

## Research Article

# CARCC: Connectivity Autorecovering via Cooperative Communication

Ulisses Rodrigues Afonseca,<sup>1</sup> Camila Feitosa Rêgo,<sup>2</sup> and Jacir Luiz Bordim<sup>2</sup>

<sup>1</sup>Instituto Federal de Goiás, Campus Luziânia, Luziânia, GO, Brazil

<sup>2</sup>Department of Computer Science, University of Brasília (UnB), Brasília, DF, Brazil

Correspondence should be addressed to Ulisses Rodrigues Afonseca; urafonseca@ifg.edu.br

Received 4 July 2016; Revised 6 October 2016; Accepted 23 October 2016; Published 15 January 2017

Academic Editor: Gianluigi Ferrari

Copyright © 2017 Ulisses Rodrigues Afonseca et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless ad hoc networks are often considered an alternative for providing communication means to support urgent and temporary tasks. In such settings, network connectivity is a primary condition to allow nodes to exchange information. In this work we explore the use of cooperative communication (CC) techniques for improving network connectivity. The proposed solution, named Connectivity Autorecovering with Cooperative Communication (CARCC), monitors potential network disruption points and proactively creates cooperative links (CC-links). These links are used to regain network connectivity when necessary. CARCC allows the selection of suitable CC-links in  $O(\Delta(G) \times \Lambda)$  time, whereas a centralized approach takes  $O(V^2 \times \Lambda)$  time, where  $\Delta(G)$  is the degree of the underline network graph  $G(V, E)$  and  $\Lambda$  is the computational cost of selecting helper nodes. Although CARCC relies on localized information, it is able to provide comparable results of other noteworthy alternatives that require global topological information. The CC-links provided by CARCC reestablished network connectivity in over 91% in the evaluated settings.

## 1. Introduction

Wireless ad hoc networks have been envisioned as an alternative to support critical and temporary tasks. The application of these networks ranges from environment monitoring to providing disaster-response teams with a fast-deployable and cost-effective communication network. Nodes in an ad hoc network communicate via short-range radio links and may cooperate in relaying packets, allowing nonadjacent nodes to exchange information. Multihop communication can be ensured as long as a path from a given source to a desirable destination node exists. In such case, the task of gathering and dispensing sensed or generated information to faraway nodes is greatly reduced. For instance, a member of a disaster-response team may use flooding to summon medical assistance. Once the appropriate team is contacted (i.e., receives the message), it may reply the caller with the necessary information on actions to be taken prior to their arrival. In the above example, network connectivity is a necessary condition for delivering the request message.

A sizable number of solutions have been proposed in the literature aiming at ensuring network connectivity. Li and Hou [1] addressed the problem of adding as few nodes as possible so that the resulting network topology is connected. They proved that this task is NP-complete and proposed a Delaunay triangulation-based heuristic to tackle it. Goyal and Caffery [2] proposed the use of limited, coordinated mobility, to recover from link and node failures. In case of network disruption, nodes would move in a coordinated way to reestablish connectivity. Articulation points (a.k.a. cut-vertices) may appear during the network operation as a result of battery depletion, node movement, or event due to link failure. As the failure of an articulation point renders the network disconnected, Yong and Hu [3] presented a mathematical model to predict the formation of articulations based on network characteristics and node activity. Afonseca et al. [4] proposed the use of packet aggregation techniques for reducing energy consumption of articulation nodes that could render the network disconnected. Khelifa et al. [5] proposed the use of redundant nodes that could be moved

or activated in case of link or node failure. When necessary, redundant nodes would be activated and moved to the appropriate locations to reinforce connectivity. Recently, Guerriero et al. [6] proposed a centralized scheme to determine an appropriate node location to execute a given task that does not require additional nodes. In the same work, a distributed heuristic based on localized information was proposed.

In spite of its benefits, the above solutions require a significant amount of time to activate, dispense, and/or move redundant nodes to the appropriate locations. The use of cooperative communication (CC, for short) has been considered an attractive alternative for enhancing network connectivity [7–10]. The research on cooperative communication has been driven by the fact that some devices, such as sensor nodes, may not be empowered with multiple antennas due to their size and power constraints. Cooperative communication is a technique that allows nodes to create a virtual antenna array without deploying multiple antennas on the same device [8, 11]. CC allows nodes to send data to the far-end receiver over long distances efficiently [8]. Song et al. [12] studied the connectivity properties of sparse and dense networks when CC is used to extend the transmission range. As expected, nodes employing CC were able to improve network connectivity even in a sparse setting. The authors show that, by properly selecting helper nodes, cooperative networks may use less power than noncollaborative networks to maintain network connectivity. However, the task of finding helper nodes that, cooperatively, reconnect network components while using the least amount of power is an NP-complete problem [13].

Yu et al. [9] proposed a centralized topology control scheme, called CoopBridges, to increase network connectivity by employing cooperative communication. Starting from an undirected, disconnected graph, CoopBridges uses a heuristic to create bidirectional, cooperative edges, connecting disjoint network components. The proposed heuristic aims at selecting a subset of nodes, called helpers nodes, requiring the least amount of power to establish a cooperative link. As redundant cooperative communication links (CC-links) may be created, a minimum spanning tree algorithm is employed to prune costly links. Neves and Bordim [7] developed a similar mechanism, called CoopSink, to interconnect network components in a wireless sensor network. CoopSink uses a modified version of the heuristic proposed in [9]. The proposed scheme provides CC-links connecting disjoint components in a low-cost path to the sink node. Note that both CoopBridges and CoopSink work under the assumption that global topological information is provided prior to their execution. However, means to collect and disseminate topological information among disjoint network components are not detailed in the above schemes.

This work proposes a localized mechanism to establish cooperative communication links to prevent network partition. The proposed scheme, called Connectivity Autorecovering via Cooperative Communication (CARCC, for short) operates by identifying possible disruption points and selecting appropriate neighboring nodes to establish backup CC-links (a preliminary version of this work appeared in

[14]). In case of node or link failure, backup CC-links become operative to reestablish network connectivity. As CC-links are directional links by nature, bidirectional links are only possible when source and destination nodes have sufficient cooperative nodes. Hence, depending on the network topology and node condition, it may not be possible to establish a bidirectional CC-link. Using localized information, CARCC explores the use of directional CC-links to provide ways to reestablish connectivity even when bidirectional links are not possible. Although CARCC shares some characteristics of the works proposed by Khelifa et al. [5] and Yu et al. [9], it does not require the use of additional nodes or previous knowledge of the network topology. Indeed, CARCC is a distributed mechanism that relies on a simple and yet effective heuristic to select appropriate nodes with least amount of energy to recover network connectivity.

Simulation results show that CARCC provides results matching those provided by centralized solutions. In addition, CARCC reduces the network requirements to create suitable CC-links. In fact, CARCC allows the selection of suitable CC-links in  $O(\Delta(G) \times \Lambda)$  time, whereas a centralized approach takes  $O(V^2 \times \Lambda)$  time, where  $\Delta(G)$  is the degree of the underline network graph  $G(V, E)$  and  $\Lambda$  is the computational cost of selecting helper nodes. Compared to a centralized scheme, the localized CC-links provided by CARCC require a slightly higher transmission power. Nevertheless, in case of failure of an articulation point, network connectivity can be successfully obtained in more than 91% in the evaluated scenarios, which is comparable to the results obtained by a centralized algorithm. It is important to note, however, that centralized algorithms require means to maintain topological information up to date. Hence, power consumption of a centralized solution highly depends on the strategy used to obtain topological information. This cost has not been accounted for in this work as investigating such strategies is outside the scope of this paper.

The remaining of this paper is organized as follows. Section 2 presents a brief overview on techniques for locating articulation points and details the cooperative communication model assumed in this work. Section 3 states the problem while the main contribution of this work is presented in Section 4. Simulation results are presented and discussed in Section 5. Finally, Section 6 concludes this work.

## 2. Preliminaries

This section presents an overview on techniques to identify articulation points and defines the cooperative communication model considered in this work.

*2.1. Identifying Critical Elements.* In this work, “critical elements” are defined as nodes and edges that, when removed, turn the graph disconnected. Let  $G = (V, E)$  be an undirected connected graph, where  $V$  is the set of nodes and  $E$  is the set of edges. Node  $v \in V$  is a *cut-vertex* (a.k.a. articulation) if its removal makes the graph  $G$  disconnected. Similarly, an edge  $e \in E$  is a *bridge* if its removal makes the graph disconnected. Articulation points are of particular importance in wireless

networks as they represent the unique link between the components they connect. These nodes usually have a higher demand in relaying packets, as compared to other regular nodes. As a consequence, articulation points are likely to experience a faster drainage of their energy resources [4].

The task of locating articulation points and bridges can be performed using the Depth First Search algorithm (DFS) [15]. Goyal and Caffery [2] and Khelifa et al. [5] employed a modified DFS to locate articulation points in WSNs. A drawback of these solutions is the fact that they require topology information to be acquired before executing the DFS algorithm. Jorgic et al. [16] proposed a distributed solution that uses  $k$ -hop neighboring information to locate articulation nodes. The proposed scheme, however, has the disadvantage of producing false positives. Chaudhuri [17] proposed a distributed algorithm to find bridge nodes and 2-connected components. This strategy was extended to locate articulation points in [18] using  $O(|V| + |E|)$  units of time. Turau [19] presented a distributed algorithm for computing bridges, articulations, and 2-connected components that runs in  $O(|E|)$  time and uses at most  $4|E|$  messages, where  $|E|$  is the number of edges.

