UNIVERSITÄT BERN

University of Bern
Institute of Computer Science

Throughput- and Cost-aware Node Relocation for MANET Resiliency Under Jamming Attacks

Bachelor thesis submitted by
Marco De Liso

Supervised by
Prof. Dr. Torsten Braun
Dr. Antonio Di Maio

Communication and Distributed Systems Research Group
Bern, Switzerland

21 August, 2023

University of Bern
Institute of Computer Science

1. **Title of the Thesis**: Throughput- and Cost-aware Node Relocation for MANET Resiliency Under Jamming Attacks

2. **Name, Designation and Institution of the Supervisor/s:**

   (a) **Prof. Dr Torsten Braun**
   Professor, CDS Research Group
   University of Bern, Switzerland
   torsten.braun@unibe.ch

   (b) **Dr. Antonio Di Maio**
   Post-doctoral Researcher, CDS Research Group
   University of Bern, Switzerland
   antonio.dimaio@unibe.ch

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

A	
ASPL	**A**verage **S**hortest **P**ath **L**ength
B	
BSA	**B**ackup **S**election **A**lgorithm
D	
DARTS	**D**istributed **A**pt **R**esource **T**ransference **S**ystem
DEBTCM	**D**ifferential **E**volution **B**ased **T**opology **C**ontrol **M**echanism
E	
ENM-LAC	**E**volving **N**etwork **M**odel based on **L**ocal-**A**rea **C**hoice
I	
IDS	**I**ntrusion **D**etection **S**ystem
ILP	**I**nteger **L**inear **P**rogramming
M	
MANET	**M**obile **A**d Hoc **NET**work
O	
OCRS	**O**bstacle-avoiding **C**onnectivity **R**estoration **S**trategy
P	
POT	**P**ython **O**ptimal **T**ransport
R	
RJR	**R**eactive **J**oint **R**elocation

RR	**R**andom **R**elocation
RSR	**R**eactive **S**equential **R**elocation
S	
SINR	**S**ignal-to-**I**nterference-plus-**N**oise **R**atio
V	
VT	**V**oronoi **T**esselation
VUE	**V**oronoi-based **U**niformity **E**valuation

List of Symbols

A list of all the symbols, their representative quantities and units has been provided below so as to familiarize the readers with the frequently used symbols within this thesis.

System Modeling

σ_{jk} The number of shortest paths from device j to device k

$\sigma_{jk}(i)$ The number of shortest paths from device j to device k passing through device i

$d(p_i, p_j)$ The euclidean distance between the positions of the devices i and j

$G(V, E)$ The graph G holding a set of V vertices and a set of E edges

$h(i, j)$ The topological distance (i.e., number of hops) between nodes i and j

s The source node

t The sink node

x_{ij} The volume of data that moves along the edge (i, j)

y_{ts} the connection volumes y_{ts} of the net information leaving from the source s and arriving at the destination t

Sets and Domains

A The set of jammed nodes

D_g The domain in which the candidate relocation positions of the jammed nodes

D_g^* The domain of relocation positions of current jammed node

E The set of edges

K The set of source-destination couples (s, t) that need to exchange information

K_V The set containing the connection volumes y_{ts}

p The set of all node positions before the jamming attack

V The set of verteces

Parameters

$\boldsymbol{\alpha}$ The coefficients' vector, contains the coefficient for each metric of the objective function

a The jammer

c The maximum flow capacity $c \in (0, +\infty)$ [bit/s]

g The grid dimension

R_{a} The radius of the jammer or jamming range

R_{d} The radius of the device or transmission range

Optimization metrics

$\boldsymbol{v^*}$ The vector of optimized jammed nodes' positions

$\boldsymbol{v}$ The vector of candidate positions for the current jammed node

τ The network's global throughput

C_b The betweenness centrality measure

C_c The closeness centrality measure

C_d The degree centrality measure

d_m The average distance walked from initial positions to relocation positions

T The maximum global throughput of the network [Mbit/s]

Abstract

Over the last decade, the use of Mobile Ad Hoc Networks (MANETs) has grown tremendously, providing us with wireless networks that can be set up on-the-fly in remote areas where there is no pre-existing network infrastructure. As the use of MANETs increases, so does the number of cyber attacks to disrupt their operation. In particular, jamming attacks can prevent critical network nodes to communicate and relay information, which seriously disrupts network performance.

In this work, we propose Reactive Sequential Relocation (RSR), an efficient relocation strategy to mitigate the impact of jamming attacks by physically moving critical nodes in the network outside the attacked area, optimizing throughput, network resiliency, and relocation cost (i.e., relocation distance). We formulate an associated joint relocation problem Reactive Joint Relocation (RJR), show that it is NP-hard (non-deterministic polynomial-time hardness), and introduce a heuristic solution in RSR. Extensive simulation studies show that RSR solves the RJR problem efficiently while simultaneously leveraging centrality measures, maximum global throughput, and mean distance to generate the most optimal final relocation outcome. Compared to baseline methods such as Random Relocation (RR) and RJR, RSR improves the maximum global throughput by up to 227% on average while decreasing the optimization runtime by up to five-folds and decreasing the average relocation distance by up to 51%.

CHAPTER 1

Introduction

Mobile Ad Hoc Networks (MANETs) are wireless networks formed by a collection of mobile devices that communicate with each other without needing a pre-existing infrastructure. MANETs emerge as a popular technology for a wide range of applications, such as military communications, disaster response, and sensor networks.

However, due to the dynamic and decentralized nature of MANETs, they are susceptible to a wide range of security threats, including node failures, malicious attacks, and communication disruptions. Ensuring the resiliency of MANETs is critical for maintaining network connectivity and providing uninterrupted communication services in the face of adverse conditions. In emergency response systems in disaster-stricken areas, MANET networks play a crucial role in enabling communication between first responders, such as firefighters and medical teams. MANETs are also extensively used in military scenarios where traditional infrastructure-based communication may be unavailable or compromised. Resiliency ensures the network can withstand disruptions, enabling secure and reliable communication during emergencies, among troops in the field, in harsh environments, or in the presence of adversaries. Therefore, there is a growing need to develop robust and efficient solutions to enhance the resiliency of MANETs.

One of the most critical attacks to a MANET is the *jamming attack*, performed by a *jamming device* that emits a noise signal to disrupt the communication link between two devices by causing partial or total signal cancellation. MANETs are particularly vulnerable to jamming attacks due to their decentralized nature and lack of centralized control, requiring distributed countermeasures to mitigate its negative impact. In addition, jamming attacks are unpredictable, making defending against those more challenging. Jamming attacks can result in dangerous losses of communication channels and isolation, leading to harsh conditions.

This work's main contributions are:

- We formalize the joint node relocation problem as an Integer Linear Programming problem, named Reactive Joint Relocation (RJR). We show its NP-hardness (non-deterministic polynomial-time hardness) and, through simulations, its unfeasible computation time (exponential growth) for real-world scenarios.
- We propose Reactive Sequential Relocation (RSR), a low-complexity, polynomial-time heuristic node relocation strategy algorithm that physically moves the jammed devices outside the jammed area while improving network resiliency and performance, minimizing relocation costs.
- Through simulations, we show that the RSR can recover the MANET from jamming attacks and provides the highest post-attack global throughput and network resiliency compared to baseline algorithms under several levels of attacker strength.
- We show that RSR dramatically reduces optimization runtime compared to RJR, up to a five-fold factor.
- Finally, we show that the more a jamming attack is powerful (i.e., the larger the jammed radius), the more RSR can provide a lower average relocation distance compared to baseline methods.

The remainder of this work is organized as follows. In Section 2, we review the existing literature on MANET resiliency, including the key challenges, limitations, and opportunities for future research. In Section 3, we discuss the system model and problem formulation used to assess the effectiveness of our MANET resiliency solution. In section 4, we explain the main joint-node relocation strategy that we develop and how it improves resiliency. Finally, we conclude the work in Section 5 and highlight the future research directions in this field.

