Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/XXXX.2022.1234567

Communication Mapping for Robot Teams

Jonathan Diller[1] (Student Member, IEEE), John G. Rogers[2] (Senior Member, IEEE), Neil T. Dantam[1], and Qi Han[1] (Senior Member, IEEE)

[1]Colorado School of Mines, Golden CO, USA
[2]DEVCOM Army Research Laboratory, Adelphi MD, USA

Corresponding author: Jonathan Diller (email: jdiller@mines.edu).

This work was supported in part by ARL DCIST W911NF-17-2-0181.

ABSTRACT Communication is a fundamental building block in almost every robotics application, acting as the thread that holds together teams and connects humans to robots deployed in the field. However, accurately predicting how well robots will be able to communicate during deployment and incorporating such information into planning algorithms is particularly challenging. In this article, we present the Link Quality Communication Map (LQCM) – a method for mapping the potential for two robots to communicate. Our communication map builds a discretized representation of the environment and uses Expected Transmission Count (ETX), a link quality metric commonly used for data routing in wireless networks, to represent the ability of two robots to communicate in the environment. We also present a method for predicting ETX for pairs of robots. This article lays out the details of building an LQCM, highlights various properties inherent to these maps, discusses how these maps can be used in a variety of robotics applications, and reviews the results and lessons learned from our own deployments of LQCMs in the field. To validate our theoretical results, we generated communication maps for a variety of environments and used them for various robotics applications including multi-robot environmental monitoring and determining regions with guaranteed communication quality to a base station. Our results show an average decrease of 18.2% in data transmission times when compared to the current community standard for representing communication in robot planning and a decrease of over 90% in extreme cases. We also ran extensive experiments on ETX datasets from our field experiments to evaluate the accuracy of our ETX prediction method and evaluated how well the method handles malicious communication jamming.

INDEX TERMS Communication Mapping, Communication-Aware Planning, Robot Networks, Robot-to-Robot Communication, Robotics

I. INTRODUCTION

Communication is required for deploying robots into the world, serving as a fundamental element in almost every field robotics application. This is particularly the case in multi-robot and robot-human collaboration scenarios, and crucial to success for many applications, including search and rescue, agriculture, mining and resource extraction, environmental monitoring and surveillance. However, accurately representing the ability for robots to communicate in planning algorithms is an open challenge in research today.

Traditional approaches for representing communication in robot planning can be divided into two main approaches: (1) the disk method where communication is assumed to be possible when within a predefined range and (2) the signal-to-noise ratio (SNR) method where algorithms attempt to maximize SNR between robots. However, decades of research in the networking community have found that the disk method is simultaneously overly conservative in prohibiting long range line of sight communications and overly optimistic in allowing short range communications through obstructions while SNR does not accurately represent the potential to transmit large volumes of data over a wireless network and cannot provide insight on multi-hop performance [1]–[3]. Over the last two decades, the networking community has developed various link quality metrics that more accurately capture wireless transmission throughput at the application level [4], [5], yet the robotics community continues to rely on limited and inaccurate approaches for incorporating communication into multi-robot planning algorithms.

Motivated by the shortcomings of methods commonly used in robotics, we pose the following research question: *What is a communication model for field robotics that accurately reflects application layer performance and how do we enable robots to build such a model?* The model should be able to inform us of how many robots are required and where to position robots to move data from robot to robot and from robot to humans. The disk method has been used to fill this need but again is not an accurate method for representing communication, particularly in cluttered environments. Furthermore, robots must be able to apply the model in areas that have not been fully explored. Many existing works in the community have attempted to do this by predicting point-to-point signal strength using machine learning. However, as previously stated, signal alone does not accurately represent the ability to communicate at the application level and these existing methods tend to require large volumes of training data. This paper provides a method for the robotics community to get on track with where the networking community has already gone.

To address this research question, we propose the Link Quality Communication Map (LQCM) - a model for representing the potential for a team of robots to communicate - and provide a method for predicting link quality in a given environment. Our LQCM uses the Expected Transmission Count (ETX) [6] link quality metric to represent the potential for robot communication. Expected Transmission Count provides application level insight on how likely a data packet is to be delivered to the receiver. Using ETX as the underlying metric gives LQCMs inherent properties, including determining where to position robots and the number of robots needed to maximize data-throughput in wireless networks formed by mobile robots. These properties make LQCMs ideal for multi-robot planning and help overcome the downfalls of previously proposed methods for representing communication in robot planning, such as making over simplifying assumptions about communication or losing application-level insight. To the best of our knowledge, we are the first to propose mapping ETX for planning large robot team deployments.

Our contributions include:

1) the Link Quality Communication Map (LQCM), a discrete representation of the environment that depicts the potential for heterogeneous robots to communicate, which we show can be used to determine the optimal robot configuration and an optimal number of robots to deploy to maximize data-throughput;
2) an aggregated regression model, termed SVRF, for predicting ETX between any two points in the environment that can be trained using sparse datasets (200 to 400 data points) and adapt to changing radio frequency (RF) environments; and
3) experimental evaluation that validates the utility of the LQCM in a variety of robotics applications and that demonstrates the accuracy of our SVRF regression model.

We review related literature on the topics of robot team deployments and current approaches to mapping communication in the following section. Notably, we make the argument for a new networking metric in robotics in Section A where we demonstrate the shortcomings of existing models. In Section III we provide relevant background on link quality in wireless networks. In Section IV we introduce the concept of an LQCM, and in Section B we describe our ETX prediction method. In Section V we discuss various properties of an LQCM and how they can be utilized in a variety of robotics applications. Section VI summarizes our field experiments and results with concluding remarks in Section VII.

II. Related Work

In this section we summarize related work on communication-aware robot team deployment, the ETX link quality metric, and mapping communication.

A. Communication-Aware Robot Team Deployment

Most works on planning communication-aware robot team deployments utilize communication models that can be divided into two main approaches. The first approach is commonly referred to as the disk method. This is an all-or-nothing approach to communication, where whenever two robots are within a predefined radius they have maximum bandwidth but are unable to communicate whenever outside of this circle. This method has been widely adopted in robot planning research [7]–[12].

The disk method relies on distance as the criteria for determining the ability to communicate. However, it is well known that data-throughput over a wireless link can vary greatly based on many factors beyond distance, including obstructing objects and the materials they are made from, multipath signals (both as a means for communication and as a source of interference), environmental noise, interference from other devices, and even the time of day [13], [14]. It has been well documented how these complex factors invalidate the disk method approach for representing communication between two wireless devices [3] and are not useful for planning communication between multiple devices [2], [15].

The second approach is the signal-to-noise ratio (SNR) method. The concept behind the SNR method is to either maximize SNR between devices [16], [17] or to set a bound on an acceptable SNR value for communicating [18]. There are several examples where SNR and its close relative Received Signal Strength (RSS) are used as a network metric for maintaining peer-to-peer communication between collaborating robots [19], [20] or for data streaming applications via multi-robot teams [17], [21], [22]. These metrics have also been found to work well for certain robotics applications not specific to data transfer, such as localization [23], [24], because their behavior can be position-specific when one of the transmitters remains static.

However, SNR fails to give us application-layer insight on how well wireless devices will be able to transmit data [1], [3]. To demonstrate this, we will review the results from our own previous work where maximizing SNR improved the packet delivery rate, but paradoxically failed to improve video streaming quality.

In [17], we deployed two robots in a network of hallways and placed a WiFi jamming device between the two robots. We fit the site-specific basic transmission loss model recommended by the International Telecommunication Union [25] with experimentally collected data and used Gaussian Process Regression for mapping noise from the WiFi jammer to predict the SNR between a relay robot and the two static robots. Using our SNR prediction method, we deployed the relay robot by maximizing the minimum SNR between any two robots and created a three-robot-long relay chain. We then measured two metrics for moving data from one end of the chain to the other: (1) the Packet-Loss-Ratio (PLR) and (2) Netflix's VMAF score for streamed video [26], which assigns a numerical value to video based on expected user experience. A smaller PLR is better while a larger VMAF score is better. Table 1 summarizes our results from using SNR as a networking metric in two different indoor environments, comparing against an agnostic approach that only attempted to maximize signal strength while ignoring noise from a signal jammer. We conducted five trials for each approach in each environment.

	Environment 1		**Environment 2**	
	Baseline	SNR	Baseline	SNR
PLR	**42.2%**	**0.1%**	59.9%	**26.9%**
Std. dev.	0.528	0.002	0.292	0.302
VMAF	**8.014**	**8.660**	0	**9.720**
Std. dev.	1.651	2.787	0	4.124

TABLE 1: Average packet loss ratio (PLR) and VMAF scores from using SNR as a networking metric, compared to an agnostic baseline method, as presented in [17].

Although utilizing SNR outperformed the agnostic approach, we still found a concerning trend in our data. Observe that in Environment 1, the VMAF scores are nearly identical yet the PLR for the agnostic method is significantly worse than the SNR method. In fact, the PLR is almost zero when maximizing the minimum SNR between any two robots in the chain, suggesting that our method for predicting SNR performed well, but this has marginal impact on data streaming. Furthermore, the SNR method found a worse PLR in Environment 2 compared to the near zero PLR found in Environment 1, but the resulting VMAF score was better in Environment 2 than it was in Environment 1. This is because SNR – and other metrics such as PLR – do not give us application-level insight on data transmission performance among multiple robots.

The networking community has since derived a variety of link quality metrics that better represent transmission throughput in wireless networks at the application level [4], [5] but the robotics community continues to use these dated and limited traditional approaches for handling communication in planning. Instead of using the agnostic disk method or an SNR approach with limited insights, we propose using the networking metric known as ETX [6] to represent the potential to communicate in multi-robot systems. The details of this metric are further discussed in Section III.

B. Predicting & Mapping Communication for Robotics

To utilize ETX in robot planning, we need a method for creating a map of ETX. Although we are focused on mapping communication, we also consider predicting communication in this section because both problems involve relatively similar communication prediction methods with the former storing the predictions in some type of data structure. We will use the two terms interchangeably here.

There are several works that address the mapping of static networks [27]–[29]. On a related topic, there also is extensive work on estimating wireless link-quality in wireless networks [30]–[32]. Both bodies of work are mainly focused on improving packet routing between static nodes and do not consider how to create a map of the potential to communicate for mobile wireless nodes (robots).

Among works that focus on mapping communication for robots, almost all existing research has focused on predicting and mapping RSS or SNR, though works that state SNR tend to predict RSS and assume RF noise is constant. We would like to note that some of the following works state that they are dealing with Received Signal Strength Indicator (RSSI) – an arbitrary and vendor specific non-negative integer to describe RSS for varying purposes on network interface controllers [33]. For clarity, RSSI is not the same as RSS and works that claim to be measuring RSSI present negative, real values in dBm (i.e. RSS measurements and not RSSI). To the best of our knowledge, no existing work predicts or maps RSSI and those that claim to do so are in fact dealing with RSS.

Works predicting and mapping RSS can be divided into two parts: (1) predicting signal strength for robot-to-fixed infrastructure communication and (2) predicting signal strength for robot-to-robot communication. Examples of predicting signal strength for robot-to-fixed infrastructure include [34], [35] where they use a distance-based method or [36], [37] where they use ray-tracing techniques in addition to distance to further refine predictions. These methods usually rely on propagation-loss models [25] and experimentally determine model parameters. Machine learning has been proposed as an alternative to using propagation-loss models, as seen in [17], [38], [39] where they use Gaussian Process (GP) regression or in [40], [41] where they train Convolutional Neural Networks. The authors in [42], [43] propose an online approach to predicting signal strength that uses filtering, though is limited to predicting signal along a robot's trajectory and cannot predict signal in areas further away. Applications involving robot-to-fixed infrastructure include robot localization [39], path planning for data collection [12], [37], supporting disaster response operations and exploration [34], [35], or evading malicious communication jamming [17].

Predicting and mapping robot-to-robot communication has also been well studied with a sizeable body of literature. The authors in [44], [45] use a distance-based model that includes obstructions while in [46] they break the environment up into regions and propose directly measuring region-to-region RSS. However, the most common approach for predicting robot-to-robot communication is to use learning-based methods, namely GP. The most common way to use a GP for robot-to-robot communication is to use pair-wise robot position (i.e. $\mathbb{R}^4$ in two-dimensional space and $\mathbb{R}^6$ for three-dimensional space) as the input to the model [47]–[50]. Although these methods have been shown to work well for predicting RSS in simplistic environments, we argue that using position alone will perform poorly in more complex environments with changing RF noise and showcases where this approach fails in our field experiments. The authors in [51] proposed using the ratio of occupied versus free cells in an occupancy grid as their model input in addition to position. In [20], the authors propose incorporating a path-loss model (such as those proposed in [25]) into the mean function prior of the GP. Breaking from GP, the authors in [52] propose using a three layer Neural Network (NN) for predicting signal strength in underground environments. Although GP and NN are powerful learning tools, we should expect them to require more data to train on as the number of inputs increase, making them challenging to use when given limited data samples. Furthermore, none of these methods address heterogenous radio types. If each robot has a unique radio, then how well two robots will be able to communicate will not only depend on the location of each robot but also the radio type of each robot. To address these shortcomings, we present a regression model for predicting ETX that considers additional model inputs beyond distance and we show empirically that our model can be trained using smaller datasets.

