2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)
August 17-21, 2025. Los Angeles, CA, US

A Max-min Tree Approach to the Automated Construction of Ad Hoc Wireless Networks in Unknown Environments

Charles Noren[1], Bhaskar Vundurthy[1], Namya Bagree[2], and Matthew Travers[1]

2025 IEEE 21st International Conference on Automation Science and Engineering (CASE) | 979-8-3315-2246-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/CASE58245.2025.11163847

Abstract— **Reliable communication networks are essential for the remote operation of automated teams of robotic agents. For unknown (no prior map) communications-deprived (no existing communication infrastructure) environments, the robotic agents must construct the network as the robots move through the terrain. We present a novel method for automated network construction tailored for mobile robotic teams that require communication with a central base station. Our key innovation is the introduction of a maximin spanning tree structure, which guarantees a minimum level of communication performance between nodes. By directly optimizing node placement based on signal-based metrics, instead of relying on geometric surrogates like distance and visibility, we also achieve significant decreases in agent utilization while maintaining coverage for the traversed area. By using the robotic agents themselves as mobile repeaters in a communication network, each robotic agent can be individually assigned to prioritize network connectivity during critical operations. Numerical simulations on common Multi-Agent Path Finding benchmarks demonstrate up to a 36% reduction in the number of required nodes compared to existing techniques. Furthermore, this work guarantees robust network connectivity in dynamic environments, outperforming strongest-neighbor approaches that are vulnerable to link disruptions. Lastly, hardware tests confirm the robustness of our method in challenging scenarios encountered in real-world deployments.**

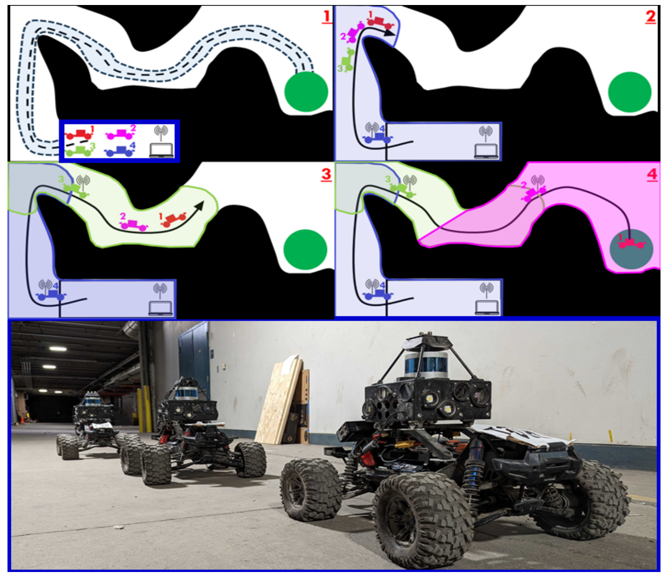


Fig. 1. A four-agent robotic convoy builds a communications network in a communications-deprived environment starting from a base station (black computer) and ending at a goal location (green circle). To ensure all agents remain in communication with the base station, the agents leave the convoy to become communications nodes as the convoy reaches the communications boundary. As the convoy leaves the communications area supported by the base station (black), Agent 4 leaves the convoy to act as a communications node (pane 2), extending the communications-accessible area further. In subsequent panes, Agent 3 and Agent 2 similarly leave the convoy, enabling Agent 1 to reach the goal location with communications.

I. INTRODUCTION

Many complex missions in dangerous environments, e.g., wildfire firefighting, battlefield triage, and search-and-rescue, will be revolutionized by the use of coordinated multi-agent robotic systems. Currently, high-performing multiple robot coordination depends on robotic agents that reliably communicate information between themselves and other decision makers. However, many environments lack the communication infrastructure to inherently support agent communications. Recent efforts have enabled robotic systems to construct this network during operations. Yet, ensuring a minimum quality of service across participants in the multi-agent system during the construction of the network remains challenging. We posit that by augmenting network construction techniques with communication-strength aware tree search on the network, a quality-ensured ad hoc robotic network topology can be established to support multi-robot operations in dangerous environments.

Abandoned buildings or subterranean caverns are examples of potentially dangerous environments with little or no existing communications infrastructure. Exploration of these environments was the focus of the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge (SubT) [1]. During exploration, vehicles were often expected to maintain contact with a central base station to communicate information back to a set of human operators. Multiple SubT performer teams approached this communication challenge by creating a mobile ad hoc wireless mesh network (MANET) by dropping “relay nodes” through the environment they traversed. These relay nodes could establish communication throughout the environment, but node placement strategies could result in variable communication quality or a large number of dropped nodes.

We introduce an automated network topology construction MANET technique for a small team of agents that aims to minimize the number of dropped nodes while ensuring quality communications (see Fig. 1). As the network grows, whether through the deployment of additional nodes by different robots or simply due to extended operation, multiple communication paths to the base station emerge. To ensure robust connectivity and maximize the minimum link quality across the diverse paths, we present a complete algorithm

[1]Charles Noren, Matthew Travers, and Bhaskar Vundurthy are from The Robotics Institute, Carnegie Mellon University, USA. {cnoren, mtravers, pvundurt}@andrew.cmu.edu
[2]Namya Bagree is from the Department of Mechanical Engineering, Carnegie Mellon University, USA. nbagree@andrew.cmu.edu

to compute a maximin metric-based spanning tree, which explicitly quantifies the connection quality of the MANET between an agent (or formation) and a base station. To our knowledge, no previous robotic MANET construction technique enables this capability.

Furthermore, the maximin spanning tree offers a reliable way to determine the connectivity of any node to the base station even when one or more intermediate nodes stop responding. This demonstrates its resilience to node failures, a key aspect of dynamic network environments. We demonstrate the effectiveness of our maximin spanning tree in improving communication quality through simulations on standard Multi-Agent Path Finding benchmark maps [2] and physical experiments involving formations of robotic agents. Therefore, to address the challenges of reliable MANET construction for teams of mobile robotic agents in dynamic environments, we contribute:

1) a quality-assured maximin tree-based communication construction algorithm for MANET generation;
2) a communication relay deployment behavior for network construction in *a priori* unknown environments.