CHAPTER 2

Related Works

In the pursuit of addressing complex research problems, scholars and researchers have navigated two overarching approaches: proactive and reactive methodologies. These distinct strategies represent divergent paths toward problem-solving, each with its unique set of strengths and limitations.

The proactive approach centers on anticipation and prevention, aiming to mitigate potential issues before they fully manifest. On the other hand, the reactive approach is characterized by its responsiveness to challenges as they arise, seeking to develop solutions in real-time.

Both approaches contribute significantly to the academic landscape, shaping the discourse on effective problem-solving paradigms. In this section, we delve into the existing related works encompassing these two methodologies, exploring their theoretical and practical applications and considering their pros and cons.

2.1 Proactive Related Works

Most of the related works considered focus on proactive MANET resiliency techniques. In these works, most researchers apply topological and performance analysis techniques, such as [1], [2], [4], [5], [7], [15], whereas others use optimization methods [6] and a fault-tolerant relocation exploration method [9]. Proactive techniques have a disadvantage as they aim to enhance the initialization of MANETs and improve the network's resilience before an attack occurs. This approach may reduce attack frequency or its power, but it cannot guarantee that they will not occur. Therefore, there is a need to implement proactive approaches that cooperate with reactive techniques to reduce the threat probability and recover from it when it occurs. Moreover, proactive measures necessitate substantial data volumes and analyzing numerous scenarios to develop effective countermeasures, which is time-consuming. Implementing all the factors to improve the MANET initialization to enhance resiliency may be too costly, which motivates the research on cost-aware MANET resiliency methods. Another consideration is that the analysis may not be able to capture all potential attack scenarios, and it may be difficult to account for the dynamic nature of MANETs.

In contrast, reactive approaches focus on providing quick recovery abilities, which can help to minimize or even nullify the damage caused by an attack. Therefore, in scenarios where attacks are unpredictable and rapidly evolving, a reactive approach may be more effective compared to a proactive approach.

2.2 Reactive Related Works

Similar to our work, some studies, such as [3], [8], [10]–[14], try the reactive approach to provide a solution against physical obstacles or cyber-attacks disrupting the communication quality of the network. In [8] and [14], the authors propose approaches to let drones complete their mission under a jamming attack. On the one hand, Tedeschi et al. [8], use a Mathematical solution based on the JAM-ME strategy, an autonomous jamming-assisted navigation system. On the other hand, Mah et al. [14] focus on optimization methods to provide a solution. These two approaches only focus on enabling drones to complete their mission autonomously during a jamming attack without attempting to reestablish communication in the network by considering important factors such as communication range. This limitation is significant, particularly when multiple devices communicate in the same MANET. Furthermore, these solutions are inadequate for addressing various jamming attack scenarios due to this limitation.

The other reactive works, such as[12] and [10], [11], are quite different. In [12], authors employ methods based on the gyroscopic force for obstacle avoidance. Other works, such as [10], [11], use a Voronoi-Tessellation-based reactive approach to enhance network resiliency by utilizing relocation to avoid physical obstacles or node failure. However, these methods cannot be applied to jamming attack scenarios for the following reasons. In the case of Mi et al. [12], their underlying assumption states that communication between devices is always available, and the positions of those devices are always known. Therefore, they are assuming that there is no jamming attack. On the other hand, solutions provided by Kusyk et al. [10], [11] might increase the number of devices in the jammed area. This increase is because their solutions are not optimized for jamming attacks and rely solely on local obstacles or communication ranges for their relocation strategies. This can lead the non-jammed devices to relocate inside the jammed area to cover the lost transmission range, as there is no apparent physical obstacle. Furthermore, since they relocate the other non-jammed devices to reestablish communication coverage, this approach only aims to recover communication range without considering reestablishing the jammed devices, which can lead to massive losses if the jammed nodes are many. Lastly, since their primary emphasis is transmission range coverage, this approach does not directly improve other critical resiliency aspects of MANETs, such as connectivity, robustness, and transmission capacities.

2.3 Motivation and Goals

The lack of reactive strategies to improve resiliency on MANETs against jamming attacks is the main reason that motivates this research. Our work aims to solve the jamming attack problem with a new approach that improves MANETs in a reactive technique called RSR for MANET resiliency. The strategy of reactive node relocation, which involves moving nodes to new positions in order to reestablish communication, is rarely utilized as a method to restore connectivity between nodes. Therefore, the study we bring will further explore this technique to solve jamming attack problems.

Table 2.1: Related Works

Article	**Approach**	**Method**
[1]	Proactive	Assurance Network and topological analysis
[2]	Proactive	Theoretical Analysis with Resiliency Evaluation
[3]	Reactive	Fault Propagation
[4]	Proactive	Evolving Network Model based on Local-Area Choice (ENM-LAC) and Theoretical analysis
[5]	Proactive	Human Walk pattern and Topology analysis
[6]	Proactive	Optimization
[7]	Proactive	Topology Analysis for Critical Nodes Detection
[8]	Reactive	Mathematical with JAM-ME
[9]	Proactive	Distributed Apt Resource Transference System (DARTS) and Intrusion Detection System (IDS)
[10]	Reactive	Rel-NSPG, Genetic Alg. and Game Theory
[11]	Reactive	Performance Analysis and Voronoi-based Uniformity Evaluation (VUE)
[12]	Proactive + Reactive	Obstacle-avoiding Connectivity Restoration Strategy (OCRS) and Backup Selection Algorithm (BSA)
[13]	Reactive	Differential Evolution Based Topology Control Mechanism (DEBTCM), Optimization and Voronoi Tesselation (VT)
[14]	Reactive	Optimization of UAV flight path
[15]	Proactive	Fault Propagation and Topology Analysis for Congestion Prevention
RSR	**Reactive**	Sequential Node Relocation

CHAPTER 3

System Model and Definitions

MANETs are being widely used in modern technologies. However, none of the solutions can effectively address the jamming attack problem when applied in an urban environment, nor can they offer an efficient relocation mechanism in the presence of communication disruptions caused by jamming attacks. The entities that we will utilize in the method to address the jamming attack are defined hereafter.

3.1 System Model

Our scenario contains a set of devices, partitioned into a set of non-jammed devices, a set A of jammed devices, and a jammer device a, located in a square scenario of unit size (1×1). We define p as the vector of all *node positions* before the jamming attack. We assume that every jammed and non-jammed device has a circular transmission range with radius R_d to communicate with their neighbors and that the jammer device has a circular transmission range with radius R_a (jamming range) that interferes with the received signal at the other devices. This means that any device $\in A$ whose position is within a disk of radius R_a around the jammer cannot receive information from any other device, becoming effectively isolated from the network. We further assume the jamming attack to be perfect, which means that the devices inside the jamming range cannot communicate with other devices. Therefore, as soon as a device is jammed, it will no longer be able to exchange information with other devices, relying on the last known information gathered from the network as a reference for the relocation protocol. The jammer's position is fixed, while we allow the non-jammed devices and the jammed devices to move within the square scenario at a significantly lower speed than the relocation algorithm's execution time scale. We assume that the devices can detect the jamming event (e.g., using a machine learning-based anomaly detection technique such as in [16]) and estimate the jammer position (e.g., using [17]) with sufficient accuracy. Network nodes periodically disseminate their position and list of neighbors to all other reachable nodes so that each device maintains an updated perception of the network topology and nodes' locations in case of a jamming attack.

