms of the number of required mobile nodes and the distance they need to travel. In the future, we plan to tackle the distributed connectivity restoration problem under resource and security constraints that may make certain locations unsafe for MN deployment and avoid making the topology structure predicable to an adversary.

Q6

Uncited reference

[27].

Acknowledgment

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

References

[1] M. Younis, I. Senturk, K. Akkaya, S. Lee, F. Senel, Topology management techniques for tolerating node failures in wireless sensor networks: a survey, Comput. Netw. 58 (Jan 2014) 254–283.
[2] X. Han, X. Cao, E.L. Lloyd, C.C. Shen, Fault-tolerant relay nodes placement in heterogeneous wireless sensor networks, in: Proceedings of INFOCOM'07, Anchorage AK, May 2007.
[3] N. Li, J.C. Hou, Flss: a fault-tolerant topology control algorithm for wireless networks, in: Proceedings of Mobicom '04, Philadelphia, PA, Sept 2004.
[4] B. Chen, et al., Span an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in: Proceedings of ACM MobiCom'01, Rome, Italy, July 2001.
[5] G. Wang, G. Cao, T. La Porta, W. Zhang, Sensor Relocation in Mobile Sensor Networks, in: Proceedings of INFOCOM'05, Miami, FL, Mar. 2005.
[6] M.F. Younis, S. Lee, A.A. Abbasi, A localized algorithm for restoring internode connectivity in networks of moveable sensors, IEEE Trans. Comput. 59 (12) (Aug. 2010) 1669–1682.
[7] F. Senel, K. Akkaya, M. Younis, An efficient mechanism for establishing connectivity in wireless sensor and actor networks, in: Proceedings of Globecom'07, Washington, DC, Nov. 2007.
[8] K. Akkaya, F. Senel, A. Thimmapuram, S. Uludag, Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility, IEEE Trans. Comput. 59 (2) (2010) 258–271.
[9] Y.K. Joshi, M. Younis, Distributed Approach for Reconnecting Disjoint Segments, in: Proceedings of Globecom'13, Atlanta, GA, Dec. 2013.
[10] S. Lee, M. Younis, Recovery from multiple simultaneous failures in wireless sensor networks using minimum steiner tree, J. Parallel Distrib. Comput. 70 (April 2010) 525–536.
[11] Y.K. Joshi, M. Younis, Straight skeleton based reconnection in a wireless sensor network, in: Proceedings of Globecom'14, Austin, TX, Dec. 2014.
[12] O. Aichholzer, F. Aurenhammer, D. Alberts, B. Gärtner, A novel type of skeleton for polygons, J. Univers. Comput. Sci. 1 (12) (1995) 752–761.
[13] O. Aichholzer, F. Aurenhammer, Straight skeletons for general polygonal figures in the plane, in: Proceedings of the 2nd Annual International Conference on Computing and Combinatorics (COCOON '96). Lecture Notes in Computer Science, no. 1090, Springer-Verlag, 1996, pp. 117–126.
[14] H. Blum, A transformation for extracting new descriptors of shape, in: W. Wathen-Dunn (Ed.), Models for the Perception of Speech and Visual form, MIT Press, 1967, pp. 362–380.
[15] D.T. Lee, Medial axis transformation of a planar shape, IEEE Trans. Pattern Anal. Mach. Intell. 4 (4) (1982) 363—369.
[16] X. Cheng, D.z. Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks, Wirel. Netw. 14 (3) (2008) 347–355.
[17] E.L. Lloyd, G. Xue, Relay Node Placement in Wireless Sensor Networks, IEEE Trans. Comput., 56 (1) (Jan 2007) 134–138.
[18] F. Senel, M. Younis, Optimized relay node placement for establishing connectivity in sensor networks, in: Proceedings of Globecom'12, Anaheim, CA, Dec. 2012.
[19] Y.K. Joshi, M. Younis, Autonomous recovery from multi-node failure in wireless sensor network, in: Proceedings of Globecom'12, Anaheim, CA, Dec. 2012.
[20] G. Robins, A. Zelikovsky, Tighter bounds for graph steiner tree approximation, SIAM J. Discrete Math. 19 (1) (2005) 122–134.
[21] S. Vemulapalli, K. Akkaya, Mobility-based self route recovery from multiple node failures in mobile sensor networks, in: Proceedings of WLN '10, Denver, CO, Oct. 2010.
[22] Y.K. Joshi, M. Younis, Mobility-based internetworking of disjoint segments, in: Proceedings of 27th Biennial Symposium on Communications (QBSC), Kingston, Ontario, Canada, June 2014.
[23] A. Abbas, Mohamed Younis, Establishing connectivity among disjoint terminals using a mix of stationary and mobile relays, Comput. Commun. 36 (13) (June 2013) 1411–1421.
[24] A. Aggarwal, L.J. Guibas, J. Saxe, P.W. Shor, A linear-time algorithm for computing the Voronoi diagram of a convex polygon, Discrete Comput. Geom. 4 (6) (1989) 591–604.
[25] L.P. Chew, Building Voronoi Diagrams for Convex Polygons in Linear Expected Time, in: Technical Report PCS-TR90-147, Dept. Math. Comput. Sci., Dartmouth College, 1986.
[26] CGAL, Computational Geometry Algorithms Library, http://www.cgal.org. Q7
[27] D. Eppstein, J. Erickson, Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions, Discrete Comput. Geom. 22 (4) (1999) 569–592.
[28] M. Tanase, R.C. Veltkamp, Straight skeleton approximating the medial axis, in: Proceedings of the 12th Annual European Symposium on Algorithms, 2004, pp. 809–821.
[29] R. Klein, Concrete and Abstract Voronoi Diagrams, vol. 400 (1989).
[30] S.-W. Cheng, A. Vigneron, Motorcycle graphs and straight skeletons, Algorithmica 47 (2) (Feb 2007) 159–182.
[31] F. Chin, J. Snoeyink, C.A. Wang, Finding the medial axis of a simple polygon in linear time, Discrete Comput. Geom. 21 (3) (1999) 405–420.
[32] H. Jiang, W. Liu, D. Wang, C. Tian, X. Bai, X. Liu, Y. Wu, W. Liu, Connectivity-based skeleton extraction in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 21 (5) (May 2010) 710–721.
[33] D. Douglas, T. Peucker , Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Canadian Cartographer 10 (2) (1973) 112–122. Q8
[34] Y. Wang, J. Gao, Y. Wang, J. Gao, J.S.B. Mitchell, Boundary recognition in sensor networks by topological methods, in: Proceedings of MobiCom '06, Los Angeles, CA, Sept. 2006.
[35] A. Kröller, S.P. Fekete, D. Pfisterer, S. Fischer, Deterministic boundary recognition and topology extraction for large sensor networks, in: Proceedings of the 17th ACM-SIAM Sympos. Discrete Algorithms, Miami, FL, Jan. 2006.