Wireless Ad Hoc Networks Resilience Through Cooperative Communication

Ulisses Rodrigues Afonseca
Instituto Federal de Goiás - IFG
Campus Luziânia
Luziânia-GO, Brazil
urafonseca@ifg.edu.br

Thiago Fernandes Neves, Jacir Luiz Bordim
Department of Computer Science
University of Brasília - UnB
Brasília-DF, Brazil
tfn.thiago@cic.unb.br, bordim@unb.br

***Abstract*—Several studies predict the use of wireless ad hoc networks in support of critical missions like search and rescue and prevention of natural disasters. In order to improve network connectivity, Cooperative Communication (CC) has been explored as an alternative to connect isolated network components in wireless ad hoc settings. However, existing cooperative communication solutions rely on global topological information which may not be feasible in more realistic scenarios. This paper presents a self-organized and distributed solution to improve network connectivity by exploring the availability of cooperative communication links. Simulation results show that the proposed scheme has a low computation cost and provides a link recovery rate comparable to those obtained by centralized solutions.**

***Keywords*–Ad hoc networks; articulation; bridge; cooperative communication; critical edge; critical node; network resilience.**

I. INTRODUCTION

In an ad hoc network, nodes cooperate in relaying packets to each other to enable communication. In such scenarios, network connectivity is crucial once faraway nodes may not be able to communicate in case of network partitioning. Link and node failure are events that may occur during the course of operation. As urgent and critical tasks, such as search and rescue and the prevention of natural disasters depend on network connectivity, ways to prevent network partitioning and node isolation are of interest. Let $G = (V, E)$ be a undirected connected graph, where V is the set of nodes and E is the set of edges. A node $v \in V$ is an *articulation* or *cut-vertex* if its removal makes the graph disconnected. Similarly, an edge $e \in E$ is a *bridge* if its removal makes the graph disconnected. Note that bridge links are connected by two articulations nodes. In this work, articulation nodes sharing a bridge are referred to "bridge nodes".

Owing to their importance in preserving network connectivity, ways to identify articulation nodes and bridges has been investigated in the literature. Goyal and Caffery Jr [1] proposed a centralized mechanism to identify articulations in wireless networks. Later, Jorgic et al. [2] presented a distributed solution, where each node performs a k-hop depth-first search to locate and identify articulation nodes and bridges using localized information. The proposed solution, however, has the drawback of false positive detections. Chaudhuri [3] and Turau el at. [4] proposed algorithms based on distributed depth-first search to determine the articulations of a graph. These solutions are optimal in time and number of messages and work with the knowledge of directly connected neighbours only.

As viable solutions to locate bridges and articulation nodes have been developed, the research community focused on alternatives to extend the availability of such nodes and links as well as in ways to reestablish network connectivity in case of failure [2][3][4]. Afonseca et al. [5] proposed ways to reduce energy consumption of articulation nodes by using packet aggregation techniques. The solution is based on the fact that energy consumption of articulation nodes is usually higher than other nodes, leading to a premature node failure. Khelifa et al. [6] propose the usage of dormant nodes that could be activated in case of link or node failure. When necessary, the dormant nodes would be activated to prevent network partitioning. In the same line, Goyal and Caffery Jr [1] proposed the usage of limited, coordinated, mobility so as to recover from link and node failures. In case of network disruption, nodes would move in a coordinated way as to reestablish network connectivity. As the location of articulation nodes are assumed to known a priory, node activation and dispatch can be employed to improve network connectivity in such areas. Yu et al. [7] employed *cooperative communication* techniques to connect disjoint network components using cooperative links. The proposed technique allows transmitting nodes to transpose the limit of maximum transmission range by allowing multiple nodes to relay the same information, thus improving network connectivity [7][8]. Neves et al. [9] exploited cooperative communications to establish power efficient links and routes to a sink node.

Despite of its benefits, all the above solutions rely on global topological information which may not be feasible to obtain and maintain due to its operational costs. Also, it seems unlikely that a network would have enough dormant nodes with the ability to be activated and dispatched to the necessary location whenever needed. Hence, cooperative communication seems to be a suitable approach to improve network connectivity. However, the aforementioned works that explore this path aimed to locate the least power link cost that can connect disjoint components before operation. This work takes a different approach by focusing on localized mechanisms to prevent network partitioning due to node and link failure. More precisely, the proposed scheme works by identifying critical elements, such as bridges and articulation nodes, and uses cooperative communication to create cooperative links to avoid network partitioning whenever possible. The proposal scheme employs cooperative communication based on localized topology information and works in a distributed, self-organizing, manner. Simulation results show that the proposed scheme has a low computation cost while link recovery rate is comparable to those obtained by centralized solutions.