As in [19], in this work a *block* is defined as maximal, nontrivial 2-connected subgraph. Figure 1, denotes a topology with two blocks and their respective members. Also, we assume that articulation points, bridges, and 2-connected subgraphs can be correctly identified by a distributed mechanism sharing similar characteristics to those in [17–19] such that the following conditions are satisfied.

*Condition 1.* A node  $v \in V$  knows whether it is an articulation point or not.

*Condition 2.* Each block  $\mathcal{B}_i$  is correctly determined and has a unique identification number  $i$  ( $\geq 1$ ). Each node  $v \in V$  knows the block id it belongs to.

The next subsection details the communication model assumed in this work.

**2.2. Cooperative Communication.** Cooperative communication (CC) is a technique in which nodes transmit the same packet, taking advantage of the broadcast nature of wireless transmission, to achieve spatial diversity. In CC, nodes create a virtual antenna array without deploying multiple antennas on the same device [8, 11]. CC research is driven by the fact that some devices, like sensors, cannot accommodate an antenna array due to size and power constraints. Besides allowing for spatial diversity gain, CC provides transmission range extension [8]. The communication model considered in this work is similar to that used in [9, 20], which is detailed next.

The communication between two nodes, say  $v_i, v_j$ , in the traditional model can be simplified in terms of the transmission power, the distance between nodes, and the rate of signal fading. A node  $v_i$  can adjust its transmit power  $P_i$  with values in the range  $[0, P_{\text{MAX}}]$ . When  $P_i = 0$ , the transceiver is turned off and when  $P_i = P_{\text{MAX}}$ , the transceiver operates at full power. Thus, a source node  $v_i$  can

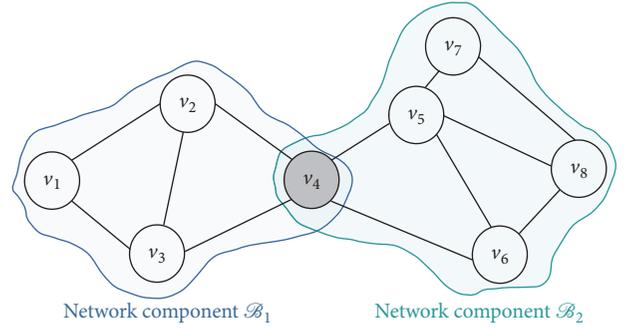


FIGURE 1: Graph representing a network topology with 8 nodes. The articulation node is shown in gray. The 2-connected blocks are  $\mathcal{B}_1 = \{v_1, v_2, v_3, v_4\}$  and  $\mathcal{B}_2 = \{v_4, v_5, v_6, v_7, v_8\}$ .

communicate directly with the destination node  $v_j$  only if its transmission power complies with the following relation:

$$P_i (d_{i,j})^{-\alpha} \geq \tau \quad (0 \leq P_i \leq P_{\text{MAX}}), \quad (1)$$

where  $\alpha$  is the exponent of signal fading, usually between 2 and 4, which is the rate of loss of the signal power with increasing distance,  $d_{i,j}$  is the Euclidean distance between  $v_i$  and  $v_j$ , and  $\tau$  is the receiver sensitivity to correctly receive a packet (i.e., the threshold of the received power so that node  $v_j$  can correctly decode the signal and obtain the original message). An edge between two neighboring nodes  $v_i$  and  $v_j$ , complying with (1), is called a *direct link* and is represented as  $\overline{v_i v_j}$ . The direct neighboring set of a node  $v_i$  is denoted as  $N(v_i)$ .

Suppose that  $v_1, v_2, \dots, v_m$  nodes cooperatively transmit signals to a particular destination node  $v_j$ . We assume that a cooperative transmission is successful if and only if the received, combined, signal at  $v_j$  is greater than or equal to  $\tau$ . For latter reference, let  $H_{i,j} \subseteq N(v_i)$  denote the *helper set* nodes of a source node  $v_i$  that cooperatively transmit packets to destination  $v_j$ . The received signal power  $P^c$  necessary for  $v_i$  and its helper nodes to communicate cooperatively with a destination node  $v_j$  can be computed by

$$P_{i \cup H_{i,j}}^c = \left( \sum_{k=1}^m P_k \cdot d_{k,j}^{-\alpha} \right) \geq \tau \quad (0 \leq P_k \leq P_{\text{MAX}}). \quad (2)$$

A cooperative communication link (CC-link) between nodes  $v_i$  and  $v_j$  is denoted as  $\overline{v_i v_j}$ . As customary, the network topology is represented as graph  $G(V, E)$ , which denotes the union of all the direct and CC-links. Let  $\overline{E}$  and  $\tilde{E}$  represent the set of direct and CC-links, respectively. Thus,  $E = \overline{E} \cup \tilde{E}$ . That is, if  $v_i v_j \in E$ , then  $v_i v_j = \overline{v_i v_j}$  if  $v_i v_j$  is a direct link and  $v_i v_j = \overline{v_i v_j}$  if  $v_i v_j$  is a CC-link. Similarly, the graph of direct and CC communication is denoted by  $\overline{G} = (V, \overline{E})$  and  $\tilde{G} = (V, \tilde{E})$ , respectively. The weight of a direct edge between nodes  $v_i, v_j$  is defined as

$$W(\overline{v_i v_j}) = \tau (d_{i,j})^\alpha. \quad (3)$$

As in [9, 14, 20], this work assumes that cooperating nodes expend the same amount of power when establishing a

cooperative link. Therefore, the weight of a CC-edge  $\widehat{v_i v_j}$  can be defined as

$$W(\widehat{v_i v_j}) = W_D(H_{i,j}) + (|H_{i,j}| + 1)W_{CC}(H_{i,j}), \quad (4)$$

where  $W_D(H_{i,j})$  is the minimum power consumption of the node  $v_i$  to communicate with the farthest node in  $H_{i,j}$  and  $W_{CC}(H_{i,j})$  is the minimum power consumption of the node  $v_i$  to communicate directly to  $v_j$ , together with its helper nodes in  $H_{i,j}$ .

In a cooperative communication from  $v_i$  to  $v_j$ , node  $v_i$  should initially send its data to helper nodes in  $H_{i,j}$  and then node  $v_i$  and its helpers must simultaneously send the same data to  $v_j$ . Thus, the weight of a CC-link is the sum of communication cost of these two steps. The cost for the first stage of communication is equivalent to  $W_D(H_{i,j})$ , while the cost of individual nodes to transmit data using CC is  $W_{CC}(H_{i,j})$ .

### 3. Problem Definition

As mentioned in the previous section, finding a suitable set of nodes that cooperatively reconnect network components using the least of amount of power is an NP-complete problem [13]. The heuristics proposed in [7, 9] are centralized and we assume that node location is known beforehand. This work takes a different approach for reestablishing network connectivity. More precisely, we focus on identifying and monitoring critical elements and, in case of failure, backup CC-links are established, whenever possible, so as to maintain network connectivity.

Suppose that node  $v_a \in V$  is an articulation node. From the definition of articulation nodes, removing  $v$  (and its corresponding edges) partitions the network into two or more blocks. Suppose that nodes  $v_w$  and  $v_z$ , located at different network blocks, are direct neighbors of articulation node  $v_a$ . In case node  $v_a$  fails, both  $v_w$  and  $v_z$  would be confined into separated network components. Clearly, a direct link  $\widehat{v_w v_z}$  cannot be established as  $P_w(d_{wz})^{-\alpha} < \tau$  and  $P_z(d_{zw})^{-\alpha} < \tau$ . Our approach relies on the fact that nodes located on distinct components (such as  $v_w$  and  $v_z$  in our example), when detecting the failure of a neighboring articulation node, may cooperate with the corresponding neighboring nodes to create a CC-link and reestablish network connectivity. Thus, our goal is to find suitable replacement nodes within  $k$ -hop distance from the failed articulation node so as to regain network connectivity.

*Problem 1.* Given nodes  $v_1, v_2, \dots, v_m$  and  $u_1, u_2, \dots, u_n$  residing at different blocks and connected to a common articulation point  $v_a$ , construct a CC-link satisfying (2) by selecting appropriate helper set  $H_{v_i, u} \subseteq N(v_i)$  ( $1 \leq i \leq m$ ) and  $H_{u, v} \subseteq N(v_j)$  ( $1 \leq j \leq n$ ) such that (4) is minimized.

As the task of finding suitable helper nodes using the least of amount of power is an NP-complete problem [13], this work proposes a heuristic to assist the neighboring nodes of an articulation in selecting helper nodes. The details of the proposed heuristic will be presented in the next section.

According to (2), as the helper set grows, less individual energy is demanded to construct a CC-link. On the other hand, the source node may expand more power to reach its direct neighbors. As discussed in the previous section, a cooperative link is a directional link by definition. Hence, it may happen that the best CC-link from a block  $\mathcal{B}_1$  to a block  $\mathcal{B}_2$  is not necessarily bidirectional (assuming that these subgraphs are disconnected due to an articulation failure). When required, two-directional CC-links may be created such that the destination node on a block may create a reverse CC-link to the source (provided that the necessary conditions are satisfied). The algorithm proposed in the subsequent section allows the establishment of both directional and bidirectional CC-links. In such case, the application developer may select the CC-link that best suits the application requirements.

Next, we present the details of the proposed Connectivity Autorecovering With Cooperative Communication (CARCC) protocol.

### 4. Network Connectivity Recovery

This section presents the Connectivity Autorecovering via Cooperative Communication (CARCC) protocol, which aims to restore network connectivity in the event of an articulation node failure. For the readers benefit, Notations show a list of the symbols used in this work. As mentioned before, CARCC works by identifying and proactively monitoring the availability of articulation nodes. In case of network disruption due to the unavailability of articulation nodes, cooperative communication is employed to reestablish connectivity via directional links. CARCC comprises two algorithms which are presented in the subsequent sections. First, a helper set selection heuristic is presented, which aims at selecting appropriate nodes to compose the helper set. Following it, the routine responsible for reconnecting the network in the event of articulation node failure is presented.

