IEEE Access
practical innovations : open solutions

Received February 1, 2013, accepted April 24, 2013, date of publication May 10, 2013, date of current version May 24, 2013.

Digital Object Identifier 10.1109/ACCESS.2013.2262013

Robust Control of Mobility and Communications in Autonomous Robot Teams

JONATHAN FINK (Member, IEEE)[1], ALEJANDRO RIBEIRO (Member, IEEE)[2], AND VIJAY KUMAR (Fellow, IEEE)[3]

[1]U.S. Army Research Laboratory, Adelphi, MD 20783, USA
[2]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA
[3]GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA

Corresponding author: J. Fink (jonathan.r.fink3.civ@mail.mil)

This work was supported by the ARL MAST-CTA under Grant W911NF-08-2-0004 and the AFOSR MURI under Grant FA9550-10-1-0567.

ABSTRACT A team of robots are deployed to accomplish a task while maintaining a viable ad-hoc network capable of supporting data transmissions necessary for task fulfillment. Solving this problem necessitates: 1) estimation of the wireless propagation environment to identify viable point-to-point communication links; 2) determination of end-to-end routes to support data traffic; and 3) motion control algorithms to navigate through spatial configurations that guarantee required minimum levels of service. Therefore, we present methods for: 1) estimation of point-to-point channels using pathloss and spatial Gaussian process models; 2) data routing so as to determine suitable end-to-end communication routes given estimates of point-to-point channel rates; and 3) motion planning to determine robot trajectories restricted to configurations that ensure survival of the communication network. Because of the inherent uncertainty of wireless channels, the model of links and routes is stochastic. The criteria for route selection is to maximize the probability of network survival—defined as the ability to support target communication rates—given achievable rates on local point-to-point links. Maximum survival probability routes for present and future positions are input into a mobility control module that determines robot trajectories restricted to configurations that ensure the probability of network survival stays above a minimum reliability level. Local trajectory planning is proposed for simple environments and global planning is proposed for complex surroundings. The three proposed components are integrated and tested in experiments run in two different environments. Experimental results show successful navigation with continuous end-to-end connectivity.

INDEX TERMS Routing protocols, path planning, wireless networks, autonomous agents.

I. INTRODUCTION

Autonomousof a group of agents that collaborate to accomplish an assigned task. As an example situation, consider a building exploration mission in which a team of robots is sent to explore a plant before the entrance of human rescuers. Some members of the team are designated as leaders and move to specified locations while some other agents provide mission support. Instrumental to task accomplishment is the availability of wireless communications which is required to exchange information between robots as well as to relay information to and from human operators. Since availability of existing wireless communication infrastructure is unlikely in the harsh environments in which autonomous robot teams are to be deployed, we want the robots to self organize into a wireless network capable of supporting the necessary information exchanges.

Note that there are two high-level challenges in this example. First, the robots must autonomously decompose the exploration task, plan trajectories to satisfy pieces of that task, and finally determine controls to execute the desired trajectories, possibly in a coordinated fashion. Second, the robots must act as nodes in a wireless network in order to estimate wireless channels and relay information accordingly to support communication that would not be possible via a direct connection. Indeed, both of these challenges have been widely studied independently by the robotics and wireless networking communities respectively. The goal of this paper is to jointly design controllers that determine trajectories

for the robots while ensuring availability of communication resources.

Communication-aware deployment of robot teams has recently received considerable attention. Substantial contributions have been made on algorithms that either constrain or modify trajectories to preserve links, or recover from disconnected topologies [3]–[8]. Related contributions take a global view and design control algorithms that optimize network-wide indicators of connectivity such as the second smallest eigenvalue or $k-$connectivity of the network's graph. All of these papers rely on disc or simple topological models that identify proximity with connectedness. While early approaches to wireless networking do rely on disc models [9], [10], it has long been recognized that proximity does not necessarily indicate reliable communication due to shadowing and fading [11], [12]. A more accurate model used in this paper is to consider link rate functions that map pairs of terminal positions to achievable communication rates. These functions can be as simple as measuring packet error probabilities [13], [14] or may involve more complex models accounting for fading and power adaptation [15], [16]. These more nuanced models of wireless connectivity have been used in [17], [18] to design communication-aware robot planning algorithms.

All of the works mentioned so far use indirect indicators of network connectivity and do not determine whether the network formed by robots is able to support desired communication rates. Recent contributions consider rates explicitly and define network integrity as the ability to support desired communication rates [19]–[21]. While this represents a significant improvement over indirect indicators of connectivity, maintaining network integrity is challenging because channel uncertainty is inherent to robot deployment. Indeed, due to shadowing and small scale fading, even small variations in robots' positions lead to significant changes in channel strength [11], [12]. Precise channel state information can be acquired through measurements, but planning algorithms necessitate access to channel quality indicators at future positions to which the robots are yet to be deployed. Channel variability translates into large changes in achievable point-to-point communication rates between pairs of nearby robots, which in turn may result in significant uncertainty about desired end-to-end communication rates are achievable or not. This paper builds from the basic idea that spatial redundancy can minimize the effect of point-to-point uncertainty in end-to-end communication rates. Realizing this idea requires a stochastic model of connectivity integrated into mobility control and motion planning algorithms as we explain next.

A. CONTRIBUTIONS AND PREVIEW

We deviate from related approaches in two fundamental ways. First, network integrity is defined by the probability of maintaining achievable end-to-end communication rates above desired basal rates. This implies more than just the optimization of stochastic models of wireless channels as we formulate our problem statement to constrain actions of the system such that probabilistic guarantees are maintained. Second, we incorporate packet routing decisions that determine the effective network topology into the control space with the explicit purpose of mitigating rapid channel fluctuations that result from node movement in rich multipath environments. These contributions were first introduced in [1], [2] where [2] provides an initial high-level view of our approach and [1] focuses on global motion planning under this architecture. This paper additionally presents a survey and analysis of methods for communication link modeling (Section III), details on the optimization problem used to solve for robust network routing (Section IV), new theoretical results on gradient-based control (Section V), extensive experimentation (Section VII) including discussion and analysis of our method in comparison to recent literature (Section VIII).

We begin the paper by discussing how end-to-end communication rates depend on the values of point-to-point rates and the manner in which packets are routed through the network. A desirable goal is to find trajectories and position-dependent routing variables that accomplish the given task while ensuring that end-to-end rates exceed basal requirements at all points in time. However, this goal cannot be guaranteed deterministically because of channel uncertainty motivating the introduction of probabilistic channel models and a probabilistic redefinition of the primary goal (Section II).

We cover probabilistic models of point-to-point channels in Section III and stochastic routing algorithms that determine (stochastic) end-to-end rates in Section IV. While probabilistic channel models are well known and beginning to have extensive adoption in communication-aware robotic deployments, we take an alternate view focusing less on the overall accuracy of the prediction—which usually incurs a large sampling overhead—and more on pragmatic concerns that allow for deployment in realistic scenarios with models that are good enough to support probabilistic guarantees when utilized with the proposed stochastic routing methods. We discuss pathloss models (Section III-A), Gaussian process models (Section III-B) and present experimental measurements (Section III-C). Using these stochastic channel models as input, we follow up with the study of the probability of network survival defined as the ability to maintain end-to-end rates above basal requirements. Packet routing variables are determined in order to maximize this probability (Section IV). By splitting traffic flows between various neighboring robots, we ensure that while failure of a particular link may reduce end-to-end communication rates, it does not interrupt them completely. Note that our approach seeks to optimize the probability of maintaining the desired network connectivity rather than explicitly optimizing the network performance in terms of bandwidth, latency, or other traditional metrics.

We finally design trajectory planning algorithms to find feasible trajectories for which these probabilities satisfy a minimum reliability constraint. We first develop local planning algorithms for simple environments and present simulation and experimental results for three and four robot teams (Section V). We then propose global planning algorithms

to handle complex environments and the local minima that plague local controllers (Section VI). These global motion planners are tested in experiments conducted in two different environments. Experimental results show successful navigation with continuous end-to-end connectivity (Section VII). We finish with a discussion and numerical analysis that compares and places our method in the context of existing approaches (Section VIII).

II. PROBLEM STATEMENT

We begin by restating the problem statement from [1], [2]. Consider a team of N robots and denote their positions as x_i, for $i = 1, \ldots, N$. We assume the robots are kinematic and fully controllable which allows us to consider simple models of the form $\dot{x}_i(t) = u_i(t)$, where $u_i(t)$ is the input. A human operator is located at the fixed operation center that we index as $i = 0$ at position x_0. Further define vectors $\mathbf{x} := (x_0, \ldots, x_N) \in \mathbb{R}^{2(N+1)}$ and $\dot{\mathbf{x}} = (\dot{x}_0, \ldots, \dot{x}_N) \in \mathbb{R}^{2(N+1)}$. The team's task is specified through a generic scalar convex potential function $\Psi : \mathbb{R}^{2(N+1)} \to \mathbb{R}$. If the potential minimum $\Psi_{\min}$ is attained at $\mathbf{x}^*$, i.e., if $\Psi(\mathbf{x}^*) = \Psi_{\min}$, the configuration $\mathbf{x}^*$ satisfies task completion. E.g., if a designated leader agent ℓ must visit a target location $x_{\ell,g} \in \mathbb{R}^2$, we can define $\Psi(\mathbf{x}) = \|x_\ell - x_{\ell,g}\|^2$. The minimum $\Psi_{\min} = 0$ is attained by any configuration $\mathbf{x}^* = (x_0, \ldots, x_\ell, \ldots, x_N)$ for which $x_\ell = x_{\ell,g}$, or equivalently by any member of the set $\mathbf{x}^* \in \{\mathbf{x} = (x_0, \ldots, x_\ell, \ldots, x_N) : x_\ell = x_{\ell,g}\}$. Irrespective of the particular form of $\Psi(\mathbf{x})$, the control problem is to find velocities $\dot{\mathbf{x}}(t)$ such that at some time t_f the team configuration $\mathbf{x}(t_f) = \mathbf{x}(0) + \int_0^{t_f} \dot{\mathbf{x}}(t)dt$ satisfies task completion in that we have $\Psi(\mathbf{x}(t_f)) = \Psi_{\min}$. Mathematically, we can write this mobility control formulation as

$$\min_{\dot{\mathbf{x}}(t), t\in[0,t_f]} \Psi(\mathbf{x}(t_f))$$
$$\text{subject to } \mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \dot{\mathbf{x}}(u)\, du. \qquad (1)$$

As robots move to accomplish their task, they maintain *end-to-end* data flows between members of the team and/or members of the team and the operation center. Information flows are indexed as $k = 1, \ldots, K$. Flows may have multiple sources and multiple destinations. The set of destinations of the k-th information flow is denoted as dest(k). For agent i and flow k, the variable $a^k_{i,\min}$ represents the required communication rate between agent i and any of the agents in the set of destinations dest(k). E.g., if the only communication of interest is from the lead robot ℓ to the operation center, we have $K = 1$. Since the flow $k = 1$ is intended to the operating center, dest(1) $= 0$ and $a^1_{\ell\,\min}$ denotes the minimum level of service for this communication. All other variables $a^k_{i\,\min} = 0$ are null.

We model *point-to-point* connectivity through a rate function $R_{ij}(\mathbf{x}) = R_{ij}(x_i, x_j)$ that determines the amount of information that agent i at position x_i can send to agent j at position x_j. Since direct communication between the source and the destination of an information flow is not always possible, terminals self-organize into a multihop network to relay packets for each other. Packet relaying is determined by routing variables α^k_{ij} which describe the fraction of time node i spends transmitting data for flow k to node j; see Fig. 1. Thus, the product $\alpha^k_{ij}R_{ij}(\mathbf{x})$ determines the rate of point-to-point information transmission from i to j. If we consider the transmission to all neighboring terminals for which $R_{ij}(\mathbf{x}) > 0$, the total rate at which packets leave agent i is $\sum_{j=0}^N \alpha^k_{ij}R_{ij}(\mathbf{x})$. Likewise, the total rate at which i receives packets from other terminals is $\sum_{j=0,j\notin\text{dest}(k)}^N \alpha^k_{ji}R_{ji}(\mathbf{x})$. The information rate $a^k_i(\boldsymbol{\alpha}, \mathbf{x})$ available for flow k at source i is the difference between outgoing and incoming rates

$$a^k_i(\boldsymbol{\alpha}, \mathbf{x}) = \sum_{j=0}^N \alpha^k_{ij}R_{ij}(\mathbf{x}) - \sum_{j=0,j\notin\text{dest}(k)}^N \alpha^k_{ji}R_{ji}(\mathbf{x}), \qquad (2)$$

where we defined the vector $\boldsymbol{\alpha}$ grouping all routing variables α^k_{ij}. Notice that the variables α^k_{ij} represent time slot shares and must therefore satisfy $0 \le \alpha^k_{ij} \le 1$ for all i, j, and k. It must also be that $\sum_{j,k} \alpha^k_{ij} \le 1$ for all i to ensure that the sum of all time shares at terminal i does note exceed 1.

Routing variables $\boldsymbol{\alpha}$ and configuration-dependent rates $R_{ij}(\mathbf{x})$ determine the set $a^k_i(\boldsymbol{\alpha}, \mathbf{x})$ of end-to-end communication rates from each node i and flow k as per (2). The task specification requires that end-to-end rates exceed the minimum threshold $a^k_{i,\min}$. Therefore, integrity of the communication network necessitates that for all i and k

$$a^k_i(\boldsymbol{\alpha}, \mathbf{x}) \ge a^k_{i,\min} \quad \text{for all } i, k. \qquad (3)$$

Rates $a^k_i(\boldsymbol{\alpha}, \mathbf{x})$ depend on positions $\mathbf{x}$ and routing variables $\boldsymbol{\alpha}$. To control end-to-end connectivity, i.e., to satisfy (3), we can resort to control of positions $\mathbf{x}$, routes $\boldsymbol{\alpha}$, or both.

Since communication is necessary for task completion, the mobility control problem as summarized in (1) is redefined. The new goal is to find algorithms and control policies that govern robot motions in order to satisfy the task specifications in (1) and (3). Reducing $\Psi(\mathbf{x})$ as per (1) and ensuring network integrity as per (3) may be conflicting requirements. We therefore replace (1) by a concurrent search of trajectories $\mathbf{x}(t)$ and routes $\boldsymbol{\alpha}(t)$ so that the task potential is minimized without

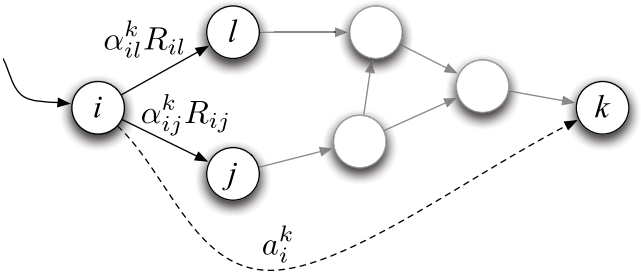


FIGURE 1. Data flows. We want to support end-to-end rate a^k_i from node i to destination k. The rate a^k_i depends on the point-to-point rates R_{ij} and the routing variables α^k_{ij} that determine the fraction of time node i sends packets to node j for flow k. We use stochastic models of both, a^k_i and R_{ij}.

ever breaking communication connectivity,

$$\begin{aligned} \min_{\boldsymbol{\alpha}(t),\dot{\mathbf{x}}(t),t\in[0,t_f]} \quad & \Psi(\mathbf{x}(t_f)) \\ \text{subject to} \quad & a_i^k(\boldsymbol{\alpha}(t),\mathbf{x}(t)) \geq a_{i,\min}^k \\ & \mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \dot{\mathbf{x}}(u)\, du \end{aligned} \tag{4}$$

where rates $a_i^k(\boldsymbol{\alpha}(t),\mathbf{x}(t))$ are given by the expression in (2) with $\boldsymbol{\alpha} = \boldsymbol{\alpha}(t)$ and $\mathbf{x} = \mathbf{x}(t)$. While continuous maintenance of communication connectivity significantly constrains our problem, there are many motivating mission scenarios including teleoperation of robotic assets or transport of latency-intolerant information such as voice or video communication.