The rest of the paper is organized as follows: Section II presents a review on cooperative communication, defines the communication model and formalizes the problem addressed in this work. Section III presents a distributed solution to recover network connectivity. In Section IV, the simulation process and the data collected are presented, and finally, Section V concludes this paper.

II. Communication Model and Problem

Cooperative Communication (CC) aims at enabling the cooperation of nodes for transmitting their messages to the destination [7][8]. Rather than operating independently, competing with each other for channel resources, nodes form a virtual Multiple Input Multiple Output (MIMO) system and simultaneously transmit the same information. In MIMO, nodes use a set of antennas to transmit and receive data to combat signal fading. Cooperative communication is similar, but uses multiple nodes in a two-step process [7][10]. In the first step, a node, called source, sends information for a subset of the nodes directly connected, called “helper nodes". In the second step, the source and helpers send the same data simultaneously. Thus, cooperative communication enjoys the same benefits of a conventional MIMO system. Further details on the characteristics of the cooperative communication can be found in Hong et al. [8]. The subsequente sections describe the cooperative communication model considered, present a brief overview of the closely related works and formally defines the problem addressed in this work.

A. Cooperative Communication Model

The communication between two nodes, in the traditional model, can be simplified in terms of the transmission power, the distance between nodes and the rate of signal fading. Thus, consider a network modelled as a planar, undirected graph $G(V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of wireless nodes and E is the set of communication links. Each node v_i can adjust its transmit power p_i with values in the range $[0, P_{MAX}]$. When $p_i = 0$, the transceiver is turned off and when $p_i = P_{MAX}$, the transceiver operates at full power. In traditional models of communication, the source v_i can communicate directly with the destination node v_j only when the transmission power of v_i complies with (1):

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

where α is the exponent of signal fading, usually around 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 τ 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.

In cooperative communication, the transmission power required by the source in conjunction with helper nodes can be determined similarly to the direct communication. Full communication between nodes v_i and v_j can be obtained with CC, if v_i transmits its signal with an auxiliary node set $H_{i,j}$ and the sum of transmission powers satisfies (2).

$$\sum_{v_k \in v_i \cup H_{i,j}} P_k(d_{k,j})^{-\alpha} \geq \tau \qquad (0 \leq P_i \leq P_{MAX}). \tag{2}$$

In the cooperative communication model, new concepts are introduced whereas the new edges in the graph can not be defined using classical concepts. Then, some important definitions on the model of cooperative communications, similar to those presented by Zhu et al. [10], are shown:

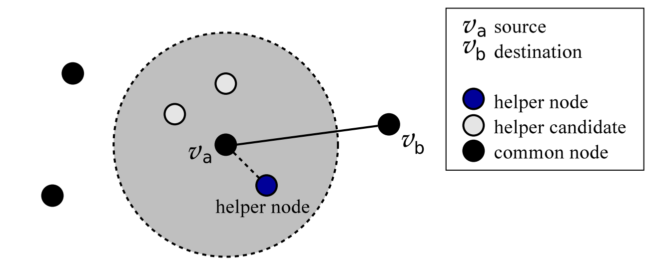


Figure 1. Example of the cooperative communication.

Definition 2.1: (Direct link): A direct link $\overline{v_iv_j}$ is an edge in E representing that the node v_i can transmit data to node v_j directly, that is, p_i is such that the node v_i can reach v_j when $p_i \leq P_{MAX}$. A solid horizontal line on the nodes represents a direct link.

Definition 2.2: (Helper node set): $H_{i,j}$ is the set of helper nodes of v_i in a cooperative communication with v_j. It is assumed that all required helper nodes are direct neighbours of v_i, that is, $H_{i,j} \subseteq N(v_i)$, where $N(v_i)$ is the set of all direct neighbours of v_i. In other words, all elements in $N(v_i)$ are candidates for helper nodes.

Definition 2.3: (CC-link): A CC-link $\widetilde{v_iv_j}$ is an edge of E representing that a node v_i can transmit data to v_j cooperatively using a set of auxiliary nodes $H_{i,j}$. A horizontal wavy line is used to denote a CC-link.

Definition 2.4: (Helper link): A helper link is an edge between v_i and one of his helpers in $H_{i,j}$.

