

**2025 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
AUTONOMY, ARTIFICIAL INTELLIGENCE, & ROBOTICS (AAIR) TECHNICAL SESSION
AUG. 12-14, 2025 - NOVI, MICHIGAN**

**COMMUNICATION NETWORK CONSTRUCTION BEHAVIORS FOR
ROBOTIC CONVOYING**

**Charles Noren^{*1}, Sahil Chaudhary^{*1}, Burhanuddin Shirose¹, Bhaskar Vundurthy,
PhD¹, and Matthew Travers, PhD¹**

¹Carnegie Mellon University, Pittsburgh, PA

ABSTRACT

We develop a set of communications-aware behaviors that enable formations of robotic agents to travel through communications-deprived environments while remaining in contact with a central base station. These behaviors enable the agents to operate in environments common in dismounted and search and rescue operations. By operating as a mobile ad-hoc network (MANET), robotic agents can respond to environmental changes and react to the loss of any agent. We demonstrate in simulation and on custom robotic hardware a methodology that constructs a communications network by “peeling-off” individual agents from a formation to act as communication relays. We then present a behavior that reconfigures the team’s network topology to reach different locations within an environment while maintaining communications. Finally, we introduce a recovery behavior that enables agents to reestablish communications if a link in the network is lost. Our hardware trials demonstrate the systems capability to operate in real-world environments.

Citation: C. Noren, S. Chaudhary, B. Shirose, B. Vundurthy, M. Travers, “Communication Network Construction Behaviors for Robotic Convoying,” In *Proceedings of the 2025 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, NDIA, Novi, MI, Aug. 12-14, 2025.

1. INTRODUCTION

Robotic teams have repeatedly demonstrated the capability to effectively search dangerous or difficult-to-reach areas. Such results were most prominently demonstrated during the Defense Advanced Research Projects Agency (DARPA) Subterranean (SubT) Challenge [1]. Given that many of the DARPA SubT test environments were not equipped with

communications infrastructure, performer teams often simultaneously completed search and rescue tasks while building a mobile wireless ad-hoc network (MANET). The constructed MANET was often dependent on “drop nodes” –communication relays that were physically “dropped” from robotic platforms– as they explored the environment. In many applications, including dismounted operations, the need for communications between robotic

* indicates equal contributions

agents is still present, but relying purely on drop-node-based strategies has many undesirable features. Such approaches can be time-consuming for human operators, as they may need to clean-up or collect the drop nodes, and the network topology can be difficult to modify once the nodes are deployed. In this work, we build on our SubT experience by truly observing the “*mobile*” name of a MANET. We introduce and integrate a series of MANET construction and modification behaviors for teams of robotic agents that travel in a convoy formation. Our aim is to begin a discussion on how to develop automated communications-aware robotic systems that can operate in communications-deprived environments without relying on human operators (e.g., the warfighter, disaster response team members, etc.). We argue that operating as a MANET is a core underlying technology to multi-agent operations and then demonstrate the application of such technologies to multi-robotic-agent formation movement problems in cluttered environments.

We consider missions where the robotic agents are required to move throughout an environment while remaining in constant contact with a central base station. Both the base station and the robotic agents are equipped with wireless communication capabilities (nodes) that allow the base station and the agents to communicate with each other at distance under nominal conditions. Agents move through the environment as part of a formation, normally a convoy, to provide mutual support to each other during the traversal. As agents move through the environment, the quality of the inter-agent communications is affected by objects in the environment and the distance between the agents. We require that connections between the base station and a multi-agent formation maintain a measure of

communications strength in order to ensure constant communication between the agents and the base station.

To achieve this operating concept, we introduce a MANET topology construction technique for robotic systems with mobile communication nodes. We introduce three behaviors: 1) the MANET construction technique for formations, 2) a formation control algorithm that maintains the network structure while the vehicles move through the environment, and 3) a set of individual agent behaviors to reconnect the network topology if a node in the network fails. We demonstrate these algorithms in simulation and on robotic hardware for a small team of robotic agents.

The remainder of this manuscript is structured as follows. We begin with an overview of the relevant literature, much of which is concerned with the DARPA Subterranean Challenge and techniques and technologies derived from it. Following this, we introduce our construction behavior and describe two “recovery behaviors”: algorithms that 1) optimize the placement of communication nodes and 2) repair the communications network in the face of node failure or loss of communications. We demonstrate these algorithms in both simulation and on robotic hardware and conclude the paper with some additional commentary and future directions.

2. RELATED WORK

As noted in the previous section, this work is the latest in a series of works derived from the DARPA Subterranean Challenge [1]. As the “Systems” SubT performer teams were required to complete a challenge with similar requirements to our work (i.e., operating in communications-deprived environments), we first provide a brief overview of the different systems developed by each of the SubT performer teams before surveying other

relevant communications-aware multi-agent systems literature.

In particular, we would like to highlight the approaches taken by Team CERBERUS [2], Team CoSTAR [3, 4], Team NCTU [5], and Team Explorer [6]. The solutions designed by these teams relied on both inter-agent and agent-operator communication in order to explore the environment and find objects of interest. The developed communication solution resulted in teams constructing a “communication backbone” that would allow information sharing (e.g., environmental maps, odometry information, detected objects, etc.) with the other agents in the team [7]. Each of the aforementioned teams relied on both robotic agents and a set of “dropped” communication nodes to ensure that the agents in the system would remain in communication with the base station. Certain teams, such as Team CERBERUS, relied on a “Human Supervisor” to command agents to drop communication nodes. This decision was attributed to the team’s observations regarding challenges with ascertaining consistent measures of communication performance in SubT environments [2]. Team CoSTAR designed a robotic behavior that considered both environmental factors, relative line-of-sight (LOS), coverage, and network conditions to determine whether a drop should occur [4]. Team NCTU’s solution evolved over the DARPA SubT challenge. The original approach designed by Team NCTU consisting of solely dropped communications nodes was augmented with mobile communication nodes mounted to small wheeled robotic platforms. The mobile nodes could be moved in response to poor communication environments, producing a flexible approach that was not dependent on large numbers of dropped nodes [5]. Team Explorer designed a set of distance-based heuristics for both LOS and non-line-of-sight (NLOS) operations, which, when combined

with a map prediction algorithm, could be used to minimize the number of dropped nodes [6, 8]. Our proposed system takes lessons learned from each of the SubT performer teams: human supervision (CERBERUS), network-conditioned-based drops (CoSTAR), mobile communications nodes (NCTU), and NLOS (Explorer) operations in order to improve overall system performance.

Clearly, the design of network construction algorithms is highly dependent on the placement of communication relays throughout the environment. This placement is influenced not only by the environment itself, but also by the current known information about the environment and the communications system. Although both our operating conditions and the environments in which SubT agents performed in were *a priori* unknown, *a priori* known environments give rise to computational geometry techniques for sensor placement strategies in wireless sensor networks [9]. Certain techniques frame the problem as a variant of the Art Gallery Problem: a visibility-based approach to predicting sensor placement. Solutions to the Art Gallery Problem minimize the number of sensors (guards) required to cover a (classically) polygonal area [10]. Different sensor models modify the expected visibility of the guards, yielding different solutions to these optimization problems. Common modifications include limiting the visible range of any placed sensor [10], distance-based visibility fading [11], and visibility through a limited number of walls (k-transmitter problem) [12]. Although some of these geometric approaches translate to robotic systems, the quality of the information that travels through the communication system also comes into play.

The different performer teams considered multiple measures of wireless

communications quality for use in the DARPA SubT Challenge. Certain teams (CERBERUS) side-stepped the issue entirely by relying on human operators, noting that common radio signal strength indicators (RSSI) were often not reliable in subterranean environments [13, 14]. However, other teams (NCTU and Explorer) relied on RSSI to determine the locations of the node drops [5, 6]. Finally, utilizing signal-to-noise (SNR) has been considered in multiple works adjacent to Team CoSTAR [3], primarily with a focus on planetary exploration. Predicting these communications metrics is challenging, with recent work relying on machine learning techniques to predict the strengths of a 5G signal in indoor environments [15]. Such techniques can be useful if direct access to SNR values is not provided by the communication system or if the expected SNR value at a position in the environment must be predicted beforehand.

Finally, while many of the approaches deployed in the DARPA Subterranean Challenge demonstrate the requirement to maintain communication with a central communication node, this requirement is often relaxed to reflect the mission context. If agents are able to operate independently or with limited oversight, this could allow for operations with intermittent connection between the agents or between the agents and a base station or human operator. This acceptability of an intermittent communication yields a fundamentally different problem from those previously considered. Although the application considered in this paper is focused on developing robotic agents that directly support dismounted operators when required, in the interest of completeness, we also wished to detail a number of recent works that allow intermittent communications. Wang et al. [16] demonstrate a unique

solution approach that develops a communication signal map online. In particular, an uncrewed aerial vehicle learns a signal-to-interference-plus-noise map while avoiding an adversarial communication jammer. Although such an approach may be extensible to a communication relay node placement problem, the presented formulation is predicated on a fixed communications environment and allowable finite-time communication service interruptions in order to facilitate exploration in the communication signal space. In a similar manner, Woosley et al. [17] use a Gaussian process to model information entropy and communication signal strength for simultaneous exploration and information collection, but again does not require the exploring agents to maintain communication with a central base station or operator. Finally, we also would like to highlight an approach that requires information transfer between agents but does so by intermittently sharing maps at scheduled rendezvous locations. This approach enables agents to share information without having to use a relay network to maintain continuous connectivity with a base station [18]. This method works well to quickly explore an unknown area, but does so at an increased operational independence of the agents themselves (i.e., the agents may not be able to communicate with each other until a rendezvous occurs).