A drawback of the formulation in (4) is the difficulty of ensuring that the constraints in (3) are satisfied. As per (2) rates $a_i^k(\boldsymbol{\alpha},\mathbf{x})$ depend on the link rates $R_{ij}(\mathbf{x})$, which are difficult to estimate; see e.g., [22] and Section III. Indeed, say that accurate channel estimates $\bar{R}_{ij}(\mathbf{x}(t))$ have been acquired for the current spatial configuration $\mathbf{x}(t)$. The challenge here is that rate estimates $\bar{R}_{ij}(\mathbf{x})$ are needed not only for configuration $\mathbf{x}(t)$, but for nearby configurations to which the robots are to move. The high variability of wireless channels makes current channel estimates $\bar{R}_{ij}(\mathbf{x}(t))$ poor predictors of rates $R_{ij}(\mathbf{x})$ even if $\mathbf{x}$ is close to $\mathbf{x}(t)$.

In order to redefine (4) in a manner that takes into account this significant uncertainty, we introduce a probabilistic formulation of channel rates. The important observation here is that if point-to-point link rates $R_{ij}(\mathbf{x})$ become random, so do the rates $a_i^k(\boldsymbol{\alpha},\mathbf{x})$ of end-to-end communication flows [cf. (2)]. Consequently, it is not possible to guarantee satisfaction of the constraints in (3). Rather, we introduce a reliability tolerance ϵ and require that for all i and k

$$\mathrm{P}\left[a_i^k(\boldsymbol{\alpha},\mathbf{x}) \geq a_{i,\min}^k\right] \geq \epsilon. \tag{5}$$

I.e., we require that the end-to-end link between all sources i and the destinations of all corresponding flows k exceed their minimum required level of service with probability larger than ϵ. Correspondingly, we reformulate the concurrent routing and mobility problem in (4) as

$$\begin{aligned} \min_{\boldsymbol{\alpha}(t),\dot{\mathbf{x}}(t),t\in[0,t_f]} \quad & \Psi(\mathbf{x}(t_f)) \\ \text{subject to} \quad & \mathrm{P}\left[a_i^k(\boldsymbol{\alpha},\mathbf{x}) \geq a_{i,\min}^k\right] \geq \epsilon \\ & \mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \dot{\mathbf{x}}(u)\, du. \end{aligned} \tag{6}$$

The focus of this work is the solution of (6).

The problem formulation in (6) inherits some standard complications from the control formulation in (1). The concurrent search in (6) is further complicated by the entanglement of the routing and mobility problems. We deal with this entanglement by fixing $\mathbf{x}$ and selecting $\boldsymbol{\alpha}$ in a manner that optimizes the reliability $\mathrm{P}\left[a_i^k(\boldsymbol{\alpha},\mathbf{x}) \geq a_{i,\min}^k\right]$ (Section IV). We follow up with local (Section V) and global (Section VI) searches on positions $\mathbf{x}$ to minimize $\Psi(\mathbf{x})$ while keeping reliabilities above the ϵ threshold. Before studying these problems, we introduce probabilistic models of point-to-point rates $R_{ij}(\mathbf{x})$ in the following section.

III. POINT-TO-POINT RATES

With robots i and j located at positions x_i and x_j we seek a probabilistic model for the supported communication rate $R_{ij}(\mathbf{x}) = R_{ij}(x_i, x_j)$. More specifically, we wish to model the expected value $\bar{R}_{ij}(\mathbf{x}) = \bar{R}_{ij}(x_i, x_j)$ and variance $\tilde{R}_{ij}(\mathbf{x}) = \tilde{R}_{ij}(x_i, x_j)$. We use simple radios that do not perform rate or power adaptation—we use 2.4 GHz Zigbee radios in our experiments; see Section III-C. In that case the communication rate $R_{ij}(x_i, x_j)$ is a function of the packet error rate of the channel $p_e(P_R(x_i, x_j))$, which in turn is a function of the received signal strength (RSSI) $P_R(x_i, x_j)$ and the noise power P_{N_0} [23]. Translation of RSSI $P_R(x_i, x_j)$ into packet error rates $p_e(P_R(x_i, x_j))$ depends on the type of modulation and the choice of error correcting codes. A generally good approximation to the packet error rate is

$$p_e(P_R(x_i, x_j)) = \operatorname{erfc}\left(\sqrt{k\frac{P_R(x_i, x_j)}{P_{N_0}}}\right), \tag{7}$$

where $\operatorname{erfc}(x)$ is the complementary error function, and k is a constant that depends on modulation and coding [23]. To determine communication rates $R_{ij}(x_i, x_j)$ we simply multiply the information rate of transmitted packets R_0 by the probability of successful decoding, $R_{ij}(x_i, x_j) = R_0\big[1-p_e(P_R(x_i, x_j))\big]$. We can then focus on models of RSSI $P_R(x_i, x_j)$ that we cascade into models of packet error rate and supported communication rate $R_{ij}(x_i, x_j)$ by means of (7).

We assume that robots perform exploratory observations of the radio frequency environment that they use to characterize the mean $\bar{P}_R(x_i, x_j)$ and variance $\tilde{P}_R(x_i, x_j)$ of RSSI values. Consider then a set of N RSSI measurements $\mathbf{z}_i$ corresponding to transmissions from a source located at $\mathbf{x}_{i,s}$ towards a receiver located at $\mathbf{x}_{i,r}$. Group in $\mathbf{y}_i := (\mathbf{x}_{i,s}, \mathbf{x}_{i,r})$ the coordinates of the source-receiver pair and denote as $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ and $\mathbf{z} = \{z_1, \ldots, z_N\}$ the set of all measurements. We consider regression techniques based on a physical pathloss model in Section III-A and a modified technique that captures spatial correlation through the use of spatial Gaussian process in Section III-B. In both cases the goal is to estimate $P_R(x_i, x_j)$ using observations $\mathbf{Y}$ and $\mathbf{z}$.

It is important to note at this point that our goal in analyzing predictive models of point-to-point signal strength, and thus rates, is not to propose new models that more accurately predict or fit the channels our system will experience. Instead, our goal in this section is to explore the tradeoffs inherent in choice of channel model with respect to factors such as number of training samples, conservative representation of uncertainty, and avoidance of computational complexity.

A. PATHLOSS MODELS

Received signal power $P_R(x_i, x_j)$ is determined by three phenomena: path–loss due to the distance from the source,

shadowing due to obstacles in the propagation path, and multipath fading that arises as a result of reflections and refractions. Of these three phenomena, path–loss and shadowing can be incorporated into a predictive model with relative ease. Fading, however, is difficult to predetermine [15], [22], [24]. Given this difficulty, we adopt the following standard model for the received power $P_{R,\mathrm{dBm}}(x_i, x_j) = 10\log(P_R(x_i, x_j))$ measured in dBm

$$P_{R,\mathrm{dBm}}(x_i, x_j) = \underbrace{L_0 - 10n \cdot \log(\|x_i - x_j\|)}_{\text{Path–loss}} - \underbrace{W(x_i, x_j)}_{\text{Shadowing}} - \underbrace{\mathcal{F}}_{\text{Fading}}, \quad (8)$$

where the term $\mathcal{F}$ is a zero-mean Gaussian random variable with variance $\sigma^2_{\mathcal{F}}$ modeling fading effects. The term L_0 is the measured power at a reference distance 1 m from the source, n is a path–loss exponent, and $W(x_i, x_j)$ is a non-smooth step function to model shadowing as a function of the number of obstacles between source and destination. The function $W(x_i, x_j)$ is set to $W(x_i, x_j) = 0$ if there is line of sight between x_i and x_j and to $W(x_i, x_j) = w$ if there are obstacles between this point. This model of shadowing requires knowledge of the environment's geometry. If this is not available, we neglect shadowing by making $W(x_i, x_j) = 0$. In either case, the problem of determining the mean $\bar{P}_R(x_i, x_j)$ and variance $\tilde{P}_R(x_i, x_j)$ of the RSSI reduces to the determination of the parameters $L_0, n, w, \sigma_{\mathcal{F}}$. Curve fitting to the experimental data $\mathbf{Y}$ and $\mathbf{z}$ is used for this purpose though it has been shown that these values can also be learned through online measurements [16], [18].

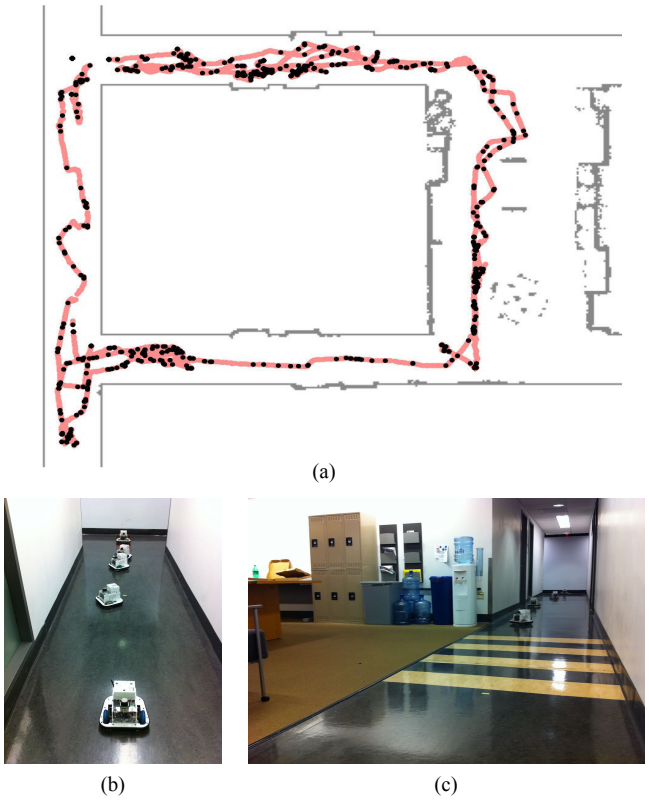


FIGURE 2. RSSI data collection in Levine building. (a) Projection of measurement dataset Y onto $\mathbb{R}^2$ for the Levine building environment. Measurements are taken between pairs of robots in a 6 robot team but we display the measurement locations by projecting onto $\mathbb{R}^2$ and only displaying the location of the receiving node for each measurement. The uniform training set Y is determined by choosing 1,000 of the measurements (black points). (b) and (c) depict environment snapshots.

B. SPATIAL GAUSSIAN PROCESS MODEL

While distance-based models provide a physics-based dimensionality reduction that is reasonably effective for modeling point-to-point signal strength, radio signal propagation is environmentally dependent and may warrant a spatially correlated estimation. We investigate here the use of a Gaussian process (GP) which is a generalization of a typical Gaussian distribution that describes a distribution over functions [25]. This allows us to generate probabilistic predictions of received point-to-point signal strength based on a data-driven model.

Gaussian process regression relies on a prior on the expected value of the process, $\mu(\mathbf{y})$, and a prior on the covariance of the process, $k(\mathbf{y}, \mathbf{y}')$. Measurements of the actual process combined with the prior functions, $(\mu(\mathbf{y}), k(\mathbf{y}, \mathbf{y}'))$ and their so–called hyper–parameters govern the behavior of Gaussian process regression. Using a GP to model signal strength is not new. Indeed, methods have been proposed to use GP-based models of signal strength for localization [24] or to support transfer learning of wireless channel capabilities [26]. We begin by extending prior work that employs GP-based regression to map received signal strength from a fixed source, i.e. in $\mathbb{R}^2$ [22], in order to consider the full point-to-point signal-strength mapping problem in $\mathbb{R}^4$.

We temporarily suspend any prior knowledge of radio signal propagation and set $\mu(\mathbf{y}) = -90$ dBm to coincide with the minimum threshold of the radios in our experimental testbed—making the prior assumption that communication is *not* possible between two arbitrary points. We choose a covariance function that is the sum of two squared exponentials

$$k(\mathbf{y}, \mathbf{y}') = \sigma^2_{k,1} \exp \frac{-d(\mathbf{y}, \mathbf{y}')^2}{2\ell_1^2} + \sigma^2_{k,2} \exp \frac{-d(\mathbf{y}, \mathbf{y}')^2}{2\ell_2^2}. \quad (9)$$

The intent is that one length scale ℓ_1 is longer and represents path–loss components of the process while the other length scale ℓ_2 relates to shadowing due to obstacles in the environment. We rely on a non-Euclidian distance function $d(\cdot, \cdot)$ to model our assumption that channels are symmetric. While it may turn out that this assumption is incorrect, it serves to significantly reduce the sampling burden in the training phase. Thus, we define the distance function between points $\mathbf{y} = (x_s, x_r)$ and $\mathbf{y}' = (x'_s, x'_r)$ as

$$d(\mathbf{y}, \mathbf{y}') = \min\left\{ \left\| \begin{pmatrix} x_s \\ x_r \end{pmatrix} - \begin{pmatrix} x'_s \\ x'_r \end{pmatrix} \right\|, \left\| \begin{pmatrix} x_s \\ x_r \end{pmatrix} - \begin{pmatrix} x'_r \\ x'_s \end{pmatrix} \right\| \right\}. \quad (10)$$

Given our choice of $\mu(\mathbf{y})$ and $k(\mathbf{y}, \mathbf{y}')$, the parameters of the

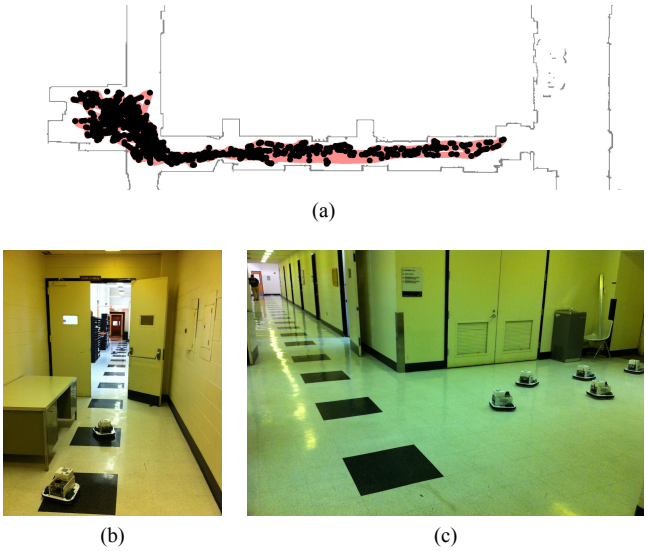


FIGURE 3. RSSI data collection in Towne building. (a) Projection of dataset Y onto $\mathbb{R}^2$ for the Towne building environment. Though measurements are made between pairs of robots in a 6 robot team, i.e., $\mathbb{R}^2 \times \mathbb{R}^2$ we display the measurement locations by projecting onto $\mathbb{R}^2$ and only displaying the location of the receiving node for each measurement. The uniform training set $\mathbf{Y}_T$ is determined by choosing 1,000 of the measurements—displayed here as black points. (b) and (c) depict snapshots from the environment.

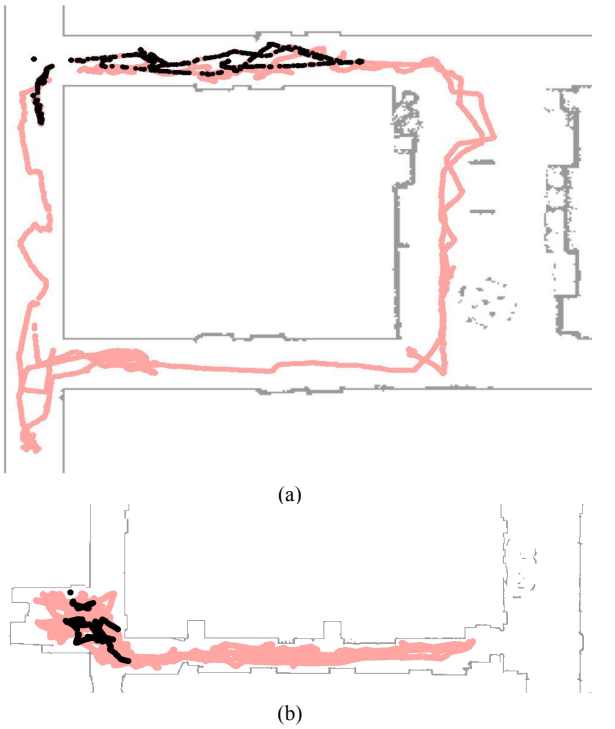


FIGURE 4. Local training locations. The local training set $\mathbf{Y}_T$ is determined for (a) Levine and (b) Towne buildings by choosing 2% of the measurements in a subset of the environment—displayed here as black points.