Definition 2.5: (Network topology): The union of all direct links and CC-links, $\overline{E}$ and $\widetilde{E}$, respectively. Similarly, the graph of direct communication and CC communication are denoted by $\overline{G} = (V, \overline{E})$ and $\widetilde{G} = (V, \widetilde{E})$, respectively. Note that $E = \overline{E} \bigcup \widetilde{E}$. Also, if $v_iv_j \in E$, then: $v_iv_j = \overline{v_iv_j}$ if v_iv_j is a direct link; and $v_iv_j = \widetilde{v_iv_j}$ if v_iv_j is a CC-link.

Definition 2.6: (Weight of direct link): The weight of a direct link $\overline{v_iv_j}$ is defined as: $w(\overline{v_iv_j}) = \tau d_{i,j}^{\alpha}$.

Definition 2.7: (Weight of a CC-link): The weight of a CC-link $\widetilde{v_iv_j}$ is defined as:

$$w(\widetilde{v_iv_j}) = w_d(H_{i,j}) + (|H_{i,j}| + 1)w_{CC}(H_{i,j}),$$

where:

- $|H_{i,j}|$: is the number of elements in $H_{i,j}$;
- $w_d(H_{i,j}) = \left(\frac{\tau}{\max_{v_k \in H_{i,j}}(d_{i,k})^{-\alpha}}\right)$: is the maximum power consumption of the node v_i to communicate with the farthest node in $H_{i,j}$;
- $w_{CC}(H_{i,j}) = \left(\frac{\tau}{\sum_{v_k \in v_i \bigcup_{H_{i,j}}}(d_{k,j})^{-\alpha}}\right)$: is the minimum power consumption of the node v_i to communicate directly to v_j, together with their 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 $w_{CC}(H_{i,j})$. Figure 1 shows a cooperative communication example. The radius of maximum transmission, represented

 ISBN: 978-1-61208-368-1

in a grey circle, shows that node v_a has three neighbours. To communicate with the destination node v_b, out of its reach, the source node v_a uses a helper candidate sharing a direct link to v_a.

B. Related Works

In wireless ad hoc networks, CC have been used as a topology control mechanism with the aim to improve network connectivity and while reducing power consumption [7][10][11]. For networks initially without full connectivity, CC was used to transpose the maximum transmission range as a mean to improve network connectivity [7][9]. Yu et al. [7] use CC as a topology control mechanism, whose purpose is to connect disjoint components through cooperative links. The proposed solution, called CoopBridges, increases the network connectivity while reducing the transmission power at each node. The authors proposed a heuristic to select power efficient helpers nodes to reduce power consumption of the nodes sharing a CC-link. Starting from an undirected, disconnected graph, CoopBridges uses the proposed heuristic to create cooperative edges to connect components in the network. In the resulting topology, the minimum spanning tree algorithm is employed within each component and between network components to prune costly links. The task of selecting power efficient helper nodes works in $O(|V|^2)$ time. Neves et al. [9] developed a similar mechanism that interconnects the components of an ad hoc network, initially with no direct connectivity with a sink node. The solution consists of four steps that uses a modified version of the heuristic proposed by Yu et al. [7]. The solution uses low cost cooperative edges to interconnect the network such that paths created should lead to the sink node. The task of selecting power efficient helper nodes takes $O(|V|^2)$ time [9].

This work presents a localized and proactive strategy to maintain network connectivity in the event of network partitioning. Unlike the previous works, that are based on coordinated mobility or dormant nodes, the proposed scheme uses cooperative communication. Localized information is used to reduce the computational cost to select helper nodes during the process of establishing cooperative links. To the best of our knowledge, this is the first work to employ cooperative communication in a proactive way to prevent network partitioning.

C. Problem Formulation

This paper address the problem of recovering network connectivity in the event of node and link failure on ad hoc networks with cooperative communication capabilities similar to those in [7][8][10][12][13]. In particular, this work focuses on monitoring articulation points and bridges nodes and, in case of unavailability, cooperative communication is employed as to reestablish network connectivity. Consider an ad hoc wireless network represented by a planar, undirected graph $G(V,E)$ such that there are a number of articulation points and bridges. By definition of articulation point and bridge, on the event of unavailability of one of these elements, the graph G becomes disconnected. Let G_a and G_b be the components of G that have been created due to the unavailability of an articulation point or a bridge node. Furthermore, let node $v_a \in G_a$ and $v_b \in G_b$ and let $H(v_a)$ and $H(v_b)$ be the set of helper nodes available to v_a and v_b, respectively. Thus, using cooperative communication, the proposed scheme aims to reconnect the graph G. In this context, the problem addressed in this work is three fold: (i) locate articulations and bridges in the network; (ii) monitor their status and; (iii) in case of unavailability, coordinate the activities of the neighbouring nodes to recovery connectivity with the aid of cooperative communication.