3. TECHNICAL APPROACH

In order to ensure communications during a robotic mission, we propose a graph-based mobile ad hoc network (MANET) construction system inspired by our experience in the DARPA Subterranean Challenge. We build on these experiences by replacing dropped “communication nodes” with a formation of cooperative ground



Figure 1: The team of robotic agents utilized in this work. The quadruped agents are compatible with the presented approaches but are not demonstrated herein.

agents. We first provide an overview of the hardware systems that constitute our approach and then discuss the proposed behaviors.

3.1. System Overview

The developed robotic system consists of a heterogeneous team of robotic platforms (agents) and an operator interface. The agent team (shown in Fig. 1) may consist of legged or wheeled platforms. Each agent is equipped with a custom-built payload that contains on-board compute, exteroceptive sensors (both a Light Detection and Ranging (LiDAR) sensor and cameras) and an inertial measurement unit (IMU). This payload enables the agents to simultaneously map their environment and compute their odometry relative to their starting location (e.g., the position of the operator interface). Each agent is also equipped with an onboard radio that allows it to communicate information to the other agents and the operator interface. The system uses Robot Operating System (ROS) and Data Distribution Service (DDS) middleware for communication between the base station and robotic agents. See [19] for a comprehensive discussion of the system architecture.

The operator interface displays information about each robotic platform (e.g., position, current waypoint, relative



Figure 2: A multi-agent convoy consisting of three vehicles. Heterogeneous convoys are also possible using our system.

experienced radio signal strength) and serves as the primary interface for sending commands to the agents. For robotic behaviors that require centralized decision-making, the operator interface also serves as the central point to coordinate the actions of each platform. For convenience, the operator interface may be abbreviated as “BST” (“base station”) in the remainder of this manuscript.

Of particular note is the system operating concept. In order to decrease operator workload and improve system redundancy, the system has been designed to perform multi-agent platooning (convoying) in a linear formation [20]. This platooning behavior requires multiple agents to form into a linear formation and travel together toward a common mission objective. The formation order is determined autonomously or by a human operator. The agents can then autonomously or in a guided manner, where an operator only controls the lead robot, travel to the mission objective. An example of a wheeled platoon is shown in Figure 2.

We desire a methodology to ensure that the platoon of agents remains in communication with the base station throughout the operation. Initially, we relied on an operator to manually “peel-off” an agent from the convoy when the convoy was about to leave communication range. Measurements of relative inter-agent communication strength (e.g., RSSI or SNR)



Figure 3: An operator commanded “peel-off” in an urban environment.

were displayed on the operator interface to assist the operator in determining “peel-off” locations. Triggering a peel-off would cause the last agent in the convoy to leave the convoy and stop. When the agent stops, it acts as a **communication node**, extending the effective communication boundary by relaying information and commands between the operator interface and all robotic convoys or agents in communications range. In order to minimize the number of network connections, convoys are treated as a single “composite” entity, where all information and commands to a convoy are filtered by and passed through the lead agent of the convoy. The lead agent then distributes commands to the remaining convoy agents until the agent leaves the convoy or the convoy is split by the operator. An example of such a peel-off is shown in Figure 3. We thus define an automated **peel-off behavior** as a behavior executed by an agent participating in a convoy that causes the agent to 1) leave the formation, 2) come to a complete stop, and 3) relay communications to other assets in the system. We modified this manual peel-off to be callable by the lead agent of an autonomous convoy. This enables the lead agent to direct the following agents to act as communication nodes if a set criterion is met. We describe this behavior as an “automated peel-off” behavior and describe the criterion for initiating a peel-off in the following section (Section 3.2).

3.2. Network Construction Behavior

Determining when to “peel-off” agents in an automated manner is fundamentally connected to the connectivity and interactions between the system assets. Graph data structures provide a convenient means to represent the interrelatedness of the assets in our system. In our communications systems, this interrelatedness of assets exists in at least two representative aspects. There is an inherent coupling between 1) the physical (spatial) representation and 2) the signal strength representation of the communication network. For a MANET, the “mobile” nature of a robotic system makes this coupling even more prevalent. As such, we pose our network construction behavior as a graph analysis problem in which the network topology changes as the robotic agents navigate through the environment.

The network topology is constructed from a set of n communications-enabled assets, including: 1) the operator interface, 2) existing friendly environmental communication infrastructure (if any), 3) individual robotic agents, and 4) formations/teams of robotic agents. For simplicity, we enumerate the total list of assets as $\mathcal{C} = \{c_0, c_1, c_2, \dots, c_{n-1}, c_n\}$ with index set $\mathcal{I} = \{0, 1, 2, \dots, n - 1, n\}$. We define the set of agents in the asset list as $\mathcal{A} \subset \mathcal{I}$ with corresponding index set $\mathcal{I}_a \subset \mathcal{I}$, and, for simplicity, denote asset c_0 as the operator interface. As these assets represent physical entities (e.g., a robotic agent), we define a mapping $p : \mathcal{C} \rightarrow \mathbb{R}^n$, which takes an asset index and returns its physical location with respect to a predetermined common reference frame (e.g., the coordinate frame of the lead vehicle).

We represent the relationships between the assets using a pair of undirected weighted graphs. The first graph, $G_d = (V, E_d, w_d)$, captures the physical distances between the assets (i.e., the relative spatial positions). In

this graph, each vertex $v_i \in V \subseteq \mathcal{C}, i \in \mathcal{I}$ represents an asset. The edges connecting all vertices are assigned a weight equivalent to the ℓ_2 -distance between agents (i.e., $w_d = d(v_i, v_j) = \ell_2(p(v_i), p(v_j)), v_{i,j} \in V$). Note that while we overload the notation of $p(\cdot)$, we intend for its meaning to remain the same.

Each asset may also be represented in the network logical layer topology, which describes the communications connections and relative signal strength the asset has to other assets. We name this representation a “**communications graph**” and define it as: $G_c = (V, E_c, w_c)$. In this definition, the vertex set (V) remains consistent with G_a (i.e., vertices represent assets), but the definition of the edge set does not. In the communications graph, the edges represent the ability for two assets to communicate. These edges are weighted by a corresponding weighting function $w_c = \text{COMM}(v_i, v_j), v_{i,j} \in V$. As discussed in Section 2, this $\text{COMM}(\cdot)$ function can represent a number of different indicators for signal strength. Finally, note that both graphs are dynamic in-so-much that the relationships between the agents may change during operation (e.g., agent motion through the environment).

Although the communications graph captures the relative communications strengths between assets, it does not inherently indicate *when* an agent should be “peel-off” from a convoy. We propose that the establishment of a communication node (i.e., a “peel-off”) should occur in response to a potential loss of communication between the agents and the base station. We model the proposed mechanism as a restriction on the allowable edge weights assumed in the communications graph. Specifically, for every agent i in the agent list \mathcal{I}_A , there must exist at least one path in the communications graph that starts at the operator interface (v_0) and ends at the agent

(v_i) where all edge weights maintain a value greater than a minimum allowable edge weight. If we define this minimum allowable edge weight as C_{THRESH} and a path in the communications graph as a set of edges between two different vertices ($\pi(v_i, v_j), v_i \in V, v_j \in V, v_i \neq v_j$), then we seek to ensure that $w_c(e) \geq C_{\text{THRESH}} \forall e \in \pi(v_0, v_i), \forall i \in \mathcal{I}_A$. For notational simplicity, we drop the dependence on v_0 and choose to represent the paths between the operator interface and an agent i using a single argument (i.e., $\pi(v_0, v_i) = \pi(v_i)$).

To ensure this restriction is observed, we construct a maximin spanning tree, $T \subseteq G_c$, on the communications graph rooted at the operator interface (v_0). The tree is constructed such that the edge weights associated with edges included in tree T maximize the strongest “weakest” link between a parent node and its child node in the tree. Concretely: for each non-root vertex, v_i , connected by edge (v_i, v_j) to its parent node, v_j , an associated edge weight $\text{CMET}(v_i, v_j) \in \mathbb{R}^+$ is chosen such that $\text{CMET}(v_i, v_j)$ represents the strongest “weakest” link for v_i in the communications graph. As the spanning tree T spans all vertices included in communications graph, and by ensuring that $\text{CMET}(v_0, v_i) \geq C_{\text{THRESH}}, v_i \in V \in T$, we ensure that at least one path between v_0 and v_i exists such that the minimum edge weight is observed for all edges in the tree.

We describe any communication nodes that represent the operator interface or an agent that has performed a peel-off behavior as a **central node**. For ease of reference, we collate these central nodes into a set: CN . Note that, initially, $CN \leftarrow \{v_0\}$ as the list of central nodes always includes the node representing the operator interface. Thus, these central nodes represent our version of the vital “communications backbone” described by many DARPA SubT works [7].

Maintaining a connection to at least one central node guarantees a communication pathway to the operator interface.

Thus, we can define the network topology \mathcal{N} of the system by considering elements of the communications graph. The network topology itself consists of a “communications backbone” of central nodes and the remaining mobile agents. There exists an effective communications boundary $\mathcal{B}(\mathcal{N})$ created by all the assets in the system. The strength of the system’s communications within this boundary as observed by the system assets is not less than C_{THRESH} . That is, \mathcal{B} forms a level set $L = \{x \in \mathbb{R}^n \mid \mathcal{B}(x) = C_{\text{THRESH}}\}$, where $x \in \mathbb{R}^n$ denotes a spatial location in the environment. This communications boundary delineates the limit of the MANET coverage area, crossing which will lead to loss of communications for the agent that crossed the boundary. The automated peel-off behavior described above enables agents to, in a methodological way, extend the communications boundary \mathcal{B} by tasking an agent to act as a communication node. This communication node extends the current communications boundary, thus guaranteeing that the remaining agents remain in contact with the operator interface with a minimum communications strength C_{THRESH} .