GP are $(\sigma^2_{\mathcal{F}}, \sigma^2_{k,1}, \ell_1, \sigma^2_{k,2}, \ell_2)$ and the resulting model is

$$P_{R,\mathrm{dBm}}(x_s, x_r) \sim \mathcal{GP}\Big(\mu(\mathbf{y}) + \mathbf{K}_{\mathbf{y},T}\left[\mathbf{K}_T + \sigma^2_{\mathcal{F}}\mathbf{I}\right]^{-1}(\mathbf{z}_T - \mu(\mathbf{Y}_T)), \mathbf{K}_{\mathbf{y}} - \mathbf{K}_{\mathbf{y},T}\left[\mathbf{K}_T + \sigma^2_{\mathcal{F}}\mathbf{I}\right]^{-1}\mathbf{K}_{T,\mathbf{y}} + \sigma^2_{\mathcal{F}}\mathbf{I}\Big). \quad (11)$$

Complete details on the derivation of (11) are available in [22]. While it is feasible to consider a dense sampling of $\mathbb{R}^2$ as a single robot explores the environment, it is impractical to make this assumption for the $\mathbb{R}^2 \times \mathbb{R}^2$ space of all point–to–point links that we are interested in. As a result, large quantities of training data are necessary before this model can make useful predictions. We address this difficulty by incorporating the basic path–loss model used in (8) into the mean function prior. Then,

$$\mu(\mathbf{y}) = L_0 - 10n\log_{10}(\|x_s - x_r\|). \quad (12)$$

Consequently, the covariance function can be simplified to be

$$k(\mathbf{y}, \mathbf{y}') = \sigma^2_{k,1}\exp\frac{-d(\mathbf{y}, \mathbf{y}')^2}{2\ell_1^2} \quad (13)$$

since it only needs to take into account shadowing due to obstacles in the environment. With the introduction of a parameterized prior, the hyperparameters of the GP are $(\sigma^2_{\mathcal{F}}, \sigma^2_{k,1}, \ell_1, L_0, n)$. The posterior for this Gaussian process has the same form as (11) where $\mu(\mathbf{y})$ and $k(\mathbf{y}, \mathbf{y}')$ are redefined in (12), (13). The idea of this model is that once a coarse model for path–loss and small–scale fading has been determined, reasonable predictions can be made anywhere in the space and local deviations are incorporated into the posterior GP by the covariance function.

C. EXPERIMENTAL RESULTS

In order to compare the basic representations or models we have proposed above, we analyze each model's ability to incorporate measurements and then make predictions in a real environment. To demonstrate the generalization of each model, we perform this comparison in two indoor environments of different design and materials on the University of Pennsylvania campus. With construction occurring in 1996 and 2003, the Levine building, depicted in Fig. 2, offers modern construction—interior walls are primarily made up of wood or metal framing with drywall. The Towne building, built in 1903–1906 and depicted in Fig. 3, offers drastically different construction materials including brick and concrete walls.

We collect data about the point–to–point links within a team of n mobile robots as they execute a pre-planned deployment within the desired workspaces as depicted in Figs. 2 and 3. Each robot is capable of self–localization in the environment and communication with its neighbors via a 2.4 GHz Zigbee radio. This means that at each measurement round, the system can collect on the order of n^2 measurements in the desired $\mathbb{R}^2 \times \mathbb{R}^2$ input space. Each robot embeds its current pose x_s in the world–frame into a message packet and broadcasts to its neighbors at a rate of 5 Hz. On arrival, the

TABLE 1. Comparison of average log-likelihoods $\bar{\mathcal{L}}$(M).

	Path–Loss	GP	GP With Path–Loss Prior
Levine uniform $\mathbf{X}_T$	−3.25	−3.14	−3.08
Towne uniform $\mathbf{X}_T$	−3.27	−3.21	−3.21

receiving robot includes its current pose x_r and the received signal strength z and logs the measurement. After an experimental trial, signal strength measurements are aggregated into a single dataset of all measurements $\{\mathbf{Y}, \mathbf{z}\}$. We use a subset of the data $\{\mathbf{Y}_T, \mathbf{z}_T\}$ to train each model. Training entails least–squares parameter estimation for the distance-based methods (8) and normal Gaussian process regression with *ad hoc* selection of maximum likelihood hyperparameters for (11).

The log-likelihood of each model with parameters based on training measurements $\{\mathbf{Y}_T, \mathbf{z}_T\}$ is then computed for the full dataset $\{\mathbf{Y}, \mathbf{z}\}$. To make this a fair comparison between the Gaussian process and parametric models, we make the following standardizations. First, we assume that each measurement is independently distributed. For all models, we note that the received power at a single point can be represented as a normally distributed random variable

$$P_{R,i}(x_s, x_r) \sim \mathcal{N}(\bar{P}_{R,i}(x_s, x_r), \tilde{P}_{R,i}(x_s, x_r)) \tag{14}$$

with mean $\bar{P}_{R,i}(x_s, x_r)$ and variance $\tilde{P}_{R,i}(x_s, x_r)$. Then, we define the likelihood for each model i based on N measurements $\{\mathbf{Y}, \mathbf{z}\}$, training $\{\mathbf{Y}_T, \mathbf{z}_T\}$, and estimated parameters $\hat{\theta}_i$ to be

$$\begin{aligned}\mathcal{L}(\mathrm{M}i) &= \log \mathrm{P}\left[\mathbf{z} \,|\, \mathbf{Y}, \mathbf{Y}_T, \mathbf{z}_T, \mathrm{M}i, \hat{\theta}_i\right] \\ &= -\frac{N}{2}\log 2\pi - \sum_{j=1}^{N}\frac{\log \tilde{P}_{R,i}(x_{s,j}, x_{r,j})}{2} \\ &\quad -\sum_{j=1}^{N}\frac{\left(y_j - \bar{P}_{R,i}(x_{s,j}, x_{r,j})\right)^2}{\tilde{P}_{R,i}(x_{s,j}, x_{r,j})}\end{aligned} \tag{15}$$

where M1 indicates the path–loss model with parameters $\hat{\theta}_1 = (L_0, n, w, \sigma_{\mathcal{F}})$, M2 indicates the GP model with parameters $\hat{\theta}_2 = (\sigma^2_{\mathcal{F}}, \sigma^2_{k,1}, \ell_1, \sigma^2_{k,2}, \ell_2)$ and M3 indicates the GP model with parameters $\hat{\theta}_3 = (\sigma^2_{\mathcal{F}}, \sigma^2_{k,1}, \ell_1, L_0, n)$. To normalize over the number of measurements, we consider the average likelihood $\bar{\mathcal{L}}(\mathrm{M}i) = (1/N)\mathcal{L}(\mathrm{M}i)$. Models with larger likelihood do a better job of explaining the captured measurements. By training on a subset of the data and then computing the likelihood over the entire set of measurements in a given environment, we test for each model's ability to provide predictions throughout the environment.

We perform two sets of analysis on the data. In the first, we choose a uniform subset of data $\{\mathbf{Y}_T, \mathbf{z}_T\}$ to train each of the candidate models in order to provide a best case comparison. Next, we go on to train each model based on a more realistic local training subset that could be collected in the beginning of a deployment into a new environment as depicted in Fig. 4. In depth results on this analysis are beyond the scope of this paper [27, Chapter 4] but we will report several key conclusions.

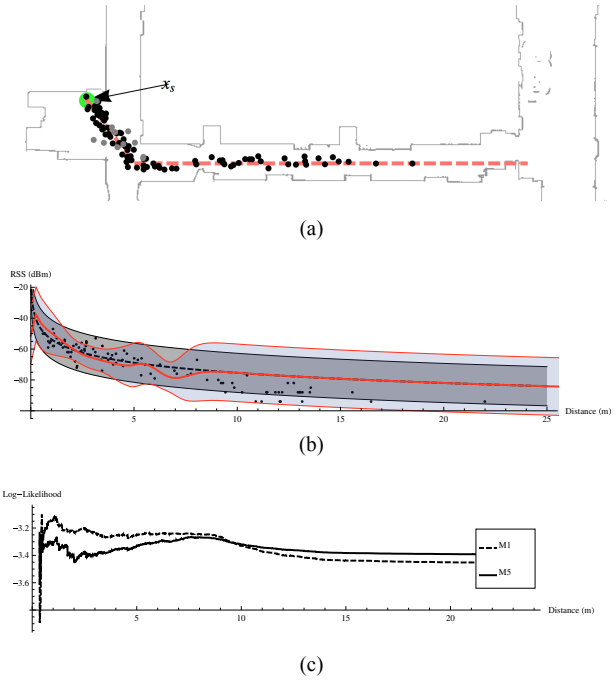


FIGURE 5. Uncertainty increases in unexplored regions for GP estimation [cf. (11)]. (a) Shows sample source location x_s, training measurements (gray dots), measurements used to validate model (black dots), and the line along which predictions are made. (b) Shows predicted output of the GP with the darker dashed line and envelope depicting the prediction from (8) while the solid line with lighter envelope depicts the prediction from the GP in (11). (c) Depicts the evolution of the likelihood of both models (8) and (11) as we consider points increasingly far away from the source and training data.

As expected, increases in model complexity as we move from M1–M3 result in increased model likelihood $\bar{\mathcal{L}}$ as indicated in Table 1. Since we train the models on a subset of the data used for likelihood comparisons, it is unlikely that this improvement is the result of over-fitting. Furthermore, we note that the $\sigma^2_{\mathcal{F}}$ parameter representing fading variance decreases as we increase model complexity. This implies that as the model becomes more expressive, deviations in the underlying process can be explicitly modeled and are not subsumed into the randomness used to model small–scale fading.

A key feature of the GP model in M3 is its representation of increased predictive uncertainty away from measurements used for training. The local training set illustrates this as depicted in Fig. 5 where we train the prior on short range line of sight measurements and then make a series of predictions. Since each prediction is given by a normal distribution, the mean of that distribution matches the performance of (8) but increased variance indicates decreased certainty about the RSSI.

Further note that as we average performance across large regions of each environment the model parameters end up having similar values. Most RSSI variations are subsumed in the noise term used to capture small scale fading. This leads

us to conclude that prediction of RSSI across large operating domains are inherently limited by the inability to model and predict the behavior of scatterers that result in fading. The robust routing methods we introduce in the following section yield probabilistic guarantees despite this uncertainty.

IV. ROBUST ROUTING

The major difficulty in solving (4) is the uncertainty in achievable transmission rates between nearby agents. Assuming that actual channels $R_{ij}(\mathbf{x})$ coincide with their estimates $\bar{R}_{ij}(\mathbf{x})$ may result in a drastic difference between presumed and actual end-to-end rates. A simple way to account for the uncertainty in $R_{ij}(\mathbf{x})$ is to discount $\bar{R}_{ij}(\mathbf{x})$ to reduce the likelihood of having actual rates smaller than the assumed value. A better way to account for this uncertainty is to recall that end-to-end not point-to-point failures are relevant. It is then possible to exploit spatial redundancy to devise robust routes that guarantee small changes in end-to-end rates despite large variability in $R_{ij}(\mathbf{x})$ [28].

To develop robust routing algorithms, start by noticing that computing the probability in (5), which is part of the problem formulation in (6), necessitates modeling the probability distribution of $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$. This is difficult in general. However, if we explicitly consider the stochastic model of point-to-point links via their means and variances, we can compute the mean $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) := \mathbb{E}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x})\right]$ and variance $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) := \text{var}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x})\right]$ of end-to-end rates $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ as

$$\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) = \sum_j \alpha_{ij}^k \bar{R}_{ij}(\mathbf{x}) - \sum_{j \notin \text{dest}(k)} \alpha_{ji}^k \bar{R}_{ji}(\mathbf{x}), \tag{16}$$

$$\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) = \sum_j (\alpha_{ij}^k)^2 \tilde{R}_{ij}(\mathbf{x}) + \sum_{j \notin \text{dest}(k)} (\alpha_{ji}^k)^2 \tilde{R}_{ji}(\mathbf{x}). \tag{17}$$

A reasonable substitution for the probability in (5) is the difference between $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ and its mean $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ normalized by its standard deviation $\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}$. Indeed, for any probability distribution, Chebyshev's inequality implies that having

$$\frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - a_{i,\min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}} \geq \sqrt{\frac{1}{\epsilon}} \tag{18}$$

is a sufficient condition for satisfying (5). Using specific assumptions on the distribution of $R_{ij}(\mathbf{x})$, tighter bounds can be obtained. If, e.g., we assume that $R_{ij}(\mathbf{x})$ has a Gaussian distribution, then (5) is equivalent to

$$\frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - a_{i,\min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}} \geq \Phi^{-1}(\epsilon) \tag{19}$$

where $\Phi^{-1}(\epsilon)$ is the inverse of the normal distribution's cumulative distribution function.

For given positions $\mathbf{x}$, we want to find routing variables $\boldsymbol{\alpha}$ that satisfy (19) or (18)—which are respectively equivalent to (5) or sufficient conditions to satisfy (5). In either case there is some indeterminacy because there may be a non-unique set of variables $\boldsymbol{\alpha}$ that satisfy the corresponding inequality. This indeterminacy provides a degree of freedom that we use to increase reliability beyond the required level. For doing so we introduce a slack variable a_Δ and write an optimization problem that seeks to maximize the minimum $a_{i,\min}^k + a_\Delta$ threshold that can be maintained with probability ϵ.

$$\boldsymbol{\alpha}(\mathbf{x}) = \underset{\boldsymbol{\alpha},\, a_\Delta}{\text{argmax}}\, a_\Delta \quad \text{subject to} \frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - (a_{i,\min}^k + a_\Delta)}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}} \geq \Phi^{-1}(\epsilon). \tag{20}$$

In (20), $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ and $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ are given as in (16) and (17), the routing variables $\boldsymbol{\alpha}$ are further constrained to $0 \leq \alpha_{ij}^k \leq 1$ and $\sum_{j,k} \alpha_{ij}^k \leq 1$, and the probability constraints are required for all i and k.

The original goal as stated in (19) is to find routing variables that make $a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k$ with probability ϵ. The formulation in (20) seeks the best margin a_Δ for which we can have $a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k + a_\Delta$ with probability ϵ. A large value of a_Δ implies that the constraints in (19) are satisfied with significant slack and that there is significant liberty to change the physical configuration without violating communication constraints. By maximizing a_Δ we find routing variables that maximize the flexibility to change the configuration for which $\boldsymbol{\alpha}$ was chosen. This flexibility is instrumental for the mobility control algorithms developed in Sections V and VI.