We model our network as a graph $G = (V, E)$, where V represents the set of vertices and E represents the set of edges $E = \{(i, j)|i, j \in V\}$. Each vertex represents a device, and the edges represent communication links between devices (i.e., the devices are *connected*). Furthermore, two devices are *connected* if they are located within each other's transmission range R_d, i.e., $d(p_i, p_j) < R_\mathrm{d}$, with $i, j \in V$ and $i \neq j$, where $d(p_i, p_j)$ is the Euclidean distance between devices i and j.

We quantify the *resiliency* of a network topology graph G through three metrics, namely the *average betweenness centrality*, the *average degree centrality*, and the *average closeness centrality*, defined hereafter. Let us define the number of shortest paths from device $j \in V$ to device $k \in V$ as $\sigma_{jk} \in \mathbb{N}$, and the number of shortest paths from device $j \in V$ to device $k \in V$ passing through device i as $\sigma_{jk}(i) \in \mathbb{N}$. The centrality measures C_b, C_d, and C_c are quantitative metrics that measure the average "importance" of nodes in a network and express the network resiliency to attacks.

In this context, the *average betweenness centrality* of a graph G is $C_\mathrm{b} = \frac{1}{|V|}\sum_{i\in V}\sum_{\substack{j,k\in V\\ i\neq j, i\neq k, j\neq k}} \frac{\sigma_{jk}(i)}{\sigma_{jk}}$, which represents how often nodes act as relay along shortest paths in the network on average.

$C_\mathrm{d} = \frac{1}{|V|}\sum_{i\in V} |\{j \in V|(i, j) \in E \vee (j, i) \in E\}|$ is the *average degree centrality* of a graph G, which represents how connected each node is to its neighbors on average.

Finally, $h(i, j)$ is the topological distance (i.e., number of hops) between nodes $i \in V$ and $j \in V$, expressed in the number of edges of the shortest path between i and j. In this context, $C_\mathrm{c} = \frac{1}{|V|}\sum_{i\in V}\left(\frac{1}{|V|-1}\sum_{\substack{j\in V\\ j\neq i}} h(i, j)\right)^{-1}$ is the *average closeness centrality* of a graph G, which represents the average length of the shortest path between a node and any other destination in the graph.

The network contains a set $K = \{(s, t) \in V^2\}$ of source-destination couples $(s, t) \in V^2$ that need to exchange information, named *connections*. We denote $K_V = \{y_{s,t} \in [0, +\infty), \forall(s, t) \in K\}$ as the set containing the *connection volumes*

$y_{ts} \in [0, +\infty)$ of the information volume leaving from the source s and arriving at the destination t, each associated with a connection $(s,t) \in K$ in the system. If source and destination nodes are not directly connected, they exchange information across a *route*, i.e., a multi-hop path from a source device $s \in V$ to a destination device $t \in V$, relayed by other intermediate nodes $\in V$. A *flow* x_{ij} on edge (i,j) is the volume of data that moves along the edge $(i,j) \in E$ for all connections in the network. In our system, all edges $(i,j) \in E$ have the same maximum flow capacity $c \in (0, +\infty)$, expressed in bit/s. We define the network's *global throughput* $\tau = \sum_{y_{ts} \in K} y_{ts}$ as the sum of all source-destination throughputs in the system. For a set of connections in a network represented by a graph G, we can define the associated *maximum global throughput* T as the maximum sum of all connection volumes $y_{ts} \in K$, obtained as the solution of the Optimization Problem 3.1 under a set of feasibility constraints, i.e., flow conservation and edge flow constraints [18]. The flow conservation constraints (Equation 3.1b) ensure that the difference between incoming and outgoing flows for each node is zero, except for nodes that are source or destination for one or more connections, where it equals the difference between incoming and outgoing connection volumes. The edge flow constraints (Equation 3.1c) ensure that edge flows cannot be negative or exceed the edge capacity c and the positive connection volume constraints (Equation 3.1d) ensure that connection volumes y_{ts} are not negative.

$$\underset{x_{ij}, \forall (i,j) \in E}{\text{maximize}} \quad \tau = \sum_{y_{ts} \in K} y_{ts} \tag{3.1a}$$

subject to

$$\sum_{\substack{j \in V: \\ (j,i) \in E}} x_{ji} - \sum_{\substack{j \in V: \\ (i,j) \in E}} x_{ij} = \sum_{\substack{j \in V: \\ (i,j) \in K}} y_{ij} - \sum_{\substack{j \in V: \\ (j,i) \in K}} y_{ji}, \quad \forall i \in V \tag{3.1b}$$

$$0 \leq x_{ij} \leq c, \qquad \forall\ (i,j) \in E \tag{3.1c}$$

$$0 \leq y_{ts}, \qquad \forall (t,s) \in K \tag{3.1d}$$

3.2 Problem Formulation: Reactive Joint Relocation

Let us assume that our scenario is covered by a square lattice of points (*grid*), where points along the two orthonormal directions are separated by a distance $g \in (0,1)$, called *grid dimension.* Therefore, the scenario is covered by a set $D_g \subset \mathbb{R}^2$ of *potential relocation positions*, where each element is a point on the $\mathbb{R}^2$ plane. The set D_g contains the only potential relocation positions of jammed nodes, and its cardinality is in the order of $\left(\frac{1}{g}+1\right)^2$. The example in Figure 3.1 shows a network example with the square lattice of points. Furthermore, it also depicts how a transmission range R_d (in this case of node 7) looks like.

The problem we aim to solve is to find the *optimal relocation positions* vector $\boldsymbol{v}^* = (v_1^*, \ldots, v_{|A|}^*) \in \mathbb{R}^{2\times|A|}$ for all jammed nodes $i \in A$ simultaneously. To do this, we need to optimize a *candidate relocation positions'* vector $\boldsymbol{v} = (v_1, \ldots, v_{|A|}) \in \mathbb{R}^{2\times|A|}$, where $|A|$ is the number of jammed nodes and each vector element $v_i \in \mathbb{R}^2$ represents the plane coordinates of the candidate relocation position for the jammed node $i \in A$. The value of $\boldsymbol{v}^*$ should maximize a utility function that considers aspects of resiliency, relocation cost, and network performance. In particular, the utility function should maximize the metrics C_b, C_d, C_c and T to enhance resiliency, robustness, and connectivity of the network, and minimize the average relocation distance d_m to reduce relocation time and energy spent.

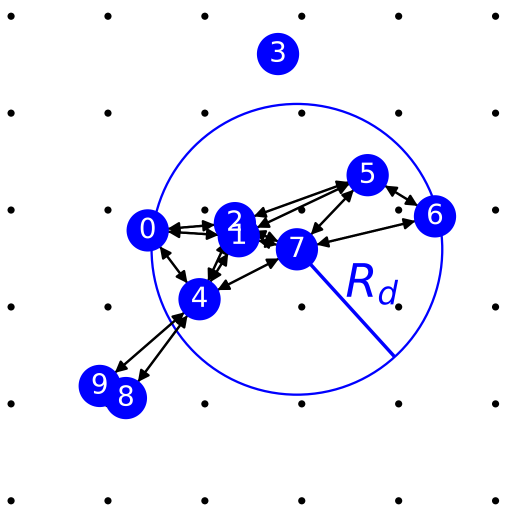

Figure 3.1: Example of a simulation scenario showing the grid lattice of points in black and the communication range R_d of node 7.

We define the *average relocation distance* $d_m = \frac{1}{|A|}\sum_{i\in A} d(p_i, v_i)$ of a set of jammed nodes A to a vector of candidate relocation positions $\boldsymbol{v}$ as the average distance between a node's current position p_i and its candidate relocation position v_i for all jammed nodes A. The relocation distance is associated with an energy and time cost to perform the relocation, therefore we only consider relocation distance as an aggregate measure of the relocation cost.