3.3. Re-optimization Behavior

As the robotic formation travels through the environment, we observe that the “peeled-off” robotic agents form a type of kinematic “chain” from the base station to the target. By analogy to a planar robotic manipulator, the agents would be synonymous to joints, and inter-agent distances synonymous to robotic links. Please see Fig. 4 for a toy example depicting our observation. In effect, the central nodes form a stationary network topology configuration for a particular mission objective. However, if the mission objective

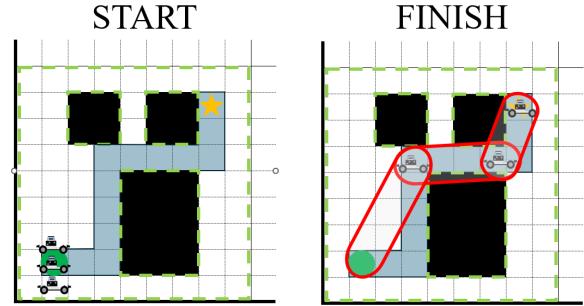


Figure 4: A toy example provided in a pair of diagrams with the starting configuration on the left and the ending configuration on the right. In the diagrams, a team of three robotic agents starts at a base station location (green circle) and are tasked with traveling to an objective (gold star) while avoiding the objects (black). The agents travel as a convoy through the dark blue squares, peeling-off at different intervals. In the final configuration, the agents are positioned similar to a kinematic chain (outlined in red).

is changed or an additional objective is provided, the central node configuration may not be adequate to provide communications coverage to that new objective. Thus, for small deviations from the initial configuration, we propose modeling the set of central nodes as a pseudo robotic manipulator. Using this model, we can then locally re-optimize the position of each robotic agent through inverse kinematic techniques and extend the communications boundary to achieve the next mission objective. An example of this technique is shown in Fig. 5.

Consider N agents participating in a convoy whose locations must be optimized in response to a new objective $x_g \in \mathbb{R}^n$ (e.g., the purple star in Figure 5). The current spatial location of each agent is given by $x_i \in \mathbb{R}^n$ with $i \in \{1, 2, \dots, N - 1, N\} = \mathcal{I}_C$. Furthermore, require the order of \mathcal{I}_C to represent adjacent agents in the chain (i.e., agents connected by the same “link”, this can be found using the maximin tree). For convenience, we also define set $\mathcal{I}_{C \setminus N} = \{0, 1, \dots, N - 2, N - 1\}$, with 0 (and the corresponding x_0) representing the

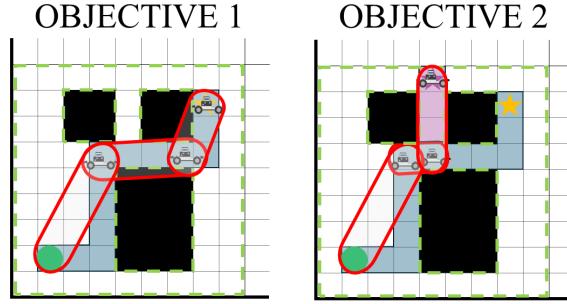


Figure 5: The toy example presented in Figure 4 extended with an additional objective (purple star). In the diagrams, a team of three robotic agents starts in the configuration resulting from their travel to the first objective (gold star) while avoiding the objects (black). The agents travel to the new location through the purple squares, changing the inter-link distances (similar to a prismatic actuator) before peeling-off at new locations intervals. In the final configuration for the the new objective, the agents are positioned into a new kinematic chain (outlined in red).

base station position. In order to optimize the set of agent positions, we concatenate the decision variables into a vector: $X \in \mathbb{R}^{(n \cdot N)}$. For convenience, we create a similar vector for the goal state, X_g , by concatenating the goal state N times except for the first element, which assumes a value x_0 . Finally, we overload function $d(x_i, x_j) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ by defining it to measure the Euclidean distance between two points x_i and x_j .

For this behavior, we consider a communications model with distance fading, but no obstacle attenuation. The maximum allowable distance between two agents such that they remain in communication is given as r_C . For sequential agents i and j , we can then form a quadratic constraint of the form

$$\text{CB}(i, j) = d(x_i, x_j) - r_C^2 \leq 0, \quad (1)$$

which captures the allowable spacing between agents to ensure that they remain within the communications boundary. Although this work only considers simple geometric constraints consistent with our communications system model, we

recommend that this constraint be modified in future studies to a more complex model.

In this re-optimization method, while the communication links are not influenced by the presence of obstacles, the agents themselves cannot achieve a position inside an obstacle. As the agents map the environment, they each contribute to a global occupancy grid, \mathcal{G} , consisting of square cells of constant size which delineates the space into object-free and object-occupied space. By iterating through the set of unoccupied cells in the occupancy grid, \mathcal{G}_U , we can enforce a constraint on the agents' positions. We do so by convexifying each occupied cell, $u \in \mathcal{G}_U$, by a capsule of constant radius r_o , where $r_o > \sqrt{\frac{l^2}{2}}$ and l represents the resolution of a occupancy grid cell. If we represent the center of a grid cell as x_u , then we can write the object avoidance constraints on the between agent i and occupied cell u as

$$\text{OB}(i, u) = r_o^2 - d(x_i, x_u) \leq 0. \quad (2)$$

Combining the above constraints yields a quadratically constrained quadratic program that can be solved on the base station to re-optimize the location of the agents. This program takes the form:

$$\min_X \frac{1}{2}(X - X_G)^T(X - X_G) \quad (3)$$

$$\text{subject to } \text{CB}(i, j), i \in \mathcal{I}_{C \setminus N}, j = i + 1, \quad (4)$$

$$\text{OB}(i, u), i \in \mathcal{I}_c, u \in \mathcal{G}_U, \quad (5)$$

$$X(0) = x_0. \quad (6)$$

Note that we further post-process the results to ensure that no path experiences large deviations from the original vehicle configurations. To bias the results away from large deviations, we perform a local change of coordinates for X , biasing the vector

towards the initial positions of the agents: $X' = X - X_{IC}$ with X_{IC} representing the initial configuration of the vehicles.

3.4. Network Repair Behavior

During real-world operations, robotic systems are prone to failure (e.g., agent onboard power loss). If such a failure occurred, the resulting communications graph may include an edge with an insufficient edge weight (i.e., below C_{THRESH}). In order to improve system robustness to such failures, we propose a communications network repair behavior that reallocates agents in order to re-establish communications with the root node of the maximin communications tree.

We adopt an approach that repositions at least one agent back to the communications boundary of the failed node. The proposed communications network repair behavior is described in Algorithm 2. The algorithm runs locally on each agent (v_a) and is activated when the signal strength between the agent and the agent's parent node (v_p) in the maximin communications tree falls below the threshold value (i.e., $CMET(v_a, v_p) < C_{THRESH}$). Note that this implies that central nodes may also move to repair the network if their signal strength falls below C_{THRESH} . Given that Algorithm 2 runs on each agent, the motion of an agent acting as a central node could activate the recovery behaviors of any agents dependent on v_a , causing a potentially large number of nodes to reposition in response to a singular node failure. We recommend further study of repair methodologies that minimize the number of agents impacted by the repair behavior (further discussed in Section 6).

Before providing an overview of the algorithm, we outline the underlying assumptions of our approach. We first assume that the root node of the maximin communications tree (i.e., the base station)

Algorithm 1 Closest Central Node (CCN)

Require: CN_L, \mathbb{V}, x
Ensure: $x_{cn} \in CN_L \triangleright$ Closest central node
1: $i = \arg \min(d(x, x_i) \forall i \in CN_L, i \notin \mathbb{V})$
2: **return** $x_{cn} = CN_L[i] \triangleright$ get i from CN_L

Algorithm 2 Network Repair Behavior

Require: $x_a, x_p, CN_L, t, \text{COMM}(v_a, v_p)$
1: Initialize $\mathbb{V} \leftarrow \emptyset, x_g = x_p$
2: **while** $\text{COMM}(v_a, v_p) < C_{THRESH}$ **do**
3: **if** $d(x_a, x_g) < d_{min}$ **then**
4: **if** $d(x_a, x_0) < d_{min}$ **then**
5: **break** \triangleright reached root node
6: $\mathbb{V} \leftarrow \mathbb{V} \cup \{v_p\}$
7: $x_g \leftarrow \text{CCN}(CN_L, \mathbb{V}, x_a)$
8: $v_p = CN[g]$ \triangleright update v_p to v_t
9: Move towards x_g \triangleright for t seconds

does not fail and that all nodes have the same communications capabilities. To enable an agent to return to the communications boundary of its parent central node, each agent stores the locations of all central nodes in the network. Define the set of central node locations as $CN_L = \{x_0^{cn}, \dots, x_n^{cn}\}$, where $x_i^{cn} \in \mathbb{R}^n$ represents the location of central node i (e.g., $x_i^{cn} \in \mathbb{R}^2$ for a planar environment) and x_0 represents the position of the root node. In the case of central node failure, these locations represent the most likely locations to be able to reestablish the network. Finally, we assume that all locations in CN_L are reachable by all robotic agents from any location in the environment.

For convenience, we define a subroutine (Algorithm 1) to find the central node closest to robotic agent a . Define agent a 's position as $x_a \in \mathbb{R}^n$ and the set of central nodes visited by agent a as $\mathbb{V} \subset CN_L$. We first find the index of the central node closest (based on Euclidean distance) to a position x , (e.g., agent x_a) that is yet to be visited (Line 1), and then return that node's location (Line 2).