*4.1. Distance Based Helper Set Selection.* This subsection presents the HSS (helper set selection) heuristic. The main objective of the HSS is to allow neighboring nodes of an articulation point to compute backup CC-links that, in case of need, can be rapidly established to regain network connectivity.

Let node  $v_a$  be an articulation point. Then, according to [19],  $N(v_a)$  comprises nodes located in  $l \geq 2$  distinct blocks  $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_l$ . Clearly, the neighboring nodes of an articulation, say  $v_i$  and  $v_j$ , each residing at different blocks, are separated by a distance  $r < d_{i,j} \leq 2r$ , where  $r$  denotes the maximum transmission range of a node complying with (1). Recall that (2) allows for computing the transmission power necessary to create a CC-link between a source node  $v_i$  and a destination node  $v_j$ . The link  $\widehat{v_i v_j}$  is created cooperatively by  $v_i$  and its helper set  $H_{i,j} \subseteq N(v_i)$ . Thus, when computing (2), the exact distance  $d_{h,j}$ ,  $h \in H_{i,j}$ , is necessary. However, this information may not be available in a distributed environment as the nodes in  $H_{i,j}$  are not direct neighbors of node  $j$ .

Clearly, a node  $h \in H_{i,j}$  could assume a distance of  $3r$  to reach node  $v_j$  when computing (2). The above estimate, however, imposes a higher than necessary energy burden on the helper nodes. To overcome this, an estimated distance  $\mathcal{E}_{h,j}$  is used while computing (2). Let  $v_i$  and  $v_j$  be adjacent nodes of articulation point  $v_a$  residing at different blocks such that  $N(v_i) \cap N(v_j) = v_a$ . By the model assumed in this work,  $v_i$  knows  $d_{i,h}, \forall h \in N(v_i)$ . Likewise,  $v_a$  knows  $d_{a,i}$  and  $d_{a,j}$ . Node  $v_h$  can compute an estimated distance  $\mathcal{E}_{h,j}$  by exchanging distance information, as follows:

$$\mathcal{E}_{h,j} = d_{h,i} + d_{i,a} + d_{a,j}. \quad (5)$$

Clearly,  $r < \mathcal{E}_{h,j} \leq 3r$ . Hence, this estimation can be used by the HSS heuristic to compute (2) based on 1-hop distance information.

Now, we turn our attention to (4). Assume that node  $v_i$  is the closest neighboring node of articulation node  $v_a$  in block  $\mathcal{B}_i$ . Clearly, the node with least distance  $d_{i,h}$ , where  $h \in N(v_i)$ , is closer to node  $v_a$  than any other node  $u \in N(v_i)$ . Hence, node  $v_i$  selects the closest nodes to node  $v_a$  to form the helper set  $H_{i,j}$ . According to (4), the closer a node  $h \in N(i)$  is to  $v_i$ , the lower the energy will be demanded to establish the CC-link  $\widehat{v_i v_j}$ . The HSS heuristic uses a distance estimation rather than a target node location. Hence, in effect, the obtained CC-link  $\widehat{v_i v_j}$  connects  $v_i$  to an arbitrary node  $v_j$ , where  $v_j$  resides at an estimated distance  $\mathcal{E}_{i,j}$  from node  $v_i$ . The details of the HSS heuristic are presented in Algorithm 1.

The HSS takes as input the source node  $v_i$ , the articulation node  $v_a$ , and the estimated distance  $d_{a,j}$  between  $v_a$  and node  $v_j$ . In the following description we consider that  $v_i \in \mathcal{B}_k$  and  $v_j \in \mathcal{B}_l$ , for  $k \neq l$ . The **while**-loop of the HSS adds nodes to the helper set  $H_{i,j}$  until the destination  $v_j$  can be reached by a cooperative link. In case the nodes in  $N(i) \setminus v_a$  are exhausted and a  $\widehat{v_i v_j}$  link cannot be established, HSS returns an empty set. The correctness of the HSS routine can be easily verified. Furthermore, HSS uses no broadcast as the  $d_{i,h}, h \in N(i)$  are known to  $v_i$ . Clearly, such information can be obtained while articulation points begin to be computed [19]. Hence, node  $v_i$  is able to compute  $H_{i,j}$  and the estimate  $\mathcal{E}_{i,j}$  based only on local information.

In case node  $v_i$  succeeds in computing the CC-link  $\widehat{v_i v_j}$  (i.e.,  $H_{i,j} \neq \emptyset$ ) to an arbitrary node  $v_j$ , a directional CC-link connecting block  $\mathcal{B}_k$  to block  $\mathcal{B}_l$  can be created. As any node in  $N(a)$  is at a distance of at most  $r$  from node  $v_a$ , by a proper selection of the estimate  $\mathcal{E}$ , the CC-link  $\widehat{v_i v_j}$  can reach the farthest node in  $N(a)$ , independently of the block where node  $v_j$  is located. In other words, the CC-link can reach all the nodes in  $N(a)$ . The reverse link  $\widehat{v_j v_i}$  can be computed by node  $v_j \in N(v_a)$  in a similar way.

As the HSS uses no broadcast, our next task is to evaluate its computational cost. The HSS adds direct neighbors to the helper set until the estimated distance can be reached via a CC-link (**while**-loop, lines (2)–(10)). The estimated distance is computed by function  $\mathcal{F}$ . Clearly, the **while**-loop runs for at most  $\Delta(G)$  times, where  $\Delta(G)$  is the maximum degree of  $G$ . Ordering the nodes by their respective distance takes  $O(\Delta(G) \times \log \Delta(G))$  time using efficient sorting algorithms. Hence, HSS runs in  $O(\Delta(G) + \Delta(G) \times \log \Delta(G)) = O(\Delta(G) \times$

$\log \Delta(G))$  time. For latter reference, we summarize the above discussion in following lemma.

**Lemma 2.** *Given a source node  $v_i$ , an articulation node  $v_a$ , and an estimated distance  $d_{a,j}$  to a destination node  $v_j$ , the task of computing the helper set  $H_{i,j}$  complying with (2) and (4) can be done in  $O(\Delta(G) \times \log \Delta(G))$  time.*

**4.2. Connectivity Monitoring Algorithm.** The proposed Connectivity Autorecovering via Cooperative Communication (CARCC) aims at recovering network connecting in case of an articulation node failure. For this purpose, direct neighbors of an articulation point monitor its availability. CARCC assumes that Conditions 1 and 2, discussed in Section 2, are satisfied prior to its execution. This task can be accomplished by using a protocol with similar characteristics of those presented in [17–19]. In case the nodes work on a sleep/awake cycles, CARCC may take advantage of a Medium Access Control (MAC) protocol such that proposed in [21] to schedule monitoring cycles.

The details of the CARCC are shown in Algorithm 2. CARCC is composed of three main tasks, which are performed in sequence. The first task (lines (1)–(3)) is to compute a suitable distance estimation to create CC-links able to connect nodes in neighboring blocks. If the articulation node fails, these blocks may still be connected via CC-links. To compute the CC-links, CARCC uses the HSS, which demands a distance estimation so that the necessary helper nodes can be selected at each block. Let  $l$  denote the number of blocks connected to articulation point  $v_a$  such that  $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_l$ . Thus, for each block  $\mathcal{B}_i (1 \leq i \leq l)$  the articulation node  $v_a$  computes  $D_i = \min(d_{a,u})$ , where  $u \in \mathcal{B}_i$ . That is, the articulation node computes the minimum distance to reach the closest node in each block. Next, the minimum distance to reach nodes in the farthest blocks is computed. The two largest values,  $D_1$  and  $D_2$ , are informed to each node  $v \in N(v_a)$ . Once the distance estimation is available to the neighboring nodes of the articulation point, these nodes execute routine HSS using the estimate  $D_1$  or  $D_2$ . If  $d_{a,u} = D_1$ , then  $u \in N(v_a)$  uses estimate  $D_2$ ; otherwise  $u$  uses estimate  $D_1$ . This is necessary since  $u$  is at distance  $D_1$  and the estimated distance to reach the second most distant node in a different block is  $D_2$ . Those nodes having sufficient helper nodes to create a CC-link for the given estimated distances report the CC-link cost back to the articulation  $v_a$ . This task is executed in lines (5)–(7).

Note that a node in  $u \in \mathcal{B}_i \cap N(v_a)$  may refrain to send its CC-link cost (line (6)) when it overhears a cost lower than its own cost. Nevertheless, the articulation node may have several candidates per block  $\mathcal{B}_i, 1 \leq i \leq l$ . The next task is to select the backup node having the least cost. This task is performed in lines (8)–(9). At this stage, each block  $\mathcal{B}_i$  has a single backup node whose cost is the least among the nodes in that block. Let  $v_i$  denote the selected backup node in block  $\mathcal{B}_i$ . As  $v_i$  uses an estimated distance to compute the CC-link, the link cost may still demand more power than necessary to reach nodes in different blocks. Hence, a power optimization phase can be executed so as to reduce the CC-link cost.