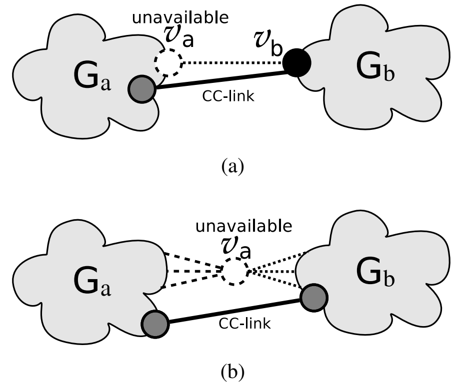


Figure 2. Representation of the problem of connectivity recovering using CC after a (a) bridge edge failure and (b) an articulation node failure.

III. PROPOSED SOLUTION

This section presents a distributed algorithm that allows the network to reestablish connectivity in case of articulation node failure. The main idea is to proactively identify suitable CC-links and to ensure that these CC-links are created in case of network connectivity disruption. Figure 2 shows an example of connectivity recovery in cases of bridge and articulation failure. The dotted edges represent topology changes that effect the communication links. When a bridge node fails, according to Figure 2a, collaborative communication is used to recover connectivity by establishing a cooperative link among the remaining bridge node and nodes in the vicinity of the failed bridge node. When there is only one articulation node, identified by v_a in Figure 2b, that connects two components, collaborative communication is employed as an alternative to reconnect the graph.

To achieve the above, the proposed scheme uses two-hop information to allow articulation nodes to periodically update their neighbours so that they can create CC-links to maintain network connectivity in the case of articulation node failure. To perform power efficient selection of helper nodes, the Greed Helper Set Selection ($GHSS$) heuristic, proposed by Yu et al. [7], is employed as a routine in the main algorithm, similarly to [10]. A call to the heuristic $GHSS$ takes as input parameters the pair (v_s, v_d), where v_s is the source node and v_d is the destination node, the output of this call is the cost of the CC-link $\widetilde{v_s v_d}$ or ∞ if it is not possible to create the CC-link. Note that, despite the related works that propose the increase of network connectivity using CC, the goal of the proposed solution focus on connectivity recovery. Another important aspect of the proposed solution is the processing type and amount of information used to reconnect the network. The proposed solution is distributed and uses localized information, while other proposals in the literature are centralized and require global topological information. The subsequent section details the proposed scheme.

A. Reconnecting Components (RC)

The proposed solution, called *ReconnectComponents* (RC), is detailed in Algorithm 1 (Figure 3). The algorithm considers that each node knows (i) whether it is part of a bridge or not,

Algorithm 1 RC($articulation$, $bridge$, S)

\# Articulation node forming a bridge notifies a
\# neighbouring nodes to replace it
1: **if** ($articulation = TRUE$ and $bridge = TRUE$) **then**
2: Let v_a and v_b be the nodes sharing a bridge. G_a and G_b are the network component connected to v_a and v_b, respectively;
3: Let $N_{v_a}(G_a)$ be the set of neighbours of v_a in G_a;
4: **for** each S seconds **do**
5: v_a computes $GHSS(v_i, v_b)$ and $GHSS(v_i, v_b)$, $\forall v_i \in N_{v_a}(G_a)$, and finds v_k, $v_k \in N_{v_a}(G_a)$, such that the combined cost of CC-links $\widetilde{v_k v_b}$ and $\widetilde{v_b v_k}$ are minimum;
6: v_a send $RECOVER(v_k, v_b)$ to nodes v_k and v_b;
7: **end for**
8: **end if**

\# Articulation node computes the CC-link cost to
\# connect its neighbouring nodes using local information
9: **if** ($articulation = TRUE$ and $bridge = FALSE$) **then**
10: Let v_a be an articulation node running the algorithm;
11: Let G_i and G_j be two network components that are connected to v_a such that $G - v_a$= $G_i \bigcup G_j$;
12: Let $N_{v_a}(G_a)$ be the set of direct neighbours of v_a in component G_a;
13: **for** each S seconds **do**
14: v_a computes $GHSS(v_i, v_j)$ and $GHSS(v_j, v_i)$, $\forall v_i \in N_{v_a}(G_a)$ and $\forall v_j \in N_{v_a}(G_b)$, and find v'_i and v'_j, $v'_i \in N_{v_a}(G_a)$ and $v'_j \in N_{v_a}(G_b)$, such that the combined cost to create the CC-links $\widetilde{v'_i v'_j}$ and $\widetilde{v'_j v'_i}$ are minimum;
15: v_a send $RECOVER(v'_i, v'_j)$ to nodes v'_i and v'_j;
16: **end for**
17: **end if**