As previously discussed, signal is not a good indicator of application-level performance. To the best of our knowledge, data transfer rate is the only other metric that has been mapped for robot-to-robot communication. The authors in [53] propose using the Minstrel algorithm – a rate control algorithm for mac80211 in the Linux kernel [54] – to predict the data transfer rate between a mobile robot and a static wireless device from direct measurements. However, this work does not address robot-to-robot communication, it requires taking point-to-point measurements for all region pairs, and it is not clear how data transfer rate alone can be used to plan multi-hop communication over large robot teams.

Among works that store communication information (i.e. construct a communication map) and use it for robot-centric planning problems, the most common method for representing this information is as a graph. The authors in [7], [48], [49] form a graph where physical locations are represented as vertices and non-directional edges are added to the graph when they predict that two robots will be able to communicate when positioned at the location pair represented by the vertices. In [45], they form a fully connected graph with edge weights based on predicted RSS. However, these works assume that communication quality is bidirectional. That is, they assume the rate that robot a at position z_i can send data to robot b at position z_j is equal to the rate that robot b can send data back to robot a. This assumption does not always hold true, especially when one of the robots is in an area with excessive RF noise. In this work, we propose a communication map in the form of a directed graph with multiple layers, where robot-to-robot communication is not assumed to be bidirectional and different robot radio types are represented through new layers to the graph.

III. Background on Cumulative Link Quality Metrics

A cumulative link quality metric is a metric common in wireless networks where a numerical value is assigned to point-to-point links to describe the cost of transmitting data over that link. For multi-hop routes through wireless networks, the cost of the route is the sum of costs of all links in the route [5]. A cumulative link quality metric termed ETX was proposed in [6] and shown to outperform the traditional hop-count approach for data packet routing. ETX is a real number from 1 to ∞ and is defined as the expected number of transmissions required to send a data packet over a network link. Mathematically,

$$ETX = \frac{1}{p_f p_r} \tag{1}$$

where p_f is the probability that the data packet is delivered to the receiving node and p_r is the probability that an acknowledgement packet is returned to the sending node. In multi-hop wireless networks, the ETX of a data packet route is the sum of ETX for each link in the route.

Expected Transmission Count is commonly used in multi-hop routing [4] and many variations of ETX have been proposed for specific wireless protocols [5]. We use ETX as our metric for this study because it is one of the fundamental building blocks for most other proposed cumulative link quality metrics and we argue that these more advanced methods can be substituted into our LQCM formulation and used in the same manner as presented here.

Furthermore, we chose ETX as our metric for mapping robot communication because it allows us to make theoretical assertions on networking performance. From [6] we get:

Assumption 1. *The highest data-throughput route through a wireless network is the route with the smallest ETX.*

Many experimental works have shown that this assumption holds true in most cases [6], [55], [56]. If we can predict ETX between arbitrary points in the environment, then *Assumption* 1 implies that we can determine robot configurations that optimize network performance. Neither RSS nor SNR can provide a guarantee on performance and it is not clear how such metrics could be used to determine an optimal multi-hop data-throughput route in wireless networks.

Although we focus on IEEE 802.11 (WiFi) in this article, ETX can be utilized to characterize peer-to-peer communication for other RF protocols, such as LoRaWAN (Long Range Wide Area Network) [57], [58] – a communication protocol that operates in the 863-928 MHz range [59]. Beyond RF, ETX has been proposed and investigated in underwater acoustic communication networks [60], [61] and in optical communication [62]. Although not investigated in this work, our proposed LQCM should also be applicable for robot-to-robot communication when using acoustic or optical communication technology.

IV. Link Quality Communication Map

In this section we formally define a Link Quality Communication Map (LQCM) and propose a method for building an LQCM using aggregated regression models.

A. Link Quality Communication Map

Conceptually, an LQCM is a representation of the potential for robots to communicate in a given environment. In practice, an LQCM is a directed, weighted, fully-connected graph. The vertices of the graph represent physical locations in the deployment environment and static, wireless infrastructure. Edge weights of the graph are the ETX value (either measured or predicted) between two robots if they were each located at the respective physical locations represented by the vertices on either end of the edge. Figure 1 shows a toy-example of building an LQCM for a simple environment.

Given continuous space $\mathcal{X}$ (Fig. 1a), we build an LQCM by discretizing $\mathcal{X}$ into a set of cells Z where each cell is a unique robot-accessible location in $\mathcal{X}$ (as depicted in Fig. 1b). The granularity of the cells is up to the user and can be based on the particular use case. Robots with identical wireless radio equipment (e.g. WiFi cards and antennas) can be assumed to be homogeneous in the LQCM. However, robots with different wireless radios will have varying ETX values between them when compared to a homogeneous radio configuration. Because of this fact, we add layers to the LQCM for each unique radio setup. Suppose we are given a team of robots with K unique radio setups. For each $z_i \in Z$, create K vertices $v_i^1, v_i^2, \cdots v_i^K$ where each vertex v_i^j represents a possible location for a robot with radio setup type j at physical location z_i (as depicted in Fig. 1c and 1d). Additionally, any static, wireless-enabled infrastructure, such as base stations, will have different ETX values with each unique robot radio setup. Let S be the set of all wireless infrastructure where each $s_i \in S$ represents the location in $\mathcal{X}$ where wireless device i is located.

Mathematically, an LQCM is the digraph $G = \{V,\ A,\ q\}$ defined as:

- $V = \bigcup_{j=1}^{K} V^j \cup S$,
- $A = V \times V$, and
- $q(v_i, v_j)$ is the ETX weight of arc $(v_i, v_j) \in A$.

where each $V^j = \{v_1^j, v_2^j, \cdots v_n^j\}$ is the set of deployment vertices for robot radio setup j. In Fig. 1d, V^1 is the top set of blue vertices that represent radio type 1, V^2 is the lower set of vertices, and S contains the single orange vertex that represents the base station in the environment (depicted as a laptop).

For simplicity, we disregard occupied cells and cells that are inaccessible by a robot in $\mathcal{X}$. We assume that the ETX between two robots in a pair of cells is the same regardless of where the robots are positioned within each cell.

Observe that the size of the map depends on the size of the environment, the granularity of the cells, the number of robot radio types, and the number of static devices. Suppose that the number of cells required for a given environment is ϕ. For each K unique radio setup, we will add an additional ϕ vertices to G while each static device in S will add a single node to the graph. This means that the number of vertices in G will be $\phi K + |S|$. If the area of each cell remains fixed, then doubling the size of the environment will double the size of ϕ. Alternatively, if we fix ϕ and allow the granularity (i.e. the area of each cell) to change, then doubling the size of the environment will not impact the size of the graph. We note that the number of robots does not impact the size of G, only the number of unique robot radio setups (K).

Determining the granularity of the discretization of $\mathcal{X}$ depends on both the environment and the application for which the LQCM will be applied. More feature-dense environments will likely require a finer cell dimension. For example, an LQCM for Clearpath Jackal robots operating in a cluttered office space could consist of cells that are one-half square meters (roughly the size of a Jackal robot). Alternatively, large environments and applications where the robots communicate over long ranges can relax the size of each cell. For example, Clearpath warthog robots operating in large, open fields for an agricultural monitoring application could use cells with an area of 25 square meters. Furthermore, we use square cells in this work for convenience, but the cells could be any cellular decomposition of the environment [63].

We chose to use a discretized representation of the environment because it lends itself well for planning the actions of multi-robot teams. Examples of this are seen in [7], [48], [49], where communication is represented as a discrete

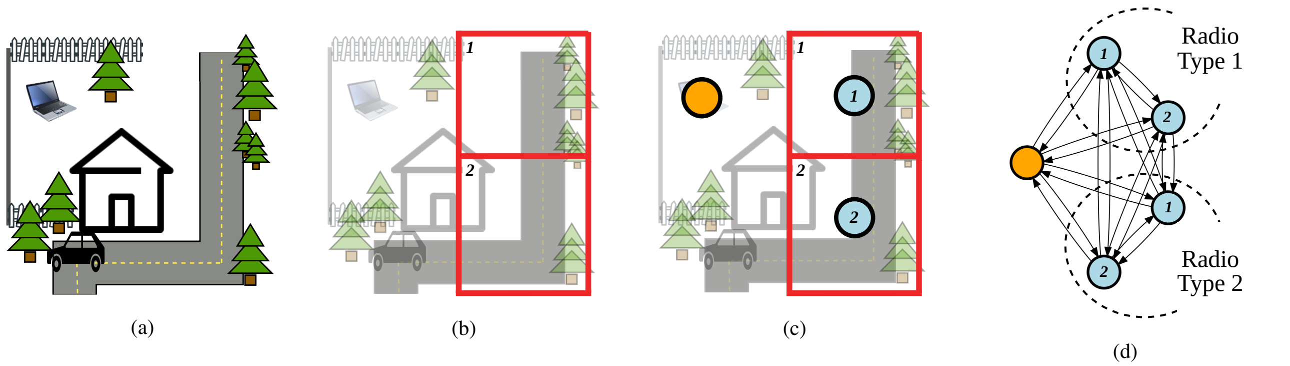


FIGURE 1: A simple example of a given environment $\mathcal{X}$ (1a), a discretization of $\mathcal{X}$ into Z (1b), generating vertices from cells in Z (1c), and a complete LQCM with a layer of vertices for each unique robot radio setup (1d).

graph to facilitate planning and coordinating robot actions. However, one possible way to build an LQCM in continuous space would be to remain in $\mathcal{X}$. Each robot would be represented as a discrete point in $\mathcal{X}$ and we would build a fully connected graph where each robot and static device would be vertices in the graph and edge weights would be defined as the ETX between each robot. A similar setup is seen in [45]. We further discuss advantages and drawbacks to a continuous-space representation of an LQCM in Section B.

A limitation to using an LQCM is that it requires an accurate function $q(v_i, v_j)$. Directly measuring ETX for all possible arcs in A would require $n(n-1)$ measurements at $\frac{n(n-1)}{2}$ unique position pairs, where $n = |V|$. Alternatively, one could estimate $q(v_i, v_j)$ by predicting the ETX between points z_i and z_j, as discussed below.

B. Predicting ETX Between Robots

To build our LQCM, we need a method for predicting ETX between any two points in $\mathcal{X}$. In general, our prediction model extracts features from an occupancy grid and uses them as inputs for aggregated regression models.

In Section III, ETX is defined mathematically as the inverse of the product of p_f and p_r. Observe that ETX is a rational function that grows to ∞ as p_f and p_r decrease. Rather than predicting ETX, we propose predicting $p_d = p_f p_r$, the probability that the data packet is received and acknowledged, which is defined in the range $[0, 1]$.

To predict p_d, we extract various features from occupancy grids to use as inputs to a regression model. Given two cells z_i and z_j in Z, let d_{ij} be the straight line distance between z_i and z_j. In most environments, the straight line between two random points will be obstructed by obstacles. Let o_{ij} be the number of occupied cells and u_{ij} be the number of unknown cells in an occupancy grid between z_i and z_j. By unknown cells, we mean cells that are neither known to be occupied nor known to be empty (often represented as '-1' in the grid). Both o_{ij} and u_{ij} can be found using Bresenham's line algorithm [64]. Let N_i and N_j be the RF noise at positions z_i and z_j, which can be measured directly or predicted using methods such as the one seen in [17], where Gaussian Process Regression is used to create a map of noise. Note that interference and RF chatter from other devices is included in both N_i and N_j. Let $\mathcal{G}_i$ be the radio gain of the robot type that we plan to deploy at z_i.

In many cases the obstacles between z_i and z_j will prevent RF signals from traveling directly from z_i to z_j. In such cases, if there still exists a connection between two robots located at z_i and z_j it is because the RF signal is following a multi-path route that goes around the obstacles between z_i and z_j. Let d^r_{ij} be the distance of the freespace path between z_i and z_j found using A* search from point z_i to z_j in the occupancy grid. To determine the likelihood that d^r_{ij} is more representative to the distance that a signal must travel from sender to receiver, we also consider the ratio between the straight-line and the freespace path distance, $\delta_{ij} = \frac{d^r_{ij}}{d_{ij}}$. The closer δ_{ij} gets to 1, the more likely it is that the two robots have line-of-sight with one another. Conversely, a larger value for δ_{ij} suggests that there is a large object between the two robots.

We define our prediction model's training set as $\{(\boldsymbol{x}_1, \boldsymbol{y}_1) \quad \cdots \quad (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, where $\boldsymbol{x}_l = [d_{ij}, o_{ij}, u_{ij}, \mathcal{G}_i, \mathcal{G}_j, N_i, N_j, d^r_{ij}, \delta_{ij}]^T$ is an input to our model and $\boldsymbol{y}_i = p_d$ is the desired model output.