Therefore, we design the optimization problem's utility function as a linear combination of the *metrics vector* $(T, -d_m, C_\mathrm{b}, C_\mathrm{d}, C_\mathrm{c}) \in \mathbb{R}^5$, scaled by a *sensitivity vector* $\boldsymbol{\alpha} \in [0,1]^5$ (with $||\boldsymbol{\alpha}||_1 = 1$). Each component α_i of the sensitivity vector scales the importance of the corresponding metric in the utility function, either amplifying or reducing the impact of a particular variable relative to others.

Finally, we can define the optimization problem RJR (which is integrated into the Algorithm 1) in Equation 3.2.

$$\boldsymbol{v}^* = \underset{\boldsymbol{v} \in D_g^{|A|}}{\arg\max} \quad \boldsymbol{\alpha}^\top (T, -d_m, C_\mathrm{b}, C_\mathrm{d}, C_\mathrm{c}) \tag{3.2}$$

The optimization problem in Equation 3.2 is NP-hard: its solution requires a number of operations in the order of $\left(\frac{1}{g}+1\right)^{2|A|}$, leading to an exponential worst-case time complexity $\mathcal{O}\left(1/g^{2|A|}\right)$.

Algorithm 1: Reactive Joint Relocation (RJR)

```
Input: Network topology graph $G$, set of candidate relocation positions
       $D_g$
Output: Optimal relocation positions $\boldsymbol{v}^*$
// Evaluate optimality for every combination of relocation
   positions
1  forall $s \in D_g^{|A|}$ do
2      $U^* \leftarrow -\infty$
       // Generate new graph moving jammed nodes $\in A$ to positions
          in combination $s$
3      $G_\mathrm{new} \leftarrow$ CreateGraphStructure($G, s$)
       // Compute maximum global optimal throughput on new graph
4      $T \leftarrow$ ThroughputOptimization($G_\mathrm{new}$)
5      $\{C_\mathrm{b}, C_\mathrm{d}, C_\mathrm{c}\} \leftarrow$ GetCentralities($G_\mathrm{new}$)
6      $d_m \leftarrow$ GetRelocDistance($p, s$)
       // Compute utility for the combination of relocation
          positions $s$
7      $U \leftarrow \boldsymbol{\alpha}^\top(T, -d_m, C_\mathrm{b}, C_\mathrm{d}, C_\mathrm{c})$
8      if $U > U^*$ then
9          $U^* \leftarrow U, \quad \boldsymbol{v} \leftarrow s$
// Update all jammed nodes' position
10 $p \leftarrow \boldsymbol{v}^*$;
11 return $\boldsymbol{v}^*$
```

The Algorithm 1 presents a method to compute the optimal solution of RJR and shows the computational unfeasibility at line 1. In fact, line 1, the RJR algorithm generates all possible combination positions associated with the jammed nodes, since we aim to find the best set of positions that maximizes Equation 3.2. In lines 3-6, RJR iterates over all candidate combinations of relocation positions ($s \in D_g^{|A|}$) and finds for each of them the optimization metrics. Finally, in lines 7-9, RJR determines the utility of each candidate combination of relocation positions $s \in D_g^{|A|}$ and picks the s associated with the highest utility among all candidate combinations in $D_g^{|A|}$.

The worst-case time complexity depends on $|A|$ and the number of relocation positions. Therefore, the optimization problem's time complexity dramatically increases as the grid density and number of jammed nodes increase, due to the larger number of available relocation positions and number of jammed nodes, respectively. Solving such scenarios becomes computationally unfeasible for large-scale problems. For this reason, we propose a heuristic method to solve the RJR expression in Equation 3.2 with polynomial complexity, detailed in Chapter 4.

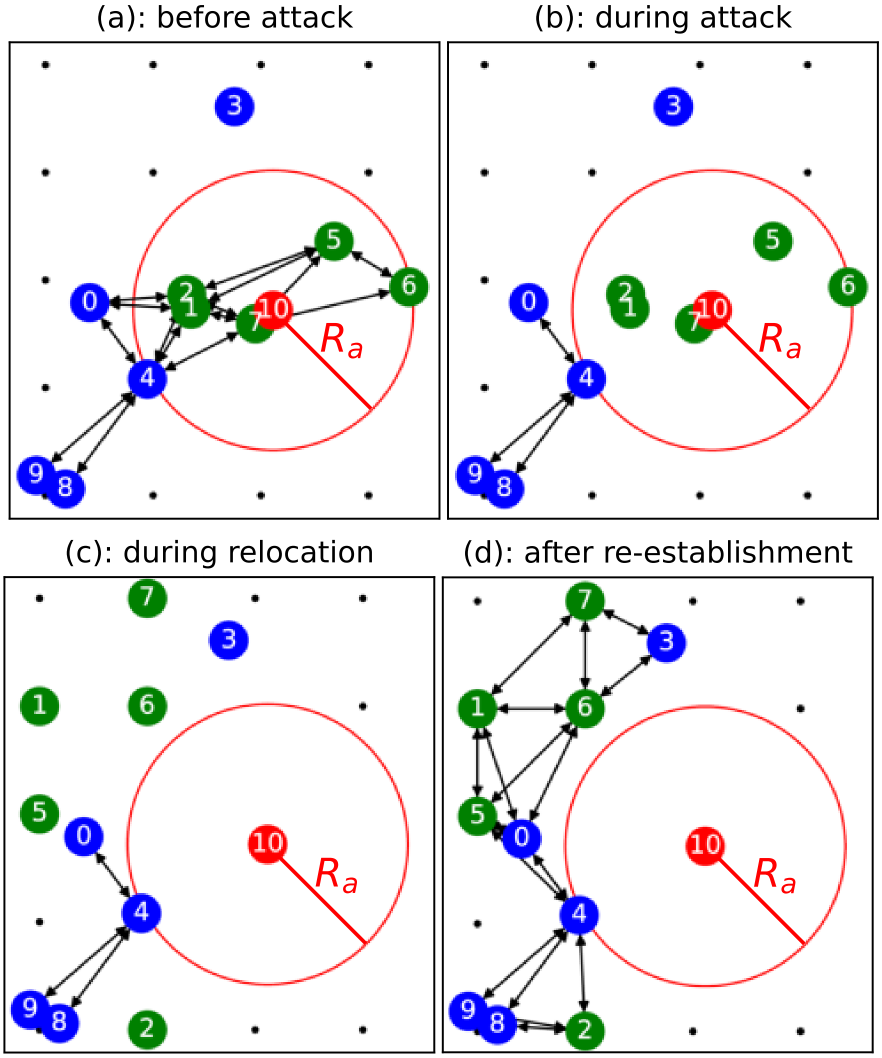

Figure 3.2: A simulated example scenario of a jamming attack. Subfigure (a) shows the network before the jamming attack. Subfigure (b) shows the network during the jamming attack with all jammed nodes (in green) disconnected. Subfigure (c) shows the network at the end of the relocation procedure. Finally, subfigure (d) shows the network after the links are re-established.

As an example of the optimization problem's goal, we consider the network in Figure 3.2a where the jammer is depicted in red, its jamming range R_a is represented as a circular area, the non-jammed devices are depicted in blue, the jammed devices are shown in green, and the potential relocation positions are indicated with a grid of black points. The optimization problem's objective is relocating the isolated jammed nodes in Figure 3.2b into an optimal configuration that involves the jammed nodes migrating in their optimal relocation position as shown in Figure 3.2c and finally recovering the communication links with the neighborhood (as depicted in Figure 3.2d), also connecting previously isolated nodes (see node 3 in Figure 3.2a).

CHAPTER 4

Reactive Sequential Relocation

This section details our proposed RSR scheme, a heuristic relocation strategy that overcomes the NP-hardness limitation of the joint relocation problem through a sequential relocation of individual nodes.