Given these assumptions and Algorithm 1, we propose the Network Repair Behavior in Algorithm 2. We first target the parent node of the agent in the maximin communications tree (Line 2), which is guaranteed to exist via our earlier assumptions. We then check to see if we are “close enough” (less than a threshold, d_{min}) to the target node’s location, x_g , on Line 3, and continue moving towards x_g if we are not (Line 9). If the distance between the target node x_g (initially, x_p , (Line 1)) and agent a becomes less than d_{min} and $CMET(v_a, v_p) < C_{THRESH}$, we assume the current central node location is not a suitable location to repair the network and we must target a different central node. We then check on Line 4 to ensure we are not at the root node (and terminate the algorithm if we are), before adding node v_p to the list of visited central nodes (Line 6). Following this, we then get the next closest central node position (Line 7) and update the next closest central node (Line 8). Consequently, the agent sequentially explores all these locations until it either establishes communication or all locations have been explored (i.e., $CMET(v_i, v_j) > C_{THRESH}$ or $\mathbb{V} = CN_L$).

An important point to note is that the algorithm is guaranteed to recover communications as long as a path exists to the root node of the maximin tree. This is due to the fact that the agent will ultimately reach the location of the base station (root node, x_0) if it cannot regain communications elsewhere.

4. NUMERICAL SIMULATIONS

The proposed communications network construction techniques and recovery behaviors were implemented in simulation. The numerical simulations are performed in environments that are representative of different real-world scenarios, including

buildings, cities, and natural environments. Unless noted otherwise, none of the test environments include existing communications infrastructure that the robotic team could utilize during its movement through the environment.

We first demonstrate our network construction capabilities on a set of maps from a Multi-Agent Pathfinding (MAPF) benchmark [21]. These results demonstrate the evolution of the network topology in complex environments using a communications model to guide the placement of the communication nodes. We also demonstrate both repair behaviors in simulation. The network repair simulation environments are modeled in Gazebo. These studies reflect how communication node failure is addressed using Algorithm 2.

4.1. Network Construction Studies

We first simulate a series of missions that require a team of robotic agents to travel through different environments with no existing communications infrastructure. The objective of this set of tests is to determine the required number of robotic agents needed to construct a communications network between an initial start location and a goal location. The environments are drawn from Stern’s Multi-Agent Pathfinding benchmark [21]. Each environment is represented as an occupancy map, where the measure of occupancy denotes whether a robotic agent may traverse a cell in the occupancy map. Although the network construction algorithms described in Section 3 are not dependent on an *a priori* possession of each environment’s occupancy map, each map is assumed to be known beforehand in order to compute the shortest path between the start and goal states (e.g., via Dijkstra’s Algorithm). Given this shortest path between the start and goal states, a communications model is then propagated

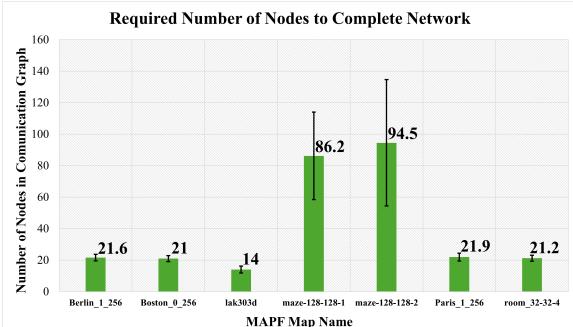


Figure 6: A summary of the network construction tests demonstrating the required number of nodes needed to construct a communications network in the MAPF environments.

from an agent (or the base station at the initial start state) along the path. Every robot that is deployed as a communication node then acts as a further communications source in this communication model, enlarging the communication-accessible area for the other robotic agents.

Although robotic agents are incapable of traversing through occupied cells, in these studies, we assume that the communications model is capable of such obstacle penetration. We represent our communications system using a communication model with distance fading: wherein the communications signal strength is inversely proportional between the source and receiver. This model is further augmented with additional signal attenuation arising from occupied cells. This additional signal attenuation is assumed at a rate of $20 \text{ [cell}^{-1}]$ (e.g., a 20 [unit] reduction in signal strength every 1 [cell] encountered).

We provide an overview of the results of our simulations in Figure 6. We also include a number of reference images for three maps, including: 1) *lak303d*, 2) *Boston_0_256*, and 3) *Berlin_1_256*, to depict the constructed network topologies. For each map, we generate one-hundred different start-goal configurations on the same occupancy map. Each starting position is denoted with a green circle, and each ending position is denoted

with a red circle. Using the communication model above, communication nodes are deployed whenever the signal strength falls below the threshold of 10 (i.e., $C_{\text{THRESH}} = 10$). Each communication node that is deployed is represented by a gold circle. The path along which the team travels is shown in orange.

Each environment demonstrates a varying amount of object clutter that must be navigated by the robotic agents. Large numbers of deployed nodes appear to arise in environments with sharp corners and on objects that must be traversed around that continuously block line-of-sight (*Berlin_1_256*, Fig. 7b). We also see that tight corridors, such as near the goal location in Fig. 7a, generally require large numbers of communication nodes to ensure communication quality along the route.

Remark (Re-optimization). *We additionally include a re-optimization of the nodal positions considered in Figure 7b to a new configuration in Figure 7d. The configuration change is motivated by the new goal location that the agents are required to visit. The contraction of the kinematic chain reflects a difference in required number of nodes (29 to 26), but the remaining agents largely remain in place.*

4.2. Network Repair Studies

This section demonstrates the efficacy of Algorithm 2 in different simulation environments. As in the previous simulation section, we adopt a scaled inverse model for our communications system. The communication metric, $\text{COMM}(\cdot)$, was defined to be inversely proportional to distance and scaled by a scaling factor k . This scaling factor is a tunable parameter to change the effective range of the communications model, with a larger scaling

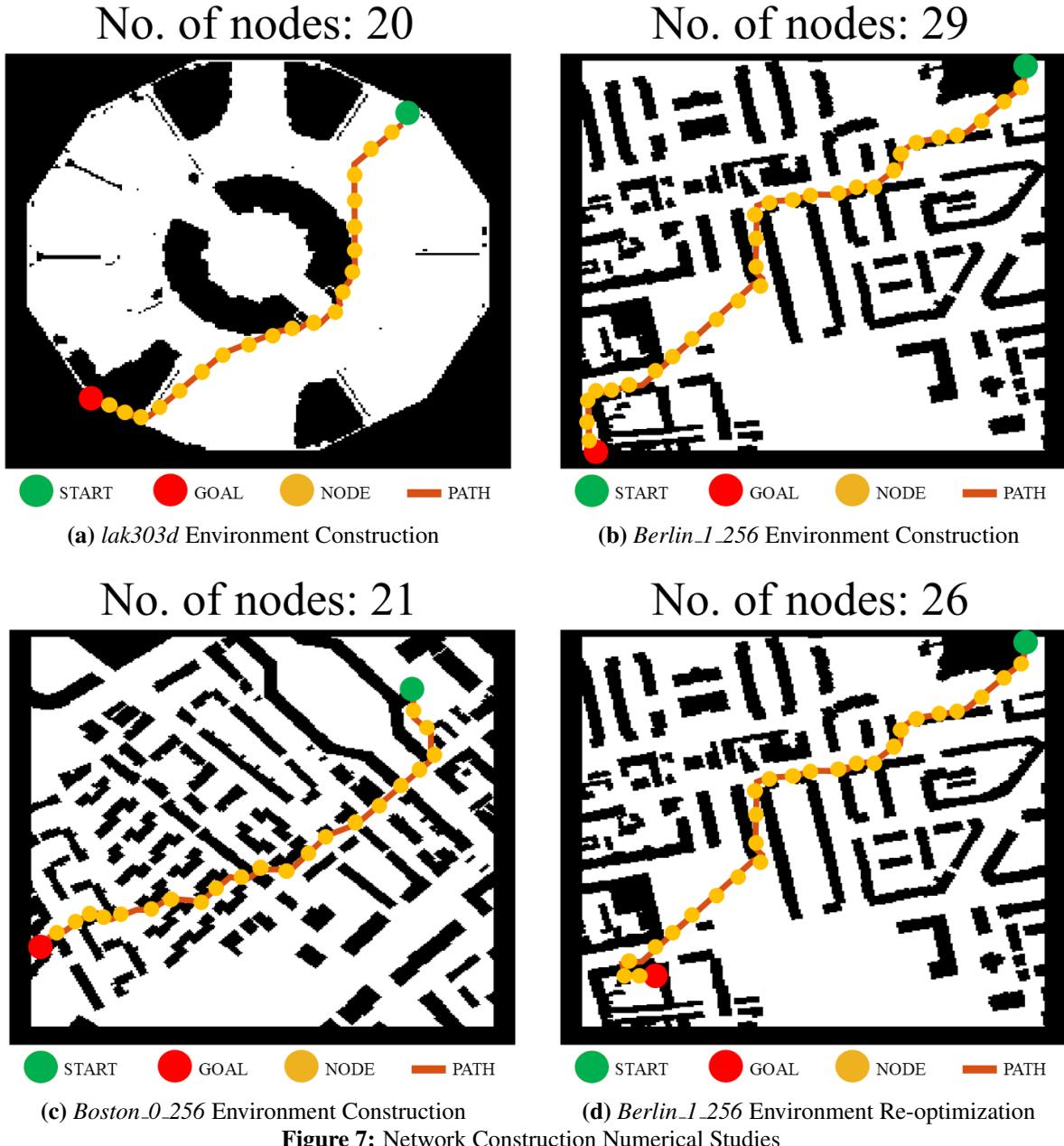


Figure 7: Network Construction Numerical Studies

factor correlated to a larger communications boundary. For example, for an agent 10 [m] away from a communications signal source with $k = 500$, our model would predict a signal strength of $\frac{1}{10} \times 500 = 50$. For each test demonstrated in this section, the value of the scaling factor and the threshold C_{THRESH} are specified.