II. Literature Review

Given the lack of available communications infrastructure in the DARPA SubT Challenge, multiple performer teams used network construction techniques to ensure communications. These techniques would construct a "communication backbone" that supported robotic operations by "dropping" communication nodes that extended the effective coverage of a wireless communication network [3]. For Team CERBERUS, a human operator was primarily responsible for determining the drop points [4]. For Team Explorer, this "dropping" behavior was primarily controlled by distance-based limits and line-of-sight (LOS) requirements between nodes [5]. Given their simplicity, geometric communications models are quite common for robotic systems, where physical distance and LOS take precedence over communications-based signal strength metrics [6]. However, SubT Teams also considered communication-based metrics, including a Radio Signal Strength Indicator (RSSI) threshold, as one of the determining factors for deploying both mobile and stationary communication nodes [7], [8]. As an alternative to RSSI, Team CoSTAR used the signal-to-noise ratio (SNR) alongside other environmental and communication factors [9].

The aforementioned approaches and metrics investigated in the DARPA SubT Challenge broadly capture aspects of the sensor coverage problem. For known environments, computational geometry approaches that frame the sensor coverage problem as a variant of the Art Gallery Problem (AGP) have proven extremely successful [8]. To address the computational complexity of the AGP, suboptimal polynomial-time approximations, such as polygonal decomposition or partitioning, are frequently employed [10]. Many variants of the AGP place additional restrictions on the coverage model used in the AGP, including limited range [10], range fading [11], or k-visibility through boundaries (i.e., the k-transmitters problem) [12]. In accordance with the AGP, these approaches often seek to provide coverage of polygonal areas and may rely on geometry-based communication models to derive the optimal sensor placements. Such models lie in contrast to a recent model presented by [6], which specifies that only certain areas of the state space must be made "communication-accessible." Such an approach provides the unique ability to represent realistic (i.e., non-polygonal) environments that robotic systems operate in, and often more closely aligns with the robotic operations (i.e., non-coverage problems).

Finally, the relay placement problem does not exist far outside of previous work in the communications-aware motion planning space. Many of these works focus on the development of control or planning algorithms that place requirements on maintaining specific distances (e.g., coverage area, visibility) or on metrics of network connectivity [13], [14]. These works are often structured around a fixed network topology and check for connectivity quality or reachability using a tree-based analysis. However, these works do not address the crucial aspect of constructing extensible network topologies that guarantee a minimum communication strength criterion [15].

III. PROBLEM DEFINITION

In this work, we consider a variant of the relay placement problem detailed in [6]. The objective of the relay placement problem is to find a minimal set of communication relay locations needed to form a valid network topology that covers an area of interest in the environment. In order to solve the problem, we require: 1) a defined area of interest, 2) a coverage model, which describes the conditions for communications coverage, and 3) a network model, which states the set of constraints required to form a valid network topology. The area of interest (denoted by $\mathcal{I}$) in the relay placement model represents an area that is to be made "communications-accessible" to the agents trying to traverse or explore the environment [6].

Following [6], consider a planar environment $\mathcal{W} \subset \mathbb{R}^2$ that is divided into object-free space ($\mathcal{F}$) and object-occupied space. All elements of the environment can be delineated into one of the two spaces through the use of an occupancy function $F : \mathcal{W} \to \{0, 1\}$. Here, "0" denotes that a spatial element is contained in the object-free space. Using this categorization, we can define $\mathcal{F} = \{x \in \mathcal{W} : F(x) = 0\}$. Our defined area of interest, $\mathcal{I}$, is a subset of the object-free space $\mathcal{F}$. Specifically, it is the region within which robotic agents will traverse for mission objectives, necessitating uninterrupted communication with the base station. For convenience, we define another indicator function $I : \mathcal{F} \to \{0, 1\}$, which maps elements of the object-free space to the area of interest (a value of "1" indicates an area must be covered). This indicator function enables us to define the area of interest $\mathcal{I} = \{x \in \mathcal{F} : I(x) = 1\}$, such as the light blue region around the planned path in pane 1 of Fig. 1. Note that $\mathcal{W}, \mathcal{F}, \mathcal{I}$ are connected spaces.

In order to maintain communications coverage, we consider a communication criterion inspired by our field experiences in the DARPA SubT Challenge [5]. Unlike [6], which considers an ℓ_2-boundedness and a visibility constraint between a pair of relays to maintain connectivity, our communication criterion depends directly on the measured communications strength between a pair of relays. For a total of N deployable relays, the relay set $\mathcal{R} = \{r_1, \dots, r_N\}$ is defined as a set of 3-tuples: $r_i = (i, x_i, y_i) \in \mathbb{N} \times \mathbb{R} \times \mathbb{R}$, where i is the relay index and $p_i = (x_i, y_i) \in \mathbb{R}^2$ is the relay position. To measure the communication strength between relays in the relay set, we define a function, $\mathrm{C}_{\text{val}}(r_i, r_j) \rightarrow \mathbb{R}^+$, that returns a measure of communication strength (e.g., SNR, RSSI) between nodes $r_i \in \mathcal{R}$, and $r_j \in \mathcal{R}$. If $\mathrm{C}_{\text{val}}(r_i, r_j) = 0$, then relays r_i and r_j are not connected.

We consider a pair of relays r_i and r_j to be mutually covered by each other if the observed communication strength, $\mathrm{C}_{\text{val}}(r_i, r_j)$, is greater than a threshold value: $\mathrm{c}_{\text{thresh}} \in \mathbb{R}^+$: $(\mathrm{C}_{\text{val}}(r_i, r_j) > \mathrm{c}_{\text{thresh}})$. For convenience, we extend our definition of coverage to include positional arguments as well: $\mathrm{C}_{\text{val}}(p, r_j) \rightarrow \mathbb{R}^+$, which captures the signal strength between a (potentially fictitious) relay placed at location $p \in \mathbb{R}^2$ and a relay $r_j \in \mathcal{R}$.