RSR approximates RJR optimization problem's optimal relocation while reducing its computational complexity by applying two heuristic mechanisms. First, it finds the optimal relocation position for each individual jammed device sequentially instead of simultaneously relocating all jammed nodes in a single joint optimization procedure. Second, it reduces the size of the candidate relocation position set D_g, by generating a *filtered relocation position set* D_g^* using three performance-oriented heuristic filters, detailed hereafter in this section.

Algorithm 2 summarizes RSR's operation. First, RSR identifies the set $|A|$ of jammed devices by selecting the nodes located within the jamming range R_a (i.e., $A = \{i \in V | d(p_i, p_\mathrm{a}) < R_\mathrm{a}\}$), then it proceeds to find the optimal relocation position v_i for each jammed device $i \in A$. Then, in Algorithm 2 lines 15-25, RSR identifies the set D_g^* of filtered relocation position by applying three heuristic filters on the set of all relocation positions D_g: (1) all filtered relocation positions must be within the communication range of at least one non-jammed device in the network (line 19); (2) the devices should be accessible from all other nodes in the network to prevent network partitioning (line 21); (3) none of the filtered relocation positions should fall within the jamming area (line 23). After computing D_g^*, in the inner loop of the Algorithm 2 lines 4-9, RSR determines the utility of relocating each jammed device to every potential filtered relocation position $\in D_g^*$. To determine

Algorithm 2: Reactive Sequential Relocation (RSR)

Input: Network topology graph G, set of candidate relocation positions D_g

Output: Optimal relocation positions $\boldsymbol{v}^*$

```
// Compute the set of filtered relocation positions
$D_g^* \leftarrow$ FilterPositions($D_g$)
// Repeat for every jammed node
forall $i \in \{1, \dots, |A|\}$ do
    $U^* \leftarrow -\infty$
    // Explore every candidate relocation position
    foreach $s \in D_g^*$ do
        // Generate new graph moving jammed node $i$ to position $s$
        $G_{\text{new}} \leftarrow$ CreateGraphStructure($G, s$)
        // Compute maximum global optimal throughput on new graph
        $T \leftarrow$ ThroughputOptimization($G_{\text{new}}$)
        $\{C_{\text{b}}, C_{\text{d}}, C_{\text{c}}\} \leftarrow$ GetCentralities($G_{\text{new}}$)
        $d_m \leftarrow$ GetRelocDistance($p, s$)
        // Compute utility for relocation of $i$ to $s$
        $U \leftarrow \boldsymbol{\alpha}^\top(T, -d_m, C_{\text{b}}, C_{\text{d}}, C_{\text{c}})$
        if $U > U^*$ then
            $U^* \leftarrow U, \quad v_i^* \leftarrow s$
    // Update the $i$-th node position
    $p_i \leftarrow v_i^*$
return $\boldsymbol{v}^* = (v_1^*, \dots, v_{|A|}^*)$
```

```
// Compute set of possible relocation positions for the current
   jammed nodes
Function FilterPositions($D_g$):
    $D_g^* \leftarrow \emptyset$
    foreach $k \in D_g$ do
        if
          // Position $k$ is within at least one non-jammed device's
             transmission range
          $\exists\ j \in V \setminus A :\ d(p_j, k) < R_d$ and
          // Position $k$ has at least one path to any other node in
             network (connected graph)
          $\forall\ j \in V \setminus A : \sigma_{jk} > 0$ and
          // Position $k$ outside of jammer range
          $d(p_a, k) > R_a$ then
            $D_g^* \leftarrow D_g^* \cup k$
    return $D_g^*$
```

the utility of relocating node i from its original location p_i to the candidate filtered relocation position v_i, RSR must compute the values of each component of the metrics vector. Specifically, for the *candidate graph* G_{new} resulting by relocating node i to the candidate position v_i, RSR computes the metrics T, d_m, C_{b}, C_{d}, and C_{c}, and then combines them linearly using a fixed *sensitivity vector* $\boldsymbol{\alpha}$ (done in the Algorithm 2 lines 6-9). Finally, in lines 10-11, RSR picks the relocation position associated with the highest utility among all positions in D_g^* and repeats the procedure for the next jammed node $\in A$. After RSR terminates and assigns an optimal relocation position to all jammed nodes, the device proceeds to relocation or self-relocation toward their assigned destination position.

While the centrality measures C_{b}, C_{d}, C_{c}, and the mean distance d_m can be computed algebraically from G_{new} and d_m, the value of the maximum global throughput T must be obtained through solving a flow-allocation sub-problem, formulated as the Linear Programming (LP) Optimization Problem 3.1, using any LP solver method, such as the simplex method (see line 6 in the Algorithm 2).

We now show that RSR's implementation in Algorithm 2 has a sub-exponential time complexity. In the worst-case scenario, RSR does not exclude any possible grid point from D_g, and each jammed node in A is relocated sequentially. Therefore, RSR requires a number of operations in the order of $|A|\left(\frac{1}{g}+1\right)^2$ leading to worst-case time complexity of $\mathcal{O}\left(|A|/g^2\right)$ that is no longer exponential (as in RJR) but polynomial, namely linear in the number $|A|$ of jammed nodes and quadratic the grid density $1/g$.

CHAPTER 5

Performance Evaluation

5.1 Experiment setup

Table 5.1: Simulation Parameters

Parameter	**value/range**
Grid dimension g	$\{0.2, 0.3, 0.4, 0.5\}$
Number of jammers a	1
Number of Devices $\lvert V\rvert$	$10 + \lvert A\rvert$
Jamming range R_a	$\{0.10, 0.11, ..., 0.30\}$
Communication range R_d	0.25
Edge capacity c	15 Mbit/s
Coefficients vector $\boldsymbol{\alpha}$	$(1.2, 1.2, 0.2, 0.2, 0.2)/3$
Number of source-dest. connections $\lvert K\rvert$	10

Through simulations, we compare the performance of RSR with those of RJR and a Random Relocation (RR) of the jammed nodes beyond the jamming range. RR identifies candidate relocation positions right outside the jammer area and assigns them randomly to the jammed nodes. By comparing the outcomes of our approach with those of the RR algorithm, we want to highlight the performance and efficiency of RSR.

In the simulated scenarios, we select an optimizer's utility function that moderately prioritizes the maximum global throughput T and average relocation distance d_m over the centrality measures of resiliency. This is done because T and d_m determine the quality of the communication and the relocation costs in terms of distance, respectively. To achieve this, we assigned higher values to the components α_i of the sensitivity vector $\boldsymbol{\alpha}$ associated with T and d_m. We chose the specific value of $\boldsymbol{\alpha}$ as in Table 5.1 through empirical observations, which showed the desired tradeoff between throughput, relocation distance, and network resiliency. We evaluated RSR's performance in different simulated scenarios, measuring T, C_{b}, C_{d}, C_{c}, and d_m, for each scenario, namely before, during, and after the jamming attack. We evaluated the algorithms' performance, considering resiliency, robustness, energy costs (according to d_m), topology, and recovery capabilities when they operate in a range of random simulated scenarios, each characterized by different parameters.

The simulation scenarios contain one jammer device and ten legitimate devices. In addition, the network nodes' positions are determined by sampling from a Normal Distribution centered in the middle of the scenario and truncated at the scenario's boundaries. The positioning is designed to mimic real-world networks found in urban environments. We run 500 independent simulations to obtain asymptotic confidence intervals of the selected metrics' averages. Table 5.1 shows the simulation parameters. The devices operate using the Open Shortest Path First (OSPF) routing protocol and employ the IEEE 802.11-OCB (Outside the Context of a Basic service set mode) standard for communication, which is defined as the direct device-to-device communication without the need for device association with an Access Point belonging to a fixed wireless network infrastructure. Furthermore, OSPF ensures that the devices are constantly aware of the network situation until jamming is performed.