We form our prediction model by combining a Support Vector Regression (SVR) model [65] and a Random Forest Regression model [66]. We use a Radial Basis kernel function [67] for the SVR. We run each model separately then average together the output of each model to get a final prediction. We chose a Random Forest because parts of our problem appear to be discrete decisions (e.g. does $o_{ij} = 0$?). However, we found that the Random Forest was prone to over fitting our data so we selected an SVR to help generalize our model.

We term this the Aggregated SVR+Random Forest (SVRF) model. Given a prior dataset, we can train our SVRF model to make p_d predictions for location pairs that we have not previously taken ETX measurements in. These predictions allow us to fill in weight values $q(v_i, v_j)$ for the LQCM digraph G.

V. Applying Link Quality Communication Maps

In this section we look at various properties of LQCM and discuss how to find paths in the maps. We then summarize how LQCMs can be applied in various robotics applications.

A. Properties of Link Quality Communication Maps

In wireless networks, the ETX of a data packet route is the sum of ETX for each link in the route, as discussed in Section III. In the LQCM graph G, the ETX of a path through G is the sum of weights of the links in the path.

Suppose we would like to transmit data from z_i to z_j and need to determine where to place robots to transmit the data. Let the ordered sequence of vertices W_{ij} be the least ETX path in G from a vertex representing z_i to a vertex representing z_j. Building on *Assumption* 1, we get:

Proposition 1. *The robot configuration (i.e., where to place each robot) that maximizes data-throughput from a source device at z_i to a sink device at z_j is at the vertices in W_{ij}.*

Proof:
By contradiction, assume that there exists a robot configuration $W_{ij}^{'} \neq W_{ij}$ that achieves higher data-throughput than the vertices in W_{ij}. From Assumption 1, $W_{ij}^{'}$ must be the least ETX path from z_i to z_j, but this contradicts the definition of W_{ij}. $\square$

Furthermore, we can also determine how many robots are required to transmit data from z_i to z_j. We define $|W_{ij}|_r$ as the number of vertices in W_{ij} that represent physical locations. That is, $|W_{ij}|_r = |W_{ij} \cap (\bigcup_{j=1}^{k} V^j)|$. With $|W_{ij}|_r$, we get:

Proposition 2. *The optimal number of robots to deploy to maximize data-throughput from a source device at z_i to a sink device at z_j is $l = |W_{ij}|_r$.*

Proof:
By contradiction, assume that the optimal number of robots to deploy to maximize data-throughput is $l^{'} \neq l$. From Assumption 1, the least ETX path from z_i to z_j must therefore have $l^{'}$ vertices. However, this contradicts the definition of W_{ij}. $\square$

In other words, if we find W_{ij} in G, then the vertices in W_{ij} will represent the physical locations to position robots, with specific radio types, to minimize ETX from a source device at z_i to a sink device at z_j. Because putting robots at the locations in W_{ij} will minimize the ETX from the source device to the sink device, then the cardinality of W_{ij} tells us the exact number of robots needed to minimize ETX. *Assumption* 1 tells us that the highest data-throughput route through a wireless network is the route with the smallest ETX. This means that finding a robot configuration that minimizes ETX from the source device to the sink device will maximize data-throughput from the source to the sink. Determining this robot configuration simply requires us to find W_{ij} in G.

As an example of *Propositions* 1 and 2, consider the environment shown in Figure 1. We have two robots, a and b, where robot a has radio type 1 and robot b has radio type 2, and we want to send a robot to z_2 to stream data back to the base station. Suppose that the least cost path from the orange node to a vertex representing z_2 in the graph in Fig. 1d is $W_{ij} = \{s_1, v_1^2, v_2^1\}$. This tells us that we should place robot b (with radio type 2) at z_1 and robot a (with radio type 1) at z_2 to maximize data-throughput from z_2 back to the base station. *Propositions* 1 and 2 can be restated to include wireless infrastructure in S by finding $W_{ij'}$ from a vertex representing z_i to infrastructure vertex $s_{j'}$, or in reverse by finding $W_{j'i}$.

Note that the optimal number of robots to transmit data from z_i to z_j is not the same as the minimum number of robots to transmit data from z_i to z_j. As shown in our field experiments in Section VI, using more robots can sometimes improve data transmission performance compared to using fewer robots. Furthermore, communication quality can change based on a variety of factors, as discussed in Section A, including the time of day. This means that *Propositions* 1 and 2 depend on an accurate prediction of ETX.

Propositions 1 and 2 assume that there are enough robots of the correct type for W_{ij}. However, in practice we will not always have enough robots for W_{ij}. Let R_k be the set of robots with radio setup k and $m_k = |R_k|$. We assume that $R_k \cap R_k = \emptyset$. Let $|W_{ij}|_r^k$ be the number of vertices in W_{ij} that represent robot radio setups of type k. That is, $|W_{ij}|_r^k = |W_{ij} \cap V^k|$. Let $\tilde{W}_{ij}$ be the least cost path from a vertex representing z_i to a vertex representing z_j in G where the following constraint holds:

$$|\tilde{W}_{ij}|_r^k \leq m_k, \ k \in \{1, 2, \cdots K\}. \tag{2}$$

Given $\tilde{W}_{ij}$, we get the following corollaries to *Propositions* 1 and 2:

Corollary 1. *Given robot team $R_1 \cup R_2 \cup \cdots R_k$, the robot configuration that maximizes data-throughput from a source device at z_i to a sink device at z_j is at the vertices in $\tilde{W}_{ij}$.*

Corollary 2. *Given robot team $R_1 \cup R_2 \cup \cdots R_k$, the optimal number of robots to deploy to maximize data-throughput from a source device at z_i to a sink device at z_j is $|\tilde{W}_{ij}|_r$.*

The proofs for *Corollaries* 1 and 2 are identical to the proofs for *Propositions* 1 and 2. Note that these corollaries are generalized to include wireless infrastructure. That is, $\tilde{W}_{ij}$ may include existing wireless infrastructure, along with multiple robots, as intermediate relay nodes to maximize data-throughput.

B. Finding Minimum Cost Paths in an LQCM

Propositions 1 and 2 require us to find minimum cost path W_{ij} through G with no restrictions on the length of the path or which nodes are included in the path. This can be done using Dijkstra's algorithm.

We acknowledge that the A* search algorithm is more efficient than Dijkstra's algorithm for least-cost path problems when there exists an admissible heuristics function $h(v_i, v_j)$ (a function that guarantees a lower bound on the cost from vertex v_i to vertex v_j). However, it is challenging to define an admissible $h(v_i, v_j)$ for G that is meaningful due to the stochastic nature of wireless communication. Notably, $q(v_i, v_j)$ will not be an admissible heuristic because there may exist an alternative path through G from v_i to v_j that is less than $q(v_i, v_j)$. For example, suppose that there is a large metal object between the cells z_i and z_j that are represented by v_i and v_j, respectively, in G and $q(v_i, v_j) = \infty$, but a third cell $z_{i'}$ has line-of-sight with both z_i and z_j and $q(v_i, v_{i'}) + q(v_{i'}, v_j) = 3$. Another possible heuristics function would be to predict ETX between all pairs of cells in Z based on distance while assuming all cells have line-of-sight. However, this strategy again does not guarantee admissibility because ETX with line-of-sight at a given distance will change depending on the environment. This is seen in our field experiments in Section VI where there is a notable difference in ETX between robots and the base station in an outdoors environment compared to an indoors environment. At the time of writing, the only guaranteed lower bound on ETX is a value of 1 (i.e. assuming there exists a perfect connection between z_i and z_j).

Corollaries 1 and 2 require us to find $\tilde{W}_{ij}$, a minimum cost path through G that adheres to the limited number of robots of each radio type. If $K = 1$ (i.e. all robots have the same radio setup) then $\tilde{W}_{ij}$ can be found using a variation of weighted breadth-first-search that limits the depth of the search to m_1 (the total number of robots available). However, if $K > 1$ (i.e. we are giving multiple radio types) then Constraint 2 makes it more challenging to find $\tilde{W}_{ij}$.

Theorem 1. *Finding $\tilde{W}_{ij}$ in G when $K \geq 2$ is at least as hard as NP-Complete problems.*

Before proving *Theorem* 1, we would like to remind the reader of the 0-1 knapsack problem: Given a set I of n items, each with weight w_i and value p_i, form a subset of items I^s such that $\sum_{i \in I^s} w_i \leq W$ and $\sum_{i \in I^s} p_i \geq P$, for some given weight limit W and desired value P. The 0-1 knapsack problem is known to be NP-Complete [68]. The idea to our proof is to create a sequence of widgets that represent items in the 0-1 knapsack problem. We impose a "total cost" for traversing the widget sequence and give a cost discount for selecting items (activating widgets – as defined below) but we limit the number of widgets that can be activated using W. Figure 2 shows the basic concept of the widget sequence used in our proof.

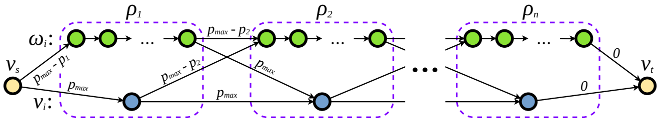


FIGURE 2: Graph G_k constructed from the 0-1 knapsack problem. Vertices grouped in purple represent a single widget, where green vertices are ω_i and blue vertices are v_i.

Proof:
We reduce the 0-1 knapsack problem into finding $\tilde{W}_{ij}$ in a special graph G_k as follows: For each item i, create a widget $\rho_i = \{\omega_i, v_i\}$, where ω_i is w_i sub-vertices in a directed tree with no branches and edge weights of 0 and v_i is a singular vertex that is not connected to ω_i. Form G_k by creating a start vertex v_s, an end vertex v_t, and then lining up all widgets in an ordered row between v_s and v_t. Add directed edges from v_s to the first ρ_1, where the edge weight going into v_1 is p_{max} and the edge weight going into the root of ω_1 is $p_{max} - p_1$, where $p_{max} = max(p_1, p_2, \ldots, p_n)$. For each ρ_i and ρ_{i+1}, add directed edges from ω_i and v_i to both ω_{i+1} and v_{i+1} with weights as described for edges going into ρ_1. The last pair ρ_n is then connected to v_t with a weight of 0. Designate all ω_i as robot type 1 and accompanying vertices v_i as robot type 2. With $m_1 = W$ and $m_2 = n$, find $\tilde{W}_{ij}$ in G_k. We consider widget i to be "activated" if $\tilde{W}_{ij}$ includes ω_i.

The activated widgets in $\tilde{W}_{ij}$ represent the items to include in I^s for the 0-1 knapsack problem. If $np_{max} - cost(\tilde{W}_{ij}) \geq P$, then there exists a subset (I^s) that meets the requirements of the given 0-1 knapsack problem. If $np_{max} - cost(\tilde{W}_{ij}) \ngeq P$, then no subset of I exists that meets the requirements of the given 0-1 knapsack problem because $\tilde{W}_{ij}$ will be a minimum cost path from v_s to v_t.

We have found a polynomial time reduction from the 0-1 knapsack problem to finding $\tilde{W}_{ij}$ in G when $K \geq 2$. Therefore, finding $\tilde{W}_{ij}$ in G when $K \geq 2$ is at least as hard as any NP-Complete problem. ■

We note that the reduction to the knapsack problem is helpful to show the hardness of finding $\tilde{W}_{ij}$ in G while a more intuitive approach to finding $\tilde{W}_{ij}$ would be to search for constrained shortest paths (CSP) in G rather than modeling our problem as an instance of the knapsack problem. Finding CSPs in a graph is also known to be NP-hard [69], with well documented exact and approximate solutions [70]. In practice, we recommend attempting to find W_{ij} in G using Dijkstra's algorithm first. That is, attempt to find a robot configuration without restricting the number of robot radio types. If the found path breaks constraint (2) (i.e. the path requires more robots with a certain radio type than what are available), then an alternative method must be used to find a CSP in G. In our field experiments in Section VI, we found that using Dijkstra's algorithm is usually sufficient for finding $\tilde{W}_{ij}$.

Given the complexity of finding $\tilde{W}_{ij}$, it could be advantageous to remove vertices from G that are unlikely to be used. For any location $z_i \in Z$, it is unlikely that more than one vertex in G that represents z_i will be in $\tilde{W}_{ij}$. However, in the general case, removing vertices will break optimality guarantees from *Corollaries* 1 and 2. A possible case where one could remove cells is when robots with a certain radio type are not able to access an area of the environment that other robots with a different radio type are able to access. The layer that represents the restricted robot class could be trimmed down based on the accessibility of those robots without losing optimality guarantees.

In Section A we briefly described one possible representation of an LQCM in continuous space, where each robot and static device would form vertices in a fully connected graph. The advantage to this approach is that we would move away from the accuracy limitations of discretizing our deployment space. The disadvantage of using continuous space is that determining where to deploy robots and how many robots to deploy to maximize data-throughput (that is, achieving *Corollaries* 1 and 2) becomes more complicated. To keep the guarantees of *Corollaries* 1 and 2 in continuous space, we would need to define a function, $f(z_i, z_j) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, that takes the positions $z_i, z_j \in \mathcal{X}$ of robots i and j and determines the ETX of transmitting from robot i to robot j. To determine an optimal robot configuration when moving data from location z_i to location z_j would require us to minimize $f(z_i, z_1) + f(z_1, z_2) + \cdots f(z_n, z_j)$ by selecting the location of $z_1, z_2, \cdots z_n$, determining a value for n, and determining which robot should be placed at each position z. This is a new combinatorial problem that is further complicated when considering physical obstacles. The main advantage of our discretized space representation for the LQCM is that it simplifies this combinatorial problem and in most cases finding $\tilde{W}_{ij}$ in an LQCM can be done using Dijkstra's algorithm, as shown in Section VI.