We first present a scenario with five robotic agents and a base station (the root node) in

an indoor environment. The results of this test are shown in Figure 8. For this test, $k = 500$ and $C_{\text{THRESH}} = 20$. Figure 8a shows both the starting configuration of the agents in the communications graph alongside the $\text{COMM}(\cdot)$ values for each node with its parent. In this scenario, as “RC1”, “RC2”, and “RC5” have already peeled-off, the agents are included in the list of central nodes. To test the network repair behavior,

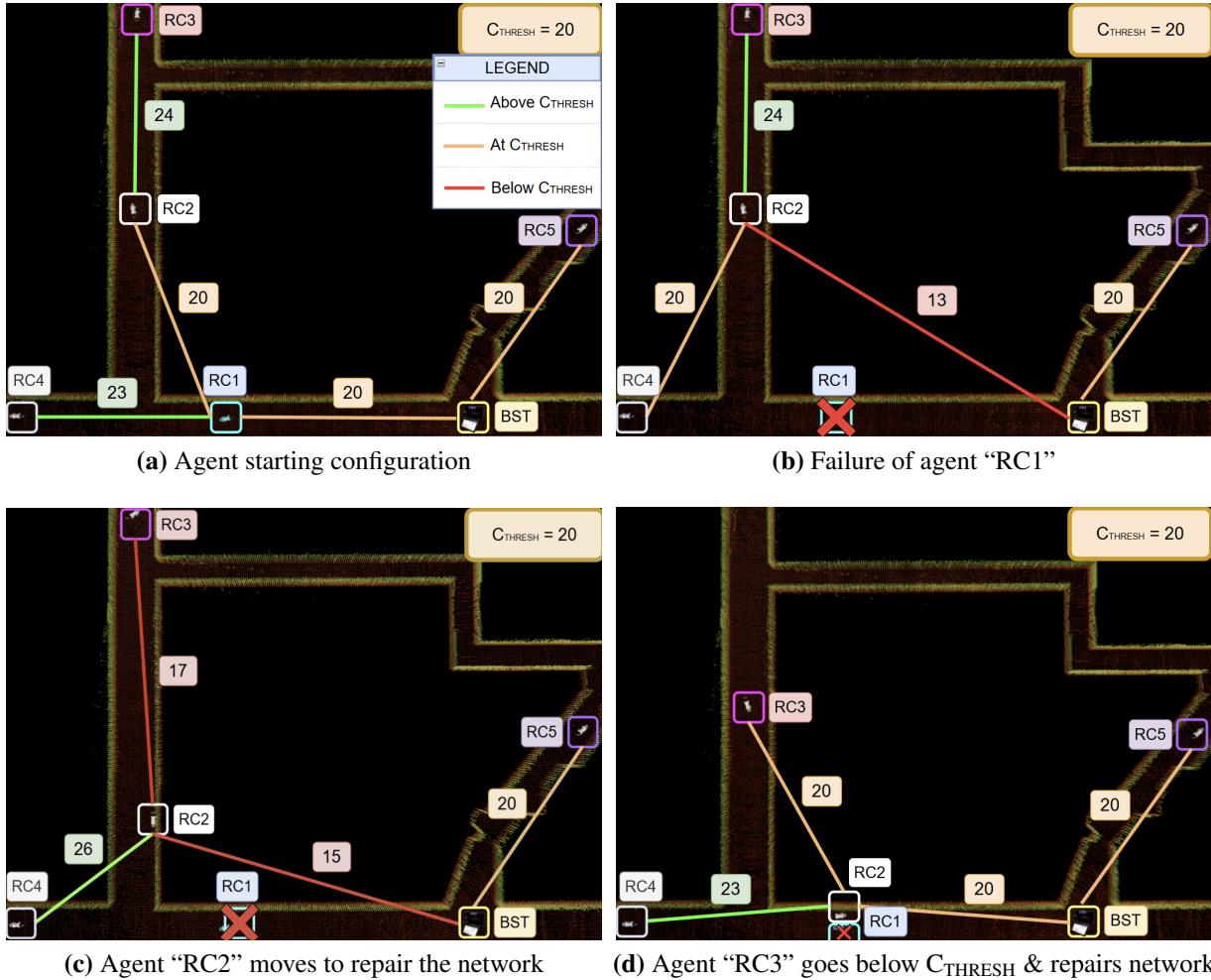


Figure 8: The first simulation study demonstrates the network repair behavior described in Algorithm 2. The study demonstrates the cascading effect of the network repairing process on multiple agents (“RC2” and “RC3”).

we simulate the failure of “RC1” which causes the agent to lose connectivity with all other agents and the base station. As seen in Fig. 8b, this sudden failure of “RC1” causes the $\text{CMET}(\cdot)$ value of “RC2” to drop below C_{THRESH} . Given the logic of Algorithm 2, the network repair behavior is activated in Agent “RC2”, compelling “RC2” to move towards the next closest central node (in this case, “RC1”). As seen in Fig. 8d, “RC2” takes the place of the failed “RC1” agent, effectively repairing the network. However, as seen in Fig. 8c, as agent “RC2” moves the $\text{CMET}(\cdot)$ value of “RC3” decreases below C_{THRESH} . This, in turn, triggers the network repair behavior in “RC3”, causing “RC3” to

take the previous position of “RC2”. While this experiment contains only five agents, it demonstrates the expected cascading effect associated with the network repair behavior on all agents influenced by the failure of a central node.

The second simulation study is shown in Fig. 9. For this test, $k = 1000$ and $C_{\text{THRESH}} = 25$. As in the previous simulation experiment, Fig. 9a shows the initial configuration of the communications graph along with the $\text{COMM}(\cdot)$ values for each node with its parent. In this study, “RC5” acts as the only non-root central node (i.e., excluding the base station) in the network. As such, we simulate the failure

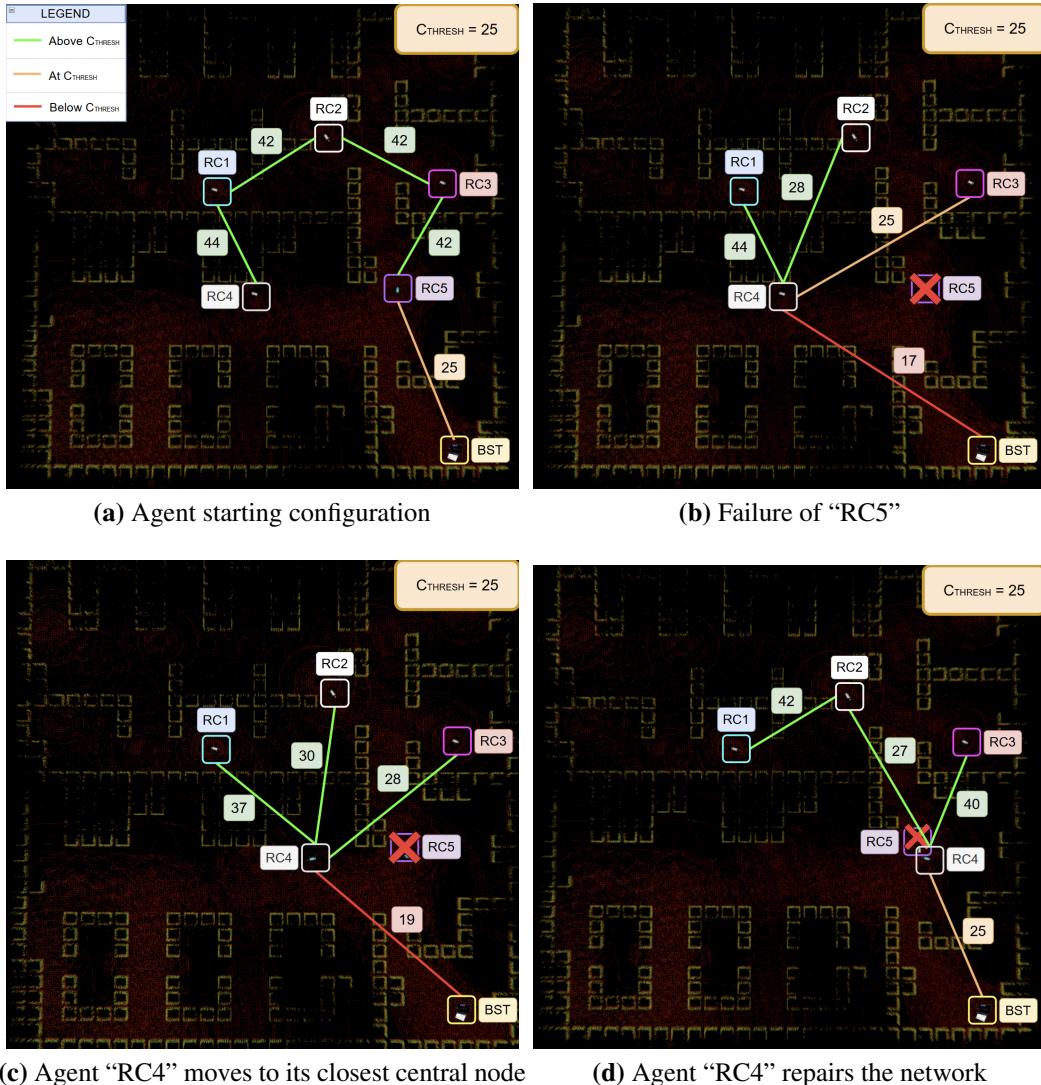


Figure 9: The second simulation study demonstrates how the network repairing process only is initiated by agents affected by the communication dropout ("RC4").

of agent "RC5" to investigate the reaction of the system. Figure 9b demonstrates the effect of the failure of "RC5". The failure of "RC5" triggers the network repair behavior in "RC4", causing it to take the place of "RC5" and repair the network (Fig. 9c). Note that none of the other robots were affected by the failure of "RC5" as "RC4" moved to repair the network, enabling the other agents to continue their tasks. This experiment demonstrates the effectiveness of the network repair behavior in scenarios where only a single communication node fails.