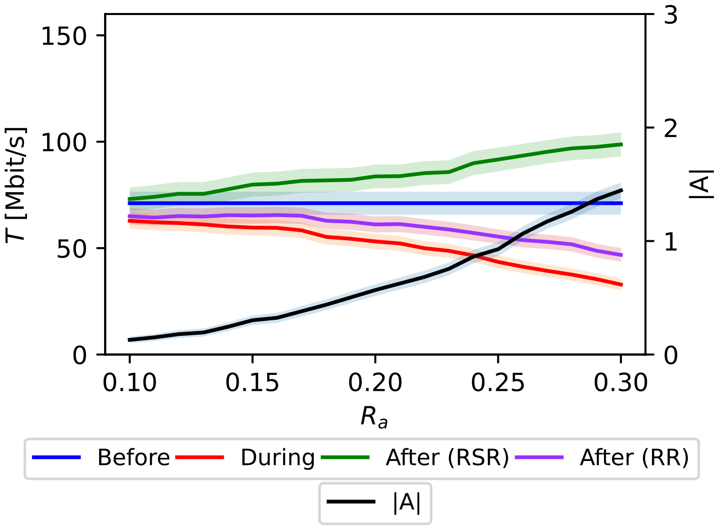

Figure 5.1: Maximum global throughput T performance for RSR and RR, under varying jamming radii R_a. Average of all grid sizes g.

5.2 Results Analysis

Metrics improvement: RSR compared with RR with $g = 0.2$					
Metric Name	$R_\mathrm{a} = 0.10$	$R_\mathrm{a} = 0.15$	$R_\mathrm{a} = 0.20$	$R_\mathrm{a} = 0.25$	$R_\mathrm{a} = 0.30$
T	15%	46%	78%	142%	227%
d_m	-574%	-340%	-148%	-44%	51%
C_b	19%	26%	46%	65%	97%
C_d	8%	12%	18%	26%	38%

Table 5.2: Algorithm's Resiliency and Performance comparison between RSR and RR. The values show the improvement or worsening of RSR compared to RR

Figure 5.1 shows the network maximum global throughput for all grid densities. The picture shows the average maximum global throughput improvement of RSR, which is up to 111% higher compared to RR in the worst-case scenario where the jammer has maximum power ($R_\mathrm{a} = 0.30$).

Figures 5.2 and 5.3 show the maximum global throughput in three different situations: before, during, and after the jamming attack (similarly to the four moments in Figure 3.2). The results demonstrate a general improvement in the network conditions compared to those during the jamming attack. In addition, the increase in performance of the network is higher with a denser grid dimension, which ranges from $g = 0.5$ (Figure 5.2a) until $g = 0.2$ (Figure 5.3b).

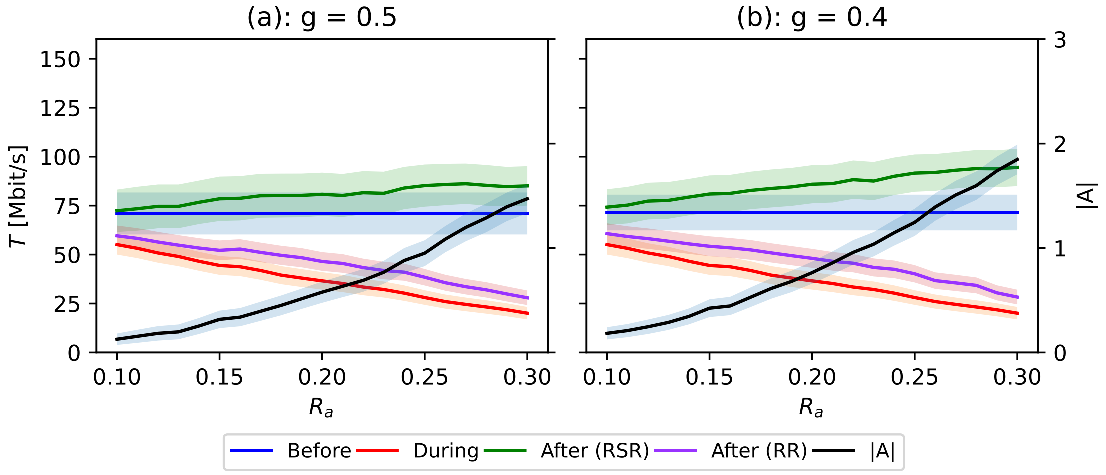

Figure 5.2: Global throughput T varying jamming radius R_a with grid dimensions $g = 0.5$ (a) and $g = 0.4$ (b). Plots show the performances of both relocation algorithms: RSRs and RRs.

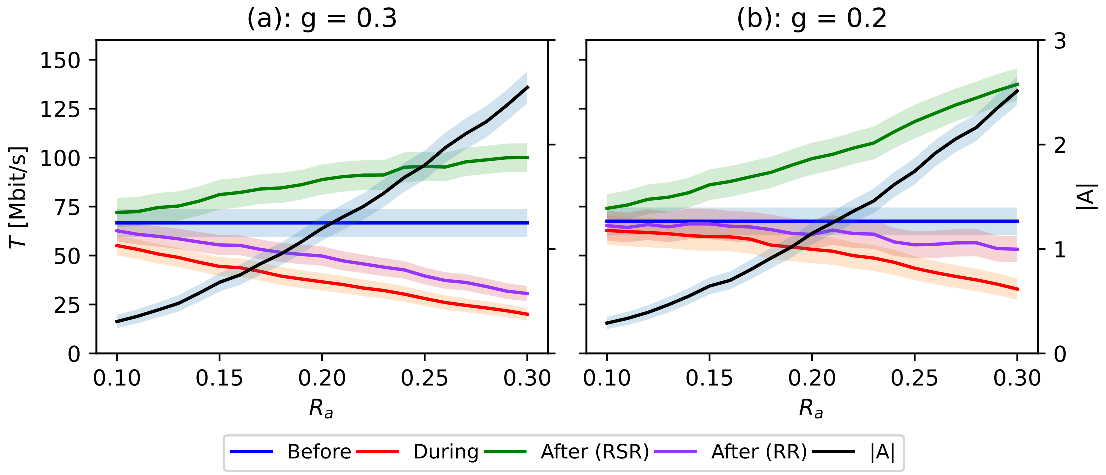

Figure 5.3: Global throughput T varying jamming radius R_a with grid dimensions $g = 0.3$ (a) and $g = 0.2$ (b). Plots show the performances of both relocation algorithms: RSRs and RRs.

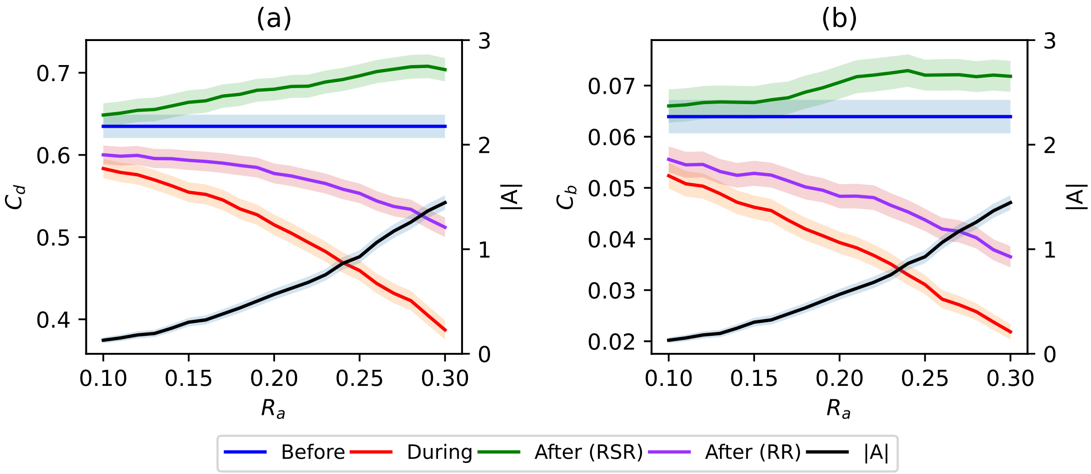

Figure 5.4: Network resiliency for RSRs and RRs, estimated using degree centrality C_d (a) and betweenness centrality C_b (b) under increasing jamming radii R_a and considering all grid dimensions.