We then construct a network from the relays in the relay set $\mathcal{R}$ by placing the relays at different locations in the object-free space. Our goal is to create a valid network topology $\mathcal{N} = \{r_1, \dots, r_n\}$, $n \leq N$ to cover space $\mathcal{I}$. Space $\mathcal{I}$ is covered if $\forall\, x \in \mathcal{I}$, $\exists\, r \in \mathcal{N} : \mathrm{C}_{\text{val}}(x, r) > \mathrm{c}_{\text{thresh}}$. Finally, we define a valid network topology as one that is "connected." Simply put, a network is connected if, for all pairs of relays $r_i \in \mathcal{N}, r_j \in \mathcal{N}$, there exists a progression of relays starting from r_i and ending at r_j such that relays are sequentially mutually covered.

In this work, we are interested in enabling communications between a starting position, x_s, and a goal location, x_g, in the environment. We define the optimal planar planning problem over a state space $\mathcal{W}$ with permissible states $\mathcal{F}$, requiring $x_s \in \mathcal{F}$ and $x_g \in \mathcal{F}$. We assume that there is at least one relay at x_s. We define $\sigma : [0,1] \rightarrow \mathcal{W}$ as a sequence of states (a path) from the set of paths Σ. The optimal planning problem is therefore to find a path σ^* that minimizes a cost function $s : \Sigma \rightarrow \mathbb{R}^+$ and connects x_s to x_g. We define the condition for optimality in a general manner as our approach is not preconditioned on any specific measure. We define the optimal path as follows:

$$\sigma^* = \arg \min_{\sigma \in \Sigma} \{ s(\sigma) |\, \sigma(0) = x_s, \sigma(1) = x_g, \\ \forall t \in [0,1], \sigma(t) \in \mathcal{F} \}. \quad (1)$$

In a known environment, where a robotic team aims to travel between two points while maintaining continuous communication with the base station, the area of interest $\mathcal{I}$ can be defined as the optimal path σ^* connecting these points. While selecting $\mathcal{I} = \sigma^*$ does not necessarily minimize the number of communication nodes required, it guarantees connectivity between the base station and the goal position. Minimizing the number of communication nodes in a known environment is a variant of the minimum covering set problem, which is NP-hard and falls outside the scope of this work [16].

In this work, we do not assume any prior knowledge of the structure of the environment, specifically the decomposition of the workspace $\mathcal{W}$ into free or occupied space. Thus, given a motion planning policy $\pi(x_s)$ that yields a feasible path σ between x_s and x_g, we propose a network construction algorithm that procedurally covers the area of interest $\mathcal{I}$. By employing a placement strategy that ensures a connected network topology, we enable the robotic agents to traverse the environment while maintaining continuous communication with the relay at x_s.

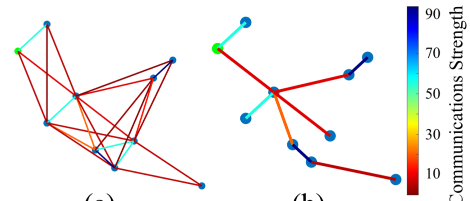
IV. METHODOLOGY OVERVIEW

A communication network topology can be modeled as a simple weighted undirected graph $G = (V, E)$. In this work, the vertices V of this graph represent physical assets participating in a network topology $\mathcal{N}$, including deployed relays and any communications-enabled agents. These assets are collectively referred to as "communication nodes." While we define any given communication node as $v_n \in V$, we split the set of vertices V into two unique sets: 1) the set of mobile "non-deployed" agents P and 2) the set of static "deployed" nodes S (i.e., $V = P \cup S$). For a team of p robotic agents $P = \{p_1, p_2, \dots, p_p\}$ and $S = \{s_{p+1}, s_{p+2}, \dots, s_N\}$ static nodes in the communication network, we define $\mathcal{J}_N = \{1, \dots, p, p+1, \dots, N\}$. Here, $n \in \mathcal{J}_N$ represents a specific communication node index. We also overload the definition of $v_n \in V$ to be equivalent to the physical location of $p_n \in \mathbb{R}^2$. The communication graph is initialized with v_1 as the only starting node, which often represents a central base station (see Fig. 1).

The edges of the communication graph have an associated positive weight, $e(v_i, v_j) \rightarrow \mathbb{R}^+$, $v_i, v_j \in V$, which represents a communication link between the vertices. For example, in Team Explorer's original approach and in [6], a positive distance function $\mathrm{d}(v_i, v_j) : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, which measured the physical ℓ_2 distance between two communication nodes in the graph, was utilized for $e(\cdot)$. Although physical distance may be used as a surrogate measure of communication strength, recent advances in communication technology have enabled the direct use of communication strength metrics (e.g., RSSI, SNR) for relay placement. For ease of notation, we denote all communications strength metrics between two vertices (v_i, v_j) using our communication coverage model notation: $\mathrm{C}_{\text{val}}(v_i, v_j) \rightarrow \mathbb{R}^+, v_i, v_j \in V$. Note that in this work, as $e(\cdot)$ may take the form of $d(\cdot)$ or $\mathrm{C}_{\text{val}}(\cdot)$, the particular use will be indicated as required.

A. Maximin Communications Graph Spanning Tree

In order to ensure that communications are maintained between the base station and all assets, we design a reactive node placement behavior that deploys a communications node in response to an imminent loss of communications coverage. To ensure the connectivity of all assets $v_n \in V$ to the base station (v_1), we require the existence of at least one nodal path in graph G that starts at the base station v_1



(a) (b)

Fig. 2. A communications graph (G) on the left and its corresponding maximin tree (T) on the right.

and ends at every asset v_n where all relays are progressively sequentially mutually covered. By defining a path through G between the base station v_1 and any vertex $v_n \in V$ as $\pi(v_1, v_n) = v_1, \ldots, v, \ldots, v_n,\ v \in V, v_1 \neq v_n$, we impose the condition $\mathrm{C}_{\text{val}}(v_i, v_{i+1}) \geq \mathrm{c}_{\text{thresh}} \,\forall\, (v_i, v_{i+1}) \in \pi(v_1, v_n),\ \forall\, v_n \in V$ where the pairing (v_i, v_{i+1}) represents every pair in the sequence of nodes defined by the path π. This condition generates a tree structure T on G, i.e., $T \subseteq G$, with vertex v_1 as the root vertex of the tree. Given this structure, we choose to simplify the path notation between node v_1 and v_n into a single argument ($\pi(v_1, v_n) = \pi(v_n)$).