The optimization problem in (20) can be reformulated as a second order cone program (SOCP) as long as $\epsilon > 0.5$ so that $\Phi^{-1}(\epsilon) > 0$. The route allocation constraints $0 \leq \alpha_{ij}^k \leq 1$ and $\sum_{j,k} \alpha_{ij}^k \leq 1$ are linear in $\boldsymbol{\alpha}$. Further note that the probability constraints in (20) can be rewritten as

$$\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})} \leq \frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}{\Phi^{-1}(\epsilon)} - \frac{a_\Delta}{\Phi^{-1}(\epsilon)} - \frac{a_{i,\min}^k}{\Phi^{-1}(\epsilon)}. \tag{21}$$

Define the $N \times N$ matrix $A_k = (\alpha_{ij}^k)$ and $\boldsymbol{\alpha} = (\text{vec}(A_1), \ldots, \text{vec}(A_K))$. Likewise, define the rate matrix $R = (R_{ij})$ which yields vectors $\bar{\mathbf{r}} = \text{vec}(\bar{R})$ and $\tilde{\mathbf{r}} = \text{vec}(\tilde{R})$ that represent the aggregate rate means and variances respectively. Then, define the matrix $B = \text{diag}\left((\sqrt{\tilde{\mathbf{r}}}, \ldots, \sqrt{\tilde{\mathbf{r}}}, 0)\right)$ with K instances of $\sqrt{\tilde{\mathbf{r}}}$ and the variable $\mathbf{y} = (\boldsymbol{\alpha}, a_\Delta)$. For each node, define the vector $c_i = (\text{vec}(\bar{R} \cdot S), \ldots, \text{vec}(\bar{R} \cdot S), -1)/\Phi^{-1}(\epsilon)$ with K instances of $\text{vec}(\bar{R} \cdot S)$ where $\bar{R} \cdot S$ is the component wise multiplication of $\bar{R}$ with a sign matrix such that $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) = \Phi^{-1}(\epsilon) c_i^{\mathrm{T}} \mathbf{y}$. Finally, define the constant $d_i = -a_{i,min}/\Phi^{-1}(\epsilon)$ and rewrite (21) as

$$\|A\mathbf{y}\| \leq c_i^T \mathbf{y} + d_i. \tag{22}$$

The constraint in (22) defines a second order cone because it constrains the norm of a vector with a linear function. A problem with conic and linear constraints and a linear objective is, by definition, a SOCP. SOCPs are a particular class of convex optimization problems that can be efficiently solved with primal-dual potential reduction or interior point methods [29]. The computational complexity of solving (20) is of order $\text{O}\left((K \cdot N^2)^{3.5}\right)$ where the product $K \cdot N^2$ of the

number of flows K and the square of the number of agents N represents the total number of variables in (20). In practical implementations, the N^2 term can be reduced by eliminating links where R_{ij} is below a certain threshold.

Remark 1: The intuitive backing of robust routing algorithms is to reduce end-to-end uncertainty by exploiting spatial redundancy through traffic splitting among various different routes. We expect to obtain this type of solution from (20). Indeed, increases in a_Δ can be brought about by either increasing the mean $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ or decreasing the variance $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$. Since the mean is a linear function of $\boldsymbol{\alpha}$, traffic splitting has a minor effect on $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$. Since the variance is a quadratic function of $\boldsymbol{\alpha}$, traffic splitting reduces $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ by a factor proportional to the splitting—recall that $\alpha_{ij} \leq 1$. Thus, traffic splitting tends to increase a_Δ because it keeps $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ more or less constant but reduces $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ significantly.

V. LOCAL CONTROL

As per (6), the objective of mobility control is to decrease $\Psi(\mathbf{x})$ while satisfying the probability constraints in (19)—which are equivalent to $\mathrm{P}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k\right] \geq \epsilon$. To check for the feasibility of a configuration $(\boldsymbol{\alpha}, \mathbf{x})$, we define the probability margin as the minimum slack in probability constraints across all flows and sources,

$$\nu(\boldsymbol{\alpha}, \mathbf{x}) := \min_{i,k} \left[\frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - a_{i,\min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}')}} - \Phi^{-1}(\epsilon) \right]. \tag{23}$$

Notice that a necessary and sufficient condition for feasibility of the *physical* configuration $\mathbf{x}$ is to have $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) \geq 0$, with routing variables $\boldsymbol{\alpha}(\mathbf{x})$ as given by the solution of the optimization problem in (20). A sufficient condition for feasibility of physical configuration $\mathbf{x}'$ is the existence of a network configuration $\boldsymbol{\alpha}$ for which $\nu(\boldsymbol{\alpha}, \mathbf{x}') \geq 0$. In particular, for $\mathbf{x}'$ close to $\mathbf{x}$ we expect to have $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}') \geq 0$ since the channel statistics at $\mathbf{x}$ and $\mathbf{x}'$ are close.

In general, local controllers define velocities $\dot{\mathbf{x}}(t)$ proportional to the negative gradients $-\nabla\Psi(\mathbf{x}(t))$ of the task potential. Since the problem in (6) is subject to communication constraints, a local controller shall be based on potential gradients projected onto the feasibile set $\nu(\boldsymbol{\alpha}, \mathbf{x}) \geq 0$. The complex description of the feasible set, however, precludes computation of projected gradients. Instead, we consider the probability margin $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ and modify the potential $\Psi(\mathbf{x})$ by adding the probability margin constraint into the objective through a barrier function,

$$\Omega(\mathbf{x}) := \Psi(\mathbf{x}) - \log\big(\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})\big). \tag{24}$$

Since non-negativity is necessary and sufficient for feasibility of physical configuration $\mathbf{x}$, the potential $\Omega(\mathbf{x})$ in (24) is defined if and only if physical configuration $\mathbf{x}$ is feasible.

The local control law is defined to implement gradient descent on the modified potential $\Omega(\mathbf{x})$ introduced in (24)

$$\mathbf{u}(t) = -\nabla\Psi(\mathbf{x}(t)) + \mu \frac{\nabla_{\mathbf{x}}\nu\big[\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)\big]}{\nu\big[\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)\big]}. \tag{25}$$

The term $\nabla\Psi(\mathbf{x}(t))$ in (25) drives the system to satisfy the task potential. The term $\nabla_{\mathbf{x}}\nu\big[\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)\big]/\nu\big[\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)\big]$ serves as a barrier that drives robots away from configurations for which there is a low probability of exceeding the desired reliability in end-to-end rates. Robots are driven so that $\dot{\mathbf{x}}(t) = \mathbf{u}(t)$ from (25) until we reach a stationary point for which $\mathbf{u}(x(t)) = \mathbf{0}$. By considering the fact that we must have $u(t) = 0$ at the fixed points of the local controller in (25) we can prove that they satisfy one of the two conditions stated in the following proposition.

Proposition 1: Solving (6) by application of (25) and (20) represents a family of algorithms that are parameterized by μ. Each instance will converge to a stationary point $\mathbf{x}_\infty(\mu) := \lim_{t\to\infty} \mathbf{x}(t)$ such that $\mathbf{u}(\mathbf{x}_\infty(\mu)) = 0$. The limit configuration $\mathbf{x}_\infty(\mu)$ satisfies one of the following two statements

(i) The task is accomplished, i.e., $\Psi(\mathbf{x}_\infty(\mu)) = \Psi_{min}$.
(ii) For all configurations $\mathbf{x} \in \mathcal{B}_{\mathbf{x}_\infty(\mu)}$ in the ball

$$\mathcal{B}_{\mathbf{x}_\infty(\mu)} := \left\{ \mathbf{x} : \nu(\boldsymbol{\alpha}, \mathbf{x}) \geq \mu \frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}\big)}{\nabla_x \Psi(\mathbf{x})} \right\} \tag{26}$$

it holds $\Psi(\mathbf{x}_\infty(\mu)) \leq \Psi(\mathbf{x})$.

Proof: The stationary point of the controller, $\mathbf{u}(\mathbf{x}_\infty(\mu)) = 0$, occurs when

$$\nabla_x \Psi(\mathbf{x}_\infty(\mu)) = \mu \frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big)}{\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big)} \tag{27}$$

for all components of $\mathbf{x}$. Case (1) occurs when the system globally minimizes the task potential function and locally maximizes the probability margin so that $\nabla_x\Psi(\mathbf{x}) = 0$, i.e., $\Psi(\mathbf{x}_\infty(\mu)) = \Psi_{min}$ and $\nabla_x\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) = 0$.

Case (2) represents a local minima. Since $\nabla_x\Psi(\mathbf{x}) = 0$ only at the global minima of the task potential function, i.e. case (1) above, the condition of local optimality occurs when (27) is true and $\nabla_x\Psi(\mathbf{x}_\infty(\mu)) > 0$. We can rearrange (27) to be

$$\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big) = \mu \frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big)}{\nabla_x \Psi(\mathbf{x}_\infty(\mu))}$$

for all components of $\mathbf{x}$. Given the definition of $\boldsymbol{\alpha}(\mathbf{x})$ as the unique argument that maximizes the SOCP in (20), it is true that

$$\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big) = \mu \frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big)}{\nabla_x \Psi(\mathbf{x}_\infty(\mu))} \geq \nu\big(\boldsymbol{\alpha}, \mathbf{x}_\infty(\mu)\big)$$

for all $\boldsymbol{\alpha}$ and all components of $\mathbf{x}$. This is a necessary condition for $\mathbf{x}_\infty(\mu)$ to be a stationary point. Therefore a sufficient condition for $\mathbf{x}$ to be a non-stationary point is that

$$\nu\big(\boldsymbol{\alpha}, \mathbf{x}\big) \geq \mu \frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}\big)}{\nabla_x \Psi(\mathbf{x})}. \tag{28}$$

Thus, gradient-based optimization with (25) will converge to a point where (28) doesn't hold true. As a consequence we must have that $\Psi(\mathbf{x}_\infty(\mu)) \leq \Psi(\mathbf{x})$ for all the $\mathbf{x}$ around $\mathbf{x}_\infty(\mu)$ for which (28) holds as we wanted to prove. ■

Algorithm 1 Search for feasible node velocities

Require: Routing solution at $\mathbf{x}(t)$, $\boldsymbol{\alpha}(\mathbf{x})$. Prediction function for link rates, $R(x_i, x_j)$. Desired velocity for each node $\dot{x}_i^{des}$. Initial position of nodes at time t, $\mathbf{x}(t)$. Increment to use when scaling velocities, Δv

```
v_j = 0 for all j ∈ J {Initialize all node velocities to zero}
for k = 0, ..., K do {each flow k}
    Q = Empty queue
    M = Empty set {To store scaled nodes}
    Enqueue(Q, k)
    while Q not empty do
        j = Dequeue(Q)
        u_j = ẋ_j^des / ||ẋ_j^des||
        v_j = ||ẋ_j^des||
        while v_j ≥ 0 and ν(α(x(t)), x(t) + v(t) T) < 0 do
            v_j = v_j − Δv {Scale velocity of node j}
            Add(M, j) {Mark node j as done}
        end while
        for i ∈ {J : α_ij^k > 0} do {each child of j}
            if i ∉ M then
                Enqueue(Q, i)
            end if
        end for
    end while
end for
```

The typical stopping point is case (ii) in Proposition 1. This case corresponds to a condition of local optimality for (24). As $\mu \to 0$, the condition in (26) represents a network configuration that cannot move closer to the goal configuration without violating the network survivability constraint $\nu(\boldsymbol{\alpha}, \mathbf{x}) \geq 0$.

In implementation, the proposed local controller operates in discrete time intervals of duration T. At times $t = kT$ for $k = 1, 2, \ldots$ optimal routes $\boldsymbol{\alpha}(\mathbf{x}(kT))$ are computed as per (20) and control inputs $\mathbf{u}(kT)$ are computed as per (25). Routing variables $\boldsymbol{\alpha}(\mathbf{x}(kT))$ are used to route packets through the network of robots and the physical positions of the robots are updated as

$$x_i((k+1)T) = x_i(kT) + \beta_i T u_i(kT) \tag{29}$$

where constants β_i scale control inputs $u_i(kT)$ to ensure $\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), \mathbf{x}((k+1)T)) \geq 0$. I.e., constants β_i guarantee that routing variables $\boldsymbol{\alpha}(\mathbf{x}(kT))$, optimized for configuration $\mathbf{x}(kT)$, are feasible operating point for configuration $\mathbf{x}((k+1)T)$. The algorithm in Sec. V-A describes a method for finding β_i. Robots are driven as per (29) until we reach a stationary point for which $\mathbf{u}(kT) = \mathbf{0}$.

Remark 2: Computing $\nabla\Psi(\mathbf{x}(t))$ is simple. Computing $\nabla_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ is difficult because: (i) Values of $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ depend on rates $R_{ij}(\mathbf{x})$ that are not known in analytic form but queried from the channel estimators discussed in Section III. (ii) The margin $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ depends on $\boldsymbol{\alpha}(\mathbf{x})$ that is computed as the solution of the SOCP in (20). To approximate $\nabla_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ we consider a finite set of perturbations $\mathbf{X}'$ of the position $\mathbf{x}$ and define $\widehat{\nabla}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ as the average of the finite difference ratios,

$$\widehat{\nabla}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) = \sum_{\mathbf{x}' \in \mathbf{X}'} \frac{\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) - \nu(\boldsymbol{\alpha}(\mathbf{x}'), \mathbf{x}')}{\|\mathbf{x} - \mathbf{x}'\|}(\mathbf{x} - \mathbf{x}'). \tag{30}$$

To compute $\widehat{\nabla}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ in (30) it is necessary to solve the SOCP in (20) for all positions $\mathbf{x}' \in \mathbf{X}'$. To reduce computational cost we define a simplified gradient estimate $\widehat{\nabla}^{(s)}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ as

$$\widehat{\nabla}^{(s)}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) = \sum_{\mathbf{x}' \in \mathbf{X}'} \frac{\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) - \nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}')}{\|\mathbf{x} - \mathbf{x}'\|}(\mathbf{x} - \mathbf{x}'), \tag{31}$$

where the routing solution $\boldsymbol{\alpha}(\mathbf{x})$ is used as an approximation to $\boldsymbol{\alpha}(\mathbf{x}')$. Though the lower compuatational cost of $\widehat{\nabla}^{(s)}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ facilitates real-time implementation, use of the gradient approximation $\widehat{\nabla}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ will yield final configurations with lower task potential.

A. VELOCITY SEARCH

Constants β_i in (29) are chosen so that the predicted probability margin $\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), \mathbf{x}((k+1)T))$ after time horizon T remains positive. A simple solution is to uniformly chose all $\beta_i = \beta$ such that

$$\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), \mathbf{x}(kT) + T(\beta\dot{\mathbf{u}}(t))) \geq 0. \tag{32}$$

A suitable value for β can be found through line search algorithms. However, values of β tend to be prohibitively conservative. To see this consider the case where one link in the communication network is utilized such that any decrease in its reliability will invalidate $\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), \mathbf{x}(kT) + \delta) \geq 0$. Uniform scaling would force $\beta = 0$ and stop the motion of all nodes.

A method to select β_i separately is shown in Algorithm 1. We consider the nodes participating in each flow k according to the tree induced by the routing solution $\boldsymbol{\alpha}(\mathbf{x}(t))$ in the following breadth-first fashion. Starting at dest(k) for the first flow, we set $\beta_{\text{dest}(k)} = 1$ and then scale β_i for all nodes $i \in \{j : \alpha^k_{j,\text{dest}(k)} > 0\}$, *i.e.* those sending data directly to dest(k), so that

$$\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), x_i(kT) + \beta_i u_i(kT)) \geq 0. \tag{33}$$

As above, the search for each β_i is a line search. However, after scaling β_i for immediate neighbors of dest(k), the algorithm procedes with the 2-hop neighbors and so forth in a breadth-first search facilitated with a first in, first out queue.

B. SIMULATION & EXPERIMENTAL RESULTS

We implement the local controller with position updates as in (29), mobility control inputs given by (25), communication variables obtained from (20), and using the backtracking search in Algorithm 1. Computing controls based on local optimization of the network-level end-to-end rates allows for a method of realizing team deployment while maintaining the necessary level of network connectivity. Fig. 6 depicts an example deployment with three robots for a time-varying task potential

$$\Psi(\mathbf{x}(t)) = \begin{cases} x_{2,goal} = (4, 0) & t < 40 \\ x_{2,goal} = (8, 0) & t < 60 \\ x_{2,goal} = (6, 5) & t \geq 60. \end{cases} \tag{34}$$

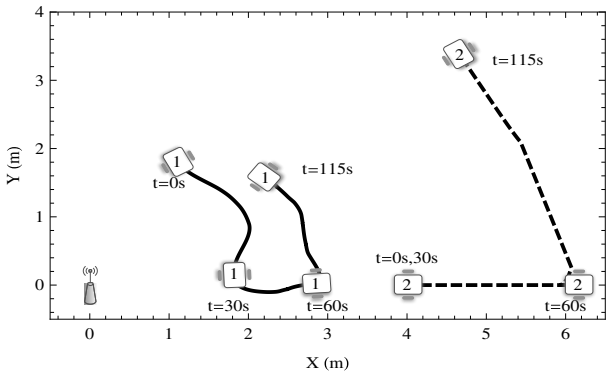


FIGURE 6. Deployment via local control law (25) for a system with a fixed access point, relay node x_1 and lead node x_2 which is controlled by a time-varying task potential $\Psi(\mathbf{x}(t))$.