**Input:** Source  $v_i$ , articulation node  $v_a$ , estimated distance  $d_{a,j}$ ;  
**Output:** A helper set  $H_{i,j}$  complying with (2) and (4) or  $\emptyset$ ;  
 Total transmission power  $W_{CC}(H_{i,j})$  or 0 if  $H_{i,j} = \emptyset$ .  
 #  $\mathcal{F}(\psi)$  is a function returning the node with the least  $d_{i,h}$ ,  $\forall h \in \psi$ ;  
 # Let  $v_j$  be the destination node such that  $v_j \in N(a)$  with  $d_{a,j} \leq r$ ;

- (1)  $\psi \leftarrow N(i) \setminus v_a$ ;  $H_{i,j} \leftarrow \emptyset$ ;
- (2) **while** ( $\psi \neq \emptyset$ ) **do**
- (3)      $H_{i,j} \leftarrow H_{i,j} \cup \mathcal{F}(\psi)$ ;
- (4)      $\psi \leftarrow \psi \setminus \mathcal{F}(\psi)$ ;
- (5)     Using (5), node  $v_i$  obtains  $\mathcal{E}(i, j)$ ;
- (6)     Node  $v_i$  uses  $\mathcal{E}(i, j)$  to compute  $P_{i \cup H_{i,j}}^c$  and  $W_{CC}(H_{i,j})$ ;
- (7)     **if** ( $P_{i \cup H_{i,j}}^c \geq \tau$ ) **then**
- (8)         **return** ( $H_{i,j}, W_{CC}(H_{i,j})$ );
- (9)     **end if**
- (10) **end while**
- (11) **return** ( $H_{i,j} \leftarrow \emptyset, 0$ );

ALGORITHM 1: HSS( $v_i, v_a, d_{a,j}$ ).

**Input:** Articulation point  $v_a$ ;  
**Output:**  $\forall H_{i,j} \neq \emptyset$ , a CC-link  $\overline{v_i v_j}$  connecting  $\{v_i, v_j\} \in N(v_a)$  such that  $v_i \in \mathcal{B}_k$  and  $v_j \in \mathcal{B}_m$ ,  $k \neq m$ ;

# Computing the min distance to reach the farthest block

- (1)  $\forall u \in \mathcal{B}_m$ ,  $1 \leq m \leq l$ ,  $v_a$  computes  $D_m = \min(d_{a,u})$ ;
- (2)  $v_a$  retrieves  $D_1$  and  $D_2$  satisfying  $D_1 \geq D_2 \geq \dots \geq D_m$ ;
- (3)  $v_a$  broadcasts  $(D_1, D_2)$  to all  $v \in N(v_a)$ ;

# Upon receiving  $D$ , check if a CC-link can be set up;

- (4)  $\forall v \in N(v_a)$  compute HSS( $v, v_a, D_1 \mid D_2$ );
- (5) **if**  $H_{v,j} \neq \emptyset$  **then**
- (6)      $v$  sends  $\langle W_{CC}(H_{i,j}), v \rangle$  back to node  $v_a$ ;
- (7) **end if**

# Selecting a suitable backup node in each block  $\mathcal{B}_m$ ;

- (8) Node  $v_a$  collects the messages and computes  $\min_{v_i \in \mathcal{B}_m} (W_{CC}(H_{i,j}), v)$  for each block  $\mathcal{B}_m$ ;
- (9) Node  $v_a$  informs to node  $v_i \in \mathcal{B}_m$  that it holds de minimum cost among the nodes in that block. Node  $v_i$  becomes the backup node for block  $\mathcal{B}_m$ ;

ALGORITHM 2: CARCC( $v_a$ ).

A CC-link power optimization strategy is presented in the next subsection.

Figure 2 shows a graph composed of one articulation and two network blocks  $\mathcal{B}_1 = \{v_1, v_2, v_3, v_4\}$  and  $\mathcal{B}_2 = \{v_4, v_5, v_6, v_7, v_8\}$ . Solid and dotted arrows represent, respectively, messages used to select a backup node in cluster  $\mathcal{B}_1$  and cluster  $\mathcal{B}_2$ . In Figure 2(a), the articulation executes Algorithm 2 and computes the distance estimation, which is sent it to its direct neighbors (lines (1)–(3)). In Figure 2(b), each node checks if the CC-link can be established to overcome the estimated distance (lines (4)–(7)). In case HSS returns a nonempty helper set for a node  $v \in N(v_a)$ , the required transmission power is informed to the articulation node. Next, the articulation point selects, for each block, the node with least cost to be its backup node in that block. Figure 2(c) represents the message used by the articulation to define the backup node (lines (8)–(9)).

With the correctness of the proposed CARCC being easily verified, our next task is to evaluate its running time. As

discussed in Section 2.1, CARCC assumes that Conditions 1 and 2 are satisfied prior to its execution. That is, after the articulation points and blocks are correctly identified, the first task (lines (1)–(3)) is executed by the articulation node and consists in computing distances  $D_1$  and  $D_2$ . Clearly, this task can be executed in  $|N(v_a)| \leq O(\Delta(G))$  time, where  $\Delta(G)$  represents the degree of the underline graph. Computing of the helper set and the communication cost (lines (4)–(7)) is performed by the direct neighbors of the articulation point and runs in  $O(\Delta(G) \times \log \Delta(G))$  time according to Lemma 2. The task of selecting the best source to connect each block (lines (8)–(9)) takes  $O(\Delta(G))$  time. Thus, overall, CARCC runs in  $O(\Delta(G) \times \log \Delta(G))$  time to identify suitable backup nodes and establish, whenever possible, CC-links connecting distinct blocks.

Next, we discuss the communication cost of the proposed algorithm. As can be observed in Algorithm 2, message transmission occurs in lines (3), (6), and (9). In line (3), the articulation point broadcasts the distance estimation

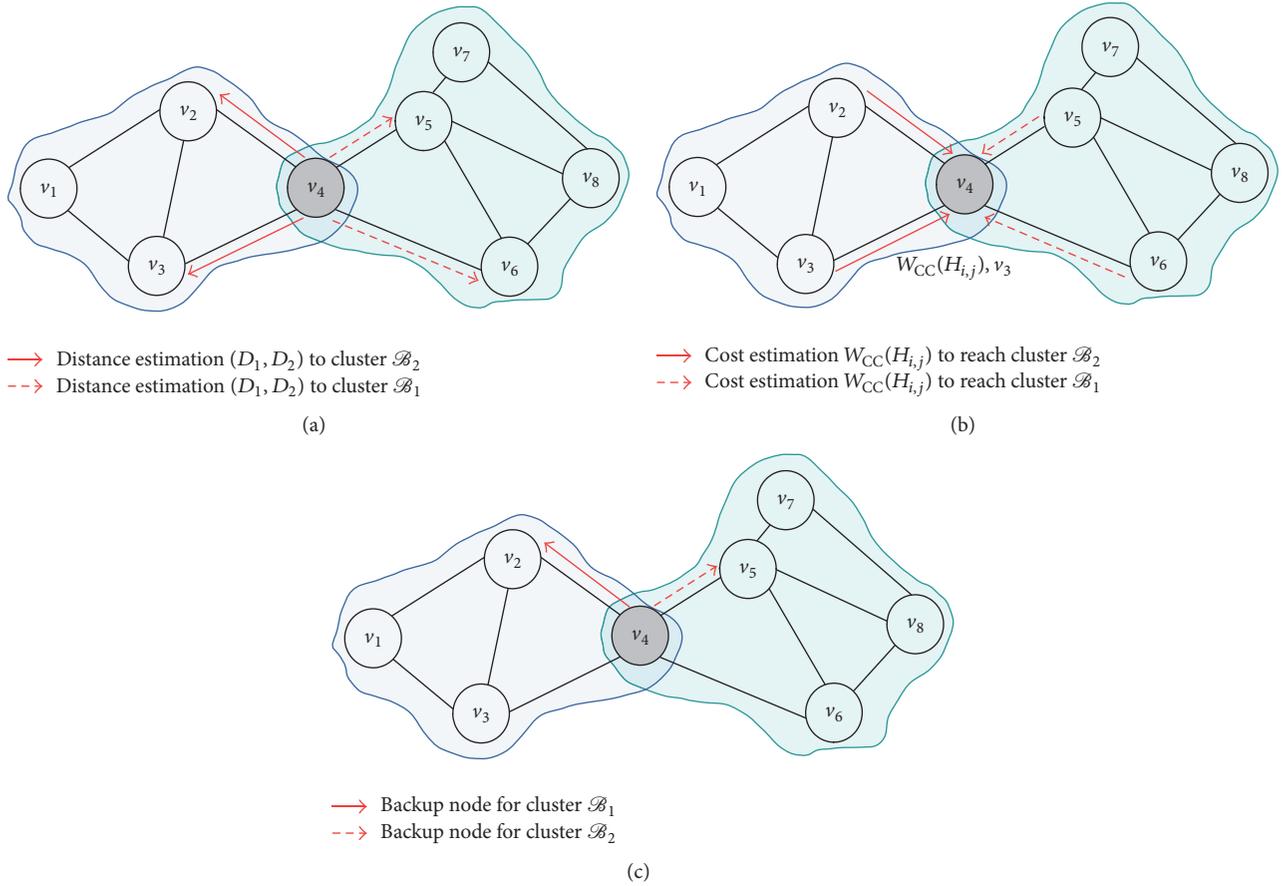


FIGURE 2: Example of CARCC operation.

to its direct neighbors. These nodes, in turn, compute the communication cost to create a CC-link to overcome the estimated distance. After that, the nodes in  $N(v_a)$  that have enough helper nodes to create a CC-link send back to the articulation node the cost of establishing the cooperative link (line (6)). Hence, line (6) requires at most  $\Delta(G)$  messages to be transmitted to the articulation point. Obviously, nodes having higher cost may refrain of sending their values when they overhear a cost being transmitted to the articulation point that is lower than its own cost. After collecting these messages, the articulation point selects the best source node in each block. Clearly, as there are at most  $l$  nodes, the articulation point may combine these values in a single message and broadcast the node ID of the selected backup node in each block  $\mathcal{B}_i$ ,  $1 \leq i \leq l$ . Hence, overall, CARCC requires  $O(|N(v_a)|) \leq O(\Delta(G))$  messages for selecting backup nodes for a given articulation point. Thus, we have the following.