C. Applications of Link Quality Communication Maps

We argue that LQCMs can be applied to any multi-robot scenario where the robots rely on wireless communication between one another and between robots and human operators. In this section we highlight how an LQCM could be applied to a variety of common multi-robot applications where high-level planning must account for wireless communication.

The maps can be used to determine how to place robots for data relaying and streaming applications such as [17], [71]–[74]. From *Propositions* 1 and 2, we can determine the optimal number of robots to deploy and where to place each robot for streaming applications by finding $\tilde{W}_{ij}$.

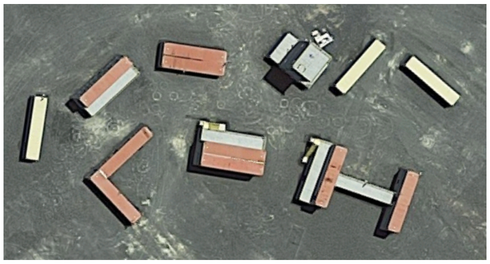
The map can also be used to determine regions of the environment that a team of robots can explore while guaranteeing a minimum wireless connection to a base station, similar to the problems addressed in [18], [53]. Suppose we have determined that an acceptable connection to the base station is an ETX of 3. Any cell in Z with a vertex in G that has a least cost path of 3 or better to the base station is a cell that can be explored while guaranteeing an acceptable wireless connection to the base station.

Link Quality Communication Maps can also be used when planning data muling problems where teams of robots are used to collect data from sensors, as seen in [7], [10], [17]. The algorithms discussed in these works could be further augmented by adding an LQCM and determining data collection and relaying configurations that reduce the time required to deliver data back to a base station.

VI. Evaluation in the Field

To evaluate the utility of our LQCM and our proposed ETX prediction method, we ran field experiments in three outdoor environments (*Mock-Town*, *Courtyard*, and *Camp* shown in Fig. 3) and one indoor environment (*Office*). The *Mock-Town* is a robotics experimental arena set up to replicate an urban environment [75]. The *Courtyard* is located on the Colorado School of Mines campus and is an open space with small concrete barriers situated between academic buildings. The *Camp* is a group of camp lodgings in a heavily wooded area located in West Point, New York. The *Office* is a long hallway that connects office spaces and class rooms and also located on the Colorado School of Mines campus. In the *Mock-Town* and *Camp* we used a team of three Clearpath Jackal robots and then added a Raspberry Pi 3 as a base station for the *Courtyard* and the *Office*. Figure 4c shows two of the robots streaming data in the *Courtyard* and Fig. 5c depicts two more Jackal robots in the *Camp* environment. Each robot was equipped with an USRP B205mini-i software-defined radio (SDR) that functions as a spectrum analyzer to measure RF noise. We lowered the gain of the TX power on the base station to reduce the distance that a robot needs to travel to break connection with the base station. The Raspberry Pi 3 had its TX power reduced to 2 dBm for the *Courtyard* and 1 dBm for the *Office*. The further reduction in the *Office* was because the Raspberry Pi was able to communicate noticeably further indoors. At this setting, the robots would lose communication with the base station at roughly 40 *m* in line-of-sight conditions. In the *Camp* environment we also deployed another USRP SDR to broadcast RF noise, similar to the Wi-Fi jammer setup described in [17]. We used OmniMapper [76] on a single Jackal robot to create occupancy grids for each environment.

We created two networks for our experiments: a test network and an administrative network. The test network consisted of the Jackal on-board Intel WiFi cards provided by Clearpath and the WiFi peripheral built into the Raspberry Pi. The test network was used to take ETX measurements and stream data. The potential to communicate over this network is what we mapped out in an LQCM for each scenario. The administrative network consisted of a powerful UniFi access point, a commercial WiFi router, and an additional Waveshare WiFi card on each Jackal robot. We used the



(a) *Mock-Town* Environment

(b) *Courtyard* Environment

(c) *Camp* Environment

FIGURE 3: Locations of outdoor field experiments.

administrative network to control the robots and coordinate their actions. The motivation for two separated networks is to move the robots beyond the limits of their communication range (introducing communication failure) over the test network while still allowing us to safely control the robots on the stronger administrative network.

To measure ETX between two robots, or between a robot and the Raspberry Pi, we had the sending node transmit 1,000 data packets. The receiving node would then send 1,000 acknowledgement packets back to the transmitter, regardless of how many packets successfully arrived at the receiver. We then counted the total number of data and acknowledgement packets received and used these numbers to calculate p_f and p_r, respectively, by dividing the number of packets received by 1,000. The data packets were each 1,400 bytes large and the acknowledgement packets were 2 bytes. We used User Datagram Protocol (UDP) to transmit packets for ETX data collection because we wanted to count the number of packets that successfully transmitted over connectionless sockets without any handshake dialogue.

In each environment, we used two or three robots to take point-to-point ETX measurements over the test network and (when applicable) had each robot measure ETX with the Raspberry Pi base station. We manually drove the robots into different locations in each environment, seeking to find configurations where we got representative samples of both extremes for each model input discussed in Section B. We took ∼200 measurements in the *Mock-Town*, ∼290 in the *Courtyard*, ∼400 in the *Camp* (300 with the Wi-Fi jammer and 100 without) and ∼260 in the *Office*.

A. ETX Prediction Accuracy

To evaluate our proposed ETX prediction method, we trained our proposed Aggregated SVR+Random Forest (SVRF) model on the ETX measurements collected from all three environments. As discussed in Section B, we predict p_d, the probability that a data packet is successfully delivered and acknowledged, as oppose to ETX directly because ETX is defined on the interval $[1, \infty)$ where as $p_d \in [0, 1]$.

We compare our SVRF model against an adaptation of the ETX model found in [20], where they suggest that p_d can be derived from a signal prediction. We predict signal between two robots using a similar method to [17] by line fitting data to the signal-to-distance function $f_{dB}(d) = -10nlog_{10}(d) + c$, proposed in [14], [25], where n and c are constants that fit $f_{dB}(d)$ to our data. We then predict p_d from bidirectional SNR, assuming a linear mapping between the two. We fit $f_{dB}(d)$ and our SNR-to-p_d mapping using our experimental data and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [77], [78]. We term this approach the Signal-Model (sigMod). We also constructed a Bayesian Ridge Regression (BR) model [79] and a Gaussian Process Regression model (GP) with a radial basis function kernel [80], both using the same inputs discussed in Section B. For a fourth baseline approach, we trained a second GP model that only uses location-pairs as model inputs (GP-pos). We chose the GP methods because using a GP model was proposed in other works to map RSS between robots using position [47]–[50] and position with obstruction [51], and for mapping RF jamming signals [17]. To compare against an NN approach, we implemented the simple NN proposed in [52] for predicting signal strength in a mine. The NN takes the same input as our SVRF model and has a single, fully connected 16 unit hidden layer. Additionally, we trained a Deep NN (DNN) that has four fully connected hidden layers (a 16 unit layer, two 32 unit layers, and another 16 unit layer), all using a restricted linear activation function. We also compare our model against the individual SVR and Random Forest (R-F) models without averaging together the prediction, as discussed in Section B. We used TensorFlow [81] to build both the NN and the DNN models and used scikit-learn [82] to build the other models.

1) Model Accuracy

To assess the accuracy of the various ETX prediction models, we trained all approaches 50 times on randomized training-test set splits and averaged together the Mean Squared Error (MSE) and R-Squared score (R2). A smaller MSE score is better while a higher R2 score is better. The MSE metric tells us how accurately each approach can predict p_d (the probability that a data packet is successfully delivered and acknowledged), penalizing predictions that are further away

	Mock-Town				*Courtyard*			
	Training Set		Test Set		Training Set		Test Set	
	MSE	R2	MSE	R2	MSE	R2	MSE	R2
sigMod	$4.47{\cdot}10^{-2}$	0.083	$4.71{\cdot}10^{-2}$	0.022	$3.55{\cdot}10^{-2}$	0.106	$3.51{\cdot}10^{-2}$	0.089
GP	$9.17{\cdot}10^{-3}$	0.812	$3.60{\cdot}10^{-2}$	0.242	$1.28{\cdot}10^{-2}$	0.678	$1.66{\cdot}10^{-2}$	0.556
GP-pos	$9.55{\cdot}10^{-3}$	0.804	$3.55{\cdot}10^{-2}$	0.254	$1.18{\cdot}10^{-2}$	0.704	$\mathbf{1.50{\cdot}10^{-2}}$	**0.600**
NN	0.200	-3.13	0.214	-3.37	0.347	-7.83	0.390	-9.39
DNN	$2.06{\cdot}10^{-2}$	0.577	$3.15{\cdot}10^{-2}$	0.345	$1.07{\cdot}10^{-2}$	0.731	$1.61{\cdot}10^{-2}$	0.565
BR	$3.12{\cdot}10^{-2}$	0.360	$3.35{\cdot}10^{-2}$	0.302	$1.99{\cdot}10^{-2}$	0.500	$1.98{\cdot}10^{-2}$	0.474
R-F	$5.22{\cdot}10^{-3}$	0.893	$\mathbf{2.98{\cdot}10^{-2}}$	**0.377**	$3.39{\cdot}10^{-3}$	0.915	$1.78{\cdot}10^{-2}$	0.527
SVR	$1.52{\cdot}10^{-2}$	0.688	$3.53{\cdot}10^{-2}$	0.259	$1.21{\cdot}10^{-2}$	0.695	$1.69{\cdot}10^{-2}$	0.552
SVRF	$8.23{\cdot}10^{-3}$	0.831	$\mathbf{3.00{\cdot}10^{-2}}$	**0.371**	$6.17{\cdot}10^{-3}$	0.845	$1.58{\cdot}10^{-2}$	0.579

TABLE 2: Prediction Accuracy at *Mock-Town* and *Courtyard*

from the desired output. An MSE value of $2.0{\cdot}10^{-2}$ equates to roughly 14% error when the true p_d value is 1 while an MSE value of $6.0{\cdot}10^{-2}$ equates to roughly 26% error. We would like an MSE value close to or below $2.0{\cdot}10^{-2}$ and do not want values above $6.0{\cdot}10^{-2}$. The R2 score is a "goodness of fit" metric that tells us how well the inputs predict the desired output. We prefer an R2 score above 0.5, which tells us that there is positive correlation between the model inputs and the outputs, and we do not want a negative R2 score, which indicates that the model is a poor choice for the data. Due to the limited number of measurements, we chose an 80-20 train-test split ratio.

Tables 2 and 3 summarize our results, listing the average MSE and R2 values from both the training and test datasets. In all four environments, both the sigMod and BR approaches struggled to capture meaning from the data, with both having negative R2 scores in the *Office*. The GP model performed well in certain environments, such as the *Office* where it tied for best, and had similar results at times between the training set and test set, as seen on the *Courtyard* dataset, which suggests that the model is robust against over fitting. However, this model did not perform particularly well in the other environments, with performance comparable to the BR model and noticeably struggling on the *Camp* dataset. The GP-pos model performed better than the other models on the *Courtyard* dataset. This environment is fairly open and using position information was sufficient to determine peer-to-peer ETX here. However, the GP-pos model did not perform as well on the *Mock-Town* and *Office* data and had a negative R2 score in the *Camp* environment with changing RF noise. Interestingly, both the GP and GP-pos models performed the best on the training dataset in the *Camp* environment then saw a major drop in performance on the test data, suggesting major over fitting. Although GP models are powerful regression tools, we believe that they are struggling to predict ETX in most environments due to a limited number of data points. The GP-pos model specifically is failing in the *Camp* environment due to the changing RF noise that was not position dependent.