5. HARDWARE TRIALS

This section details a set of hardware trials that were conducted on a team of wheeled robotic platforms. We demonstrate both our network construction technique and network repair behavior on the small robotic team. We conclude the section with a discussion on computation time.

5.1. Convoy & Network Repair Trials

We choose to demonstrate both the network construction behavior and the network repair behavior on a three-agent

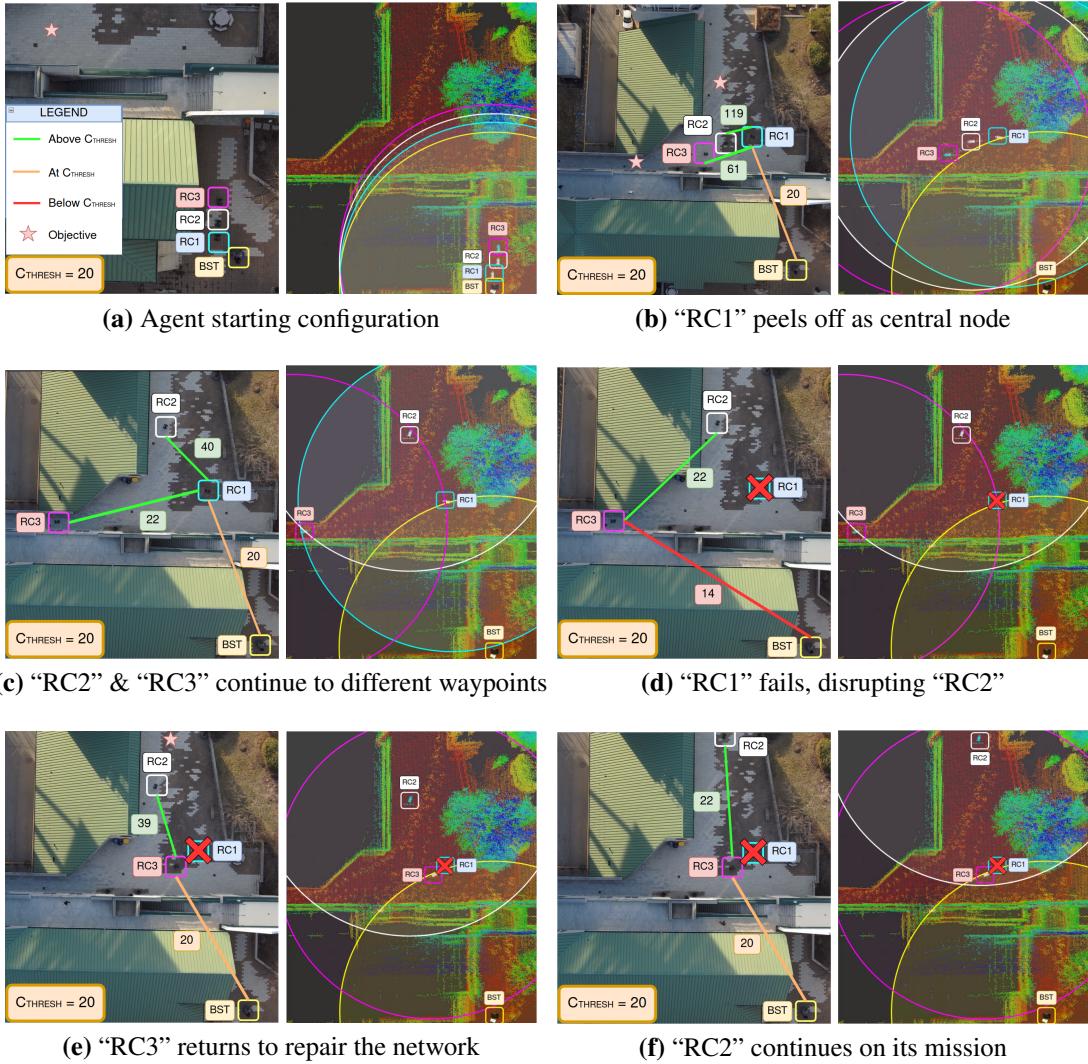


Figure 10: A system hardware test for the network construction and network repair behaviors. Each image is composed of a full-color aerial image on the left and a corresponding point cloud map –visualized in RViz– that is seen by the operator at the base station. The RViz images show how each node contributes to furthering the communications boundary, enabling the team to reach objectives not initially within reach of the communications

robotic team. Fig. 10 shows the time lapse of a mission in a city environment using the hardware described in Section 3.1. For this hardware trial, $k = 500$ and $C_{\text{THRESH}} = 20$.

In Fig. 10, each real world image is accompanied by the corresponding point cloud map –visualized in RViz– that is seen by the operator at the base station. The RViz images show how each node contributes to furthering the communications boundary, enabling the team to reach objectives not initially within reach of the communications

system. The subfigures also demonstrate the progression of the network construction throughout the mission. Specifically, the different images show how the agents change their behavior to act as communications nodes in order to ensure communication with the base station.

Figure 10a shows the agents starting as a convoy (formation) in the vicinity of the base station. The agents are tasked with achieving the waypoints marked by the pink stars in Fig. 10. The behaviors and formation

structure associated with the agent convoy are further detailed in our previous work [19, 20]. As the mission progresses, “RC1” “peels-off” as it reaches the communications boundary created by the base station and establishes itself as a central node (Fig. 10b). This enables “RC2” and “RC3” to continue on the mission, as they can now use “RC1”, which is stationary, to relay messages to the base station. “RC1” effectively “extends” the communications boundary, enabling “RC2” and “RC3” to reach the first waypoint.

Agents “RC2” and “RC3” then encounter a fork in the road that requires the two agents to travel in different directions to achieve two new waypoints. The agents diverge to explore each route, as shown in Fig. 10c.

To demonstrate the network repair behavior (Algorithm 2) during active operations, we then trigger a failure in “RC1” that causes it to drop from the communications network. This is shown in Fig. 10d. The failure of “RC1” causes “RC2” and “RC3” to leave the communications-accessible area, as the signal strength experienced by both robots drops below C_{THRESH} . The network repair behavior causes “RC3” to replace “RC1”, enabling “RC2” to continue its mission (Fig. 10e & Fig. 10f). This experiment demonstrates the functionality of both the network construction and network repair behaviors in real time on robotic hardware for a simple real-world mission.

5.2. Network Construction Trials

This section discusses the results of the network construction trials using both the “automated peel-off” behavior described in Section 3.1 and the maximin criterion described in Section 3.2.

Fig. 11 demonstrates the “automated peel-off” behavior for a convoy of robots. For this hardware trial, we set $k = 500$ and $C_{\text{THRESH}} = 24$. Fig. 11a shows

the convoy starting in the vicinity of the base station. Next, Fig. 11b shows the convoy progressing towards the objective (represented as a pink star). As the convoy progresses to the objective, “RC1” peels-off as its $\text{COMM}(\cdot) = C_{\text{THRESH}}$. Fig. 11c then shows the convoy’s progression until it reaches the communications boundary, causing “RC2” to peel-off. Finally, Fig. 11d shows “RC3” reaching the objective, using “RC1” and “RC2” as communication relays.

The behavior of the maximin spanning tree is demonstrated in Fig. 12. For this hardware trial, $k = 500$ and $C_{\text{THRESH}} = 25$. Each image shows the $\text{COMM}(\cdot)$ values between pairs of nodes and the $\text{CMET}(\cdot)$ value for each node given by the maximin tree. Fig. 12c and Fig. 12d show the two different outcomes of the experiment, with the former being when all nodes are active and the latter being when “RC4” fails.

In Fig. 12a, “RC1” connects to the base station via “RC3”. As “RC1” moves towards its objective, it finds a better connection through “RC2” as seen in Fig. 12b before reaching its objective as seen in Fig. 12c. However, if “RC4”, which acts as the link between “RC2” and the base station, fails, it weakens the connection between “RC2” and the base station. This causes “RC1” to connect to “RC3” instead, causing $\text{COMM}(\cdot) = C_{\text{THRESH}}$, triggering a peel-off. This causes “RC1” to fail to reach its objective, which is reflected in Fig. 12d.

The difference between the two cases is evident in Fig. 12b and Fig. 12d. In Fig. 12b the $\text{CMET}(\cdot)$ value of “RC2” is 29 whereas it is 26 for “RC3”. Hence, as “RC1” approaches “RC2”, it connects via “RC2” because its $\text{CMET}(\cdot)$ value is stronger than the “RC3” $\text{CMET}(\cdot)$ value. However, in Fig. 12d the failure of “RC4” causes “RC2” to connect to “RC1” instead, as the “RC1” $\text{CMET}(\cdot)$ value is 25, which is higher than any of the other possible values for “RC2”. As a