\# Actions taken by nodes receiving a RECOVER msg
18: Let $\{v_i, v_j\}$ be the set of nodes receiving a $RECOVER(v_i, v_j)$ message;
19: Let v_a be the articulation node the sent the $RECOVER$ message;
20: **while** true **do**
21: **if** v_a is unavailable **then**
22: Create the CC-links $\widetilde{v_i v_j}$ and $\widetilde{v_j v_i}$;
23: **end if**
24: **end while**

Figure 3. Algorithm Reconnect Components

and (ii) whether it is an articulation or not. This knowledge can be obtained by running algorithms such as those proposed in [2][3]. Besides these requirements, the algorithm takes as input a parameter S that indicates the time interval in which the cooperative links are computed and updated to accommodate eventual topological changes.

Bridge nodes calculates the bidirectional cooperative link having the least cost between the adjacent articulation and its direct neighbours (lines 1-5). A message is sent to the elected nodes that compose the cooperative link (line 6). Nodes receiving the $RECOVER(v_i, v_j)$ message should monitor the source of the message (lines 18-24). Should the articulation node become unavailable, the cooperative link is then used to maintain connectivity (line 22).

When a node is an articulation and does not compose a bridge, it calculates the cooperative link with least power cost between each pair of neighbouring nodes in the components interconnected by it (line 9-14). After that, a message is sent to the selected nodes (line 15). Note that, up to this point, the articulation only elects nodes to replace it and the links supported by the articulation node. It should be noted that after an articulation node failure, that is not associated to a bridge, a new bridge is created and therefore two new bridge nodes appear in the graph. In this case, the connectivity can

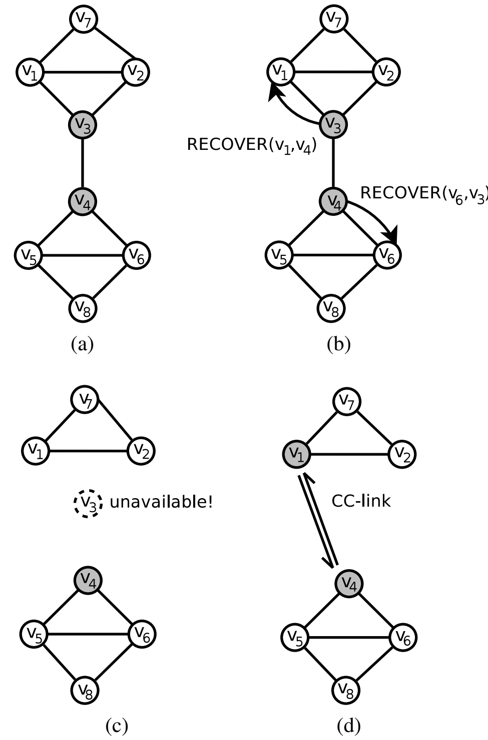


Figure 4. Example of connectivity recovery after a bridge node failure.

be maintained continuously using the same strategy.

B. A Working Example

Figure 4 and 5, respectively, show the sequence of events for recovering the connectivity when a bridge and an articulation become unavailable. Figure 4a presents the initial topology in which nodes v_3 and v_4 are bridge nodes. In Figure 4b, these nodes notify neighbours that should take over its function in case of failure. In Figure 4c, articulation v_3 fails and in Figure 4d a cooperative link is created between nodes v_1 and v_4. In Figure 5b, the articulation node v_4 notify nodes v_2 and v_6 that they have been elected to create a cooperative link. In Figure 5c, when the articulation v_4 fails, a cooperative link is created and the new topology is presented in Figure 5d. In the resulting topology, there are two new articulation nodes and a bridge.