The NN model performed worse than all other models on every dataset, with consistently high MSE and a negative R2 score in every environment. This same model was found to perform quite well in [52], where they had a dataset with over one million measurements and trained with a batch size of 2,048, suggesting that this NN would have performed better for predicting p_d if we had a substantially larger dataset. The

	Camp				*Office*			
	Training Set		Test Set		Training Set		Test Set	
	MSE	R2	MSE	R2	MSE	R2	MSE	R2
sigMod	0.355	-1.19	0.358	-1.29	$6.28{\cdot}10^{-2}$	-0.29	$6.49{\cdot}10^{-2}$	-0.37
GP	$4.43{\cdot}10^{-3}$	0.973	0.112	0.286	$1.28{\cdot}10^{-2}$	0.738	$\mathbf{2.03{\cdot}10^{-2}}$	**0.581**
GP-pos	$5.15{\cdot}10^{-3}$	0.968	0.223	-0.425	$1.44{\cdot}10^{-2}$	0.704	$2.42{\cdot}10^{-2}$	0.490
NN	0.468	-1.89	0.493	-2.12	0.245	-4.01	25.3	-430
DNN	$4.97{\cdot}10^{-2}$	0.694	$7.15{\cdot}10^{-2}$	0.548	$1.88{\cdot}10^{-2}$	0.615	4.07	-65.6
BR	$9.71{\cdot}10^{-2}$	0.400	0.107	0.321	$2.15{\cdot}10^{-2}$	0.560	0.432	-6.875
R-F	$1.34{\cdot}10^{-2}$	0.917	$6.88{\cdot}10^{-2}$	0.563	$4.07{\cdot}10^{-3}$	0.917	$2.25{\cdot}10^{-2}$	0.534
SVR	$4.83{\cdot}10^{-2}$	0.702	$7.35{\cdot}10^{-2}$	0.535	$1.35{\cdot}10^{-2}$	0.723	$2.07{\cdot}10^{-2}$	0.574
SVRF	$2.39{\cdot}10^{-2}$	0.852	$\mathbf{6.39{\cdot}10^{-2}}$	**0.596**	$7.17{\cdot}10^{-3}$	0.853	$\mathbf{2.02{\cdot}10^{-2}}$	**0.583**

TABLE 3: Prediction Accuracy at *Camp* and *Office*

DNN performed well at times, ranking second best in the *Camp* and third in the *Mock-Town* and *Courtyard*. However, the DNN struggled greatly with the *Office* data. We used the *Courtyard* dataset when designing and initially evaluating the DNN. We believe that this causes the DNN's design to work well for outdoor environments but not for indoor environments, suggesting that the design of the model must change from environment-to-environment for this approach to work well.

The Random Forest model performed well on the *Mock-Town* and *Office* datasets but was not as consistent on the other datasets. This model outperformed all of the other models on the training data for the *Mock-Town*, *Courtyard*, and *Office* datasets but sees a large drop in performance on these test sets. This suggests that this model is over fitting to the data and does not generalize well. The SVR model does not appear to suffer from over fitting and had a reasonable drop in performance from the training data to the test data for all datasets. However, the SVR model by itself does not uniformly perform better than the other models.

The trend in the Random Forest and SVR performance motivates our proposed SVRF model. The SVRF model outperformed all other models on the test data in the *Camp* and *Office* environments, while performing second best in the *Mock-Town* and *Courtyard* environments, with a near tie on the former of the two. These results show that the SVRF model provides a nice balance between the superior results of the Random Forest and the generalizability of the SVR model.

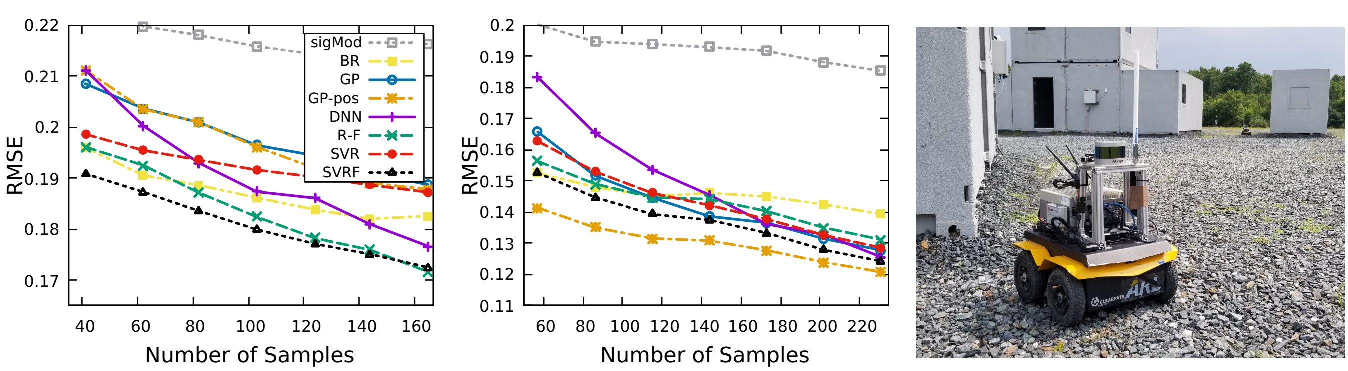
2) Training on Sparse Datasets

To evaluate how data sparsity impacts performance, we varied the size of the training set, training each regression model 50 times at each interval, and measured the average Root Mean Squared Error (RMSE). The RMSE tells us the average prediction error in p_d, where a lower RMSE is better. We prefer an RMSE at or below 0.15 and do not want a value above 0.25, which is an error of roughly 15% and 25%, respectively, when p_d is close to 1.

Figures 4a, 4b, 5a, and 5b graph our results for each of the datasets. We note that some of the models had a noticeably worse RMSE, causing them to not appear on the graph. For example, the NN performed very poorly in the *Mock-Town* environment, where the minimum achieved RMSE was 0.39, so the results for the NN do not appear in any of the figures while the sigMod and GP-pos models did not perform well enough to appear in the figures for the *Camp* and *Office* datasets.

Our SVRF model reaches the desired accuracy (0.15) with around 80 to 100 measurements on the *Courtyard* and *Office* datasets. No model reached our preferred error level on the *Mock-Town* and *Camp*datasets, with the latter only reaching an acceptable error level (0.25) after 300 measurements. However, our SVRF model is trending downward on all datasets as the number of measurements increase, suggesting that the model would eventually hit the preferred error level if given more data points.

The model-versus-model performance is mostly the same as already discussed in Section 1, with the SVRF model consistently stronger than the other approaches on the *Mock-Town*, *Camp*, and *Office* datasets and second best on the *Courtyard* dataset. The only model that has a notable performance trend different from the rest is the DNN, which is trending downwards at a steeper rate than most of the other models on the outdoor datasets (*Mock-Town*, *Courtyard*, and *Camp*). The same trend is seen with the GP model on the *Office* dataset. These trends suggest that the DNN and GP models could eventually outperform the SVRF model if given enough data in the environments where they perform well (outdoor environments for the DNN and indoors for the GP). All of the considered datasets each took several hours to collect due to the process required to measure ETX, meaning that the SVRF model's robustness to sparse datasets makes it more ideal than the DNN and GP. Furthermore, neither of

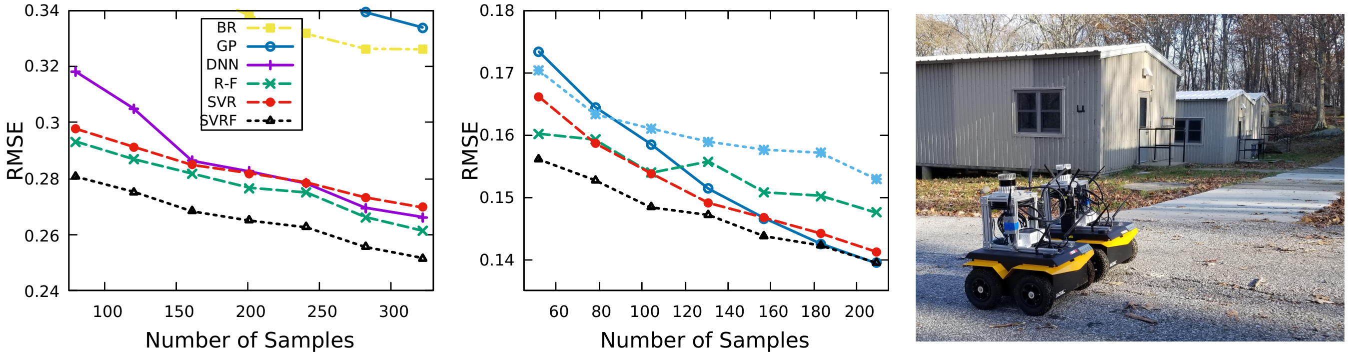


(a) *Mock-Town* Dataset

(b) *Courtyard* Dataset

(c) *Mock-Town* Environment

FIGURE 4: Graphs (a) and (b) show the average RMSE as the size of the training set varies on the *Mock-Town* and *Courtyard* datasets, respectively. Image (c) shows two of the Jackal robots (second robot at top right) in the *Mock-Town*.



(a) *Camp* Dataset

(b) *Office* Dataset

(c) *Camp* Environment

FIGURE 5: Graphs (a) and (b) show the average RMSE as the size of the training set varies on the *Camp* and *Office* datasets, respectively. Image (c) shows two of the Jackal robots in the *Camp*.

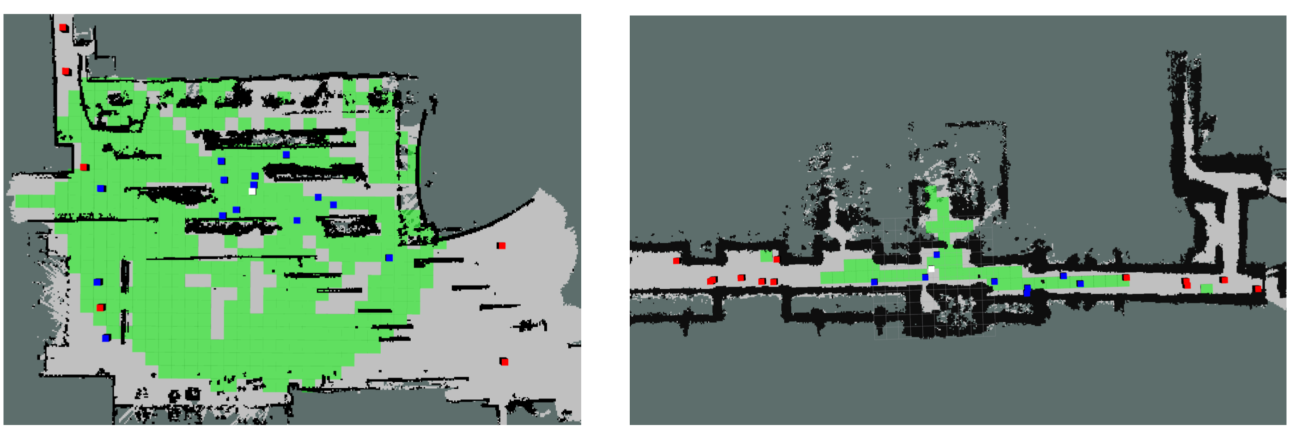
these alternative approaches are as universally adaptable to different environments as the SVRF model.

B. Maintaining Connectivity with Another Device

Our LQCM can be used to determine what subregions of the environment a robot can manuever through while maintaining a desired communication link with another device. The other device could be another robot or static infrastructure. To demonstrate this, we used an LQCM in the *Courtyard* and *Office* environments to determine which areas in the environment a robot could operate in while still maintaining a desired ETX connection with the base station. Possible applications for this setup include a robot exploring the environment while streaming sensor data back to the base station or when a robot is remotely operated from the base station, as seen in works such as [83], [84].

In the *Courtyard*, we cycled through every vertex in the LQCM and checked the predicted probability of delivery, p_d, from the *robot* to the *base station*. If the p_d was above 0.5 (an ETX of 2) we highlighted the cell green that is represented by that vertex in the LQCM. In a similar manner, we checked the predicted p_d from the *base station* to the *robot* vertices in the *Office* environment and highlighted all corresponding cells green where $p_d \geq 0.75$ (and ETX of 1.33). For each setup, we randomly selected 20 locations throughout the environment and measured p_d (from the robot to the base station for the *Courtyard* and from the base station to the robot in the *Office*).

Figure 6 shows the subregions of each environment (shaded in green) where the robot can maintain the desired communication link with the base station. The blue markers show where p_d was above the desired threshold and the red



Courtyard

Office

FIGURE 6: Connectivity results from the *Courtyard* and *Office*. Green shading are subregions that are predicted to meet the desired communication quality, blue markers are locations where p_d was measured above the desired threshold, red markers are locations where p_d was measured below the threshold, and the white markers are base stations.

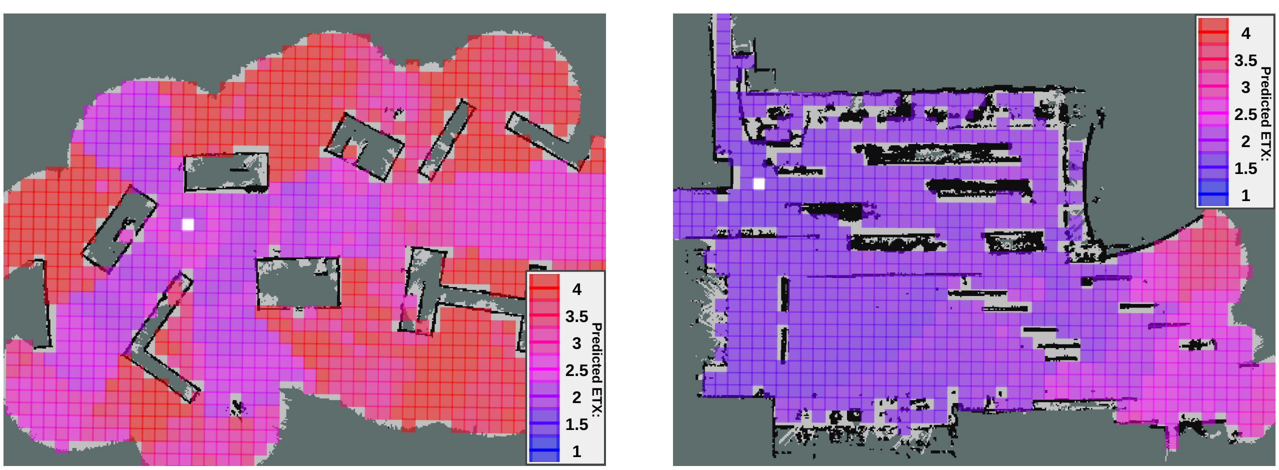
markers are where p_d was below the threshold. For clarity, we desire blue markers in shaded regions and red markers outside of shaded regions. In the *Courtyard* only two out of 20 measurements were not as predicted while in the *Office* only a single measurement out of 20 did not match the prediction. These experiments show how an LQCM can be used for identifying subregions of the environment where a robot can operate while still maintaining a desired communication link with another device.