To enforce the single-path communications coverage constraint to all mobile agents, we pose T as a minimum spanning tree where the edge weights capture the strongest "weakest" link (widest-path) on a path between the root vertex and a non-root vertex in the tree. This is referred to as the "maximin communications metric." For each non-root vertex, v_n, connected by a path $\pi(v_n)$ to the root vertex v_1, a value $\mathrm{C}_{\text{met}}(v_1, v_n) \in \mathbb{R}^+$ is calculated using Algorithm 1. This is done to find the maximum minimum edge value $\mathrm{C}_{\text{val}}(\cdot)$ for all nodal paths between v_1 and v_n in the communication graph G. By definition, the spanning tree T spans all vertices included in the communications graph. Thus, by ensuring $\mathrm{C}_{\text{met}}(v_1, v_n) \geq \mathrm{c}_{\text{thresh}},\ v_n \in V$, we ensure the existence of a path between v_1 and v_n such that the minimum edge weight is observed for all edges in the tree.

We enlist a number of helper data structures to build the maximin tree. The first structure is a real-valued n-tuple, $\mathrm{CV} \in \mathbb{R}^N$, which represents the minimum communication value experienced between the n^{th} communication node and the root node v_1. Next, we define another n-tuple, $\mathrm{SPT} \in \{0, 1\}^N$, which tracks the inclusion of vertices in the spanning tree. Finally, $(\mathrm{PATH}, \mathrm{TEMP}) \in \mathbb{Z}^{+,N}$ tracks the structure of the tree. All tuple indices correspond to node indices, with index 1 representing the central base station.

We additionally introduce a supporting routine, $\mathrm{MCV}(\mathrm{CV}, \mathrm{SPT}, V)$ that parses CV for the maximum value of a node not already in the tree and returns that node's index or -1 if all nodes are already included in the tree. Algorithm 1 exists in the family of single-source shortest path algorithms (making it a complete algorithm). With a naive array implementation, the time complexity of the presented algorithm is $O(|V|^2)$. This logic takes the set V and the root node v_1 as input where $1 \in \mathcal{J}_N$, determines the maximin tree for the graph, and then publishes a message identifying the vertices with a $\mathrm{C}_{\text{met}}(\cdot)$ below $\mathrm{c}_{\text{thresh}}$. For convenience, define: $\mathrm{MIN}(\cdot) = \mathrm{CV}[y] \leftarrow \min(\mathrm{CV}[x], \mathrm{C}_{\text{val}}(v_x, v_y))$.

Algorithm 1 Maximin Communication Spanning Tree

Define $\mathrm{MCST}(v_1, V)$
Input v_1, V ▷ Index 1 is the assumed root
Output CV
Require: $\mathrm{CV} = \{0\}^N$, $\mathrm{CV}[1] = \infty$, $\mathrm{SPT} = \{0\}^N$
Require: $\mathrm{PATH} = \{1\}^N$, $\mathrm{TEMP} = \{1\}^N$

for $i \in \mathcal{J}_N$ **do**
  if $\sum_{j \in \mathcal{J}_N, i \neq j} \mathrm{C}_{\text{val}}(v_i, v_j) = 0$ **then**
    $\mathrm{SPT}[i] = 1$, $\mathrm{CV}[i] = 0$, $\mathrm{PATH}[i] = i$
  else ▷ if not disconnected
    x = $\mathrm{MCV}(\mathrm{CV}, \mathrm{SPT}, V)$
    if $x \neq -1$ **then** ▷ i.e., still have nodes in tree
      $\mathrm{SPT}[x] = 1$, $\mathrm{PATH}[i] = \mathrm{TEMP}[i]$
      for $y \in \mathcal{J}_N$ **do**
        if ($\mathrm{C}_{\text{val}}(v_x, v_y) > 0$, $\mathrm{SPT}[y] = 0$, and
          $\mathrm{CV}[y] < \mathrm{MIN}(\cdot)$) **then**
          $\mathrm{CV}[y] = \mathrm{MIN}(\cdot)$, $\mathrm{TEMP}[y] = x$
    else ▷ if all nodes already in the tree
      for $y \in \mathcal{J}_N$ **do**
        if $\mathrm{SPT}[y] = 0$ **then**
          $\mathrm{SPT}[y] = 1$, $\mathrm{CV}[y] = 0$,
          $\mathrm{PATH}[y] = y$ ▷ Set self as parent

By running Algorithm 1, the value of $\mathrm{C}_{\text{val}}(\cdot)$ may be monitored to ensure communication capabilities. An example of the maximin spanning communications tree for a communications graph is shown in Fig. 2. In Fig. 2a, the vertices in the communications graph are demonstrated in both green (v_1) and blue ($v \in (V \setminus \{v_1\})$), with edges demonstrating connection strengths greater than $\mathrm{c}_{\text{thresh}}$. The edge colors reflect varying communications strengths, from low strength in red to high strength in blue. Using Algorithm 1, a spanning tree structure can be imposed on the graph presented in Fig. 2a, which is shown in Fig. 2b.

Algorithm 2 Communication Relay Deployment Behavior

Input P, G, v_1, σ, t

while $p_p \neq \sigma(1)\ \forall\, p \in P$ **do** ▷ while no agent at x_g
  $\mathrm{C}_{\text{met}} \leftarrow \mathrm{MCST}(v_1, V(G))$
  for $k \in P$ **do**
    if $\mathrm{C}_{\text{met}}[k] < \mathrm{c}_{\text{thresh}}$ **then**
      agent k deploys a communication relay
  Move along σ ▷ for t seconds

B. Communication Relay Deployment Behavior

For this work, each agent of the multi-agent team may act as at most one communication node (as opposed to the carrier agents in [5]). Once the maximin communications spanning tree is found, the communication relay deployment behavior in Algorithm 2 periodically checks to ensure that no agent has a C_{met} value below $\mathrm{c}_{\text{thresh}}$. Note that if multiple agents (e.g., $i, j \in P$) have a C_{met} value below $\mathrm{c}_{\text{thresh}}$, and if $\mathrm{C}_{\text{val}}(i, j) > \mathrm{c}_{\text{thresh}}$, then before both agents deploy nodes, only the agent with the lowest ID (e.g., $i < j$ implies agent

i) deploys a relay. This is to ensure that multiple agents do not redundantly deploy nodes at the same location.