C. Computational Cost

The $GHSS$ routine is used to perform the required helper selection in the RC algorithm (lines 5 and 14) and is also used, as in previous works, to measure the computational cost of the proposed algorithm. For this purpose, let $\Delta(G)$ denote the maximum degree of a node in G. Also, let v_a and v_b denote two nodes connected by a bridge. According to previous definitions, node v_a and v_b have $N_{v_a}(G_a)$ and $N_{v_b}(G_b)$ neighbours, respectively. Note that $N_{v_a}(G_a) \subset N(v_a)$ and $N_{v_b}(G_b) \subset N(v_b)$. Then, the task of computing the best set of helper nodes to create CC-links (line 5) makes at most $2[N(v_a) + N(v_b)]$ calls for the $GHSS$ routine. In the case where a node v_a is an articulation node such that $G - v_a$= $G_i \bigcup G_j$, node v_a computes the best set of helper nodes among its one and two-hop neighbouring nodes. Hence, in the worst case, the direct neighbours of v_a, say $v_i \in N(v_a)$ has a degree of at most $\Delta(G)$. As the heuristic to compute the best set of helper nodes needs to verify all alternatives among the

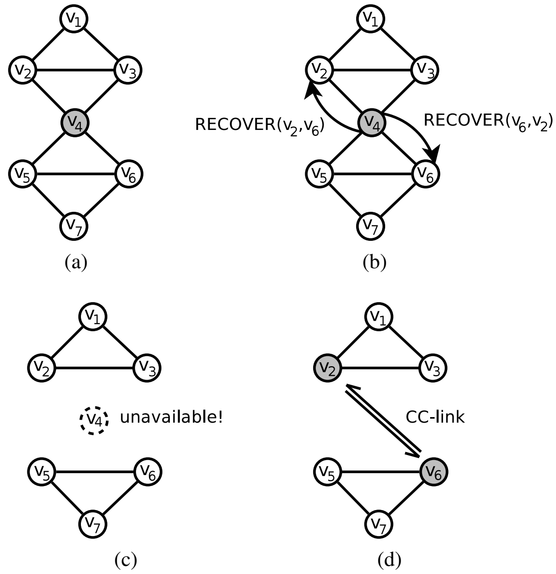


Figure 5. Example of connectivity recovery after a failure in an articulation.

nodes in $N(v_i)$ that connect to $v_j \in G_j$ and vice-versa, node v_a makes, in the worst case, $O(\Delta(G)^2)$ calls to the $GHSS$ routine. Considering that $2[N(v_a) + N(v_b)] \leq O(\Delta(G)^2)$, the RC algorithm uses, in the worst case, $O((\Delta(G))^2)$ calls to the $GHSS$ to select the best helper set to establish a CC-link.

IV. Evaluation and results

The proposed solution has been evaluated by simulation. The validation process consists in: (i) generate random topologies; (ii) identify articulation nodes and bridge nodes; (iii) employ the RC algorithm to compute the best CC-links and check whether these links are able to reconnect the network in case of articulation and bridge nodes failure. To assess the goodness of the proposed solution, the resulting CC-link cost is compared with those produced by the centralized scheme presented by Yu et al. [7].

The evaluation scenarios are based on the following parameters (similarly to those in [9][10][13]): a set of nodes $n = 20, 30, ..., 60$ are randomly positioned in a 300×300m area. Equation (1) can be easily adapted to a more suitable path loss models. Hence, in what follows, Free Space Path Loss Model [14] is considered, where the maximum transmitting power (P_{MAX}) is set to 6dBm and the receiver threshold (τ) is set to -71dBm, allowing to a maximum transmission range (R_{MAX}) of $\approx$ 70m on the 2.4GHz frequency band [14]. To compare the performance of the proposed solution, the following metrics were considered:

M1: **Computational cost:** Aims to evaluate the overhead (in terms of calls to $GHSS$) to select the helper set with least power cost to create CC-link using distributed and a centralized solutions;

M2: **Power cost:** The amount of transmission power needed to establish the CC-links;

M3: **Percentage of recovered connectivity**: identifies the percentage of graphs that had connectivity recovered;

To compute **M1**, first a random topology is generated using the defined parameters. Then, the articulation nodes and bridges are identified in the graph and the best helper sets are computed. For a defined node density, the simulation results are drawn from an average of a three hundred random topologies. Note that, as the purpose of **M1** is to compare the computational cost, the parameter S has no effect in this case. Table I presents the simulation results for metric M1. The column "density" corresponds to the number of nodes in the graph. Columns "Articulations" and "Bridges" correspond to the number of calls to the $GHSS$ heuristic. The column "Global" shows the same results when the best cooperative bi-direction link among all the nodes is computed. Note that the degree of each node is random. Thus, as the node density gets higher, nodes tend to have a larger node degree and, consequently, increasing the computational cost. However, even by increasing the degree of the graph, the proposed solution still has a scalable computational cost. As can be seen in the Table, the proposed algorithm obtained, for the evaluated cases, a reduction of up to 67 times the number of calls to the GHSS routine.