C. Generating and Searching Maps

We generated LQCMs for each of the environments. As discussed in Section A, we discarded occupied cells and cells that are inaccessible by a robot to avoid generating an excessive number of cells for locations that are not feasible to deploy a robot. The outdoor environments (the *Mock-Town*, the *Courtyard*, and the *Camp*) have several open areas and each cover a large space, so we chose to use 2×2 meter-sized cells when discretizing these environments. This created 1,197 cells for the *Mock-Town*, 819 cells for the *Courtyard*, and 1,158 for the *Camp*. We chose a cell size of 0.6×0.6 for the *Office* because the hallways are much more narrow, with several small spaces that a robot can access but are too small to be accurately represented with a 2×2 meter cell. This created 1,046 cells for the *Office*.

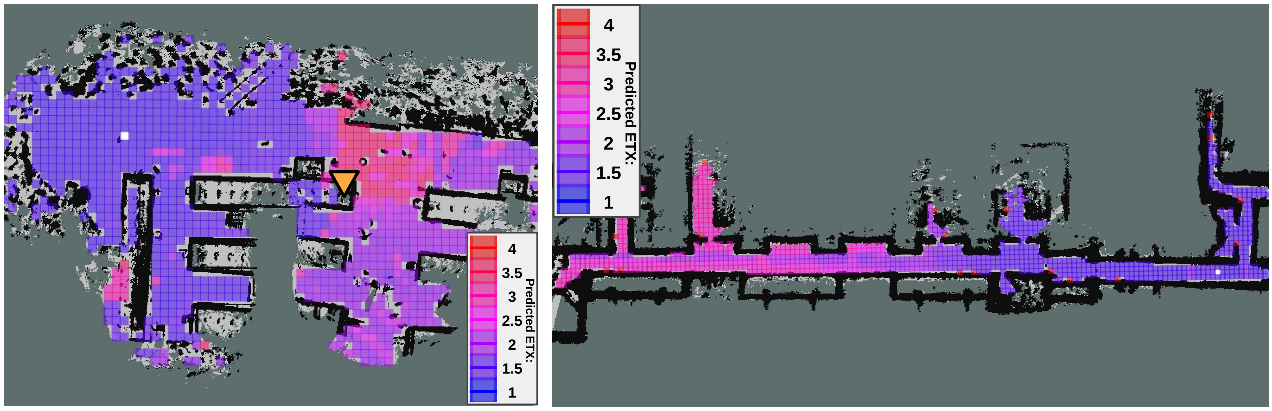
Figures 7 and 8 show the LQCMs for each environment. The color of each cell depicts the predicted ETX between a robot at the white cell and a robot at that specific cell. Blue represents a perfect connection (ETX = 1) and red is a weak connection (ETX = 4). In Fig. 7a, we can see the strong shadowing of each building in the *Mock-Town*. These buildings are all made out of metal and are extremely hard to penetrate with an RF signal. The affect of obstacles is much lower in the *Courtyard* because the small concrete barriers are not as difficult to communicate around, as seen in Fig. 7b. In Fig. 8a, we can see the impact of the WiFi jammer at the *Camp*, located at the orange triangle in the figure. These buildings are easier to communicate through than in the *Mock-Town*, so the impact of obstacles is not as noticeable but the impact of the WiFi jammer is very pronounced. Figure 8b shows the *Office* environment, which is mostly a narrow hallway. There are two sets of double doors that separate the two main segments of the hallway, which are not fully aligned. This misalignment is what creates the noticeable shadowing along the sides of the hallway on the left side of the figure, which is what we expect to see.

To evaluate the computational cost of searching for paths in an LQCM, we randomly selected robot source and sink locations in each of the environments and used Dijkstra's algorithm to get robot configurations from the LQCM, as discussed in Section B. We chose Dijkstra's algorithm instead of other CSP algorithms because we argue that, in most cases, Dijkstra's algorithm is sufficient for finding valid robot configurations, as discussed in Section B and further investigated in Section D. We used the disk method as an alternative approach for finding robot configurations. For the disk method, we assume that any two robots will be able to communicate when within a predefined distance, which was determined experimentally. When required, we selected additional robot deployment locations by minimizing the maximum distance between any two neighboring robots. We chose this baseline because it is often used to plan robot-to-robot communication [7], [10] and to plan robot-to-infrastructure data collection applications [9], [11]. For a second baseline we used the SNR method, as presented in [17]. For this approach, we predicted robot-to-robot signal strength



(a) *Mock-Town* (b) *Courtyard*

FIGURE 7: Link Quality Communication Maps for the *Mock-Town* and *Courtyard* environments. Each cell's color represents the predicted ETX between a robot at the white cell and a robot at that specific cell, where blue is a perfect connection (ETX = 1) and red is a weak connection (ETX = 4).



(a) *Camp* (b) *Office*

FIGURE 8: Link Quality Communication Maps for the *Camp* and *Office* environments. Each cell's color represents the predicted ETX between a robot at the white cell and a robot at that specific cell, where blue is a perfect connection (ETX = 1) and red is a weak connection (ETX = 4).

using the method described in Section A and predicted noise using k-nearest neighbors and noise measurements collected when exploring the maps. To determine if additional robots are needed to connect the source and sink robots, we set a minimum SNR threshold required for two robots to be considered connected. We then incrementally increased the number of relaying robots, searching the map for positions that maximize the minimum connection between any two robots, until the predicted robot-to-robot SNR was above the predefined minimum.

Table 4 shows the results of searching for 20 robot configurations in each environment. The disk method was the most computationally efficient and always computed in under one millisecond. The SNR method took around six milliseconds to find configurations in the *Mock-Town* and *Courtyard* but noticeably struggled in the *Camp* and *Office*. The latter two areas a very cluttered and the SNR method is very sensitive to obstructions, preferring line-of-sight when possible. This caused the algorithm to struggle to find satisfactory configurations at times in the *Camp* and *Office* where there is less room for additional robots to be positioned with line-of-sight between one another. Searching the LQCM with Dijkstra's algorithm was not as fast as the disk method but consistently solved in under 10 milliseconds. Our envisioned use for LQCMs is for them to be integrated into the mission planning phase of a robot deployment, where the search only runs periodically. Therefore, we argue that a computation time at or below 10 milliseconds is reasonable for the intended use.

D. Environmental Monitoring Application

We ran data transmission experiments in the two outdoor environments to demonstrate how an LQCM can be used for relaying data in an environmental monitoring application. We first looked at robot-to-robot data relaying in the *Mock-Town* then looked at streaming data from a robot to a base station in the *Courtyard*. In both environments, we transmitted large chunks of data and recorded the time required to move the data. We used total transmission time as our metric because this encompasses other common networking metrics (e.g. packet latency and packet error rate) while focusing on application-layer performance. We kept the robots stationary during data transfer because we are interested in comparing how well the LQCM represents robot-to-robot communication when compared to other classical methods. However, depending on the application, the robots may need to continue moving while communicating. For these types of scenarios, our LQCM would be incorporated into application-specific path planning algorithms, similar to how communication graphs have been used in works such as [7], [45], [49].

We again used the disk method and maximizing SNR as baselines. We were able to find valid robot configurations in the LQCM for every scenario using Dijkstra's algorithm

Environment	*Mock-Town*	*Courtyard*	*Camp*	*Office*
DM	< 1 / < 1	< 1 / < 1	< 1 / < 1	< 1 / < 1
SNR	6.05 / 0.22	6.2 / 0.41	4,516 / 3,114	1,946 / 3,047
LQCM	8.65 / 2.25	4.0 / 0.32	8.65 / 0.88	6.75 / 1.02

TABLE 4: Average computation time / standard deviation (both in milliseconds) to search for robot configurations.

	Mock-Town Environment				*Courtyard* Environment			
Scenario	1	2	3	4	5	6	7	8
DM	12.4	66.6	**1.7**	291.85	48.1	37.2	168.4	**17.5**
SNR	-	-	-	-	∞	40.0	207.2	27.4
LQCM	**7.3**	**4.1**	3.9	**11.6**	**38.6**	41.7	**23.4**	27.1

TABLE 5: Transmission Times for Environmental Monitoring Application

(avoiding the need to solve for CSPs as discussed in Section B).

For robot-to-robot data relaying, we designated one robot as a data source and another robot as a sink to replicate a scenario where robots are collecting and relaying data between each other. We moved the source robot and sink robot around the *Mock-Town* and used the LQCM to determine if additional robots were required to improve data-throughput and to decide where to position additional relay robots when needed. We only considered scenarios where the LQCM proposed a different robot configuration than the configuration found using the baseline. For each considered configuration, we had the source robot send the sink robot 100 data packets, each 110 kB in size for a total of 11 MB, and recorded the time required to complete the transmission. We repeated this experiment 10 times for each setup. This data relay experiment is meant to imitate transmitting larger data files, such as data muling applications where robots are used to collect sensor data [7], [10], [11].

Figure 9 shows the various source-to-sink scenarios we experimented with. Table 5 shows the results for each scenario. Except for *Scenario 3*, using our LQCM resulted in an increase in the data transmission rate for all scenarios in the *Mock-Town* environment. In *Scenario 1*, we see an example of where the disk method added an extra robot that slowed down data streaming while *Scenarios 2* and *3* show examples of where the disk method did not add an additional robot when the LQCM did. The additional robot was needed in *Scenario 2*, where the LQCM clearly outperforms the disk method, but was not needed in *Scenario 3*. We hypothesize that our method failed to outperform the baseline in *Scenario 3* because the large building (marked red in 9) was positioned perfectly to reflect radio waves directly from the source to the sink robot and our ETX prediction method was unable to accurately predict this behavior. Interestingly, the fastest data-transfer rate found by the LQCM was in *Scenario 3*, suggesting that using the LQCM is still a competitive method in scenarios where the ETX prediction may have some error. Finally, *Scenario 4* shows an example where both the LQCM and the disk method chose the same number of robots but placed them in different locations. The disk method was agnostic to how the buildings would impact communication and placed a relay robot between two buildings. This caused the disk method to struggle greatly, averaging over 290 seconds to transmit the data, while using the LQCM only took 11.6 seconds, a transfer rate comparable to the other scenarios. On average, using the LQCM decreased transmission time by 26.6% in the *Mock-Town* environment, which extreme cases (*Scenarios 2* and *4*) seeing over 90% decrease in transmission time.

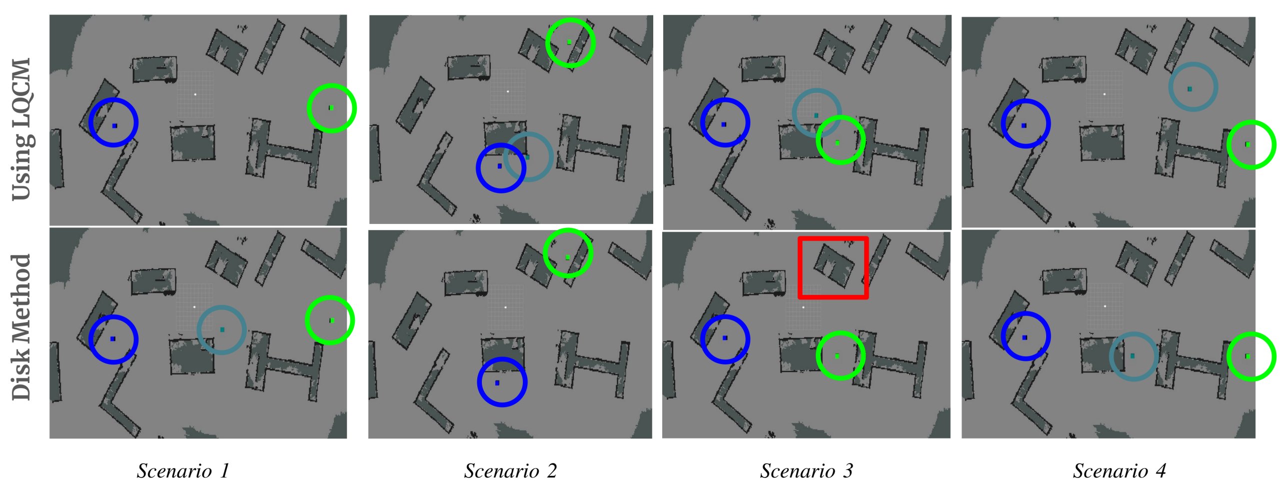


FIGURE 9: Environmental monitoring application in the *Mock-Town* environment. Blue markers show the source robot, green markers show the sink robot and turquoise are relay robots. The red box marks a building discussed in Section D.