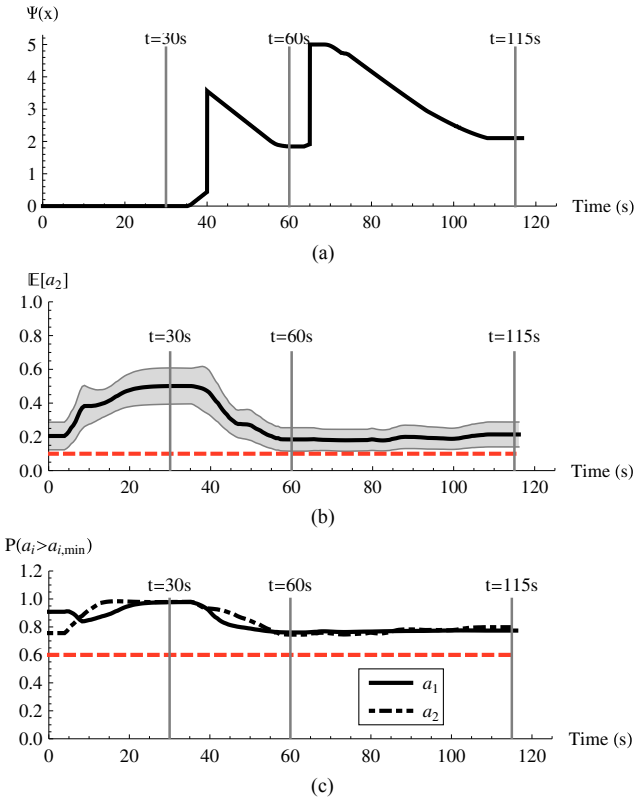


FIGURE 7. Performance of the local control law (25) demonstrating convergence of $\Psi(\mathbf{x}(t))$ in (a), the maintenance of expected end-to-end rate greater than the threshold of $a_{2,min} = 0.1$ in (b), and the $P(a_i \geq a_{i,min}) > 0.6$. The envelope surrounding $\mathbb{E}[a_2]$ in (b) depicts the 60% confidence interval for realizations of the end-to-end rate with stochastic $R_{ij}(\mathbf{x})$.

However, it is also interesting to observe that when task potential is minimized, *e.g.* $t \leq 30$ s, the local control law (25) maximizes the probability of each end-to-end rate exceeding its minimum threshold. When the task potential switches so that $\Psi(\mathbf{x})$ is no longer minimized, the probability margin is reduced so that the primary objective, minimization of $\Psi(\mathbf{x})$, is prioritized. Finally, Fig. 7(b) depicts the end-to-end rate of the node x_2 that must remain above $a_{2,min} = 0.1$. Remember that $R_{ij}(\mathbf{x})$ is a stochastic rate that affects the end-to-end rate. The envelope around $\bar{a}_2$ in Fig. 7(b) depict the effect that different realizations of communication channels $R_{ij}(\mathbf{x})$ will have on the end-to-end rate. Since the pursuit of minimization on $\Psi(\mathbf{x})$ is constrained to have a probability margin $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) > 0$, the end-to-end rate exceeds its threshold in the presence of deviations to $R_{ij}(\mathbf{x})$.

Most importantly, this example demonstrates convergence of the task potential $\Psi(\mathbf{x})$ while maintaining $P(a_i \geq a_{i,min}) > \epsilon$ as depicted in Fig. 7(a) and (c).

As shown through simulation in [1], the addition of a second relay robot introduces local maxima to the potential function (24). Fig. 8 depicts a similar experimental trial now with physical robots and real radio measurements—see Section VII for a description of the experimental platform. In this trial, the task potential function is $\Psi(\mathbf{x}) = \|x_3 - (4, 0)\|$. The scaling of velocities through selection of β_i values is as described above with an additional requirement that inter-robot collisions be avoided, i.e. Algorithm 1 must ensure that $\|x_i - x_j\| < R$ for all i, j of $\mathbf{x}(t + T) = \mathbf{x}(t) + \mathbf{v}(t)T$.

At $t = 80$ s, the control law (25) has reached a stationary point that does not allow the task potential function to be globally minimized as depicted in Fig 9(a). This is due in part because of the inter-robot collision avoidance as well as the local maxima that are introduced to the potential function (24) as in [1]. Throughout the trial, the controller maintains the probabilistic guarantee for end-to-end communication of the lead robot as depicted in Fig. 9(b). This is further reflected in the measurements of actual achievable end-to-end rates as depicted in Fig. 9(c).

In general, as the number of agents increases, the frequency of local minima in $-\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ becomes more and more of an issue for local control. The addition of obstacles into the environment and considerations for robots of physical size add further difficulties as they not only affect feasible $\mathbf{x}$ due to collision constraints, but also introduce non-smooth components in the underlying point-to-point communication links $R_{ij}(\mathbf{x})$.

VI. GLOBAL PLANNING

Local control drives the system towards local minima of (24). With larger teams and more complex network topologies and environments, local minima become inefficient. To succeed in high-level situational awareness tasks, a global search of (6) is necessary. To implement this global search we recast (6) in terms amenable to application of motion planning algorithms.

Let X be a bounded connected open subset of $\mathbb{R}^{2N}$ representing the joint state space for the team of robots with $\mathbf{x}_{init} \in X$ denoting the initial configuration of the team. The team's goal is to reach the configuration space region $X_g = \{\mathbf{x} : \Psi(\mathbf{x}) < \Psi_{min} + \delta\} \subset X$. In a telepresence application where a lead agent must visit $x_{\ell,g}$, we set $X_g = \{\mathbf{x} : \|x_\ell - x_{\ell,g}\| < \delta\}$ which is the set of configurations that minimizes the potential $\Psi(\mathbf{x}) = \|x_\ell - x_{\ell,g}\|^2$ within δ

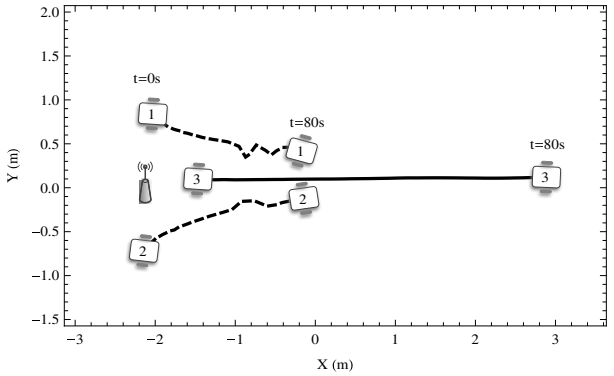


FIGURE 8. Deployment via local control law (25) for a system with a fixed access point, relay nodes x_1, x_2, and lead node x_3 which is controlled by a task potential $\Psi(\mathbf{x}(t)) = \|x_3 - (4, 0)\|$.

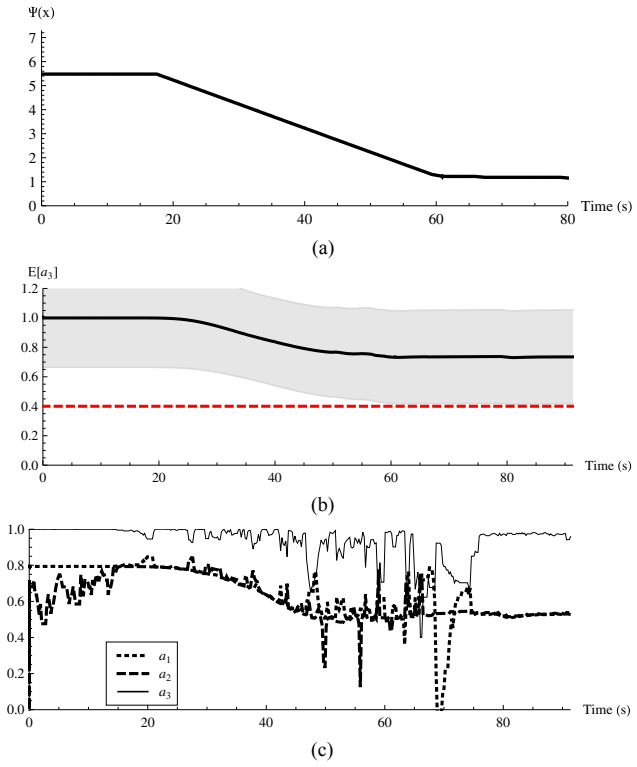


FIGURE 9. The (a) convergence of $\Psi(\mathbf{x})$, (b) expected end-to-end rate of the leader, and (c) actual rates of all agents for the four robot trial depicted in Fig. 8. Note that in (c), a_1 and a_2 must remain above 0 while a_3 must remain above 0.4 with probability 0.8.

tolerance. We further define the physical obstacle region X_{obs} consisting of those configurations that place any given robot on a physical obstacle and the communication infeasibility region X_{inf} as those configurations for which it is impossible to satisfy the network survivability constraint in (19),

$$X_{inf} = \left\{ \mathbf{x} : \frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - a_{i,min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}} \geq \Phi^{-1}(\epsilon), \right.$$
$$\left. \forall\, \boldsymbol{\alpha} \in \left\{ \alpha_{ij}^k : 0 \leq \alpha_{ij}^k \leq 1, \sum_{j,k} \alpha_{ij}^k \leq 1 \right\} \right\}. \quad (35)$$

Algorithm 2 Rapidly exploring random tree algorithm

Require: Initial state $\mathbf{x}_0$, goal region X_g, representation of the bounded configuration space X.

$\mathcal{T}.init(\mathbf{x}_0)$
while $i < N$ **do**
  $\hat{\mathbf{x}} \leftarrow$ RandomState$(X, \mathcal{T})$
  $\mathbf{x}_{\min} \leftarrow$ Nearest$(\mathcal{T}, \hat{\mathbf{x}})$
  if $\mathbf{x}_{new} \leftarrow$ Extend$(\mathbf{x}_{\min}, \hat{\mathbf{x}})$ **then**
    $\mathcal{T}$.add_vertex$(\mathbf{x}_{new})$
    $\mathcal{T}$.add_edge$(\mathbf{x}_{\min}, \mathbf{x}_{new})$
    if $\mathbf{x}_{new} \in X_g$ **then**
      return $\mathcal{T}$
    end if
  end if
end while
return $\mathcal{T}$

The joint obstacle region is given by the union $X_{obs} \cup X_{inf}$ of the physical obstacle region and the virtual obstacle region defined by the network survivability constraints. We want to navigate the free space $X_{free} := X \setminus (X_{obs} \cup X_{int})$ defined as the complement of the joint obstacle region. We also define a path in X as a continuous vector function $\boldsymbol{\sigma} : [0, s] \rightarrow X$ from the interval $[0, s]$ to the joint configuration space X. Concatenation of paths is defined as $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1 | \boldsymbol{\sigma}_2$. Solving the global planning problem is then tantamount to finding a path $\boldsymbol{\sigma} : [0, s] \rightarrow X_{free}$ such that $\boldsymbol{\sigma}(0) = \mathbf{x}_{init}$ and $\boldsymbol{\sigma}(s) \in X_g$.

The dimensionality of our problem and the high computational cost of verifying a state is not part of X_{inf} makes deterministic search algorithms, such as A^* and its variants, impractical. Trajectory optimization methods such as CHOMP [30] and STOMP [31] are inapplicable since our problem statement yields infinite cost and does not permit gradient calculations from inadmissible states. Instead we turn towards the large class of probabilistic motion planners. Generally speaking, probabilistic roadmap methods rely on randomly sampling from the space of admissible configurations X_{free} and then finding feasible paths between states. This approach will perform poorly in our problem space since X_{free} can be thought of as a very thin slice of X—this is a common limitation of probabilistic roadmap methods [32].

We therefore turn to the use of the rapidly exploring random tree (RRT) algorithm [33]. This probabilistic search method offers good space filling properties and efficient exploration of an unknown space. The basic structure of a RRT, as detailed in Algorithm 2, is to start with an initial point $\mathbf{x}_0$ and expand to explore the workspace, adding states in a tree-like structure until a feasible point is added such that $\mathbf{x} \in X_g$. At each step of the RRT algorithm we pick a random state $\hat{\mathbf{x}} =$ RandomState$(X, \mathcal{T})$ and select the point $\mathbf{x}_{\min} =$ Nearest$(\mathcal{T}, \hat{\mathbf{x}})$ that is closest to $\hat{\mathbf{x}}$ among those that have already being added to $\mathcal{T}$. We then attempt a virtual drive from $\mathbf{x}_{\min}$ to $\hat{\mathbf{x}}$ using the subroutine $\mathbf{x} =$ Extend$(\mathbf{x}_{\min}, \hat{\mathbf{x}})$. The point $\mathbf{x}$ is the first intersection of this virtual path with the border

of the free space region X_{free}, or, if the border is not reached, the random point $\hat{\mathbf{x}}$. The point $\mathbf{x}$ is then added to the tree $\mathcal{T}$ as a branch connected to the point $\mathbf{x}_{\min}$ that was closest to $\hat{\mathbf{x}}$ among the preexisting elements of the tree. The algorithm terminates when a point $\mathbf{x} \in X_g$ in the goal configuration is added to the tree.

A common problem encountered when applying RRT algorithms to high-dimensional state spaces is that computation of $\textsc{Nearest}(\mathcal{T}, \hat{\mathbf{x}})$ is inefficient for increasing tree sizes. We adopt the strategy of storing the tree $\mathcal{T}$ in a KD-tree data structure which stores states in $\mathbb{R}^d$ by recursively subdividing based on alternating axis-aligned hyperplanes [34]. This enables approximate nearest neighbor calculations that maintain performance even as the dimension increases. The subroutines $\textsc{Extend}(\mathbf{x}_{\min}, \hat{\mathbf{x}})$ and $\textsc{RandomState}(X, \mathcal{T})$ present challenges that are specific to the high dimensional network connectivity problem in (6). The verification of feasible states as $\textsc{Extend}(\mathbf{x}_{\min}, \hat{\mathbf{x}})$ is used to expand the tree towards $\hat{\mathbf{x}}$ is costly as it requires the solution of the SOCPs in (20). The cost of uniformly exploring X_{free} in the subroutine $\textsc{RandomState}(X, \mathcal{T})$ is prohibitive because the infeasibility space X_{inf} has a complex geometry in $\mathbb{R}^{2N}$ that cannot be decomposed into cartesian products of obstacles in $\mathbb{R}^2$ as is common in navigation problems. We discuss these two issues in the following two sections.

A. EFFICIENT VERIFICATION OF FEASIBLE STATES

The $\textsc{Extend}(\mathbf{x}_{from}, \mathbf{x}_{to})$ algorithm attempts to virtually drive the system from $\mathbf{x}_{from}$ towards $\mathbf{x}_{to}$ by successively verifying that points along the line connecting $\mathbf{x}_{from}$ and $\mathbf{x}_{to}$ are in X_{free}. It returns the state $\mathbf{x}_{new}$ as the closest state to $\mathbf{x}_{to}$ such that all states sampled with precision $\Delta\mathbf{x}$ between $\mathbf{x}_{from}$ and $\mathbf{x}_{new}$ are in X_{free}. In traditional motion planning applications, verification that $\mathbf{x} \in X_{free}$ is based on an algebraic constraint or collision query with a multitude of efficient methods for doing so [35]–[37]. While the necessary computation to determine $\mathbf{x} \notin X_{obs}$ is typically small, computation of $\mathbf{x} \notin X_{inf}$ requires a solution of the SOCP (20) and can be costly.