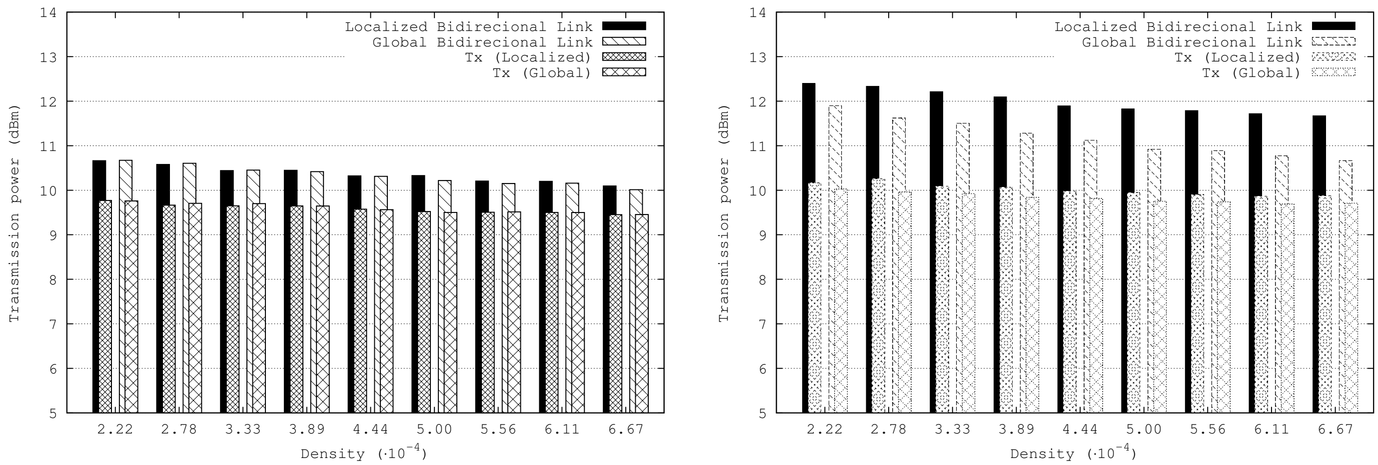
The simulation results for metric **M2** are shown in Figure 6. In the figure, the x-axis represents the graph density (number of nodes per area) and the y-axis represents the power consumption for the computed CC-links. As can be seen in Figure 6a, the proposed algorithm has an power consumption slightly higher than then that provided by the global bi-directional link when an articulation connecting two components fails. This trend is also verified in Figure 6b that shows the power consumption to reestablish connectivity in case of an bridge node failure. The average power consumption reduces as the network density increases. This occurs as the articulation and bridge nodes have potentially more nodes that may act as helper nodes. With closer helper nodes, the cost $w_d(H_{i,j})$ decreases and the weight of the cooperative link $w(\widetilde{v_i v_j})$ tends to reduce as well. Despite the slightly higher power consumption of the proposed solution, it is important to state that the RC relies solely on local information.

Suppose that the nodes could increase the transmission power beyond the P_{MAX} up to the limit necessary to reestablish network connectivity without the aid of a CC-link. Although this is an unlikely situation, it provides a lower bound on the minimum amount of power necessary to reestablish communication. Also in Figure 6b, the bar `Tx(Localized)` corresponds to the minimum transmission power needed to reestablish connectivity without resorting to CC-links in the resulting topology using localized information. Similarly, the `Tx(Global)` bar corresponds to the amount of power necessary to reestablish connectivity using global information. When a direct link are considered, the nodes selected in both cases present comparable results in terms of the required transmitting power to reestablish network connectivity.

For the metric **M3**, percentage of graphs that had con-

TABLE I. NUMBER OF CALLS TO THE GHSS ROUTINE (M1).