For the robot-to-base-station streaming scenarios in the *Courtyard* environment, we designated one robot as a data source and positioned the other robots to help relay the data back to the base station (as needed). We again used an LQCM to determine when any additional robots were needed to connect the source robot to the base station. For each configuration, we had the source robot send the base station 714 data packets, each 1.4 kB in size for a total of 1 MB, and recorded how long it took to transmit the data. We repeated the experiment 10 times for each robot configuration. This experiment imitates streaming sensor readings, such as a video feed as seen in [17], [85], [86].

Figure 10 shows the various robot-to-base-station scenarios and Table 5 shows the results for each scenario. The disk method outperformed using an LQCM in *Scenario 8* while the LQCM outperformed the disk method in *Scenarios 5* and *7*, and both methods performed comparable in *Scenario 6* despite finding different robot configurations. The disk method was more eager to add additional relay robots than the LQCM, doing so in all but *Scenario 7*. We believe that the LQCM failed to outperform the disk method in *Scenario 8* because the prediction model is pessimistic on how well the robot could communicate with a relay robot and determined that communicating directly with the base station would be more efficient. On average, using the LQCM still decreases the transmission time by 9.7% when compared against the disk method in the *Courtyard* environment and by a total of 18.2% across all scenarios in both environments. The SNR method found the same robot configuration as using an LQCM in *Scenarios 6* and *8* but differed from the LQCM in *Scenarios 5* and *7*, using one less robot than the LQCM. In *Scenario 5*, the SNR method picked a robot configuration where the source robot was unable to connect to the base station at all. In *Scenario 7*, using the LQCM decreased transmission time by 88.7% compared to using the SNR method.

These environmental monitoring experiments highlight the downfalls of each baseline and demonstrate how using an LQCM helps avoid these downfalls and improves multi-robot team performance. The disk method tends to be too conservative while also being agnostic to how obstacles will impact communication, which became particularly problematic in the *Mock-Town* environment where the buildings are more difficult to communicate through. On the other hand, the SNR method is responsive to the environment but tends to be too liberal, leading to scenarios where using SNR results in complete communication failure. Using the LQCM gives us a balance between the other two methods, where we are aware of how obstacles will impact communication while not being too optimistic about peer-to-peer performance.

VII. Conclusions, Limitations, and Future Work

This work presents the Link Quality Communication Map (LQCM), a way of mapping the potential for two robots to communicate using the Expected Transmission Count (ETX) link quality metric. We presented various properties of LQCM and highlighted how they can be used in a variety of distributed, multi-robot applications. Our field test

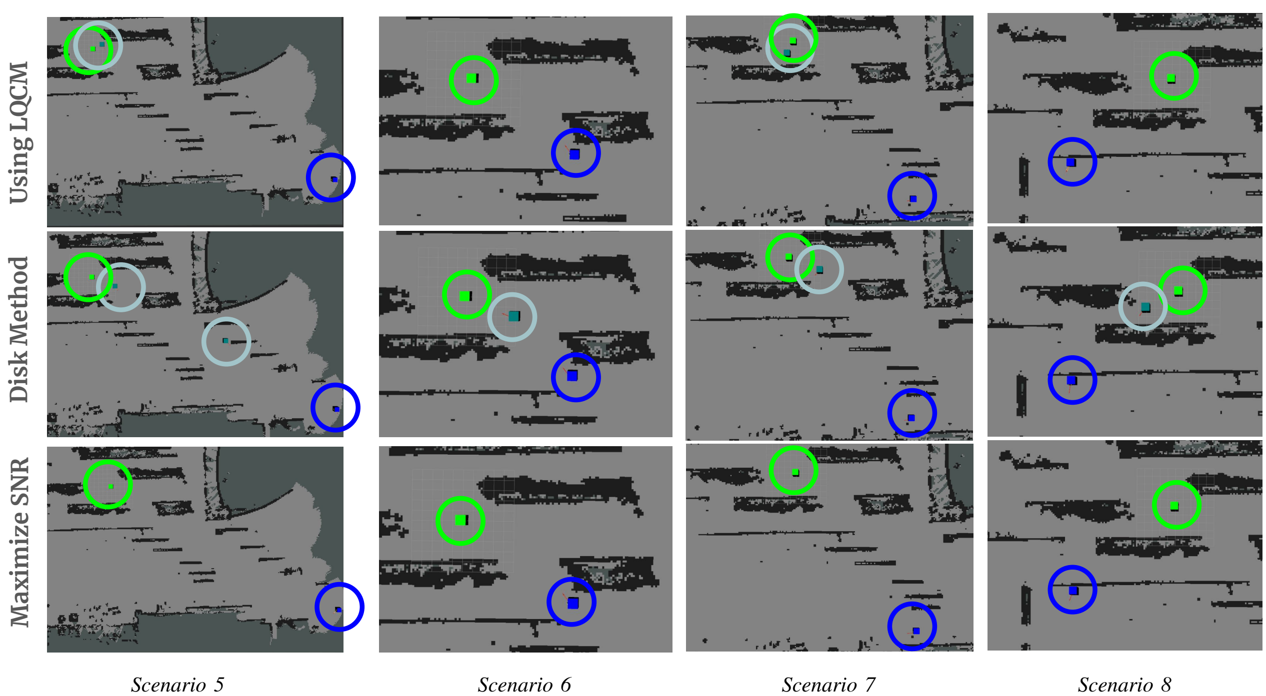


FIGURE 10: Data streaming in the *Courtyard* environment.

results support our claims for why communication-aware robot planning should use an LQCM instead of traditional methods for representing communication. We also presented an aggregated regression model, termed SVRF, for predicting ETX between points by extracting features from an occupancy grid, including robot-to-robot distance, obstruction, RF noise, radio power settings, and a prediction on signal multi-path distance. Using data collected on physical robots, we showed that our ETX prediction model outperforms other approaches. To promote research integrity, we have made the ROS package used for generating an LQCM and our experimental data publicly available[1].

This work shows great promise in closing the gap between agnostic communication models used in robot planning and real-world robot-to-robot communication but has limitations. In this work, we assumed that the physical environment is static and known a priori. This limitation could be addressed by rapidly updating the structure of an LQCM online, as changes to the environment are discovered. Future work should focus on incorporating transfer learning methods (for example, see [87]) to overcome this limitation and make our prediction models more robust to changing domains.

Another limitation is that collecting ETX measurements to train our SVRF model is time consuming – each of the presented datasets took several hours to collect. Possible approaches to overcome this limitation include adapting prior knowledge to novel environments with similar characteristics, using simulations to generate training data as suggested in [40], [41], or using filtering methods such as those seen in [31], [42], [43] to speed-up the data collection process. If given a larger dataset, more powerful learning approaches, such as Gaussian Process Regression or Deep Neural Networks, may prove to be better ETX predictors. Additionally, other factors, such as the material type of obstacles and the time of day, impact ETX but were not incorporated into our ETX prediction model. One possible extension to this work is to create scene graphs, similar to the ones found in [88], that hold information on predicted object material types, which could be incorporated into our ETX prediction process.

To simplify the process of determining how many robots are required and where to position robots to move data from robot to robot and from robot to humans, we chose to discretize the environment. We showed that this discrete representation provides optimality guarantees when selecting robot configurations but at the limitation that we may be losing communication quality accuracy through the coarseness of the regions or from a poor decomposition of the environment. Future work could further investigate methods for building communication maps using ETX in continuous space. Another limitation of our proposed LQCM design is that it currently has no mechanism for handling how a single robot can be used for multiple data paths through the graph. The ETX along any given data path would depend on how strong the connection is along the other data paths, the size of the data packets, and the interval of communication (a continuous data stream versus periodical data transfer). To the best of our knowledge, this topic has not been fully investigated.

In addition to addressing the current limitations of LQCM as presented here, we see many areas for future work. We plan to investigate how to integrate building an LQCM with the multi-robot Simultaneous Localization and Mapping problem. We also plan to expand our LQCM for three-dimensional maps and include multi-modal robots (e.g. accommodating both ground vehicles and drones). As discussed in Section B, finding certain types of paths in an LQCM is hard and more work is needed in identifying and evaluating algorithms for finding such paths in an LQCM.

[1] https://github.com/JonD07/CommunicationMap

Acknowledgements

We would like to thank Yee Shen Teoh for his help collecting training data. We would also like to thank the members of the Pervasive Computing Systems Research Group (PeCS) for their feedback on this manuscript. This work is supported in part by the ARL DCIST W911NF-17-2-0181.

REFERENCES

[1] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 1–13.

[2] D. S. De Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of multihop wireless networks: Shortest path is not enough," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 83–88, 2003.

[3] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11 b mesh network," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*, 2004, pp. 121–132.

[4] N. Chakchouk, "A survey on opportunistic routing in wireless communication networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2214–2241, 2015.

[5] Y. Qu, B. Ng, and W. Seah, "A survey of routing and channel assignment in multi-channel multi-radio wmns," *Journal of Network and Computer Applications*, vol. 65, pp. 120–130, 2016.

[6] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 134–146.

[7] J. Scherer and B. Rinner, "Multi-uav surveillance with minimum information idleness and latency constraints," *Robotics and Automation Letters*, vol. 5, no. 3, pp. 4812–4819, 2020.

[8] C. Mao, J. Liu, and L. Xie, "Multi-uav aided data collection for age minimization in wireless sensor networks," in *International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2020, pp. 80–85.

[9] M. Chen, W. Liang, and S. K. Das, "Data collection utility maximization in wireless sensor networks via efficient determination of uav hovering locations," in *International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021, pp. 1–10.

[10] M. A. Schack, J. G. Rogers, Q. Han, and N. T. Dantam, "Robot team data collection with anywhere communication," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 705–711.

[11] J. Diller, P. Hall, and Q. Han, "Holistic path planning for multi-drone data collection," in *19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, 2023.

[12] M. A. Schack, J. G. Rogers, and N. T. Dantam, "The sound of silence: Exploiting information from the lack of communication," *Robotics and Automation Letters*, 2024.

[13] I. T. Union, "Effects of building materials and structures on radiowave propagation above about 100 mhz," 2015.

[14] M. Safak, *Digital communications*. John Wiley & Sons, 2017.

[15] E. Yanmaz, S. Hayat, J. Scherer, and C. Bettstetter, "Experimental performance analysis of two-hop aerial 802.11 networks," in *Wireless Communications and Networking Conference (WCNC)*. IEEE, 2014, pp. 3118–3123.

[16] M. Sweatt, A. Ayoade, Q. Han, J. Steele, K. Al-Wahedi, and H. Karki, "Wifi based communication and localization of an autonomous mobile robot for refinery inspection," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4490–4495.

[17] J. Diller, N. Dantam, J. Rogers, and Q. Han, "Communication jamming-aware robot path adaptation," in *19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, 2023.

[18] J. N. Twigg, J. R. Fink, P. L. Yu, and B. M. Sadler, "Efficient base station connectivity area discovery," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1398–1410, 2013.

[19] M. M. Zavlanos, A. Ribeiro, and G. J. Pappas, "Network integrity in mobile robotic networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 3–18, 2012.

[20] J. Fink, A. Ribeiro, and V. Kumar, "Robust control of mobility and communications in autonomous robot teams," *IEEE Access*, vol. 1, pp. 290–309, 2013.

[21] C. Dixon and E. W. Frew, "Maintaining optimal communication chains in robotic sensor networks using mobility control," *Mobile Networks and Applications*, vol. 14, pp. 281–291, 2009.

[22] B. Ning, Q.-L. Han, Z. Zuo, J. Jin, and J. Zheng, "Collective behaviors of mobile robots beyond the nearest neighbor rules with switching topology," *IEEE transactions on cybernetics*, vol. 48, no. 5, pp. 1577–1590, 2017.

[23] M. Ocana, L. Bergasa, M. Sotelo, J. Nuevo, and R. Flores, "Indoor robot localization system using wifi signal measure and minimizing calibration effort," in *International Symposium on Industrial Electronics*, vol. 4. IEEE, 2005, pp. 1545–1550.

[24] J. Biswas and M. Veloso, "Wifi localization and navigation for autonomous indoor mobile robots," in *international conference on robotics and automation*. IEEE, 2010, pp. 4379–4384.

[25] I. T. Union, "Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 300 mhz to 450 ghz," 2019.

[26] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. Cock, "Vmaf: The journey continues," *Netflix Technology Blog*, vol. 25, no. 1, 2018.

[27] A. D. Wood, J. A. Stankovic, and S. H. Son, "Jam: A jammed-area mapping service for sensor networks," in *24th Real-Time Systems Symposium*. IEEE, 2003, pp. 286–297.