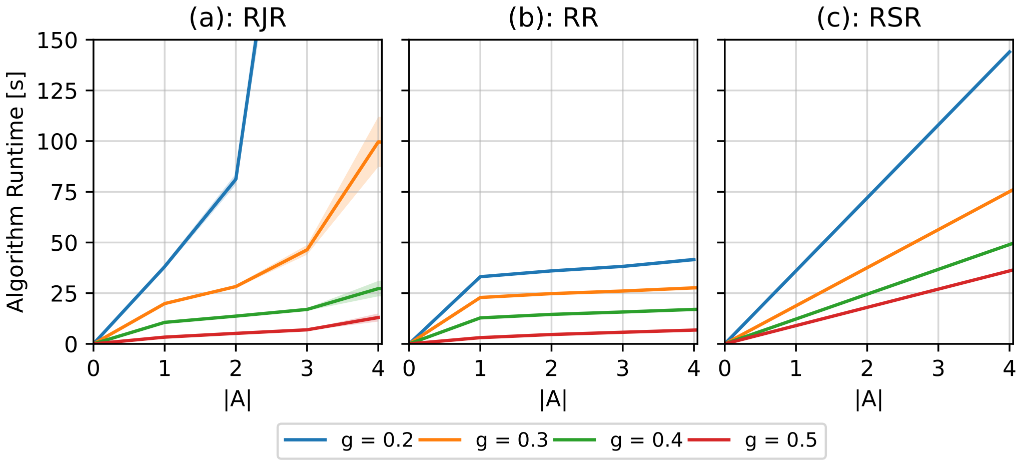

Figure 5.5: Runtime of RJR (a), RR (b) and RSR (c) for increasing number of jammed nodes $|A|$ and grid size g.

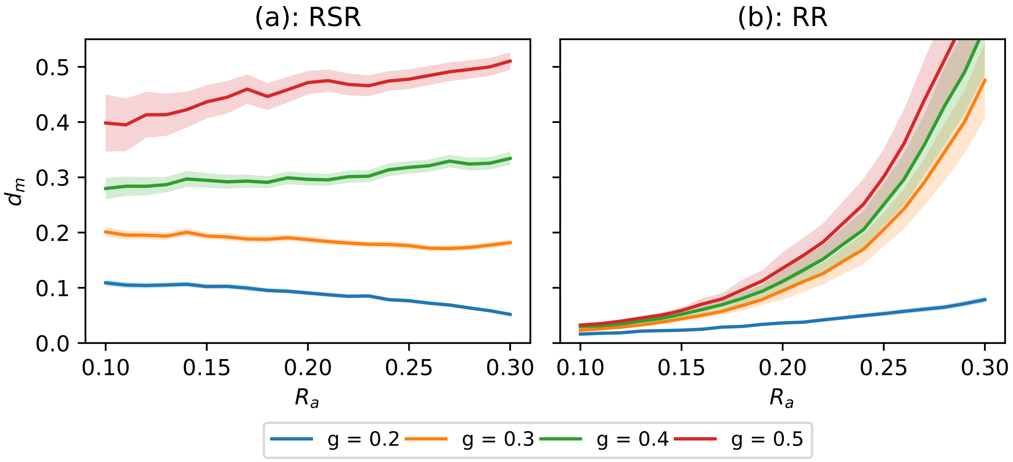

Figure 5.6: Average relocation distance d_m performance for (a) RSR and (b) RR for increasing jammer radii R_a.

In this context, the maximum global throughput improvement of RSR, as depicted in Table 5.2, is by up to 227% higher on average compared to RR in the worst-case scenario where the jammer has maximum power ($R_\mathrm{a} = 0.30$).

Figure 5.4 shows two of the centrality measures used for the optimization: the betweenness centrality and degree centrality. These plots demonstrate the algorithm's capability to improve the network's connectivity and resiliency, even under the harsh conditions brought by the jammer. Nodes with a high degree centrality have more connections in the network, which can improve overall connectivity. In the presence of a jammer, having high-degree devices can help to maintain alternative paths and bypass the affected areas, increasing the reliability and robustness of the network, hence the ability to withstand disruptions caused by the jammer. On the other hand, nodes with high betweenness centrality lay on many shortest paths between other devices in the network. Therefore, by optimizing their placement or allocation of resources, which was our intention with RSR, the algorithm can exert greater control over the flow of information and enhance network connectivity in the face of jamming attacks, improving robustness and reliability. Finally, as for the previous case, the RSR algorithm always outperforms the RR algorithm and the initial conditions of the network. The results for C_b and C_d, as illustrated in Table 5.2, provide additional evidence of the resilience outperformance of RSR in comparison to RR.

Figure 5.5 compares runtimes of the algorithms RJR, RR, and RSR. In particular, Figure 5.5a shows that the RJR's runtime grows exponentially with the number of jammed nodes $|A|$, highlighting its unfeasibility in large-scale realistic

Runtime comparison: RSR compared with RJR with $g = 0.2$				
Metric Name	$\|A\| = 1$	$\|A\| = 2$	$\|A\| = 3$	$\|A\| = 4$
Runtime	6%	13%	201%	573%

Table 5.3: Algorithm's runtime comparison between RSR and RJR. The values show the improvement of RSR compared to RJR.

scenarios regardless of the relocation's optimality in terms of performance. Figure 5.5b shows that the runtime of the selected baseline (RR) is linear with $|A|$, even though its performance is not robust against powerful jammers with increasing range R_{a}. Finally, Figure 5.5c shows RSR's runtime performance growing sub-exponentially (i.e., linearly with $|A|$ and quadratically with the grid density $1/g$), reducing up to five-folds the runtime compared to RJR while achieving the best performance of the relocated network. Table 5.3 shows further evidence of the fact that RSR outperforms RJR, illustrating the runtime reduction of RSR compared to the runtime of RJR.

Figure 5.6 shows the comparison of the average relocation distance of the jammed nodes between RSR (Figure 5.6a) and RR (Figure 5.6b). Since RR is solely based on the random relocation of the jammed nodes right outside the jamming range, the average relocation distance will, in most cases, be the minimum possible because no other factors are considered. Nevertheless, the results of RSR show neither a linear nor exponential increase of the mean distance while augmenting the jamming range, hence showing that the decision is not based only on the average distance itself. Furthermore, with denser grids (i.e., $g = 0.2$), RSR outperforms RR by up to 51% as depicted in Table 5.2, resulting in decreased average relocation distances for each jammed node. This is because while RR identifies the nearest relocation positions, these positions are randomly assigned to the jammed nodes, which means they do not always represent the shortest relocation distance for the node itself. Instead, the algorithm ensures the best possible solution given the objective function.

5.3 Discussion

The simulations demonstrate that denser grids yield better results, surpassing even the initial metrics' values of the network. This is due to the fact that the networks are randomized and do not follow a specific scheme, except for the function we used to simulate urban scenarios. This can lead to a not optimal node repositioning when talking about the resiliency and performance metrics we considered.

Furthermore, in scenarios with less dense grids (approximately 0.3 to 0.5), the algorithm can re-establish the network in most cases without significant runtime delays. However, the limited number of available positions restricts the extent to which connectivity and network quality can be improved. Furthermore, the algorithm successfully reconnects disconnected devices (as depicted in Figure 3.2) unless other priorities, such as multi-partitioning, take precedence.