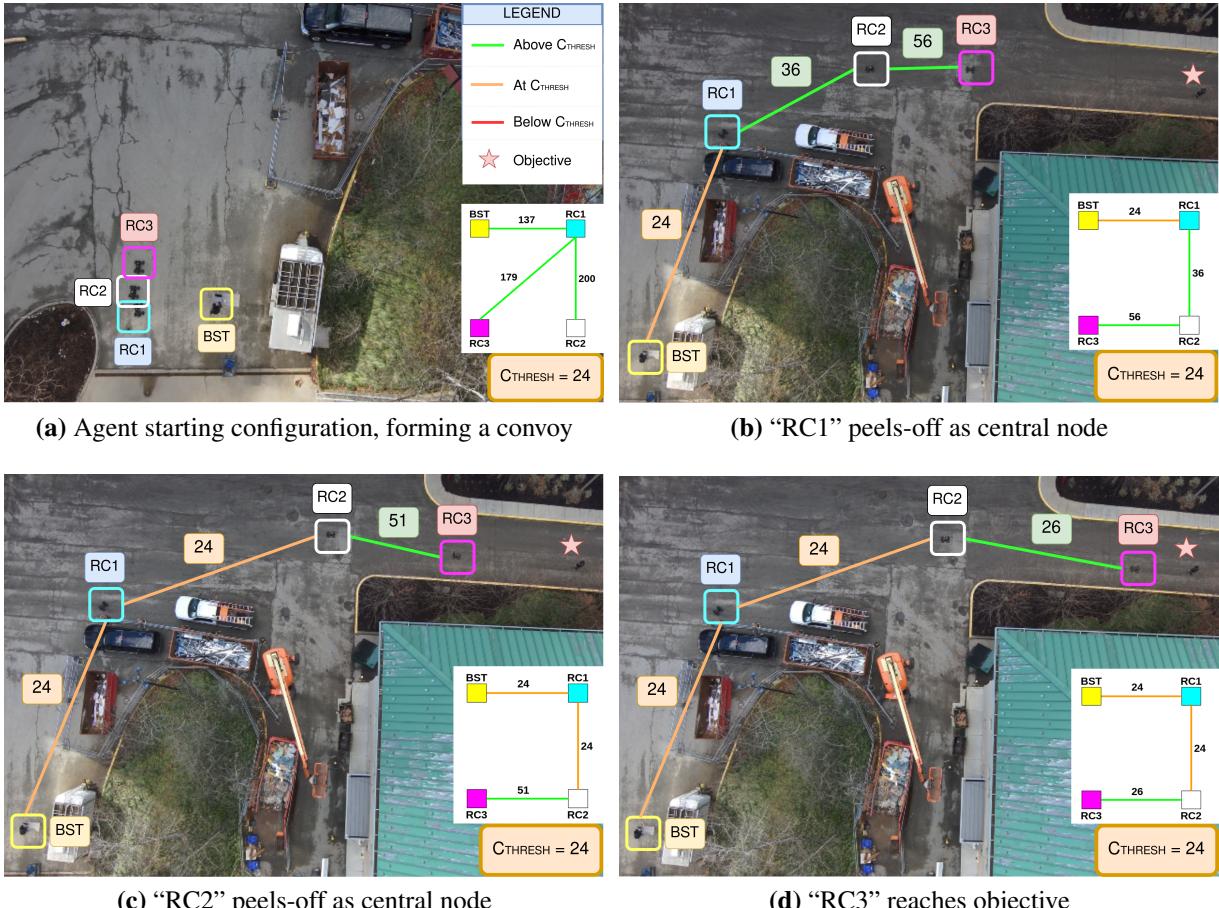


Figure 11: A depiction of the automated peel-off behavior showing how robots in a convoy “peel-off” and act as relays to enable mission completion. This behavior forms a crucial part of the network construction technique. Each image showcases the communication graph representing the network topology on the bottom right, and the physical layer representation of the same tree overlaid on the images.

result, “RC1” remains connected to “RC3” and peels-off.

5.3. Computation Time Trials

Runtime performance is an important consideration for system operation. We demonstrate the scalability of our algorithms in a series of empirical tests, the results of which are tabulated in Fig. 13. We benchmarked the increase in computation time for the network construction algorithm as a function of network size (number of nodes). Fig. 13 reports the time taken to construct the maximin spanning tree from the communications graph. As expected, the time increases considerably as the number of

nodes in the graph increases. As the network construction algorithm has been implemented in Python, significant speed gains are expected to be seen if the implementation is changed to use C++.

6. CONCLUSIONS

The work presented in this manuscript only scratches the surface of the many challenges in developing communications-aware robotic teaming behaviors. We have demonstrated how collaborative mobile robotic teams can be utilized to develop a mobile ad-hoc network (MANET) that enables communications in communications-deprived environments.

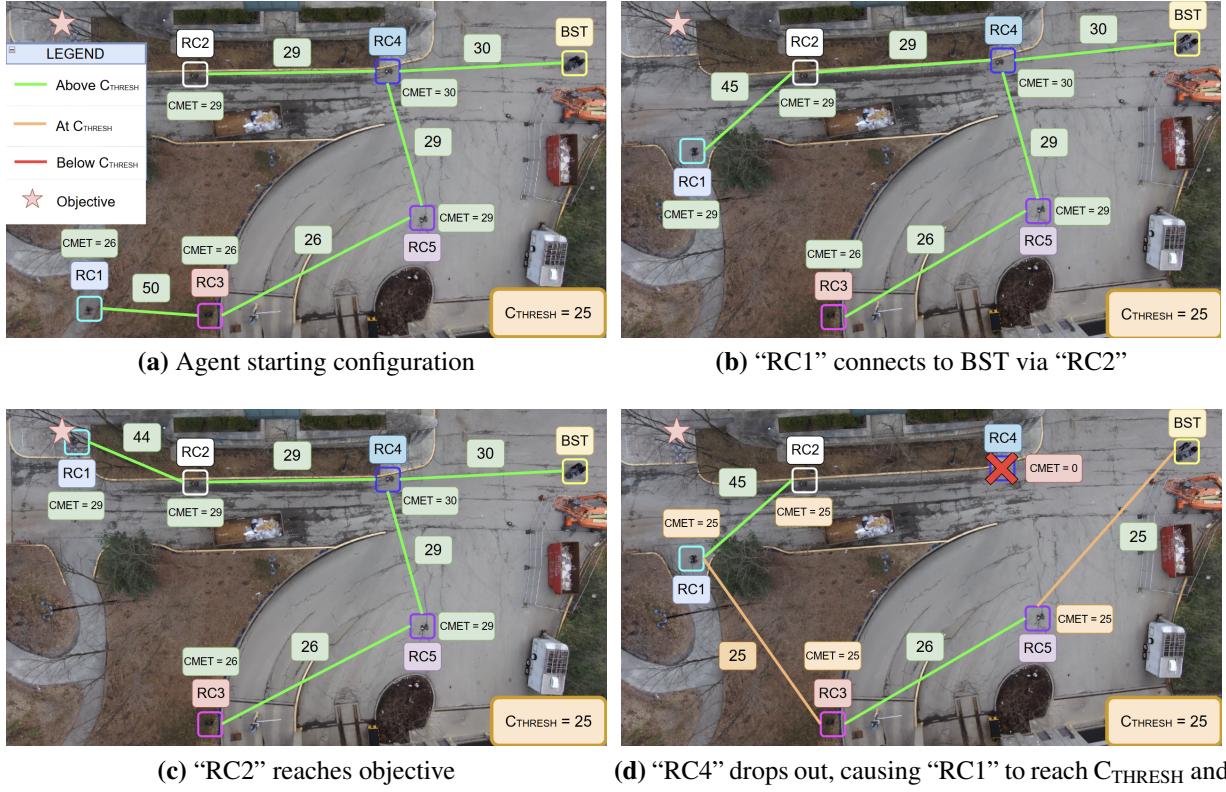


Figure 12: Demonstration of the maximin criterion for constructing the tree. This criterion considers weak links in the chain of relays (represented by the $CMET(\cdot)$ values for each node), and not just the strongest connection. Fig. 12d shows “RC1” stopping despite having a strong direct connection with “RC2”. This is because the connection between “RC2” and “RC5” is weak, causing “RC2” to connect to “RC1” instead. Note that links with zero signal strength have not been shown.

After demonstrating the ability to form a network during a mobile robotic operation, we then demonstrated a number of network recovery and repair behaviors that we have developed to address challenges associated with real-world operations. While the presented hardware demonstrations denote the technical competence of the system, multiple avenues of improvement can be made to increase the robustness and capabilities of the system.

Our experiments have shown that accurate communications and signal strength models can be used to guide the creation of a communications network. Realistic models of a robot’s communications systems are often highly coupled to the hardware

utilized in the system. As simple geometric models are presented in this manuscript, the robotics and communications community should continue to strive to develop increasingly accurate models of signal and communications strength. Specifically, incorporating a more accurate model of the communications system that accounts for object interference could help identify better locations during the re-optimization process.

Second, but along the same line of reasoning, the lack of an available environmental map before system operation requires the presented approach to be reactive in nature. By this we mean that agents must monitor the strength of the communications signal and then “peel-off” as a reaction to a

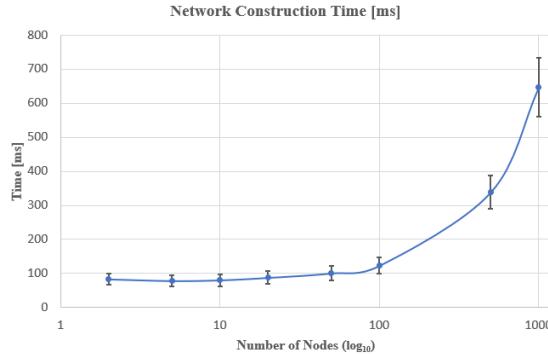


Figure 13: A summary of the computation times of the network construction algorithm. The reported times are averaged over 1000 runs, and signify the time taken by the most computationally intensive part of the algorithm (the maximin spanning tree construction.)

poor signal strength. Developing a system that predicts the environment map from a partial observation of the environment could be utilized to decrease the required number of agents by allowing the agents to predict the edge of the communications boundary before reaching it. Finally, developing an adaptive construction policy that enables the system to estimate a communications model online could provide additional benefits if environmental conditions cause the “true” communications model to be out of distribution with the deployed model.