Thus, the experienced automated behavior is that an agent stops to become a stationary communication repeater (relay) if its experienced $\mathrm{c_{thresh}}$ value falls too low. In practice, the "true" minimum value of $\mathrm{c_{thresh}}$ is buffered by some amount, $\delta \in \mathbb{R}^+$, to conservatively approximate a communications boundary. This ensures that the system does not experience an unexpected dropout or cause oscillatory behaviors near the communications boundary (i.e., $\mathrm{c'_{thresh}} = \mathrm{c_{thresh}} + \delta$). The magnitude of δ is highly dependent on the system hardware and the rate t at which Algorithm 2 is checked.

V. COMPARATIVE SIMULATIONS

To evaluate the performance of our proposed maximin communication-metric tree construction technique, we conducted simulations within Multi-Agent Path Finding (MAPF) benchmark environments [2]. These benchmarks are employed solely to generate realistic communication strength layouts, enabling the simulation of communication graphs without relying on prior environmental knowledge. Crucially, from a communication perspective, we are still operating within unknown environments. We begin by comparing our maximin approach with the network construction method used by Team Explorer, demonstrating that our method yields a reduction in the number of deployed nodes. We then demonstrate that the maximin tree's ability to condense multiple nodal paths into a single representative value provides a more informed assessment than a myopic approach.

A. Simulation Environment

In cluttered environments, the effectiveness of communication networks is significantly impacted by distance and obstacles. To model this, we adopt an inverse square law for signal attenuation, approximating signal strength as inversely proportional to the square of the distance with an additional linear decline through the obstacles. The communication model used in this simulation section is given as

$$\mathrm{C_{val}}(r_1, r_2) = c_1 \cdot \frac{1}{\overline{r_1 r_2}^2} - c_2 \cdot d_{obst}.$$

Note that this model is adopted solely for simulation purposes and is not utilized to preprocess the environment for communication node placement. The algorithm's performance with a real communications system will be demonstrated in the subsequent hardware trials section. The simulations in this section assume a value of $c_1 = 100$ for obstacle-free space and an additional constant $c_2 = 20$ when considering obstacles. For example, a robot 2 units away from the source with an obstacle that is 0.5 units thick in between experiences a signal strength of $25 - 10 = 15$ units.

B. Visibility vs. Communications-Metric Graph Construction

This section compares the performance of two communication network construction approaches: the visibility-based method used by Team Explorer in SubT [5] (Algorithm 3) and our maximin communications-metric-based construction approach (Algorithm 2). Team Explorer's approach adds nodes to graph G depending on their relative distance and visibility to previously deployed nodes, as detailed in Algorithm 3. To check visibility, we define a procedure, $\mathrm{LOS}(v, p), v \in V$, which returns "True" if p is within the line of sight of any static node in V and otherwise returns "False." Furthermore, two hyperparameters: $\mathrm{d_{LOS}} \in \mathbb{R}^+$ and $\mathrm{d_{NLOS}} \in \mathbb{R}^+$, were defined to capture the maximum allowable distance (e.g., in an ℓ_2 sense) for a node to be placed with respect to any previous node in G. The first parameter, $\mathrm{d_{LOS}}$, represents the furthest allowable distance an agent p is allowed to be from any static vertex $v \in V(G)$ while $\mathrm{LOS}(v, p)$ is "True." Similarly, $\mathrm{d_{NLOS}}$, represents the furthest allowable distance an agent p is allowed to be from any static vertex $v \in V(G)$ if $\mathrm{LOS}(v, p)$ is "False." The full algorithm logic is presented in Algorithm 3.

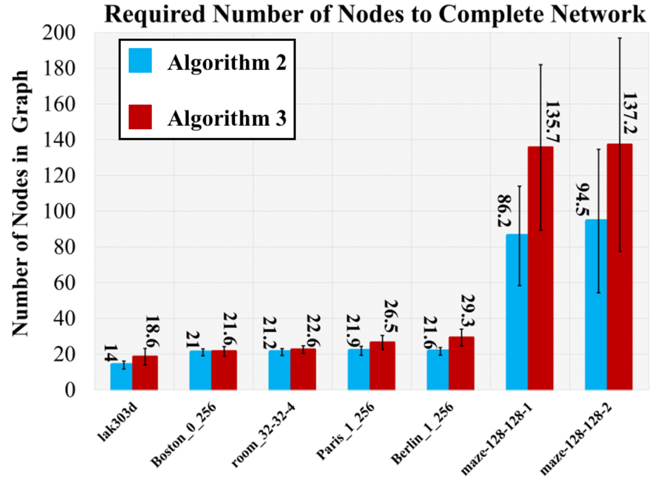


Fig. 3. A summary of network construction tests demonstrating that communication-based metrics decreases total node usage.

Algorithm 3 Explorer Communication Graph Construction

Input N, P, $\mathrm{d_{LOS}}$, $\mathrm{d_{NLOS}}$
Require: $N \geq 1$, $\{v_1\} = S$

1: **while** $n \leq N$ **do** ▷ Exit if run out of nodes
2: **for** $p \in P$ **do**
3: $s = \arg\min \mathrm{d}(s_i, p), s_i \in S$ ▷ Find closest node
4: **if** $\mathrm{LOS}(s, p)$, and $\mathrm{d}(s, p) > \mathrm{d_{LOS}}$ **then**
5: $S = S \cup p,\ P = P \setminus \{p\},\ \ n = n + 1$
6: "Establish Node" ▷ Drop Node for SubT
7: **else if** $\mathrm{d}(s, p) > \mathrm{d_{NLOS}}$ **then**
8: $S = S \cup p,\ P = P \setminus \{p\},\ \ n = n + 1$
9: "Establish Node" ▷ Drop Node for SubT

We evaluate their effectiveness by navigating a robot between random start and goal locations. Along the path, communication nodes are deployed whenever the signal strength falls below a threshold ($\mathrm{c_{thresh}}$) of 10. A key difference lies in how signal attenuation is modeled. To ensure connectivity in Algorithm 3, we model a visibility-based approach that assumes an immediate and complete signal loss (infinite attenuation, $\mathrm{d_{NLOS}} = 0$) when the line of sight between the communications nodes is broken.