Tables 5.2 and 5.3 reveal significant network condition enhancements when contrasting RSR to RR and RJR. Compared to the deteriorated network conditions (during the jamming attack) and the RR algorithm, the relocation protocol significantly enhances network connectivity, robustness, routing capabilities, and overall throughput. Each device is assigned to its nearest position in the relocation set, minimizing walking costs and lowering energy and resource expenses. Thanks to the filtering and optimization steps in the RSR algorithm, the relocation distance surpasses the RR algorithm in specific cases only.

CHAPTER 6

Conclusions

6.1 Conclusions of the Present Work

This work proposes RSR, a jamming attack countermeasure for MANETs, which physically relocates nodes to positions outside the jammed area that maximize network throughput and resiliency, and minimizes cost (relocation distance). We formulate the relocation problem as an optimization problem named RJR, prove its NP-hardness, and show its unfeasibility for large-scale scenarios experimentally. We show the influence of various metrics, including T, d_m, C_b, C_d, and C_c, in three distinct scenarios: pre-jamming attack, during the attack, and after the recovery protocol. Through a simulation study, we show that RSR outperforms RR performance by increasing maximum global throughput by up to 227% on average, improving resiliency by up to 97% of the average node centrality in the relocated network, reducing optimization runtime compared to RJR up to five times, with a negligible performance penalty, and reduce relocation distance by up to 51%.

6.2 Future Works

Future improvements to the methodology involve reducing the time complexity associated with the joint relocation and introducing additional obstacles to enhance the realism of scenarios, such as incorporating physical obstacles. The current optimization model focuses solely on global throughput; however, it can be expanded to the entire implementation, allowing the optimization of all relevant factors and identifying the optimal combination based on the assigned positions. This addition could lead to a further reduction in the time complexity of RSR, enabling the simulation of even larger scenarios and considering a relocation protocol in a continuous space, instead of discrete. In this context, RSR provides a relocation protocol based on a grid lattice of points with variable density, hence discrete, since the nodes are limited to relocate only in these points. However, there is no real physical obstacle that limits the jammed nodes. Therefore, they could be relocated also within the continuous space in which the scenario is defined (i.e. in between two points of the grid). This approximation is made only to reduce the time complexity of the algorithm, leading to an approximated relocation protocol. Overcoming the time complexity issue, the algorithm would perform well enough to allow a continuous space for the relocation protocol.

Bibliography

[1] Y. Kakuda and M. Malek, "A unified design model for assurance networks and its application to mobile ad hoc networks," in *2011 Tenth International Symposium on Autonomous Decentralized Systems*, Tokyo, Japan: IEEE, Mar. 2011, pp. 637–644, ISBN: 978-1-61284-213-4. DOI: 10.1109/ISADS.2011.91.

[2] A. Jabbar, H. Narra, and J. P. Sterbenz, "An approach to quantifying resilience in mobile ad hoc networks," in *2011 8th International Workshop on the Design of Reliable Communication Networks (DRCN)*, Krakow, Poland: IEEE, Oct. 2011, pp. 140–147, ISBN: 978-1-61284-125-0 978-1-61284-124-3 978-1-61284-123-6. DOI: 10.1109/DRCN.2011.6076896.

[3] A. B. Cavalcante and M. Grajzer, "Fault propagation model for ad hoc networks," in *2011 IEEE International Conference on Communications (ICC)*, Kyoto, Japan: IEEE, Jun. 2011, pp. 1–5, ISBN: 978-1-61284-232-5. DOI: 10.1109/icc.2011.5962675.

[4] S. Liu, D.-G. Zhang, X.-h. Liu, *et al.*, "Dynamic analysis for the average shortest path length of mobile ad hoc networks under random failure scenarios," *IEEE Access*, vol. 7, pp. 21 343–21 358, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2896699.

[5] D. Zhang and J. P. Sterbenz, "Measuring the resilience of mobile ad hoc networks with human walk patterns," in *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, Munich, Germany: IEEE, Oct. 2015, pp. 161–168, ISBN: 978-1-4673-8050-8 978-1-4673-8051-5. DOI: 10.1109/RNDM.2015.7325224.

[6] A. Agarwal and D. Mishra, "Hovering localization and power allocation for UAV assisted DF relaying ad hoc network," *IEEE Access*, p. 6, 2020.

[7] T.-H. Kim, D. Tipper, P. Krishnamurthy, and A. L. Swindlehurst, "Improving the topological resilience of mobile ad hoc networks," in *2009 7th International Workshop on Design of Reliable Communication Networks*, Washington, DC, USA: IEEE, Oct. 2009, pp. 191–197, ISBN: 978-1-4244-5047-3. DOI: 10.1109/DRCN.2009.5340008.

[8] P. Tedeschi, G. Oligeri, and R. Di Pietro, "Leveraging jamming to help drones complete their mission," *IEEE Access*, vol. 8, pp. 5049–5064, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2963105.

[9] A. P. Lauf and W. H. Robinson, "Fault-tolerant distributed reconnaissance," in *2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, San Jose, CA, USA: IEEE, Oct. 2010, pp. 1812–1817, ISBN: 978-1-4244-8178-1. DOI: 10.1109/MILCOM.2010.5679552.

[10] J. Kusyk, E. Urrea, C. S. Sahin, M. U. Uyar, G. Bertoli, and C. Pizzo, "Resilient node self-positioning methods for MANETS based on game theory and genetic algorithms," in *2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, San Jose, CA, USA: IEEE, Oct. 2010, pp. 1399–1404, ISBN: 978-1-4244-8178-1. DOI: 10.1109/MILCOM.2010.5680141.

[11] J. Kusyk, J. Zou, S. Gundry, C. S. Sahin, and M. U. Uyar, "Metrics for performance evaluation of self-positioning autonomous MANET nodes," in *2012 35th IEEE Sarnoff Symposium*, Newark, NJ, USA: IEEE, May 2012, pp. 1–5, ISBN: 978-1-4673-1465-7 978-1-4673-1464-0. DOI: 10.1109/SARNOF.2012.6222710.

[12] Z. Mi, Y. Yang, and J. Y. Yang, "Restoring connectivity of mobile robotic sensor networks while avoiding obstacles," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4640–4650, Aug. 2015, ISSN: 1530-437X, 1558-1748. DOI: 10.1109/JSEN.2015.2426177.

[13] S. Gundry, J. Kusyk, J. Zou, C. S. Sahin, and M. U. Uyar, "Performance evaluation of differential evolution based topology control method for autonomous MANET nodes," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, Cappadocia, Turkey: IEEE, Jul. 2012, pp. 000 228–000 233, ISBN: 978-1-4673-2713-8 978-1-4673-2712-1 978-1-4673-2711-4. DOI: 10.1109/ISCC.2012.6249299.

[14] M.-C. Mah, H.-S. Lim, and A. W.-C. Tan, "UAV relay flight path planning in the presence of jamming signal," *IEEE Access*, vol. 7, pp. 40 913–40 924, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2907962.

[15] S. Hong, H. Yang, N. Huang, D. Li, G. Cao, and Z. Wu, "Fault propagation model in mobile ad hoc network based on random walk model," in *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, Hong Kong, Hong Kong: IEEE, Jun. 2014, pp. 434–439, ISBN: 978-1-4799-3669-4 978-1-4799-3668-7. DOI: `10.1109/CYBER.2014.6917503`.

[16] W. A. Gardner, "Cyclostationarity: Theory and methods - II: Examples, applications, and future directions," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 305–341, Feb. 2004. DOI: `10.1109/JPROC.2003.821915`.

[17] K. C. Ho and B. Friedlander, "Direction-of-arrival estimation of multiple jammers using cross-correlation analysis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 2, pp. 655–670, Apr. 1995. DOI: `10.1109/7.376412`.

[18] D. Bertsekas, *Linear Network Optimization.* Massachusetts Institute of Technology, 1991, p. 273.