**Theorem 3.** Given nodes  $v_1, v_2, \dots, v_m$  and  $u_1, u_2, \dots, u_n$  residing at different blocks and connected to a common articulation point  $v_a$ , the task of constructing a CC-link by selecting appropriate helper set  $H_{v_i, u} \subseteq N(v_i)$  ( $1 \leq i \leq m$ ) and  $H_{u, v} \subseteq N(v_j)$  ( $1 \leq j \leq n$ ) can be computed by CARCC in  $O(\Delta(G) \times \log \Delta(G))$  time using  $O(\Delta(G))$  messages.

In the above discussion, we have restricted our attention to articulation node failures. Clearly, when nodes in the vicinity of the articulation become unavailable or multinode failure occurs, the backup node may not be able to create the CC-link. One possible alternative to overcome this limitation is to have multiple backup CC-links composed of disjoint source and helper nodes. When the articulation fails, the remaining nodes may decide which backup CC-link among the available ones is still capable of reconnecting the network.

**4.3. Optimizing the CC-Link.** As mentioned before, the backup nodes establish CC-links based on an estimated distance informed by the articulation point. Hence, the CC-link may require more power than necessary to reach nodes in distinct blocks. Hence, this section presents an alternative to optimize the CC-link transmission power. More precisely, after selecting backup nodes in each block  $\mathcal{B}_i$ ,  $1 \leq i \leq l$ , these nodes may execute Algorithm 3 to optimize the transmission power. The idea behind Algorithm 3 is that nodes in a neighboring block, upon receiving a message via CC-link, may inform the source of the message whether the transmission power can be reduced or not. To achieve this, the selected backup node  $v_i$  in each block  $\mathcal{B}_i$ ,  $1 \leq i \leq l$ , sends a message  $M$  containing  $W_{CC}(H_{i,j})$ ,  $SrcBlockID$ , and  $SrcNode$ . All the nodes in different blocks collect  $M$  and register the

# Backup nodes adjust its  $W_{CC}(H_{i,j})$  based on  $P^c$ ;

- (1) The backup node in each block  $\mathcal{B}_i$  sends a CC message  $M$  containing  $W_{CC}(H_{i,j}), SrcBlockID, SrcNode$ ;
- (2) For each received message  $M$ , node  $v_z \in V$  records a tuple  $t$  with the  $(P_z^c, SrcBlockID, DestBlockID, SrcNode)$ ;
- (3) Node  $v_z$  sends a message  $T$  with the collected information back to the  $SrcNode$  through node  $v_a$ ;
- (4) For each block  $\mathcal{B}_i$ , node  $v_a$  computes
 
$$P_{\mathcal{B}_i}^c \leftarrow \max(P_z^c), \quad z \in \mathcal{B}_i;$$
- (5) Node  $v_a$  inform the  $P_{\mathcal{B}_i}^c$  to  $SrcNode$  in block  $\mathcal{B}_i$ ;
- (6) On receiving the  $P_{\mathcal{B}_i}^c$ , the  $SrcNode \in \mathcal{B}_i$  adjust its transmission power
 
$$W_{CC}^{opt}(H_{i,j}) \leftarrow \left( \frac{W_{CC}(H_{i,j})}{P_{\mathcal{B}_i}^c} \right) \cdot \beta, \quad (\beta > 0);$$

ALGORITHM 3: PwOpt.

received signal power ( $P_{SrcBlockID}^c$ ). This information is routed back to node  $v_i$  via direct links using a message  $T$ . As node  $v_a$  is the articulation point, all reply messages pass through  $v_a$ . Hence, by checking the reply messages, node  $v_a$  can identify the highest  $P_{\mathcal{B}_i}^c$  for each pair  $\overline{v_i v_j}$ , where  $v_i \in \mathcal{B}_i$  and  $v_j \in \mathcal{B}_j, i \neq j$ . The computation of  $P_{\mathcal{B}_i}^c$  can be done by the relaying nodes, so as to reduce the number of messages reaching the articulation point. By properly combining the reported messages, the articulation node receives at most  $\Delta(G)$  messages. To see this, let us consider the number of messages received by nodes in block  $\mathcal{B}_i$ . Clearly, as there are at most  $l$  backup nodes, the nodes in block  $\mathcal{B}_i$  receive at most  $l - 1$  messages. Each node may create a table consisting of the source node, block ID, and received transmission power and route back to the articulation point. Downstream nodes on the path to the articulation point only report the received values after collecting the messages from their neighbors. This can be achieved by allowing nodes closer to the articulation point to wait for the reply messages to be received and combined. The above approach reduces the number of messages passed to the articulation point to the number of neighboring nodes in that block. That is, the number of received messages by node  $v_a$  from block  $\mathcal{B}_i$  equals  $N(v_a) \cap \mathcal{B}_i$ . Once the articulation point collects the message  $T$  from its direct neighbors, it can determine the highest received power for each source/destination pair in each block. This information is sent via broadcast to its direct neighboring nodes. On receiving the message, node  $v_i$  can adjust its transmission power to reach node  $v_j \in \mathcal{B}_j$ . Indeed, the optimized transmission power to reach node  $v_j$  can be computed as  $W_{CC}^{opt}(H_{i,j}) = (W_{CC}(H_{i,j})/P^c) \cdot \beta$ , where  $\beta > 0$  is used as margin of error. Note that the set of auxiliary nodes  $H_{i,j}$  does not change during this task (the HSS routine is not applied again). Distance estimation based on RSSI has been reported in the literature to provide reasonable accuracy [22]. Alternatively, the path loss can be determined from frames transmitted at different power levels using the technique proposed by Kohvakka et al. [23]. Also,  $\beta$  can be used to reduce the effects of eventual fluctuations.

Figure 3 shows the power optimization steps to reduce power consumption of the cooperative links (CC-links). Figure 3(a) shows the CC-link and Figure 3(b) depicts the transmission of message  $M$  used to discover  $P^c$  of each

reachable node in an adjacent block. In Figure 3(c), the node receiving message  $M$  stores the tuple  $t$  and routes the message  $T$  to the articulation node. In Figure 3(d), the articulation node selects the highest  $P_{\mathcal{B}_i}^c$  among the received messages and forwards it to the backup node. Based on the information received, the backup node optimizes the transmission power so as to reduce energy consumption. This final step is shown in Figure 3(e), where the CC-link is optimized to reach as far as the nearest neighbor in the adjacent block  $\mathcal{B}_2$ . In the above example, a single, directional CC-link from block  $\mathcal{B}_1$  to block  $\mathcal{B}_2$  was established. The same steps are executed to create a reverse CC-link from  $\mathcal{B}_2$  to block  $\mathcal{B}_1$ . Note that, whenever possible, CARCC creates bidirectional CC-links among distinct blocks connected to an articulation point. In case of articulation node failure, these backup CC-links are used to maintain connectivity.

As can be observed, the task of the optimizing the CC-links power transmission requires at most  $\Delta(G) + 1$  messages to be handled by the articulation point. Furthermore, Algorithm 3 runs in  $O(\Delta(G))$  time, which is the time taken to combine the received  $T$  tuples and forward them to the downstream nodes.

## 5. Simulation Results

To assess the goodness of the Connectivity Autorecovering via Cooperative Communication (CARCC) protocol, we have implemented it and other prominent alternatives in Matlab [24]. For a given network topology, the simulation process begins by identifying articulation points and blocks. From this point, the HSS routine is used to identify suitable backup nodes. For each topology, the nodes are randomly positioned in a  $300 \times 300$  m area. The node density  $N_d$  assumes the following values  $N_d = 4.0 \times 10^{-4}, 4.5 \times 10^{-4}, \dots, 12.0 \times 10^{-4}$ . Only connected network topologies with at least one articulation are considered in the simulation. The transmission power, receiver threshold, and signal fading are similar to those used by Zhu et al. [20] and Yu et al. [9]. The maximum transmitting power ( $P_{MAX}$ ) is set to 4900 and the receiver threshold ( $\tau$ ) is set to 1, allowing a maximum transmission range ( $R_{MAX}$ ) of  $\approx 70$  m. The simulation results comprehend the average of 1000 simulations for each configuration set, with a 95% confidence interval.