To evaluate the performance difference, we conducted extensive simulations across a diverse set of maps from the MAPF benchmark dataset. For each map, we randomly generated 100 different start and goal locations for the robot, ensuring a comprehensive assessment across varying environmental configurations and path requirements. These experiments measure the number of nodes required to maintain communication between the start and goal points, with fewer nodes indicating a more efficient approach. The results are presented in Fig. 3 where the node construction approach from Algorithm 2 (in blue) consistently outperforms (has fewer nodes) the approach from Algorithm 3 (in red). The largest decrease in required nodes was observed in the *maze-128-128-1* environment, where the average decrease in deployed nodes across all trials was 36%.

C. Efficacy of Maximin Communication Spanning Tree

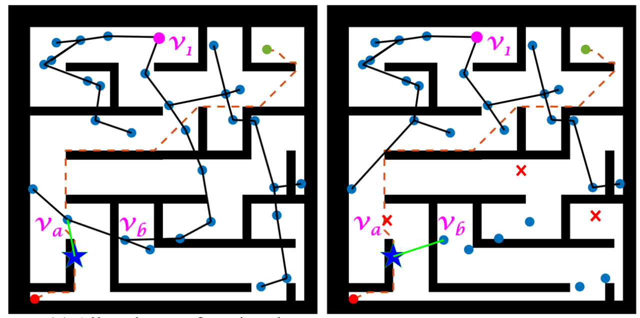
Algorithm 1 optimizes the maximin communication metric across nodal paths, ensuring the weakest link to the base station exceeds a threshold for higher network quality. While intuitively beneficial for practical robotic applications, we found no prior architectures offering comparable network quality guarantees. To evaluate the efficacy of our algorithm, we introduce a baseline approach: the strongest neighbor communication metric (Algorithm 4). This method assumes an ideal, disruption-free network, where connectivity to the strongest neighbor directly implies connectivity to the base station. Such a simplified strongest-neighbor strategy reflects common practices observed in the SubT Challenge, providing a comparative benchmark for our maximin algorithm.

Algorithm 4 Strongest Connection Metric

Input v^*, G, c_{thresh}
Output IfConnected
Require: $\text{MAXCV} = 0$
add v^* to G
$\text{MAXCV} = \max(\mathrm{C}_{\text{val}}(v^*, v_i),\ \forall\, i \in \mathcal{J}_N)$
if $\text{MAXCV} > c_{\text{thresh}}$ **then**
    IfConnected = True
else
    IfConnected = False

To illustrate the algorithmic differences, consider Fig. 4a, which depicts already deployed nodes (blue) and a base station (pink). We examine a point (blue star, which represents v^*) along the path between the red and green dots, which in turn represent x_s and x_g, respectively. In this instance, Algorithm 1 computes $\mathrm{C}_{\text{met}}(\cdot) > c_{\text{thresh}}$ indicating sufficient connectivity for v^* to the base station. In contrast to Algorithm 1's use of $\mathrm{C}_{\text{met}}(\cdot)$ to determine connectivity, Algorithm 4 relies on the strongest neighbor link (e.g., node v_a). In Fig. 4a, this also results in a correct connectivity assessment, which we denote as a true positive.

However, Fig. 4b shows a potential failure scenario. Here, three nodes, including v_a, experience intermittent connection issues, rendering them nonfunctional. Consequently, the star node's strongest link shifts to v_b, which still exceeds the threshold. Algorithm 4, in this case, incorrectly infers base station connectivity via v_b, resulting in a false positive when compared to the true connectivity determined by Algorithm 1. This example highlights the vulnerability of the strongest neighbor approach to link disruptions, which the maximin tree is robust against.



(a) All nodes are functional (b) Three nodes (X) are disconnected

Fig. 4. Comparison of Algorithm 1 and Algorithm 4.

To rigorously evaluate the performance of Algorithm 1 and Algorithm 4, we conducted an extensive study across multiple MAPF benchmark maps, each with randomized node configurations. Specifically, we generated 100 random pairs of start and goal locations (x_s and x_g) within each map and tested both algorithms at discrete points between them. Each environment is seeded with 30 relays that form a valid connected network, and we simulated network disruptions by randomly disconnecting 3 of the 30 nodes. We also randomly select a base node from the remaining set of non-disconnected relays. This methodology allowed us to assess the effectiveness of the algorithms across a diverse range of dynamic network conditions. We quantified performance by calculating the rate of false positives produced by Algorithm 4 relative to the total positives identified by Algorithm 1. This metric represents the percentage of instances where Algorithm 4 yields an incorrect connectivity assessment.

The results of this study are presented in Fig. 5. Notably, in the *den520d* map, Algorithm 4 exhibited zero errors, highlighting its potential for significantly faster computation compared to Algorithm 1 in static environments. This observation aligns with the algorithm's adoption in prior work with static network assumptions. However, across the broader dataset, we observed a substantial increase in false positives, reaching up to 100% in some cases. This dramatic decline in performance underscores the necessity of a base-station-centric critical strength-aware communication tree, as provided by Algorithm 1, for robust decision-making in dynamic and mission-critical scenarios.