Consequently, we store $\boldsymbol{\alpha}(\mathbf{x})$ for every node in $\mathcal{T}$ and reuse methods from Section V to extend new states. By relying on the fact that an optimal robust routing solution $\boldsymbol{\alpha}(\mathbf{x})$ will be feasible for neighboring states, it is often possible to extend $\mathbf{x}$ towards $\hat{\mathbf{x}}$ without the costly overhead of numerical optimization as detailed in Algorithm 3. After initialization in lines 1–3, the algorithm proceeds by evaluating the network feasibility at successive team configurations $\mathbf{x}_{new} = \mathbf{x} + \Delta\mathbf{x}$ with the current robust routing solution $\boldsymbol{\alpha}(\mathbf{x})$. Often, the system can take several steps, i.e. lines 5–6, before the networking constraints are violated as indicated by the rate margin $\nu(\boldsymbol{\alpha}, \mathbf{x}_{new}) \leq 0$. When this does occur, a new optimal robust routing solution is computed from (20) as in lines 8–9.

Algorithm 3 $\textsc{Extend}(\mathbf{x}_{from}, \mathbf{x}_{to})$

Require: Initial state $\mathbf{x}_{from}$, desired final state $\mathbf{x}_{to}$, verify segment over K steps

1: $\mathbf{x}_{new} \leftarrow \mathbf{x} \leftarrow \mathbf{x}_{from}$
2: $\boldsymbol{\alpha} \leftarrow$ argmax (20) for rates R_{ij} in configuration $\mathbf{x}_{new}$.
3: $\Delta\mathbf{x} \leftarrow (x_{to} - x_{from})/K$
4: **while** $\mathbf{x}_{new} \neq \mathbf{x}_{to}$ **and** $\boldsymbol{\alpha} \neq \emptyset$ **do**
5: $\quad\mathbf{x}_{new} \leftarrow \mathbf{x}$
6: $\quad\mathbf{x} \leftarrow \mathbf{x}_{new} + \Delta\mathbf{x}$
7: $\quad$**if** $\nu(\boldsymbol{\alpha}, \mathbf{x}_{new}) \leq 0$ **then**
8: $\quad\quad$ {Recompute $\boldsymbol{\alpha}$ if the probability margin at $\mathbf{x}_{new}$ is negative}
9: $\quad\quad\boldsymbol{\alpha} \leftarrow$ argmax (20) for rates R_{ij} in configuration $\mathbf{x}_{new}$.
10: $\quad$**end if**
11: **end while**
12: **if** $\mathbf{x}_{new} = \mathbf{x}_{from}$ **then**
13: $\quad$**return** $\emptyset$
14: **else**
15: $\quad$**return** $\mathbf{x}_{new}$
16: **end if**

B. BIASED SPACE SAMPLING

Random states $\hat{\mathbf{x}}$ are chosen to sample the space $X \subset \mathbb{R}^{2N}$ according to a probability distribution $p_{\mathbf{x}}(\mathbf{x})$ representing the belief about configuration $\mathbf{x}$ being part of a feasible path $\boldsymbol{\sigma}(s)$. If nothing is known about $\boldsymbol{\sigma}(s)$, we choose $p_{\mathbf{x}}(\mathbf{x})$ uniform in the space X; i.e., we make $p_{\mathbf{x}}(\mathbf{x}) = 1/v(X)\mathbb{I}\{\mathbf{x} \in X\}$ where $\mathbb{I}\{\mathbf{x} \in X\}$ denotes the indicator function of the set X and $v(X)$ the volume of set X. In general, at least the final configuration is known in that $\boldsymbol{\sigma}(s) \in X_g$. We can then bias the distribution towards X_g by making

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{p_g}{v(X_g)}\mathbb{I}\left\{\mathbf{x} \in X_g\right\} + \frac{1-p_g}{v(X \setminus X_g)}\mathbb{I}\left\{\mathbf{x} \notin X_g\right\}. \quad (36)$$

When $p_g = v(X_g)/v(X)$ the distribution in (36) corresponds to uniform sampling. Larger values of p_g make $\hat{\mathbf{x}}$ more likely to hit X_g than what corresponds to its volume $v(X_g)$. Goal biasing as in (36) improves efficiency of RRT algorithms by reducing the number of samples necessary to find a feasible path $\boldsymbol{\sigma}(s)$ in the high dimensional space $X \subset \mathbb{R}^{2N}$.

In some cases of interest, the volume of X_g is comparable to the volume of X. In these cases, goal biasing offers little improvement over uniform sampling. In, e.g., the telepresence application where $\Psi(\mathbf{x}) = \|x_\ell - x_{\ell,g}\|^2$ the goal position of the leader $x_{\ell,g}$ is known, but the positions of the remaining robots are free. Thus, goal biasing would reduce the exploration cost along the components associated with x_ℓ but keep the cost of exploring the remaining $2(N-1)$ dimensions fixed. To further reduce exploration cost in this case we construct a prediction $\tilde{X}_g \subset X_g$ of the final configuration and bias sampling towards this configuration prediction by making the sampling distribution

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{p_g}{v(\tilde{X}_g)}\mathbb{I}\left\{\mathbf{x} \in \tilde{X}_g\right\} + \frac{1-p_g}{v(X \setminus \tilde{X}_g)}\mathbb{I}\left\{\mathbf{x} \notin \tilde{X}_g\right\}. \quad (37)$$

Constructing a final configuration prediction $\tilde{X}_g$ is task-specific. We describe here a method applicable to the telepresence application. To determine the configuration prediction $\tilde{X}_g$ we determine configuration predictions $\tilde{X}_{i,g}$ for each robot

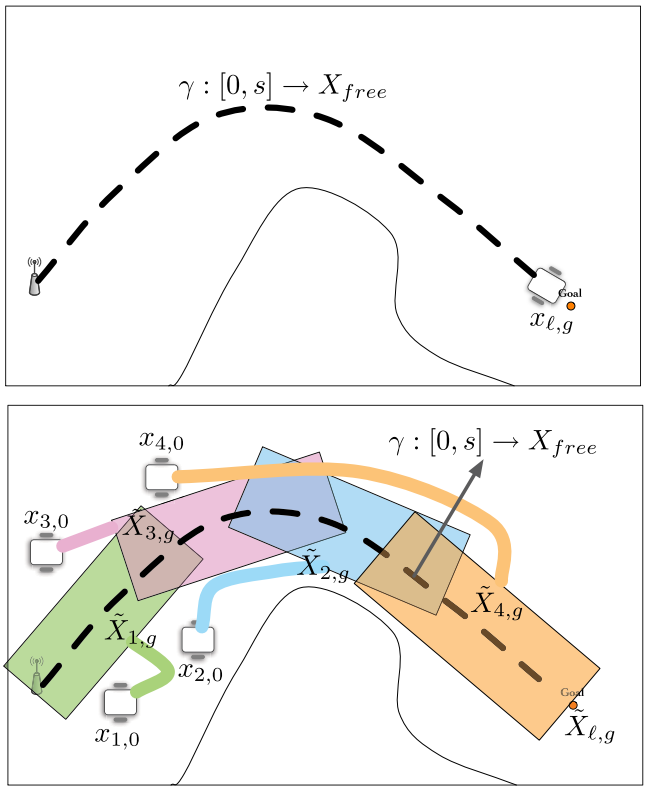


FIGURE 10. **Biased space sampling. Since we only know one component of the goal state $x_{g,\ell}$ and it is expensive to expand our search space in the high-dimensional state of the entire system, it is beneficial to bias our search towards configurations that are deemed likely to succeed. In (a), an obstacle-free path $\gamma : [0, s] \to \mathbb{R}^2$ is found between the access point and goal location. In (b), the path is divided into $N - 1$ segments and enlarged to represent a class of possible goal configurations $\tilde{X}_{i,g}$.**

and compute $\tilde{X}_g$ as the Cartesian product of these individual sets, i.e., $\tilde{X}_g = \prod_{i=1}^{N} \tilde{X}_{i,g}$. Notice that for the lead robot we can make $\tilde{X}_{\ell,g} = X_{\ell,g} = \{x_\ell \in \mathbb{R}^2 : \|x_\ell - x_{\ell,g}\| < \delta\}$.

Observe now that $X \subset \mathbb{R}^{2N}$ is the Cartesian product $X = \prod_{i=1}^{N} X_i$ of the N decoupled spaces $X_i \in \mathbb{R}^2$ corresponding to each individual robot. If we further assume a homogeneous team of robots then all robots operate in the same space $X_i = Y$, with a common set of physical obstacles Y_{obs}, and consequently a common free space $Y_{free} = Y \setminus Y_{obs}$. It follows that the joint free space X_{free} is also a Cartesian product of N identical sets Y_{free} minus those configurations for which a network cannot be established with sufficient reliability,

$$X_{free} = \left(Y_{free}\right)^N \setminus X_{inf}. \tag{38}$$

While infeasible network configurations are captured by X_{inf} as given in (35), X_{free} can otherwise be described by the free space of individual robots.

To exploit this observation, we first determine an obstacle free path $\gamma : [0, s] \to \mathbb{R}^2$ such that $\gamma(0) = x_0$ is the position of the operating center and $\gamma(s) \in X_{\ell,g}$; see Fig. 10(a). This path can be determined by a RRT algorithm [38]. Since the dimensionality of the space and the goal set $X_{\ell,g}$ are small, an RRT algorithm biased towards the goal set finds this path

Algorithm 4 RANDOMSTATE(X)

Require: Configuration space description X, obstacle-free path $\gamma(s) \to \mathbb{R}^2$ such that $\gamma(0) = x_0$ and $\gamma(s) = x_{\ell,g}$, probability p_g.

1: $\tilde{X}_{\ell,g} = \{x_\ell \in \mathbb{R}^2 : \|x_\ell - x_{\ell,g}\| < \delta\}$
2: $\tilde{X}_{i,g} \leftarrow \text{Enlarge}\left(\gamma_{k(i)}\right)$
3: $\tilde{X}_g \leftarrow \prod_{i=1}^{N} \tilde{X}_{i,g}$
4: $p \leftarrow \text{Uniform}[0, 1]$
5: **if** $p > p_g$ **then**
6: $\quad \hat{\mathbf{x}} \leftarrow \text{Uniform}(X \setminus \tilde{X}_g)$
7: **else**
8: $\quad \hat{\mathbf{x}} \leftarrow \text{Uniform}(\tilde{X}_g)$
9: **end if**
10: **return** $\hat{\mathbf{x}}$

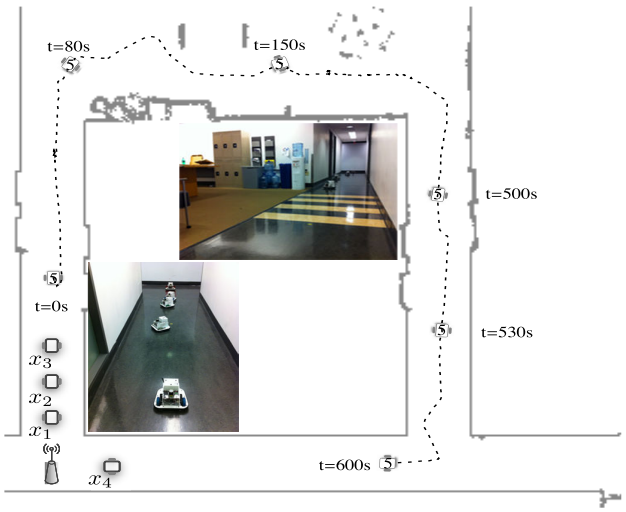


FIGURE 11. **Task specification for Levine experiment. The lead node x_5 follows a sequence of waypoints looping the environment. Initial configuration shown.**

with small computational cost. The obstacle-free path $\gamma : [0, s] \to Y_{free}$ is split into $N - 1$ equal length segments γ_k such that $\gamma_k : [0, s_k] \to \gamma : [ks/(N-1), (k+1)s/(N-1)]$. The ith robot is then assigned to a segment by the function $k(i)$ based on the Euclidian distance to its midpoint such that $\sum_{i \neq 0,\ell} \|\gamma_{k(i)}(s/2) - x_{i,0}\|$ is minimized; see Fig. 10(b). Segments are then enlarged to define the region $\tilde{X}_{i,g}$ for $i \neq 0, \ell$. Since this is heuristic for the goal configuration, the only requirement on $\tilde{X}_{i,g}$ is that $\gamma_{k(i)} : [0, s] \to \tilde{X}_{i,g}$. A typical choice is

$$\tilde{X}_{i,g} = \{x_i : \min_s \|x_i - \gamma_{k(i)}(s)\| < \tilde{d}_g\}$$

where $\tilde{d}_g$ is a parameter controlling the enlarged size of $\tilde{X}_{i,g}$. The predicted final configuration is then computed as the Cartesian product $\tilde{X}_g = \prod_{i=1}^{N} \tilde{X}_{i,g}$.

This procedure is summarized in Algorithm 4. In lines 1–3, the predicted goal configuration $\tilde{X}_g$ is constructed. A random sample $\hat{\mathbf{x}}$ is then drawn uniformly from $\tilde{X}_g$ with probability p_g or from $X \setminus \tilde{X}_g$ otherwise. It should be noted that the construction of $\tilde{X}_g$ described above is based on the

heuristic that a feasible goal configuration in an environment with obstacles will *resemble* a line-of-sight communication chain. Increasing the size of $\tilde{X}_g$ with large values of $\tilde{d}_g$ limits the implication of this assumption.

VII. EXPERIMENTAL RESULTS

We rely on a centralized implementation of the algorithms described in Section VI. Since the algorithms implicitly maintain a connected network of agents, coordinated control commands can be routed through the wireless network. A centralized implementation is possible for navigation with a small number of agents around $N = 10$ or lower since it is tractable to exchange and maintain a global state. All experiments are conducted on the Scarab ground platform at the University of Pennsylvania; see Fig. 11 inset. The robots are capable of accurate self-localization and are equipped with off-the-shelf Zigbee radios. Zigbee uses O-QPSK modulation with MSK pulse shaping. It uses 16 channels spaced at 5 Mhz and has transmission rates of 250 kb/s [39]. The nominal noise figure is $P_{N_0} = -60$ dBm.

Each experiment consists of a telepresence-type task that requires a single lead robot, indexed by ℓ, to visit one or more locations in the environment while maintaining a desired end-to-end communication rate $a_{\ell,min}$ with reliability ϵ to a fixed operating center. The algorithms introduced in this paper yield feasible configurations for the team—$\boldsymbol{\alpha}(t)$ and $\mathbf{x}(t)$ which represent the network and physical configurations respectively. During an experiment, each robot probes the communication channels with its neighbors to determine actual instantaneous measurements of the point-to-point received signal strength at a rate of 5 Hz. This data is logged locally and aggregated after each experiment to compute the supported communication rate $\hat{R}_{ij}(t)$ between node i and j at time t. Using these measurements in conjunction with the network routing solution $\boldsymbol{\alpha}(t)$, we can compute lower bounds on the actual achievable end-to-end rate at time t for each node i, $\hat{a}_i(\boldsymbol{\alpha}(t), \mathbf{x}(t))$.

Recall that the problem statement in (4) requires that $\mathrm{P}\left[a_i(\boldsymbol{\alpha}(t), \mathbf{x}(t)) \geq a_{i,\min}\right] \geq \epsilon$ for all nodes i. Thus, in our experimental verification each node must be able to maintain

$$\hat{a}_i(\boldsymbol{\alpha}(t), \mathbf{x}(t)) > a_{i,min} \tag{39}$$

with probability ϵ. To achieve the desired end-to-end rates, *all* nodes in the team must satisfy this constraint. Thus, in experimental analysis we will evaluate (39) across the duration of the experiment to determine the percent of time $\hat{a}_i(\boldsymbol{\alpha}(t), \mathbf{x}(t)) > a_{i,min}$ and use this as a metric for the success of that trial.

As in Section III, we conduct experimental trials in two buildings of different construction material to demonstrate the generality of our solution. For the first trial in the Levine building, we focus on a task that requires a shift in the basic topology of the network while robustly maintaining communication constraints. This represents a capability that will not emerge from purely reactive local control methodologies. For our second experimental trial in the Towne building, we simplify the topological constraints on the task but increase the difficulty of the scenario by training our predictive communication models on a subset of the environment.