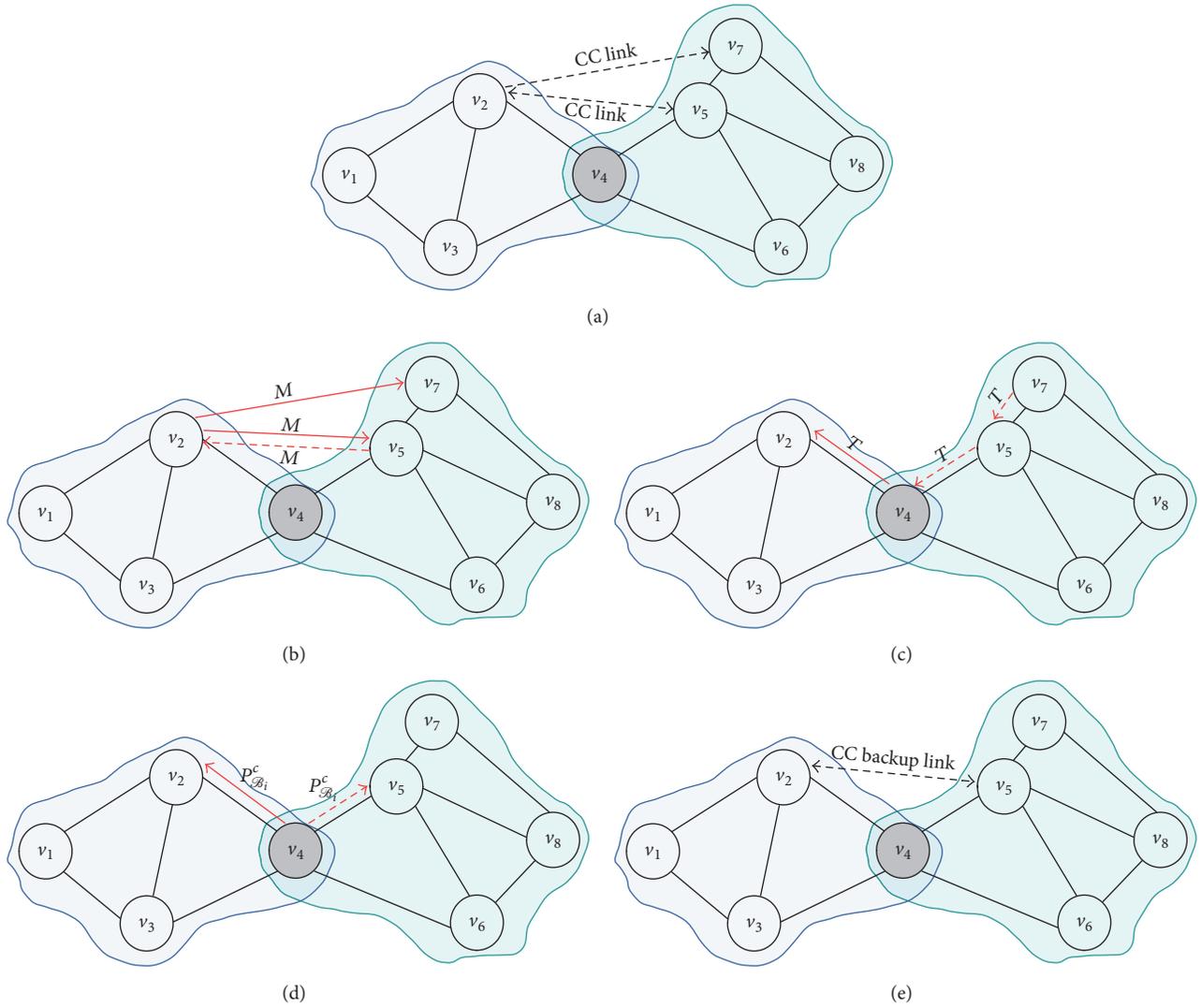


FIGURE 3: CC-link power optimization example.

Figure 4 depicts simulation snapshots of the proposed solution. Figure 4(a) shows the initial topology containing a single articulation point. Due to articulation node failure, the network becomes partitioned into two disjoint components as depicted in Figure 4(b). Figure 4(c) shows the backup CC-links computed by Algorithm 2. The dashed lines represent the CC-links and the dotted lines represent the links connecting the source nodes and the corresponding helpers nodes. Figure 4(d) shows a reconnected network topology employing a single CC-link, obtained after power optimization (i.e., after the execution of Algorithm 3).

To verify goodness of the selected backup nodes in terms of power consumption and connectivity, CARCC is compared to the following: (i) a centralized solution in which global topology information is available and (ii) a localized solution in which  $k$ -hop node distance information is known beforehand. In the simulations  $k = 2$  and the proposed HSS is applied to select the helper set. To compute the CC-link using global topology information, an exhaustive search is performed to select source and helper nodes with the aim

of reestablishing network connectivity. The resulting CC-link is equivalent to running a minimum spanning tree after the creation of all possible cooperative links in the network. This method matches the links established by CoopBridges [9, 10]. The best localized CC-link is computed in a similar way, but considering source and destination nodes as direct neighbors of the articulation node. The above strategy has been used in the preliminary version of this work [14]. Hereafter, “global” and “local” refer to the topology information used in computing a CC-link. Recall that CARCC computes the CC-links based solely on distance estimation as it does not rely on exact node location. The following metrics are used in comparing the CC-links obtained using global and local topology information to those links provided by CARCC:

- (i) Metric M1 ( $W_D(H_{i,j})$ ): the transmission power required to reach the farthest node in the helper set.
- (ii) Metric M2 ( $W_{CC}(H_{i,j})$ ): the transmission power required by each individual node to establish a CC-link.

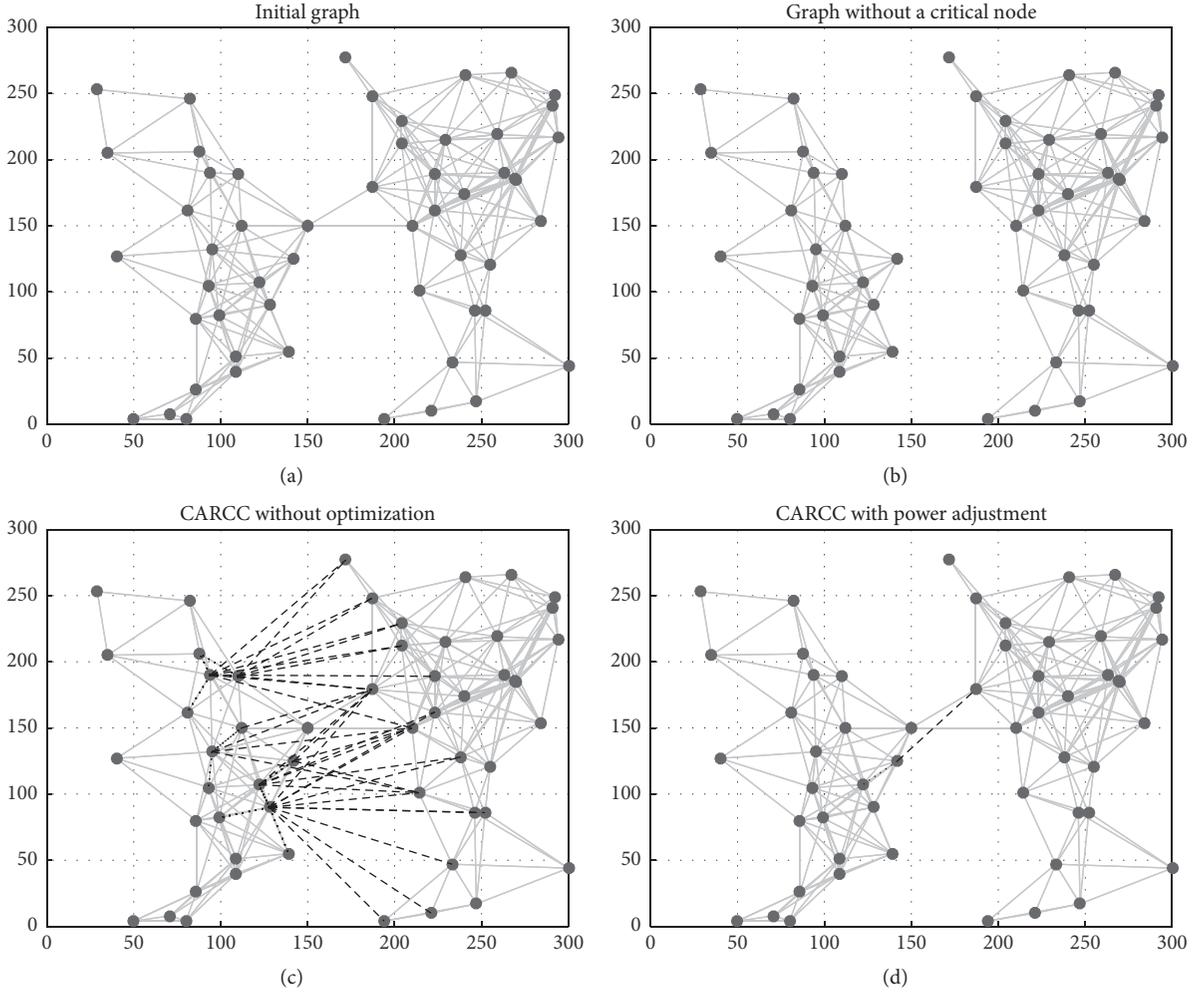


FIGURE 4: Example of CARCC simulation.

- (iii) Metric M3 ( $W_D(H_{i,j}) + (|H_{i,j}| + 1)W_{CC}(H_{i,j})$ ): the total transmission power required to establish connectivity using cooperative communication.
- (iv) Metric M4 ( $W_D$ ): the transmission power required to maintain connectivity when cooperative communication is not possible.
- (v) Metric M5 ( $|H_{i,j}|$ ): the number of helper nodes used to maintain connectivity with the selected destination node.
- (vi) Metric M6 ( $|\tilde{E}|$ ): the number of links created by means of cooperative communication.

Metrics M1–M3 show the cost of establishing a CC-link, starting from the cost of reaching the selected helper nodes (M1), the cost of taking part in a CC-link (M2), and the total cost of the CC-link (M3). Metric M4 considers the case in which nodes may increase the transmission power beyond  $P_{MAX}$ . Hence, M4 serves as the base to compare the costs of a direct link and a CC-link to reestablish communication in case on network disruption. Lastly, M5–M6 shows the average number of helper nodes and CC-links used to maintain

connectivity in case of articulation node failure. The rest of this section presents results for the above metrics. These metrics are evaluated as a function of node density (number of nodes per area) shown in the  $x$ -axis. In what follows, the transmission power with and without employing Algorithm 3 is denoted as “CARCC” and “CARCC.PwOpt,” respectively. The results for the best global and local CC-links are denoted as “Global” and “2-Hop,” respectively.