Density ($\times 10^{-4}$)	Articulation	Bridge	Global
2.22	13.22	14.78	174.04
2.78	14.52	16.38	276.98
3.33	15.62	17.18	408.20
3.89	16.74	18.86	562.78
4.44	17.78	21.98	742.00
5.00	18.90	21.36	950.72
5.56	19.26	24.96	1177.96
5.11	20.34	22.48	1431.12
6.67	22.12	25.24	1710.72



(a) (b)

Figure 6. Average power cost to recover connectivity in case of: (a) an articulation node failure; and (b) a bridge node failure.

nectivity recovered, it was observed that the proposed solution presents similar results to the global alternative. This results are not shown due to space limitation. Nevertheless, the observed results shows that, for node density up to 2.5, a success rate of 98% by employing global information while RC attains 96.5%, only 1.5 points below the extensive evaluations. For node density with values between 2.5 and 5.0, both algorithms were able to recover network connectivity in approximately 99% of the cases. On graphs with node density above 5.0, both algorithms have been able to reconnect the graphs in all evaluated cases.

V. Conclusion

Maintain connectivity in wireless ad hoc networks is a goal that has been addressed in many ways, most of them focusing on identifying critical nodes and implement mechanisms the preserve these nodes using efficient routing, packet aggregation, among other techniques [2][3][4]. This work explored cooperative communication to reconnect the network using distributed processing and localized knowledge. Bidirectional links are created between network components when articulation and bridge nodes fail. The main contribution of this work is to present an algorithm that reduces the computational cost when using localized information that offers resilience when monitoring critical elements, creating cooperative links when conventional links become unavailable. Simulation results demonstrate that the proposed solution provides similar results of more costly solutions that rely on global topological information. For the scenarios evaluated, the computational cost of the proposed scheme was 67 times lower than centralized solution, while producing comparable results.

Acknowledgment

This work was partially supported by CNPq, CAPES and UnB/DPP.

References

[1] D. Goyal and J. Caffery Jr, "Partitioning avoidance in mobile ad hoc networks using network survivability concepts," in Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on. IEEE, 2002, pp. 553–558.

[2] M. Jorgić, I. Stojmenović, M. Hauspie, and D. Simplot-Ryl, "Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks," Proceedings of the 3rd IFIP Mediterranean Ad Hoc Networking Workshop, 2004, pp. 360–371.

[3] P. Chaudhuri, "An optimal distributed algorithm for finding articulation points in a network," Computer Communications, vol. 21, no. 18, 1998, pp. 1707–1715.

[4] V. Turau, "Computing bridges, articulations, and 2-connected components in wireless sensor networks," in Algorithmic Aspects of Wireless Sensor Networks. Springer, 2006, pp. 164–175.

[5] U. R. Afonseca, P. H. Azevêdo Filho, J. L. Bordim, and P. S. Barreto, "Reducing energy consumption of articulation points in wireless sensor networks," in II Workshop de Sistemas Distribuídos Autonômicos, 2012, pp. 21–24.

[6] B. Khelifa, H. Haffaf, M. Madjid, and D. Llewellyn-Jones, "Monitoring connectivity in wireless sensor networks," in Computers and Communications, 2009. ISCC 2009. IEEE Symposium on. IEEE, 2009, pp. 507–512.

[7] J. Yu, H. Roh, and W. Lee, "Topology Control in Cooperative Wireless," IEEE Journal on Selected Areas in Communications, vol. 30, no. 9, 2012, pp. 1771–1779.

[8] C.-C. J. K. Y.-W. Peter Hong, Wan-Jen Huang, Cooperative Communications and Networking: Technologies and System Design. Springer Science & Business Media, 2010.

[9] T. F. Neves and J. L. Bordim, "Topology control in cooperative ad hoc wireless networks," Electronic Notes in Theoretical Compututer Science, vol. 302, 2014, pp. 29–51.

[10] Y. Zhu, M. Huang, S. Chen, and Y. Wang, "Energy-efficient topology control in cooperative ad hoc networks," Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 8, 2012, pp. 1480–1491.

[11] M. Cardei, J. Wu, and S. Yang, "Topology control in ad hoc wireless networks using cooperative communication," Mobile Computing, IEEE Transactions on, vol. 5, no. 6, 2006, pp. 711–724.

[12] M. Uysal, Cooperative Communications for Improved Wireless Network Transmission: Framework for Virtual Antenna Array Applications. Information Science Reference, 2010.

[13] J. Yu, H. Roh, W. Lee, S. Pack, and D.-Z. Du, "Cooperative bridges: Topology control in cooperative wireless ad hoc networks," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–9.

[14] J. C. Liberti and T. S. Rappaport, Smart Antennas For Wireless Communications: IS-95 And Third Generation CDMA Applications, ser. Prentice Hall Communications Engineering And Emerging Technologies Series. Upper Saddle River, N.J.: Prentice Hall PTR, 1999.