VI. HARDWARE TRIALS

The network construction algorithms presented in Section IV were tested on a convoy of robotic agents in the context of an automated patrol mission through a hospital-like building without any available network infrastructure. The testing environment is shown in Fig. 6, which depicts the

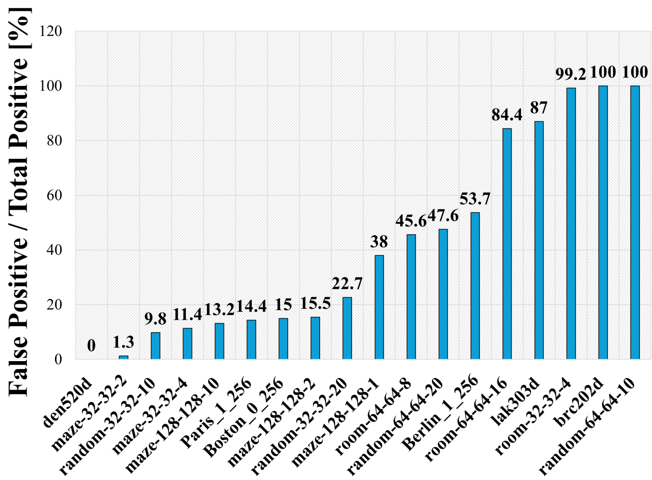


Fig. 5. Multiple MAPF environments demonstrate advantages in using Algorithm 1 over Algorithm 4.

floor plan of the building as it was mapped by the agents during the hardware trials. Figure 6 shows that this environment consists mainly of two long corridors (viewpoints 1 and 3), a sharp turn (viewpoint 2), and an exit point to the exterior of the test facility (viewpoint 4). In all experiments, the convoy started at the operator base station near viewpoint 1. The path σ was implicitly constructed using a frontier exploration algorithm governed by a greedy heuristic to minimize the total distance traveled [5], [17].

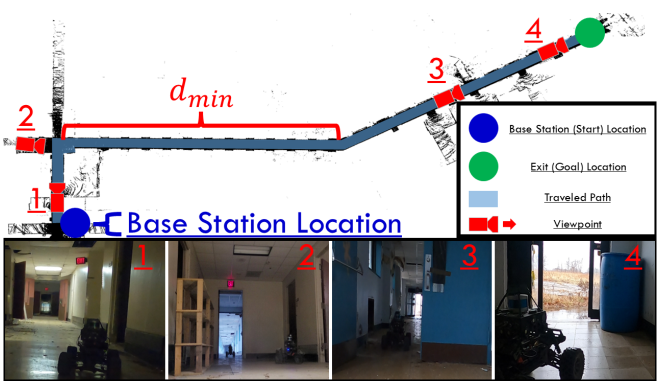


Fig. 6. Hardware Trial Environmental Overview

The convoy consisted of three mobile robotic agents (Fig. 1) that were all equipped with a communications radio. Each agent's radio enabled it to talk to all other agents or the base station if the corresponding radio was also in range. Constant communication between the base station and the convoy was ensured by procedurally building a communications network using Algorithm 2 and Algorithm 3.

A. Visibility vs. Communications-Metric Graph Construction

The results of the comparative simulations reflect that distance-based metrics may perform poorly in complex environments with high densities of line-of-sight blocking terrain features. For the given environment in this set of experiments, a singular limit could be observed such that the multi-agent team of robots would have at least one agent reaching the egress point without LOS constraints. This limit is denoted d_{min} (shown in Fig. 6), and was measured to be at least 75 [m] (thus, $\mathrm{d_{LOS}} = \mathrm{d_{NLOS}} = d_{min} = 75$ [m] for the given tests). Any value above 75 [m] would allow the agents to reach the goal. However, given a suboptimal parameter choice for $\mathrm{d_{LOS}}$ or $\mathrm{d_{NLOS}}$, the mobile agent team was unable to reach the exit point of the facility. This is reflected in Fig. 7a. Here, a suboptimal choice was selected ($\mathrm{d_{LOS}} = 65$ [m], $\mathrm{d_{NLOS}} = 0$ [m]) which caused agent "RC1" and agent "RC3" to stop at points where the system lost line-of-sight to the previous nodes. Note that line-of-sight to the base station relay was available in the starting hallway. In contrast, leveraging Algorithm 1 with a communication-based metric (selected as falling below a "signal strength" threshold) provides a different spatial distribution of deployed agents. Note that the selection of the communication-based metric also required empirical testing to determine the hyperparameter threshold, but did allow for the dropping of the line-of-sight constraint, which is advantageous in complex geometries.

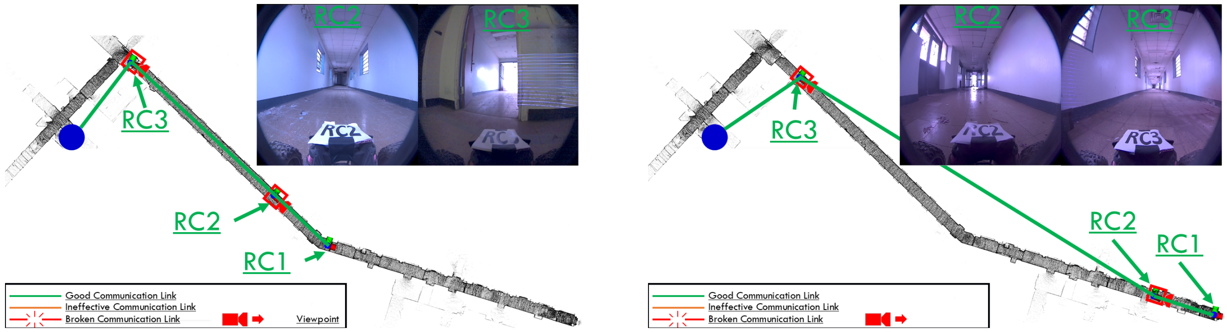
B. Failed Node Robustness Test

In addition to the performance tests, additional robustness tests were performed to handle disconnected or "failed" nodes. Here, a failed node does not communicate with the base station but can communicate with the agents in the convoy. On one agent, "RC3," the maximin communications spanning tree is used to construct the network (Algorithm 2) while "RC2" uses Team Explorer's strategy (Algorithm 3).

The position of the failed node in the environment is shown in Fig. 8 and is labeled "RC1." The placement of this node satisfies both the distance and LOS constraints required for Algorithm 3. Without the maximin communications spanning tree, "RC2" does not stop until the position marked as "RC2" in Fig. 8, as this is where the communication range (determined by $\mathrm{d_{LOS}}$) is reached. When "RC2" moved beyond the "RC1" position, communication with the agent began to fail and became intermittent. A post-trial analysis shows that the agent which stopped at the "RC2" position had almost half the "signal strength" needed to communicate with the base station ($\mathrm{c_{thresh}}$). However, with the presence of the maximin communications spanning tree, "RC3" observed this drop in signal strength near the "RC1" position, causing it to deploy as a communications node. That threshold was reached at the position listed as "RC3" in Fig. 8. Thus, the presence of the maximin communications spanning tree restrained agent "RC3" from leaving the communications-accessible area and ensured that it was not tricked by the disconnected node.