[28] Y. Peng, Y. Yang, X. Lu, and X. Ding, "Coding-aware routing for unicast sessions in multi-hop wireless networks," in *Global Telecommunications Conference GLOBECOM*. IEEE, 2010, pp. 1–5.

[29] N. Anyameluhor and E. Peytchev, "Routing and communication path mapping in vanets," in *Proceedings of the 28th European Conference on Modelling and Simulation (ECMS 2014), Brescia, Italy, 27-30 May 2014*, F. Squazzoni, F. Baronio, C. Archetti, and M. Castellani, Eds. European Council for Modelling and Simulation, 2014, pp. 359–363. [Online]. Available: http://dx.doi.org/10.7148/2014-0359

[30] K.-H. Kim and K. G. Shin, "On accurate measurement of link quality in multi-hop wireless mesh networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, 2006, pp. 38–49.

[31] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, and E. J. Coyle, "A kalman filter based link quality estimation scheme for wireless sensor networks," in *Global Telecommunications Conference*. IEEE, 2007, pp. 875–880.

[32] J. Zhou, C. Guo, P. Pawelczak, and I. Niemegeers, "Adaptable link quality estimation for multi data rate communication networks," in *69th Vehicular Technology Conference*. IEEE, 2009, pp. 1–5.

[33] IEEE, "Draft standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE P802.11-REVma/D89.0*, pp. 1–1230, 2010.

[34] H.-j. Im, C.-E. Lee, and Y.-J. Cho, "Radio mapping scheme using collective intelligent robots for teleoperation in unstructured environments," in *international symposium on robot and human interactive communication*. IEEE, 2014, pp. 856–861.

[35] M. Zoula and J. Faigl, "Wireless communication infrastructure building for mobile robot search and inspection missions," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5970–5976.

[36] C. R. Karanam and Y. Mostofi, "A foundation for wireless channel prediction and full ray makeup estimation using an unmanned vehicle," *Sensors Journal*, 2023.

[37] Z. Ji, G. Li, R. Han, S. Wang, B. Bai, W. Xu, K. Ye, and C. Xu, "Mpcom: Robotic data gathering with radio mapping and model predictive communication," *arXiv preprint arXiv:2404.10541*, 2024.

[38] C. Nieto-Granda, J. G. Rogers III, N. Fung, S. Kemna, H. I. Christensen, and G. Sukhatme, "On-line coordination tasks for multi-robot systems using adaptive informative sampling," in *Proceedings of the 2018 International Symposium on Experimental Robotics*. Springer, 2020, pp. 318–327.

[39] R. Miyagusuku, A. Yamashita, and H. Asama, "Precise and accurate wireless signal strength mappings using gaussian processes and path loss models," *Robotics and Autonomous Systems*, vol. 103, pp. 134–150, 2018.

[40] R. Levie, Ç. Yapar, G. Kutyniok, and G. Caire, "Radiounet: Fast radio map estimation with convolutional neural networks," *Transactions on Wireless Communications*, vol. 20, no. 6, pp. 4001–4015, 2021.

[41] M. Torun, H. Cai, and Y. Mostofi, "Spatial prediction of channel signal strength map using deep fully convolutional neural network," in *Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2022, pp. 553–558.

[42] R. Parasuraman, B.-C. Min, and P. Ögren, "Rapid prediction of network quality in mobile robots," *Ad Hoc Networks*, vol. 138, p. 103014, 2023.

[43] R. Parasuraman, P. Oegren, and B.-C. Min, "Kalman filter based spatial prediction of wireless connectivity for autonomous robots and connected vehicles," in *IEEE 88th vehicular technology conference (VTC-fall)*, 2018, pp. 1–5.

[44] Q. Gu, J. Yu, Z. Lin, J. Bai, B. Zhang, Y. Shen, J. Wang, and Y. Wang, "Md-radiomap: Multi-drone radio map building via single-anchor ultra-wideband localization network," in *International Conference on Automation Science and Engineering (CASE)*. IEEE, 2023, pp. 1–6.

[45] M. A. Schack, J. G. Rogers, Q. Han, and N. T. Dantam, "Optimizing non-markovian information gain under physics-based communication constraints," *Robotics and Automation Letters*, vol. 6, no. 3, pp. 4813–4819, 2021.

[46] J. Fink, J. Twigg, P. Yu, and B. M. Sadler, "A parsimonious model for wireless connectivity in robotic networks," in *IEEE Global Conference on Signal and Information Processing*, 2013, pp. 855–858.

[47] J. Banfi, A. Q. Li, N. Basilico, I. Rekleitis, and F. Amigoni, "Multirobot online construction of communication maps," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2577–2583.

[48] A. Quattrini Li, P. K. Penumarthi, J. Banfi, N. Basilico, J. M. O'Kane, I. Rekleitis, S. Nelakuditi, and F. Amigoni, "Multi-robot online sensing strategies for the construction of communication maps," *Autonomous Robots*, vol. 44, pp. 299–319, 2020.

[49] F. Amigoni, J. Banfi, N. Basilico, I. Rekleitis, and A. Quattrini Li, "Online update of communication maps for exploring multirobot systems under connectivity constraints," in *Distributed Autonomous Robotic Systems: The 14th International Symposium*. Springer, 2019, pp. 513–526.

[50] P. Ladosz, H. Oh, G. Zheng, and W.-H. Chen, "Gaussian process based channel prediction for communication-relay uav in urban environments," *Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 313–325, 2019.

[51] M. Zoula, M. Prágr, and J. Faigl, "On building communication maps in subterranean environments," in *International Conference on Modelling and Simulation for Autonomous Systems*. Springer, 2020, pp. 15–28.

[52] L. Clark, J. A. Edlund, M. S. Net, T. S. Vaquero, and A.-a. Agha-Mohammadi, "Propem-l: Radio propagation environment modeling and learning for communication-aware multi-robot exploration," in *Robotics: Science and Systems*, 2022.

[53] J. M. Gregory, J. N. Twigg, and J. R. Fink, "Enabling autonomous information-gathering and self-recovery behaviors in partially-known, communication-constrained environments," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 92–99.

[54] L. W. D. Project, "Rate control," 2023, accessed: 2024-10-15. [Online]. Available: https://wireless.docs.kernel.org/en/latest/en/developers/documentation/mac80211/ratecontrol/minstrel.html

[55] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 169–180, 2007.

[56] X. Ni, K.-c. Lan, and R. Malaney, "On the performance of expected transmission count (etx) for wireless mesh networks," in *Proceedings of the 3rd International conference on performance evaluation methodologies and tools*, 2008, pp. 1–10.

[57] P.-Y. Chen, L. Bhatia, R. Kolcun, D. Boyle, and J. A. McCann, "Contact-aware opportunistic data forwarding in disconnected lorawan mobile networks," in *40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 574–583.

[58] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch, "$free$—fine-grained scheduling for reliable and energy-efficient data collection in lorawan," *Internet of Things Journal*, vol. 7, no. 1, pp. 669–683, 2019.

[59] ITU-T, "Low power protocol for wide area wireless networks," International Telecommunication Union, Recommendation Y.4480, November 2021. [Online]. Available: https://www.itu.int/rec/T-REC-Y.4480

[60] E. A. Carlson, P.-P. Beaujean, and E. An, "Location-aware routing protocol for underwater acoustic networks," in *OCEANS*. IEEE, 2006, pp. 1–6.

[61] D. Hwang and D. Kim, "Dfr: Directional flooding-based routing protocol for underwater sensor networks," in *OCEANS*. IEEE, 2008, pp. 1–7.

[62] B. Yan, L. Wang, Y. Su, Y. Tian, and B. Lu, "A novel routing metric for underwater optical wireless communication using ns-3," in *International Conference on Big Data Computing and Communications (BigCom)*. IEEE, 2021, pp. 44–51.

[63] J.-C. Latombe, *Exact Cell Decomposition*. Boston, MA: Springer US, 1991, pp. 200–247. [Online]. Available: https://doi.org/10.1007/978-1-4615-4022-9_5

[64] J. E. Bresenham, "Algorithm for computer control of a digital plotter," in *Seminal Graphics: Pioneering Efforts That Shaped the Field, Volume 1*. New York, NY, USA: Association for Computing Machinery, 1998, pp. 1–6. [Online]. Available: https://doi.org/10.1145/280811.280913

[65] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, 1996.

[66] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[67] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear svm." *Journal of Machine Learning Research*, vol. 11, no. 4, 2010.

[68] M. R. Garey and D. S. Johnson, *Computers and intractability*. freeman San Francisco, 1979, vol. 174.

[69] L. D. P. Pugliese and F. Guerriero, "A survey of resource constrained shortest path problems: Exact solution approaches," *Networks*, vol. 62, no. 3, pp. 183–200, 2013.

[70] L. Lozano and A. L. Medaglia, "On an exact method for the constrained shortest path problem," *Computers & operations research*, vol. 40, no. 1, pp. 378–384, 2013.

[71] M. R. Souryal, J. Geissbuehler, L. E. Miller, and N. Moayeri, "Real-time deployment of multihop relays for range extension," in *Proceedings of the 5th international conference on Mobile systems, applications and services*, 2007, pp. 85–98.

[72] D. Mox, M. Calvo-Fullana, M. Gerasimenko, J. Fink, V. Kumar, and A. Ribeiro, "Mobile wireless network infrastructure on demand," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7726–7732.

[73] D. Mox, V. Kumar, and A. Ribeiro, "Learning connectivity-maximizing network configurations," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5552–5559, 2022.

[74] J. Wu, H. Lu, Y. Xiang, R. Wu, and F. Wang, "Mbr: A map-based relaying algorithm for reliable data transmission through intersection in vanets," *Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3661–3674, 2018.

[75] M. A. Childers, A. Hurwitz, J. Westrich, and J. Millemaci, *Robotics research collaboration campus: Operations for industry challenges*. DEVCOM, Army Research Laboratory, 2021.

[76] A. J. B. Trevor, J. G. Rogers, and H. I. Christensen, "Omnimapper: A modular multimodal mapping framework," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1983–1990.

[77] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2000.

[78] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[79] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[80] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of mathematical psychology*, vol. 85, pp. 1–16, 2018.

[81] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[82] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[83] J. Meyer, A. Prabhakar, A. Pinosky, I. Abraham, A. Taylor, M. Schlafly, K. Popovic, G. Diniz, B. Teich, B. Simidchieva *et al.*, "Scale-invariant specifications for human-swarm systems," *Transactions on Field Robotics*, 2024.

[84] G. Hattenberger, T. Verdu, N. Maury, P. Narvor, R. Bailon-Ruiz, G. Cayez, F. Couvreux, G. C. Roberts, and S. Lacroix, "Experimental uav flights to collect data within cumulus clouds," *Transactions on Field Robotics*, 2024.

[85] Y. Girdhar, L. Cai, S. Jamieson, N. McGuire, G. Flaspohler, S. Suman, and B. Claus, "Streaming scene maps for co-robotic exploration in bandwidth limited environments," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7940–7946.

[86] A. Kaknjo, M. Rao, E. Omerdic, L. Robinson, D. Toal, and T. Newe, "Real-time video latency measurement between a robot and its remote control station: Causes and mitigation," *Wireless Communications and Mobile Computing*, p. 8638019, 2018.

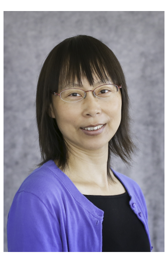
[87] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[88] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *Robotics: Science and Systems XVI*, 2020.

Jonathan Diller (Student Member, IEEE) earned a bachelor's degree in computer science from the Pennsylvania State University at Harrisburg and a master's degree in computer science from the Colorado School of Mines. He is currently a Ph.D. candidate in Robotics at Mines, where his research focuses on multi-robot systems, particularly path finding, tasking, communication, and energy sharing.

John G. Rogers (Senior Member, IEEE) received the bachelor's and master's degrees in electrical and computer engineering from Carnegie Mellon, the master's degree in computer science from Stanford, and the Ph.D. degree in robotics from the Georgia Institute of Technology. He is currently a Senior Research Scientist with the U.S. DEVCOM Army Research Laboratory (ARL), where he is also leading research efforts in coordinated tactical maneuver in complex and contested terrain for teams of ground robots.

Neil T. Dantam (Member, IEEE) is an Associate Professor of Computer Science at the Colorado School of Mines and Faculty Fellow at DEVCOM Army Research Laboratory. His work focuses on robot planning and manipulation, covering task and motion planning, kinematics, robot communication, and real-time software. Previously, Neil was a Postdoctoral Research Associate in Computer Science at Rice University working with Prof. Lydia Kavraki and Prof. Swarat Chaudhuri. Neil received a Ph.D. in Robotics from Georgia Tech, advised by Prof. Mike Stilman, and B.S. degrees in Computer Science and Mechanical Engineering from Purdue University.



Qi Han (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of California, Irvine, CA, USA, in 2005. She is currently a Professor in the Department of Computer Science at the Colorado School of Mines, Golden, CO, USA. Her research interests include networked multi-robot systems, Internet of Things, mobile sensing, edge computing, and augmented reality.