A. LEVINE BUILDING

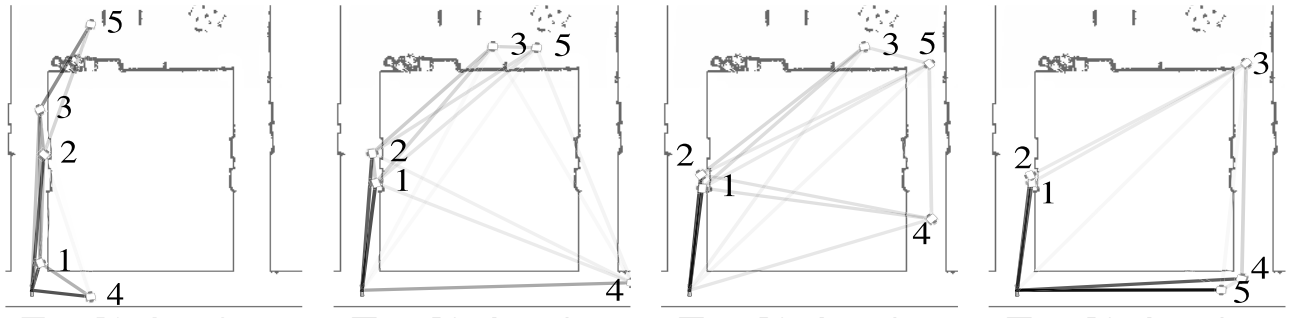
Figure 11 depicts a series of waypoints that the lead node, x_5 must visit. Four additional mobile nodes, x_1, x_2, x_3, x_4 are available to relay data back to the fixed access point indicated in the lower left of Fig. 11 with end-to-end rate of $a_{5,min} = 0.25$ with probability $\epsilon = 0.75$. Each relay node must maintain end-to-end rates greater than zero. The deployment plan is based on a priori channel measurements for model M1 from Section III that yield parameters $L_0 = -51.3$ dBm, $n = 2.07$, $w = -7.58$ dBm, $\sigma^2_{\mathcal{F}} = 31.6$. The predicted and measured end-to-end rates of each node are depicted in Fig. 13.

First, notice that the instantaneous rate $\hat{a}_5(\boldsymbol{\alpha}(t), \mathbf{x}(t))$ is almost always above its minimum threshold of $a_{5,min} = 0.25$. In fact, it drops below the minimum threshold only 2.9% of the time, well within the allowable reliability $\epsilon = 25\%$ for this problem specification. However, for that rate to be maintained in an end-to-end sense across the network, each node must be able to support the necessary rate margin $a_{i,min}$. The corresponding fraction of time spent below the minimum threshold for each of the instantaneous node rates $\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4$ is 9.2%, 0.8%, 0.3%, and 0.6%. This means the desired rates are not supported during, at most, 13.8% of the time.

Representative network configurations are depicted in Fig. 12. In Fig. 12(a), at $t = 100$ s, the predicted goal state $\tilde{\mathbf{X}}_g$ assumes the shortest line of sight path which is the left hallway, i.e. a result similar to what one would expect from reactive methods. As the system transitions to Fig. 12(b), where the lead node x_5 has been tasked to a waypoint in the right hallway, the prediction for $\tilde{\mathbf{X}}_g$ shifts to a chain of relays going through the right hallway. This shift in the basic topology of $\tilde{\mathbf{X}}_g$ refocuses exploration of the joint state space so that x_4 moves towards a configuration that will lower the performance of the network over the short term. As node x_5 completes the desired loop, it utilizes x_4 as a relay channel and is able to maintain the desired end-to-end rate. This dramatic shift in network topology would not be possible with a purely reactive method and illustrates the advantage of deliberative planning.

B. TOWNE BUILDING

Fig. 14(a) shows the environment for this experiment. The simple geometry does not require dramatic shifts in network topology. However, we complicate the problem by training the communication model on the subset of the environment labeled as training region in Fig. 14(a). The deployment plan is based on a priori channel measurements for model M1 from Section III that yield parameters $L_0 = -52.2$ dBm, $n = 1.95$, $w = -3.9$ dBm, $\sigma^2_{\mathcal{F}} = 41.3$. The leader is deployed to the end of the corridor at time $t = 900$ s—see Fig. 14(a)—and then returns to its initial position.



(a) $t = 100$ s (b) $t = 278$ s (c) $t = 482$ s (d) $t = 622$ s

FIGURE 12. Snapshots from the sequence of feasible network configurations that satisfy the task depicted in Fig. 11. Line weight indicates the expected amount of information to be transmitted over that point-to-point link.

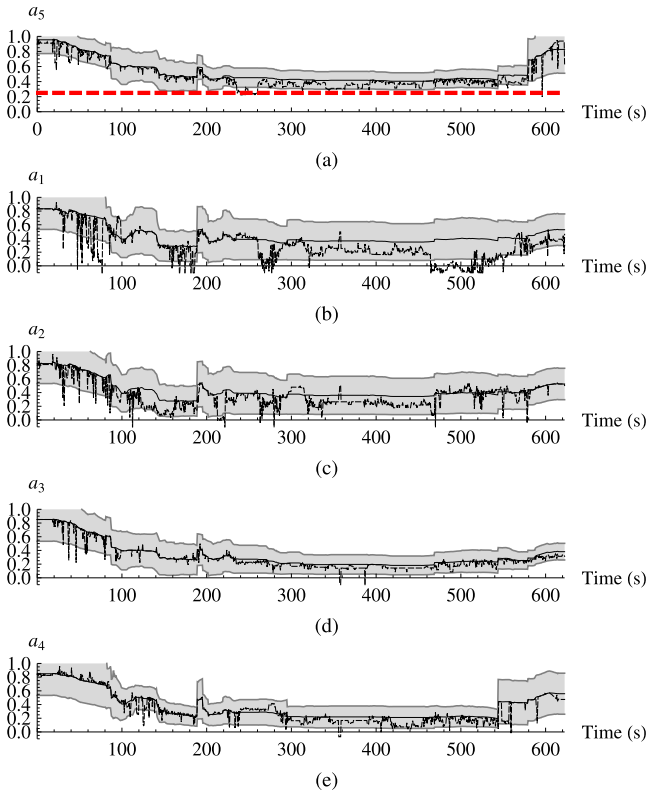


FIGURE 13. The end-to-end rates of the nodes during the Levine building experiment depicted in Fig. 11. (a) depicts the prediction, $\bar{a}_5$, $\tilde{a}_5$, and instantaneous, $\hat{a}_5$, end-to-end rate for the leader and (b)–(e) depict the instantaneous rates of the relay nodes. In each plot, the solid line with shaded envelope depicts $\bar{a}_i$ and variations that occur with probability $\epsilon = 0.75$ based on $\tilde{a}_i$. The dashed black line represents the instantaneous end-to-end rate $\hat{a}_i$. The dashed red link in (a) depicts the threshold $a_{5,min} = 0.25$.

Predicted and measured end-to-end rates of each node are depicted in Fig. 15. Despite the limited training data, the performance in this environment is comparable to the Levine experiment. The empirical failure rate for each node is 8.2%, 3.3%, 17%, 0.5%, and 16.7% for a_1 through a_5 respectively. The actual end-to-end rate, $\hat{a}_5$, for $300 \leq t \leq 900$ s is consistently offset from the prediction since this robot is operating far from the training region. This result is expected based on the communication modeling experiments described in Section III. However, it also demonstrates an important capability of the robust methods we employ since we are still able to maintain communication within the desired parameters when the model systematically overestimates the performance of point-to-point links. The use of a more complex radio communication model, such as the Gaussian process method (11), would incorporate increased uncertainty in this region and require more conservative configurations from the global planner—leading to improved performance.

C. COMPLEXITY BENCHMARKING

With regards to the running time of Algorithm 2, we note that it is difficult to characterize the performance of randomized search algorithms. One factor is the complexity of X_{free} which is determined both by physical obstacles in the environment as well as the constraints placed on feasible network configurations. Another component in determining running time is the planning horizon defined as the number of states that must be expanded in order to find a solution. However, we can study the computational complexity of the proposed approach through a benchmark task that can be solved many times with different problem parameterizations.

We choose the benchmark task, depicted in Fig. 16(a) which requires the lead robot to visit a series of positions in the environment labeled 1–8 while communicating data at a specified rate, a_{min}, to the operating center located near waypoint 1. We parameterize the task by the number of robots N and the end-to-end rate of communication that must be maintained a_{min} while fixing the desired reliability $\epsilon = 0.8$. The performance is measured by the running time to compute the series of network configurations necessary for the lead robot to visit its sequence of waypoints. The average performance is depicted in Fig. 16(b) based on 10 trials per task parameterization.

As expected, increasing the number of robots adds to the complexity of both the individual SOCP solutions as well as the randomized search algorithm. Increasing the minimum end-to-end rate, a_{min}, has a similar effect on the complexity. Intuitively, increased a_{min} increases complexity because it reduces the number of feasible configurations that can

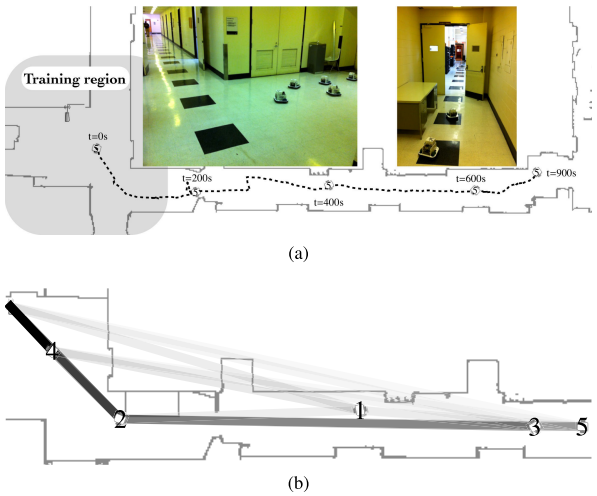


FIGURE 14. (a) Task specification for the Towne building experiment. It requires the lead node, x_5 to follow a sequence of waypoints that take it as far as possible from the fixed operating center. (b) Snapshot from $t = 550$ s of the deployment.

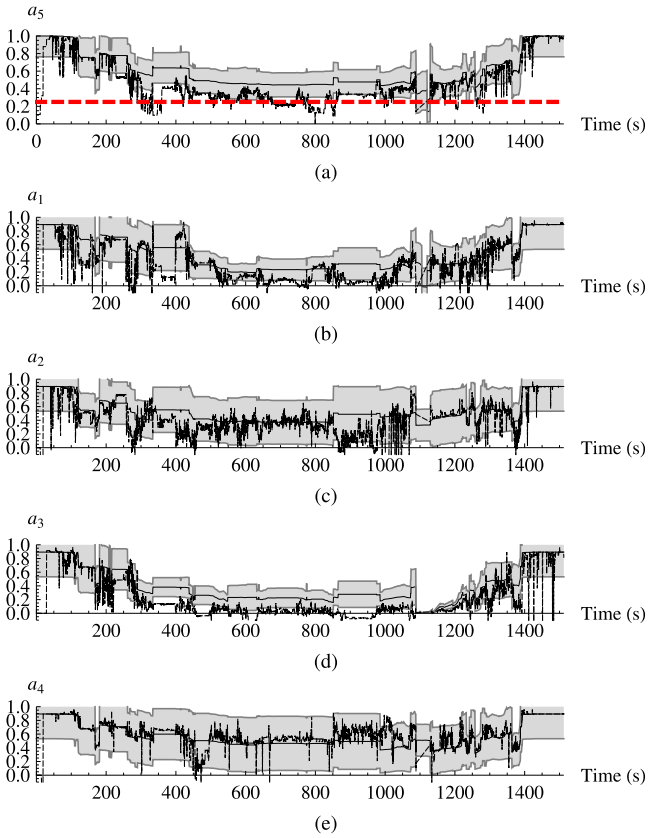


FIGURE 15. End-to-end rate performance during Towne building experiment. Predicted rates $\bar{a}_i$ and corresponding variances $\tilde{a}_i$ along with observed instantaneous rates $\hat{a}_i$ shown for leader robot (a) and relay nodes (b)–(e). Solid lines with shaded envelopes depict $\bar{a}_i$ and variations that occur with probability $\epsilon = 0.75$ based on $\tilde{a}_i$. Dashed lines represent instantaneous end-to-end rates $\hat{a}_i$. The dashed red link in (a) depicts the minimum acceptable rate $a_{5,min} = 0.25$ for the leader robot. Rates exceed minimum performance thresholds most of the time.

support the required rate a_{min}. This increases the planning effort necessary to explore the workspace and find a feasible path.

Randomized planning algorithms can only offer the guarantee of probabilistic completeness. Since there is no precise way to determine when a task cannot be solved with the current configuration, we test for task feasibility by stopping the planning process after a specified timeout period. For the purposes of this benchmarking, that timeout is 300 s for each subtask. An artifact of this timeout is that tasks in extremely complex spaces (e.g. $a_{min} = 0.51, N = 8$) are not solved though we know a solution exists (e.g. the solution for $a_{min} = 0.51, N = 7$ is a subset of the possible solutions with $N = 8$). As the complexity of the task increases, so does the variance of the running time as depicted in Fig. 16(c) for a particular task with $a_{min} = 0.61$.

VIII. COMPARISON WITH EXISTING METHODS

Contributions to communication-aware deployment of robot teams can be classified according to three axes: *communication channel models* such as fixed communication range, line of sight, pathloss, or stochastic; *network integrity metrics* like network-wide connectivity, supported end-to-end bit error rate, or maintenance of desired rate constraints; and *mobility control method* namely local gradient-based methods or global search.

Many of the early approaches to these problems adopt simple channel models that are either binary measures of connectivity or deterministic functions of distance [3]–[6]. By coupling these channel models with graph-theoretic notions of connectivity, it is possible to construct gradient-based controllers that move agents to locally optimize global measures of connectivity in an effort to maintain network integrity. While some gradient-based control methods do incorporate the avoidance of obstacles [7], [40], [41], local minima in the control space due to the environment geometry often limits their ability to maintain connectivity as explicitly demonstrated in [40]. Furthermore, the use of simple channel models and graph-theoretic notions of connectivity makes it impossible to assure the network's ability to support the team's task.

More recent work in this area extends the state of the art with respect to the axes outlined above. While still relying on a binary notion of connectivity, the approach presented in [8] does incorporate notions of deliberative global planning and is shown to operate effectively in realistic environments to provide the existence of a communication network to a mobile user with known or unknown path. Incorporating a rate-based definition of network integrity, the approach in [20] jointly optimizes network routing and mobility to support specific communication requirements based on deterministic channel models. However, the control methods presented do not consider geometric constraints that arrise due to obstacles. Finally, the recent work presented in [21] uses stochastic models for communication channels but assumes a fixed a priori network topology and gradient-based mobility control that incorporates obstacles in the same way as [40] and thus suffers from the same limitations with respect to local minima induced by environment geometry. Jitter is added to the local

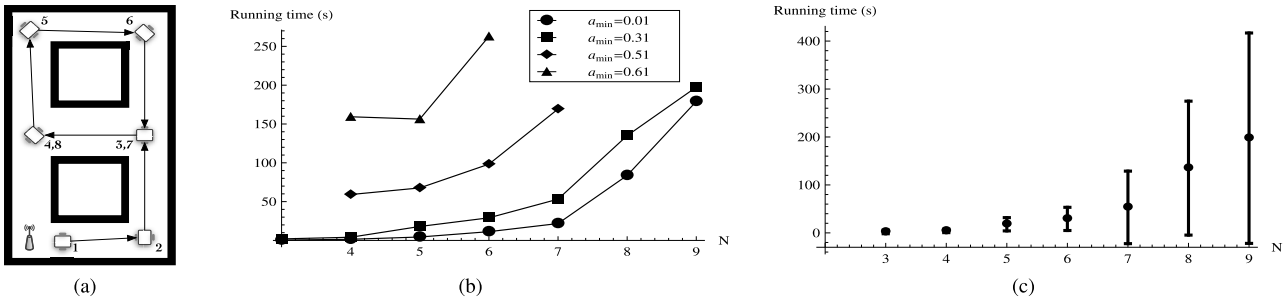


FIGURE 16. Running time for benchmark environment with global planner solving the task depicted in (a). The average running time for tasks with different a_{min} and number of robots N are depicted in (b). The variance of the running time for a particular task is depicted in (c).