Figure 5 shows the results for Metric M1. According to the CC model described in Section 2.2, with closer helper nodes, the values of  $W_D(H_{i,j})$  tend to decrease. This fact can be observed in results shown in Figure 5, where CARCC without optimization is able to cut the transmission power in approximately 50% when comparing lower ( $N_d = 4.0 \times 10^{-4}$ ) and higher node density ( $N_d = 12.0 \times 10^{-4}$ ). A similar behavior is observed for the global and local solutions as well as for CARCC with optimization. Clearly, global and local topological information allows for a better selection of source and helper nodes, resulting in a lower transmission cost for metric M1. As CARCC uses an estimate distance instead of a precise node location, the results for CARCC without optimization demand higher transmission power.

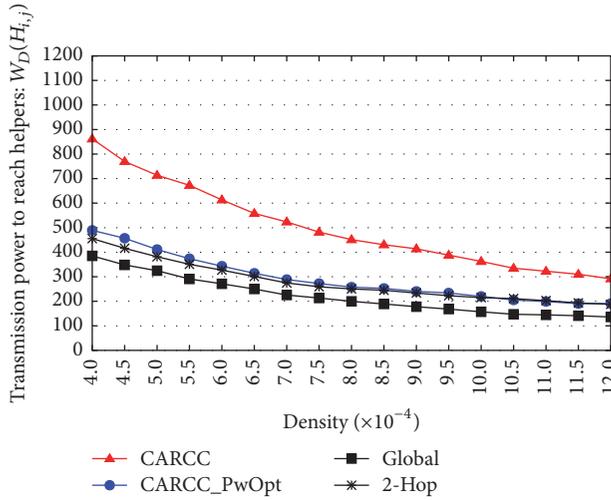


FIGURE 5: Transmission power  $W_D(H_{i,j})$  required to reach the farthest node in the helper set.

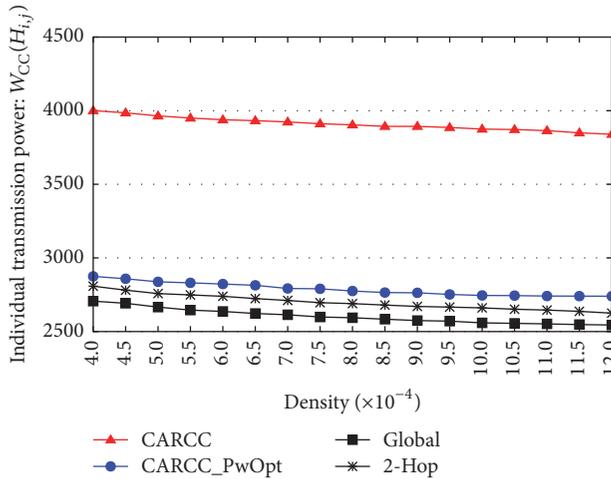


FIGURE 6: Transmission power  $W_{CC}(H_{i,j})$  required by individual node during cooperation.

However, when power optimization is used, the source and helper nodes are able to reduce the transmission power significantly. In fact, the transmission power used by CARCC with power optimization is similar to that obtained when precise node location at 2 hops is used. The results show that the transmission power difference at higher node density is negligible when CARCC with power optimization is used.

Figure 6 shows the average transmission power for individual nodes during the cooperative transmission. Unlike previous results, the node density has little effect on the transmission power for metric M2. In case of an articulation node failure, the CC-link connecting disjoint components need to overcome a distance that surpasses that of a direct link, which is independent of the node density. As can be observed, nodes participating in a CC-link require  $W_{CC}(H_{i,j}) \approx P_{MAX}$  in case of CARCC without optimization. However, when power optimization is employed, the transmission power cut

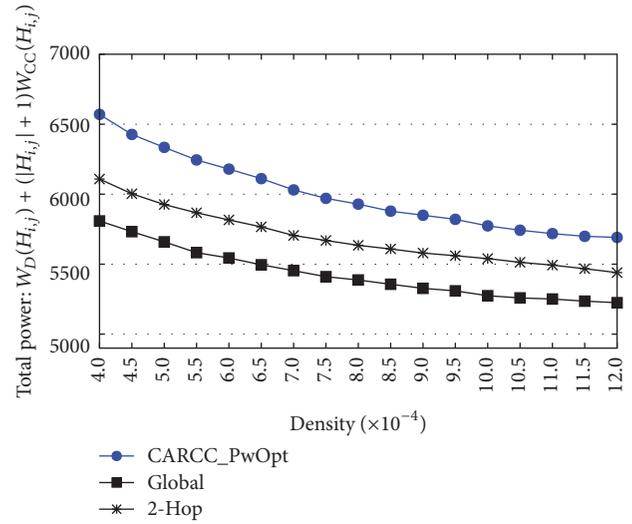


FIGURE 7: The total transmission power of the CC-link.

surpasses 30%. In fact, the global CC-link, local CC-link, and CARCC with optimization present similar results, with the latter expanding a little more power. In the worst case, CARCC with optimization requires  $\approx 7.7\%$  more transmission power than the best global CC-link and  $\approx 4.4\%$  more transmission power than the best local CC-link. The above results show that the power optimization allows CARCC to create efficient CC-links.

Figure 7 shows simulation results for Metric M3. This metric encompasses the power necessary to reach the nodes on the helper set and the transmission power required to cooperatively send data to the destination node. As can be observed in the figure, the total transmission power decreases with higher density, which comes from the reduced transmission power required to reach the helpers. The transmission power required by the optimized CARCC is about 10.35% higher than the best global CC-link and about 5.42% compared to the best local CC-link.

In general, nodes may increase their transmission power up to certain limits, which are determined by the underline hardware or by governmental regulations. Figure 8 shows simulation results for Metric M4, in which nodes may increase the transmission power without bounds. Although a node may not increase the transmission power beyond the  $P_{MAX}$ , it provides a lower bound on the amount of power required to reestablish communication when direct links are employed. The results denote that node density has a higher impact on CC-links as compared to direct links. As discussed above, this follows from the results shown in metric M1. With node density  $d = 12.0 \times 10^{-4}$ , the CARCC expands at most 14% more transmission power than a direct link computed using global topological information and at most 9.2% using 2-hop local information. This result indicates that CARCC with power adjustment can reconnect the network with a minor transmission power increase. That is, CARCC with optimization is at most 14% higher than the lower bound provided by a direct link.

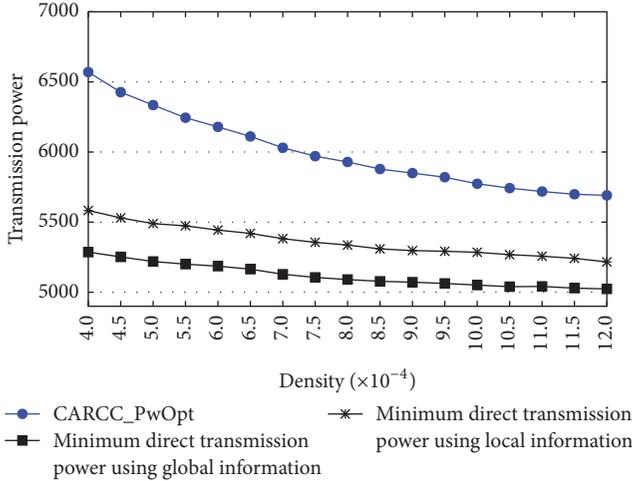


FIGURE 8: Minimum transmission power using direct link.

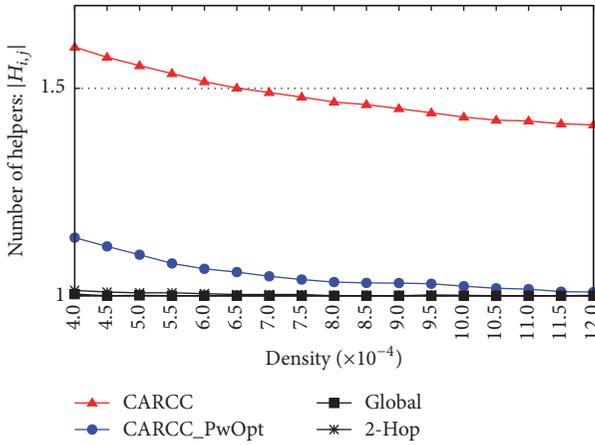


FIGURE 9: Number of nodes in the helper set.

Next, we evaluate the number of nodes in the helper set. The results are depicted in Figure 9, which shows the simulation results for Metric M5. As can be observed, the number of helper nodes is below 2 nodes independent of the strategy employed. CARCC without optimization requires a higher number of helper nodes due to higher estimated distance of the CC-links. When power optimization is employed, the number of helper nodes at higher density is close to those required by the global and local strategies.

In the example shown in Figure 3(a), one can verify that the CC-links created by Algorithm 2 may reach a number of nodes in distinct blocks of the source node. In the figure, the source node  $v_2$  established a CC-link capable of reaching nodes  $v_5$  and  $v_7$ . The simulation results in Figure 10 show that the number of nodes at the destination blocks reached by the selected source node increases linearly with node density. The results of metric M6 show an increase nearly threefold when the node density increases from  $d = 4.0 \times 10^{-4}$  to  $d = 12.0 \times 10^{-4}$ . This behavior corroborates that CARCC create links that are able to reach nodes well beyond the neighborhood of

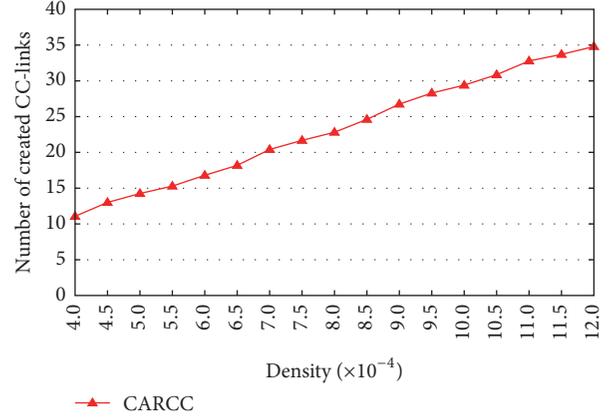


FIGURE 10: Number of created cooperative links.

the articulation node. Clearly, such links could be explored to provide an alternative path to a particular destination.