The network repair experiments clearly indicate that a “smarter” graph optimization or search approach should be utilized to decide which agent acts to repair the network. The presented approach has high agent utilization that is not reactive to or predictive of environmental threats. It is clear that exploitative adversarial strategies could be employed to compromise a team of robotic agents using the network repair strategy presented in this manuscript. Instead of requiring agents to reach the location of the parent node in the maximin communications tree, a possible avenue for future research could include re-organizing the communications graph topology by employing the computational geometry

techniques discussed in Section 2 on the already explored map. Alternatively, relaxing constraints of continual network connectivity could enable the use of techniques such as intermediate or “repair” rendezvous points similar to those discussed in [18]. This can result in a more effective agent utilization with minimal changes to the network topology. Furthermore, our approach relies on the A* path planner to find a feasible path back into the communications boundary. However, there can be scenarios where the planner does not find a valid path because of unobserved regions of the environment. This limitation may also be overcome by using the same aforementioned map prediction algorithm.

7. REFERENCES

References

- [1] DARPA, “Defense advanced research project agency, subterranean (subt) challenge (archived),” Online, 2019.
- [2] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth, S. Zimmermann, H. Nguyen, M. Fehr, L. Solanka, R. Buchanan, M. Bjelonic, N. Khedekar, M. Valceschini, F. Jenelten, M. Dharmadhikari, T. Homberger, P. D. Petris, L. Wellhausen, M. Kulkarni, T. Miki, S. Hirsch, M. Montenegro, C. Papachristos, F. Tresoldi, J. Carius, G. Valsecchi, J. Lee, K. Meyer, X. Wu, J. I. Nieto, A. Smith, M. Hutter, R. Siegwart, M. W. Mueller, M. F. Fallon, and K. Alexis, “CERBERUS: autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the DARPA subterranean challenge,” *CoRR*, vol. abs/2201.07067,

2022. [Online]. Available: <https://arxiv.org/abs/2201.07067>
- [3] T. S. Vaquero, M. S. da Silva, K. Otsu, M. Kaufman, J. A. Edlund, and A.-a. Agha-mohammadi, “Traversability-aware signal coverage planning for communication node deployment in planetary cave exploration,” *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2020.
- [4] B. Morrell, K. Otsu, A. Agha, D. D. Fan, S.-K. Kim, M. F. Ginting, X. Lei, J. Edlund, S. Fakoorian, A. Bouman, F. Chavez, T. Kim, G. J. Correa, M. Saboia, A. Santamaria-Navarro, B. Lopez, B. Kim, C. Jung, M. Sobue, O. C. Peltzer, J. Ott, R. Trybula, T. Touma, M. Kaufmann, T. S. Vaquero, T. Pailevanian, M. Palieri, Y. Chang, A. Reinke, M. Anderson, F. E. T. Schöller, P. Spieler, L. M. Clark, A. Archanian, K. Chen, H. Melikyan, A. Dixit, H. Delecki, D. Pastor, B. Ridge, N. Marchal, J. Uribe, S. Dey, K. Ebadi, K. Coble, A. N. Dimopoulos, V. Thangavelu, V. S. Varadharajan, N. Palomo, A. Rosinol, A. Chatterjee, C. Kanellakis, B. Lindqvist, M. Corah, K. Strickland, R. Stonebraker, M. Milano, C. E. Denniston, S. Sahnoune, T. Claudet, S. Lee, G. Salhotra, E. Terry, R. Musuku, R. Schmid, T. Tran, A. Kourchians, J. Schachter, H. Azpurua, L. Resende, A. Kalantari, J. Nash, J. Lee, C. Patterson, J. G. Blank, K. Patath, Y. Kubo, R. Alimo, Y. Almalioglu, A. Curtis, J. Sly, T. Wells, N. T. Ho, M. Kochenderfer, G. Beltrame, G. Nikolakopoulos, D. Shim, L. Carlone, and J. Burdick, “An addendum to nebula: Toward extending team costar’s solution to larger scale environments,” *IEEE Transactions on Field Robotics*, vol. 1, pp. 476–526, 2024.
- [5] C.-L. Lu, J.-T. Huang, C.-I. Huang, Z.-Y. Liu, C.-C. Hsu, Y.-Y. Huang, S.-C. Huang, P.-K. Chang, Z. L. Ewe, P.-J. Huang, P.-L. Li, B.-H. Wang, L.-S. Yim, S.-W. Huang, M. R. Bai, and H.-C. Wang, “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.
- [6] S. Scherer, V. Agrawal, G. Best, C. Cao, K. Cujic, R. Darnley, R. DeBortoli, E. Dexheimer, B. Drozd, R. Garg, I. Higgins, J. Keller, D. Kohanbash, L. Nogueira, R. Pradhan, M. Tatum, V. K. Viswanathan, S. Willits, S. Zhao, H. Zhu, D. Abad, T. Angert, G. Armstrong, R. Boirum, A. Dongare, M. Dworman, S. Hu, J. Jaekel, R. Ji, A. Lai, Y. H. Lee, A. Luong, J. Mangelson, J. Maier, J. Picard, K. Pluckter, A. Saba, M. Saroya, E. Scheide, N. Shoemaker-Trejo, J. Spisak, J. Teza, F. Yang, A. Wilson, H. Zhang, H. Choset, M. Kaess, A. Rowe, S. Singh, J. Zhang, G. A. Hollinger, and M. Travers, “Resilient and modular subterranean exploration with a team of roving and flying robots,” *Field Robotics*, vol. 2, pp. 678–734, 2022.
- [7] 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.

- [8] M. Tatum, “Cmu-ri-tr-20-19: Communications coverage in unknown underground environments,” Master’s thesis, Carnegie Mellon University, 2020.
- [9] H. M. Ammari, “A computational geometry-based approach for planar k-coverage in wireless sensor networks,” *ACM Trans. Sen. Netw.*, vol. 19, no. 2, Feb. 2023. [Online]. Available: <https://doi.org/10.1145/3564272>
- [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*, vol. 3, 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, N. Benbernou, P. Bose, M. Damian, E. D. Demaine, V. Dujmovic, R. Flatland, F. Hurtado, J. Iacono, A. Lubiw, P. Morin, V. Sacristan, D. Souvaine, and R. Uehara, “Coverage with k-transmitters in the presence of obstacles,” *Journal of Combinatorial Optimization*, vol. 25, 01 2013.
- [13] C. Rizzo, D. Tardioli, D. Sicignano, L. Riazuelo, J. L. Villarroel, and L. Montano, “Signal-based deployment planning for robot teams in tunnel-like fading environments,” *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1381–1397, 2013.
- [14] D. Tardioli, L. Riazuelo, D. Sicignano, C. Rizzo, F. Lera, J. L. Villarroel, and L. Montano, “Ground robotics in tunnels: Keys and lessons learned after 10 years of research and experiments,” *Journal of Field Robotics*, vol. 36, no. 6, pp. 1074–1101, 2019. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21871>
- [15] M. A. Islam, M. Maiti, Q. M. Alfred, P. K. Ghosh, and J. Sanyal, “Attenuation modelling and machine learning based snr estimation for 5g indoor link,” in *IEEE VLSI Device, Circuit and System Conference (VLSI-DCS)*, 2020.
- [16] X. Wang, M. C. Gursoy, T. Erpek, and Y. E. Sagduyu, “Jamming-resilient path planning for multiple uavs via deep reinforcement learning,” in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021.
- [17] B. Woosley, P. Dasgupta, J. G. R. III, and J. Twigg, “Multi-robot information driven path planning under communication constraints,” *Autonomous Robots*, vol. 44, pp. 721–737, 2020.
- [18] A. R. da Silva, L. Chaimowicz, T. C. Silva, and M. A. Hsieh, “Communication-constrained multi-robot exploration with intermittent rendezvous,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2024.
- [19] P. Sriganesh, J. Maier, A. Johnson, B. Shirose, R. Chandrasekar, C. Noren, J. Spisak, R. Darnley, B. Vundurthy, and M. Travers, “Modular, resilient, and scalable system design approaches - lessons learned in the years after

DARPA subterranean challenge,”
in *IEEE ICRA Workshop on Field
Robotics*, 2024.

- [20] N. Bagree, C. Noren, D. Singh, M. Travers, and B. Vundurthy, “Distributed optimal control framework for high-speed convoys: Theory and hardware results,” *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 2127–2133, 2023, 22nd IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896323015197>
- [21] 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.

8. CONTACT INFORMATION

Matthew Travers
Senior Systems Scientist
Carnegie Mellon University
Robotics Institute
School of Computer Science
5000 Forbes Ave.
Pittsburgh, PA
mtravers@andrew.cmu.edu

9. ACKNOWLEDGMENT

This work was in part supported by OUSD/R&E (The Under Secretary of Defense-Research and Engineering), National Defense Education Program (NDEP) / BA-1, Basic Research.

APPENDIX

This section contains a list of important notations used in this manuscript for reference.

Definition	Notation
List of assets	\mathcal{C}
Index set for the assets	\mathcal{I}
Set of agents	\mathcal{A}
Edges in spatial graph	E_d
Edge weights in spatial graph	w_d
Spatial graph	G_d
Operator interface node in G_d	c_0
Edges in communications graph	E_c
Edge weights in communications graph	w_c
Communications graph	G_c
Mapping from asset index to physical location	p
Signal strength between two nodes $(v_i, v_j,), v_{i,j} \in V$	$w_c, \text{COMM}(v_i, v_j,)$
Operator interface node in G_c	v_0
Minimum allowable communications edge weight	C_{THRESH}
Path between two nodes $(v_i, v_j,), v_{i,j} \in V$	$\pi(v_i, v_j)$
Maximin spanning tree	T
Strongest “weakest” link for (v_i, v_j) to operator node	$\text{CMET}(v_i, v_j)$
Set of central nodes	CN
Network topology	\mathcal{N}
Communications boundary	$\mathcal{B}(\mathcal{N})$
Parent node	v_p