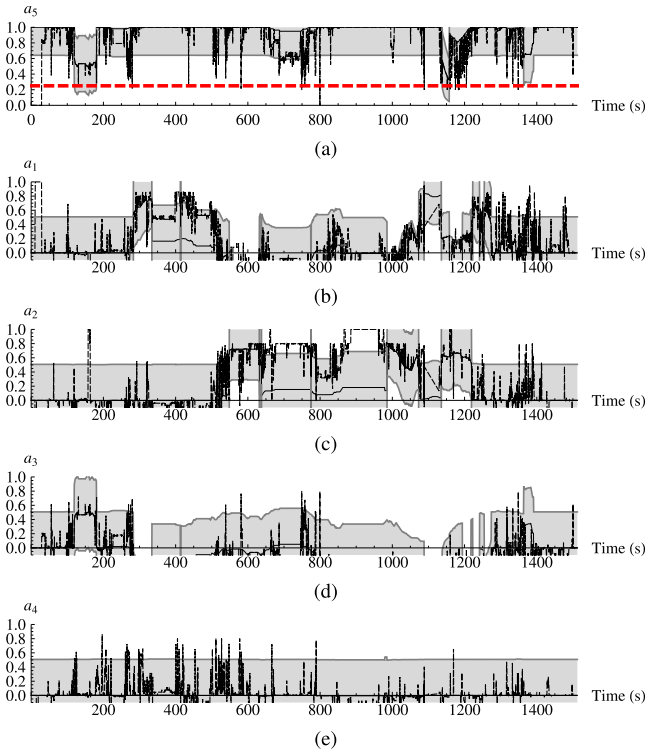


FIGURE 17. The achievable end-to-end rates of the nodes during the Towne building experiment depicted in Fig. 14(a) when utilizing a static multihop routing. (a) depicts the prediction, $\tilde{a}_5$, $\tilde{a}_5$, and instantaneous, $\hat{a}_5$, end-to-end rate for the leader and (b)–(e) depict the instantaneous rates of the relay nodes. In each plot, the solid line with shaded envelope depicts $\tilde{a}_i$ and variations that occur with probability $\epsilon = 0.75$ based on $\tilde{a}_i$. The dashed black line represents the instantaneous end-to-end rate $\hat{a}_i$. The dashed red line in (a) depicts the threshold $a_{5,min} = 0.25$.

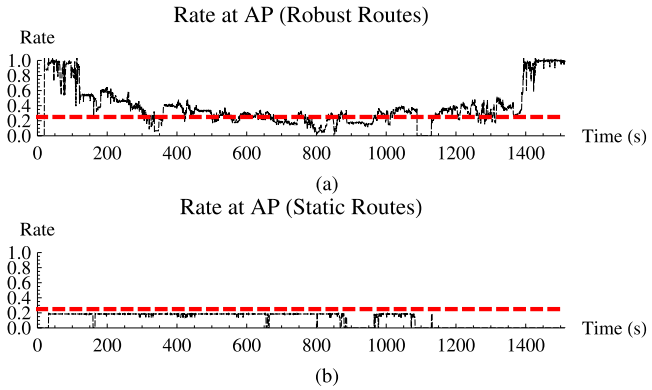


FIGURE 18. Discrete simulation of the routing queue at each node leads to a representation of the actual rate of information flow from node x_5 to the fixed operating center subject to (a) a robust routing protocol or (b) static multihop routing protocol. The dashed red line in these figures depicts the desired end-to-end rate of 0.25.

controllers, but this only aids in escaping small local minima and cannot address loop scenarios as in Fig 11.

In this work we consider stochastic communication models, network–wide rate–based connectivity metrics, and global planning. While the advantages of using global planning and network wide connectivity metrics are clear, the advantage of using stochastic communication models is less so. In particular, the work presented here assumes stochastic channel models but is agnostic to its particular form as models with less accurate channel estimates will just result in more conservative deployment plans. The question remains, however, of what are the advantages of a robust stochastic routing solution. In the remainder of this section we offer some evidence supporting the use of robust routing techniques rather than the more traditional deterministic routes often employed.

We do this by comparing the ability to support an end-to-end flow of information with robust routing approaches versus more traditional fixed multi-hop routing topologies. For the purposes of this comparison, we utilize the experimental deployment illustrated in Fig 14(a) with the associated predictions and experimentally measured wireless communication channels, i.e., the predicted and actual rates R_{ij}. We assume a static multihop topology to simplify analysis where messages are relayed in a chain $x_5 \rightarrow x_3 \rightarrow x_1 \rightarrow x_2 \rightarrow x_4 \rightarrow AP$.

Figure 17 depicts the achievable end-to-end rates of each node during the Towne building experiment described above. Remember that network integrity requires every node to satisfy the constraint on achievable end-to-end rate based on the queue stability argument presented in Section II. It is clear that the end-to-end rates based on static routing in Fig. 17 do not perform as well as those based on robust routing in Fig. 15. It is also important to note that dynamic application of ad-hoc routing protocols may improve the performance in Fig. 17, the robust routes used in Fig. 15 are computed a priori and implemented in an open-loop fashion.

We can also pursue a more generic comparison of the performance under robust routing versus traditional multihop topologies by using wireless channel measurements of instantaneous supported point-to-point rates along with routing allocations to perform a data-driven discrete simulation of the queues at each node. This provides a more tangible comparison of the end-to-end rate from node x_5 to the fixed operating center during the deployment subject to both robust routing and a fixed multihop topology. Figure 18 depicts the resulting rates due to this analysis when we assume a small finite queue size at each node. Notice that while the robust routing solution supports an actual end-to-end rate that exceeds the desired constraint with high probability, the static routing solution is unable to ever maintain this level of communication.

IX. CONCLUSION

We propose a system architecture that provides end-to-end connectivity for teams of mobile robots as they pursue operator–assigned tasks. Because point-to-point wireless communication is uncertain, we adopt a stochastic model for supported rates and develop optimal robust solutions to the wireless routing problem. As nodes move to satisfy their tasks, channel performance degrades and eventually invalidates even robust solutions to the wireless routing problem. We presented a novel approach to maintain network connectivity across a team of mobile agents that takes into account specific thresholds on the end-to-end rates of individual nodes while also considering the stochastic nature of communication channels. By specifically searching for robust solutions, our approach yields configurations with spatial diversity and increased probability of success in the face of difficult to predict communication channels.

As can be expected, local control approaches will converge to solutions that are local minima of the concurrent mobility and network routing problem. As the number of agents increase, these local minima drastically limit the performance of the system. Thus, we propose a randomized global planning approach that attempts to jointly solve the mobility and optimal network routing problems.

We present, to the best of our knowledge, the first example of an experimental verification for a mobile communication maintenance system that relies on an robust end-to-end rate constraint for network integrity. Furthermore, we are able to do this with limited assumptions about the model for point-to-point achievable rates. In fact, our experiments succeed with a coarse rate model that can be applied to a wide range of environments. Finally, our experimental results illustrate the value in pursuing global search methods rather than reactive gradient-based methods in that we are able to find a sequence of network configurations that would not emerge from local optimization of network integrity.

The use of randomized planning techniques implicitly casts our formulation as a feasibility problem. Future work will focus on the incorporation of techniques from gradient-based methods to decrease the time-to-plan and increase the optimality of our solutions.

Another avenue for future work is to further study the relationship between the number of agents, problem feasibility, and planning complexity. The benchmarking results above present an interesting scenario where the *addition* of a node shifts the complexity of the problem in such a way that it is not solved within our time constraints. In general, we maintain that the communication maintenance problem should become easier to solve as we increase the number of relay agents beyond the minimum required for feasibility.

REFERENCES

[1] J. Fink, A. Ribeiro, and V. Kumar, "Motion planning for robust wireless networking," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2419–2426.

[2] J. Fink, A. Ribeiro, and V. Kumar, "Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams," *Proc. IEEE*, vol. 100, no. 1, pp. 164–178, Jan. 2012.

[3] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, Aug. 2007.

[4] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 812–816, Aug. 2007.

[5] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie, "Maintaining limited-range connectivity among second-order agents," in *Proc. Amer. Control Conf.*, Minneapolis, MN, USA, Jun. 2006, pp. 2124–2129.

[6] M. Schuresko and J. Cortes, "Distributed motion constraints for algebraic connectivity of robotic networks," *J. Intell. Robot. Syst.*, vol. 56, nos. 1–2, pp. 99–126, Sep. 2009.

[7] D. P. Spanos and R. M. Murray, "Motion planning with wireless network constraints," in *Proc. Amer. Control Conf.*, Portland, OR, USA, Jun. 2005, pp. 87–92.

[8] O. Tekdas, W. Yang, and V. Isler, "Robotic routers: Algorithms and implementation," *Int. J. Robot. Res.*, vol. 29, no. 1, pp. 110–126, 2010.

[9] I. Stojmenovic, A. Nayak, and J. Kuruvila, "Design guidelines for routing protocols in ad hoc and sensor networks with a realistic physical layer," *IEEE Commun. Mag.*, vol. 43, no. 3, pp. 101–106, Mar. 2005.

[10] A. Neskovic, N. Neskovic, and G. Paunovic, "Modern approaches in modeling of mobile radio systems propagation environment," *IEEE Commun. Surveys Tuts.*, vol. 3, no. 3, pp. 2–12, Nov. 2009.

[11] H. Lundgren, E. Nordstrom, and C. Tschudin, "The gray zone problem in IEEE 802.11b based ad hoc networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 6, no. 3, pp. 104–105, Jul. 2002.

[12] J. Fink, N. Michael, A. Kushleyev, and V. Kumar, "Experimental characterization of radio signal propagation in indoor environments with application to estimation and control," in *Proc. Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 2834–2839.

[13] D. DeCouto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multihop wireless routing," in *Proc. Int. ACM Conf. Mobile Comput. Netw.*, San Diego, CA, USA, Sep. 2006, pp. 134–146.

[14] A. Ribeiro, Z.-Q. Luo, N. D. Sidiropoulos, and G. B. Giannakis, "Modelling and optimization of stochastic routing for wireless multihop networks," in *Proc. 26th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Anchorage, AK, USA, May 2007, pp. 1748–1756.

[15] Y. Mostofi, A. Gonzalez-Ruiz, A. Gaffarkhah, and L. Ding, "Characterization and modeling of wireless channels for networked robotic and control systems—A comprehensive overview," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4849–4854.

[16] M. Malmirchegini and Y. Mostofi, "On the spatial predictabiity of communication channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 964–978, Mar. 2012.

[17] A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Toward the deployment of a mobile robot network with end-to-end performance guarantees," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, May 2006, pp. 2085–2090.

[18] A. Ghaffarkhah and Y. Mostofi, "Communication-aware motion planning in mobile networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2478–2485, Oct. 2011.
[19] G. Hollinger and S. Choudhary, "Communication protocols for underwater data collection using a robotic sensor network," in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, Dec. 2011, pp. 1308–1313.
[20] M. Zavlanos, A. Ribeiro, and G. Pappas, "Mobility and routing control in networks of robots," in *Proc. Conf. Decision Control*, Atlanta, GA, USA, Dec. 2010, pp. 7545–7550.
[21] Y. Yan and Y. Mostofi, "Robotic Router Formation in Realistic Communication Environments," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 810–827, Aug. 2012.
[22] J. Fink and V. Kumar, "Online methods for radio signal mapping with mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 1940–1945.
[23] R. Shorey, A. Ananda, M. C. Chan, and W. T. Ooi, *Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions*. New York, NY, USA: Wiley, 2006.
[24] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 2480–2485.
[25] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[26] N. Wagle and E. W. Frew, "Transfer learning for dynamic RF environments," in *Proc. Amer. Control Conf.*, 2012, pp. 1406–1411.
[27] J. Fink, "Communication for teams of networked robots," Ph.D. dissertation, Dept. Electr. Comput. Syst. Eng., Univ. Pennsylvania, Philadelphia, PA, USA, 2011.
[28] Y. Wu, A. Ribeiro, and G. B. Giannakis, "Robust routing in wireless multi-hop networks," in *Proc. Conf. Inf. Sci. Syst.*, Mar. 2007, pp. 637–642.
[29] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 284, nos. 1–3, pp. 193–228, 1998.
[30] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 489–494.
[31] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4569–4574.
[32] D. Hsu, L. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin, "On finding narrow passages with probabilistic roadmap planners," in *Proc. Int. Workshop Algorithmic Found. Robot.*, 1998, pp. 141–153.
[33] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2000, pp. 995–1001.
[34] A. Atramentov and S. M. LaValle, "Efficient nearest neighbor searching for motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, pp. 632–637.
[35] M. C. Lin, D. Manocha, J. Cohen, and S. Gottschalk, "Collision detection: Algorithms and applications," in *Algorithms for Robotic Motion and Manipulation*, J.-P. Laumond and M. H. Overmars, Eds. Wellesley, MA, USA: A. K. Peters, 1997, pp. 129–142.
[36] B. Mirtich, "V-Clip: Fast and robust polyhedral collision detection," Mitsubishi Electronics Research Laboratory, Cambridge, MA, USA, Tech. Rep. TR97-05, 1997.
[37] M. C. Lin and J. F. Canny, "A fast algorithm for incremental distance calculation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1991, pp. 1008–1014.
[38] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 1–76, 2011.
[39] K. T. Le. (2005). *IEEE 802.15.4 and Zigbee Compliant Radio Transceiver Design* [Online]. Available: http://www.hometoys.com/htinews/feb05/articles/chipcon/zigbee.htm
[40] E. Stump, A. Jadbabaie, and V. Kumar, "Connectivity management in mobile robot teams," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, May 2008, pp. 1525–1530.
[41] P. R. Giordano, A. Franchi, C. Secchi, and H. Bÿlthoff, "Bilateral teleoperation of groups of UAVs with decentralized connectivity maintenance," in *Proc. Robot., Sci. Syst.*, Los Angeles, CA, USA, Jun. 2011, pp. 1–8.

JONATHAN FINK (M'11) is a Research Scientist with the U.S. Army Research Laboratory, Adelphi, MD, USA. He received the B.S. degree in electrical and computer systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2004. From 2005 to 2011, he was with the University of Pennsylvania, Philadelphia, PA, USA, where he received the Ph.D. degree in electrical and systems engineering. His current research interests include collaboration and planning for multi-robot systems with emphasis on communication modeling and planning for network connectivity. He received the Best Student Paper Award at RSS in 2009.

ALEJANDRO RIBEIRO (M'13) is an Assistant Professor with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA, in 2008. He received the B.Sc. degree in electrical engineering from Universidad de la Republica Oriental del Uruguay, Montevideo, in 1998. From 2003 to 2008, he was with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MI, USA, where he received the M.Sc. and Ph.D. degrees in electrical engineering. His current research interests include communication, signal processing, and networking. He is a Fulbright Scholar and a recipient of the National Science Foundation CAREER Award in 2010 and the Best Student Paper Awards at ICASSP in 2005 and ICASSP in 2006.

VIJAY KUMAR (F'08) is the UPS Foundation Professor and the Deputy Dean for Education with the School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA, USA. He received the Ph.D. degree in mechanical engineering from Ohio State University, Columbus, OH, USA, in 1987. He has been the Faculty with the Department of Mechanical Engineering and Applied Mechanics with a secondary appointment in the Department of Computer and Information Science, University of Pennsylvania, since 1987. He is a fellow of the American Society of Mechanical Engineers. He has served on the editorial boards of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, the *Journal of Franklin Institute*, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, the *ASME Journal of Mechanical Design*, the *ASME Journal of Mechanisms and Robotics*, and the *Springer Tracts in Advanced Robotics*. He is a recipient of the 1991 National Science Foundation Presidential Young Investigator Award, the Lindback Award for Distinguished Teaching, and the 1997 Freudenstein Award for significant accomplishments in mechanisms and robotics. He received the Best Paper Awards at DARS in 2002, ICRA in 2004, ICRA in 2008 RSS in 2009, and DARS in 2010. He is a Distinguished Lecturer in the IEEE Robotics and Automation Society and an elected member of the Robotics and Automation Society Administrative Committee.