## 6. Conclusion

This work explores cooperative communication to provide a localized solution to reconnect isolated network components and restore network connectivity. The proposed protocol, named Connectivity Autorecovering with Cooperative Communication (CARCC), uses single-hop topological information to proactively reconnect the network using cooperative links. The proposed scheme works by monitoring nodes and regions that are more susceptible to rupture the network. CARCC was compared against two different strategies that use global and  $k$ -hop information to compute the best CC-links to regain network connectivity. Simulation results show that the links established by CARCC require similar amount of transmission power to those that require more detailed information on node location. The transmission power is at most 13% and 5.42% higher than the best CC-link using, respectively, global and 2-hop location information. CARCC runs in  $O(\Delta(G) \times \log \Delta(G))$  time using only local information, where  $\Delta(G)$  is the degree of the graph. A global solution, on the other hand, requires  $O(|V|^2 \times \Delta(G) \times \log \Delta(G))$  time. Clearly, CARCC is suitable alternative to regain network connectivity and provide an alternative path to route sensitive and urgent information to a monitoring center or sink node. As a future work, we plan to consider more elaborated path loss models as well as other wireless channel impairments, such as fading and signal interference in the simulations.

## Notations

$G(V, E)$ :	Graph $G$ , where $V$ is the set of vertices (nodes) and $E$ is the set of edges (links)
$ V $ :	Number of vertices (nodes)
$ E $ :	Number of edges (links)
$\bar{E}$ :	Direct edge (link)
$\tilde{E}$ :	Cooperative edges (links)
$v_i v_j$ :	Edge between node $i$ and node $j$

$\overline{v_i v_j}$ :	Direct edge between node $i$ and node $j$
$\widehat{v_i v_j}$ :	Cooperative edge between node $i$ and node $j$
$\mathcal{E}_{h,j}$ :	Estimated distance between node $h$ and $j$
$\mathcal{B}_i$ :	Network block with ID $i$ (cluster of biconnected nodes)
$P_i$ :	Transmission power
$P_{MAX}$ :	Maximum transmission power
$d_{i,j}$ :	Distance between nodes $i$ and $j$
$D$ :	Minimum distance between a node and a disjoint network block
$\alpha$ :	Path loss exponent
$\tau$ :	Receiver sensitivity threshold
$H_{i,j}$ :	Helper set connecting the source node $v_i$ to destination node $v_j$
$P_{i \cap H_{i,j}}^c$ :	Received power over a cooperative link formed by $i$ and its helpers
$\psi$ :	Direct neighbors of the nodes excluding articulation nodes
$\mathcal{F}(\psi)$ :	Function returning the closest node in the set $\psi$ to a given destination node
$W_D$ :	Communication cost of a direct edge
$W_D(H_{i,j})$ :	Communication cost of node $i$ to reach the nodes in its helper set
$W_{CC}$ :	Individual cost of cooperative communication link
$W(\widehat{v_i v_j})$ :	Total cost of a cooperative communication link
$SrcNode$ :	Source node
$DstNode$ :	Destination node
$SrcNodeId$ :	Identifier of the network block holding the source node
$DstNodeId$ :	Identifier of the network block holding the destination node
$\beta$ :	Margin of error for transmission power adjustment
$T$ :	Message containing the received signal power
$M$ :	Message containing the destination transmission power and the block of the source node
$\Delta(G)$ :	The degree of the graph $G$
$N_d$ :	Node density.

## Competing Interests

The authors declared that they have no conflict of interests.

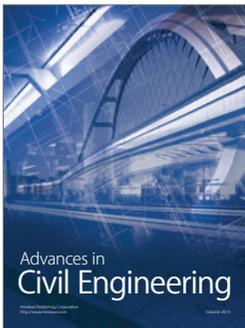
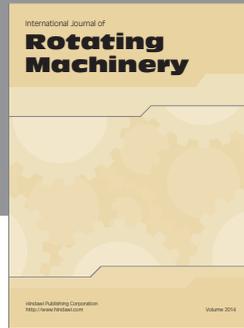
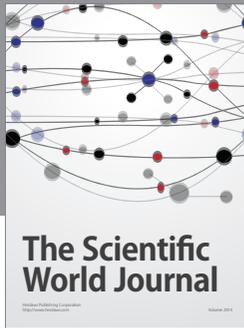
## Acknowledgments

This work has been partially supported by CNPq and CAPES.

## References

- [1] N. Li and J. C. Hou, "Improving connectivity of wireless ad hoc networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05)*, pp. 314–324, San Diego, Calif, USA, July 2005.
- [2] D. Goyal and J. Caffery Jr., "Partitioning avoidance in mobile ad hoc networks using network survivability concepts," in *Proceedings of the 7th International Symposium on Computers and Communications (ISCC '02)*, pp. 553–558, IEEE, July 2002.
- [3] T. Yong and G. Z. Hu, "The articulation nodes modeling of wireless sensor networks," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCAISM '10)*, vol. 12, pp. 12–75, IEEE, Taiyuan, China, 2010.
- [4] U. R. Afonseca, P. H. Azevêdo Filho, J. L. Bordim, and P. S. Barreto, "Localização e redução do consumo de energia em pontos de articulação em redes de sensores sem fio," in *Proceedings of the 2nd Workshop de Sistemas Distribuídos Autônomicos*, pp. 21–24, 2012.
- [5] B. Khelifa, H. Haffaf, M. Madjid, and D. Llewellyn-Jones, "Monitoring connectivity in wireless sensor networks," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC '09)*, pp. 507–512, July 2009.
- [6] F. Guerriero, A. Violi, E. Natalizio, V. Loscri, and C. Costanzo, "Modelling and solving optimal placement problems in wireless sensor networks," *Applied Mathematical Modelling*, vol. 35, no. 1, pp. 230–241, 2011.
- [7] T. F. Neves and J. L. Bordim, "Topology control in cooperative Ad Hoc wireless networks," *Electronic Notes in Theoretical Computer Science*, vol. 302, pp. 29–51, 2014.
- [8] C.-C. Jay Kuo, Y.-W. Peter Hong, and W.-J. Huang, *Cooperative Communications and Networking: Technologies and System Design*, Springer Science+Business Media, 2010.
- [9] J. Yu, H. Roh, W. Lee, S. Pack, and D.-Z. Du, "Topology control in cooperative wireless ad-hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1771–1779, 2012.
- [10] J. Yu, H. Roh, W. Lee, S. Pack, and D.-Z. Du, "Cooperative bridges: topology control in cooperative wireless ad hoc networks," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, San Diego, Calif, USA, March 2010.
- [11] M. Uysal, *Cooperative Communications for Improved Wireless Network Transmission: Framework for Virtual Antenna Array Applications*, Information Science Reference, 2010.
- [12] S. Song, D. L. Goeckel, and D. Towsley, "Collaboration improves the connectivity of wireless networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–11, IEEE, Barcelona, Spain, April 2006.
- [13] W.-T. Chen and N.-F. Huang, "Strongly connecting problem on multihop packet radio networks," *IEEE Transactions on Communications*, vol. 37, no. 3, pp. 293–295, 1989.
- [14] U. R. Afonseca, T. F. Neves, and J. L. Bordim, "Sinkpath: a cooperative communication protocol to Reestablish WSN connectivity," in *Proceedings of the 13th International Conference WWW/Internet 2014 (ICWI '14)*, pp. 299–306, IADIS Press, Porto, Portugal, October 2014.
- [15] J. Hopcroft and R. Tarjan, "Algorithm 447: efficient algorithms for graph manipulation," *Communications of the ACM*, vol. 16, no. 6, pp. 372–378, 1973.
- [16] M. Jorgic, M. Hauspie, D. Simplot-Ryl, and I. Stojmenovic, "Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks," in *Proceedings of the 3rd IFIP Mediterranean Ad Hoc Networking Workshop*, pp. 360–371, Bodrum, Turkey, June 2004.
- [17] P. Chaudhuri, "An optimal distributed algorithm for computing bridge-connected components," *The Computer Journal*, vol. 40, no. 4, 1997.

- [18] P. Chaudhuri, "An optimal distributed algorithm for finding articulation points in a network," *Computer Communications*, vol. 21, no. 18, pp. 1707–1715, 1998.
- [19] V. Turau, "Computing bridges, articulations, and 2-connected components in wireless sensor networks," in *Algorithmic Aspects of Wireless Sensor Networks*, vol. 4240 of *Lecture Notes in Computer Science*, pp. 164–175, Springer, 2006.
- [20] Y. Zhu, M. Huang, S. Chen, and Y. Wang, "Energy-efficient topology control in cooperative Ad Hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1480–1491, 2012.
- [21] J. Lin and M. A. Ingram, "SCT-MAC: a scheduling duty cycle MAC protocol for cooperative wireless sensor network," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 345–349, Ottawa, Canada, June 2012.
- [22] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 41–52, 2007.
- [23] M. Kohvakka, J. Suhonen, M. Hännikäinen, and T. D. Hämäläinen, "Transmission power based path loss metering for wireless sensor networks," in *Proceedings of the IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '06)*, pp. 1–5, September 2006.
- [24] M. Grant, S. Boyd, and Y. Ye, *Cvx: Matlab Software for Disciplined Convex Programming*, 2008.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