VII. CONCLUSIONS

This manuscript presents a maximin communications spanning tree approach for ensuring communications between a multi-agent robotic team and a base station. We demonstrate the capability of the approach in both simulation and hardware to generate network topologies in complex environments, and demonstrate how the maximin communications spanning tree provides additional robustness to disconnected "failed" nodes in the environment. Immediate future works relate primarily to two areas: 1) map-predictive communication network construction and 2) the development of



(a) Test I: Ending state in Algorithm 3 evaluation

(b) Test II: Ending state in Algorithm 2 evaluation

Fig. 7. The testing environment for all hardware experiments. Note that each camera view taken from the multi-agent team is also labeled on the map. The closest red arrow is an approximate position for the image. The image was taken in the direction indicated by the arrow's head. Hardware evaluations comparing Algorithm 3 and Algorithm 1. Figure Fig. 7a reflects the end positions for the agents using Algorithm 3 and a distance-based metric. Figure Fig. 7b reflects the end positions for the agents using Algorithm 1 and a communications-based metric.

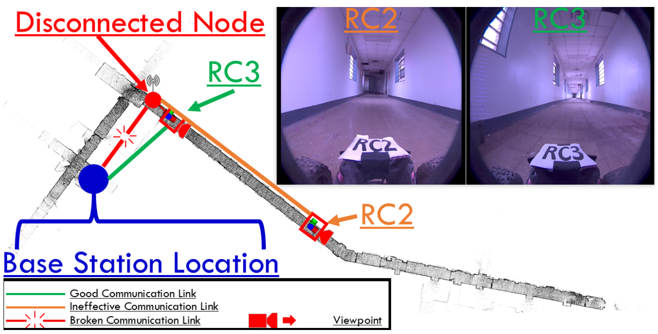


Fig. 8. The presence of the disconnected node in the environment affects each agent differently. Agent "RC2" does not realize that it cannot communicate with the base station and continues out of range. This contrasts with agent "RC3," which recognizes the communication failure and stops.

communication-quality-constrained exploration-based planning and control algorithms for robotic convoying. While the problem definition discussed in Section III dismissed the effectiveness of minimum set coverage approaches for robotic systems, the use of machine learning to predict environment layout and communications strengths would be a vital step towards generating more optimal relay placements. Works such as Tatum [8] began this effort, but more powerful environmental prediction algorithms (e.g., MapEx [18]) could provide a large improvement over Tatum's baseline. Furthermore, communication-quality-constrained motion planning and control for exploration has been explored in prior works, but its realization on hardware needs further development and testing. Specifically, utilizing perception systems that predict drops in communication quality (e.g., observing known-communication blocking materials or geometries) and then enforcing constraints in the reachable space of a low-level controller provides a unique online benefit towards ensuring the system never leaves communication range.

References

[1] DARPA, "Defense advanced research project agency, subterranean (subt) challenge (archived)," Online, 2019.

[2] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, E. Boyarski, and R. Bartak, "Multi-agent pathfinding: Definitions, variants, and benchmarks," *Symposium on Combinatorial Search (SoCS)*, 2019.

[3] J. Gielis, A. Shankar, and A. Prorok, "A critical review of communications in multi-robot systems," *Current Robotic Reports*, vol. 3, no. 4, pp. 213–225, 2022.

[4] M. Tranzatto *et al.*, "CERBERUS: autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the DARPA subterranean challenge," *CoRR*, vol. abs/2201.07067, 2022.

[5] S. Scherer *et al.*, "Resilient and modular subterranean exploration with a team of roving and flying robots," *Field Robotics*, vol. 2, pp. 678–734, 2022.

[6] M. Zoula and J. Faigl, "Wireless communication infrastructure building for mobile robot search and inspection missions," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[7] C.-L. Lu *et al.*, "A heterogeneous unmanned ground vehicle and blimp robot team for search and rescue using data-driven autonomy and communication-aware navigation," *Field Robotics*, vol. 2, pp. 557–594, 2022.

[8] M. Tatum, "Communications coverage in unknown underground environments," Master's thesis, Carnegie Mellon University, 2020.

[9] T. S. Vaquero, M. S. da Silva, K. Otsu, M. Kaufman, J. A. Edlund, and A.-a. Agha-mohaammadi, "Traversability-aware signal coverage planning for communication node deployment in planetary cave exploration," *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2020.

[10] G. Kazazakis and A. Argyros, "Fast positioning of limited-visibility guards for the inspection of 2d workspaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 2843–2848.

[11] M. Ernestus, S. Friedrichs, M. Hemmer, J. Kokemuller, A. Kröller, M. Moeini, and C. Schmidt, "Algorithms for art gallery illumination," *Journal of Global Optimization*, vol. 68, 05 2017.

[12] B. Ballinger *et al.*, "Coverage with k-transmitters in the presence of obstacles," *Journal of Combinatorial Optimization*, vol. 25, 01 2013.

[13] G. Antonelli, F. Arrichiello, S. Chiaverini, and R. Setola, "A self-configuring manet for coverage area adaptation through kinematic control of a platoon of mobile robots," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 1332–1337.

[14] R. Pandey, A. K. Singh, and K. M. Krishna, "Multi-robot exploration with communication requirement to a moving base station," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, 2012, pp. 823–828.

[15] M. Wzorek, C. Berger, and P. Doherty, "Router and gateway node placement in wireless mesh networks for emergency rescue scenarios," *Autonomous Intelligent Systems*, vol. 1, 12 2021.

[16] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 6th ed. Springer, 2018.

[17] B. Yamaguchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97*, 1997, pp. 146–151.

[18] C. Ho *et al.*, "Mapex: Indoor structure exploration with probabilistic information gain from global map predictions," 2024.