

SYSTEM ARCHITECTURES FOR COMMUNICATION-AWARE MULTI-ROBOT NAVIGATION

James Stephan[†] Jonathan Fink[‡] Alejandro Ribeiro[†]

[†]Department of Electrical and Systems Engineering, University of Pennsylvania

[‡]US Army Research Laboratory

ABSTRACT

In this paper, we present a hybrid system architecture that enables a team of robots to self-organize into a multi-hop ad-hoc network allowing for the completion of a given task while providing the desired end-to-end data rates between designated robots. This architecture consists of a two stage feedback loop in which an outer loop provides infrequent global coordination and an inner loop, operating locally, controls the motion and network routing of each robot. The resulting system is able to operate dynamically in complex environments with minimal global coordination as demonstrated through multiple experiments. We conclude with a realistic application of our system, namely patrolling a set of hallways.

Index Terms— Communication-aware robot teams, networked robots, multi-robot path planning

1. INTRODUCTION

Communication-aware robot teams must balance the goal of completing the given task with the requirement of providing the minimum end-to-end data rate. This is commonly referred to as the concurrent mobility and communication problem. The complexity of this problem is derived from the understanding that as one robot moves, the communication link between it and every other robot in the network is affected. Thus, previous systems attempt to move the robots such that specific properties of the underlying communication network are maintained. These properties range from maximizing the Laplacian's second eigenvalue [2–5], maintaining k-connectedness [6], to achieving desired end-to-end data rates [7, 8].

In order for these systems to operate effectively, a tractable model of the point-to-point link between robots must be available. This is complicated by the random phenomenon known as fading, which can dramatically affect the quality of the link. Initially, the model used was a simple binary disc, where if two robots were within a nominal distance communication was assumed, [9–12]. This allows for the communication constraints to be modeled as geometric constraints, thus simplifying the motion control portion of the system. The next set of models used, which are more complex than a binary disc, estimate the expected channel rate by a smooth function of separating distance, [2–7, 13]. These models allow for the motion control to consider the gradient of the link rate, not just its existence. The final and most complex models use an estimate of the expected link rate as well as an estimate of the variance to capture the affects of fading, [8, 14–17]. These models allow for more robust guarantees on the channel rates, thus providing more confidence in the ability of the system to maintain the desired network properties.

The main contribution of this paper is a hybrid system architecture that is able to solve the concurrent mobility and communication problem without the limitations exhibited by previous systems, while preserving their beneficial qualities (Section 3). This system is validated and compared to existing solutions experimentally (Section 4) and then used to complete a realistic task of patrolling a set of hallways (Section 5).

2. PROBLEM FORMULATION

For this paper, we use a team of N robots capable of point-to-point communication and movement through a known environment. Each robot on the team is given an index $i \in \{1, \dots, N\}$. For robot i , its location in the environment is denoted by $x_i(t) \in \mathbb{R}^2$ and the collection of all N robot locations is called a formation, $\mathbf{x}(t) \in \mathbb{R}^{2N}$. At the initial time t_0 , the team is in the formation $\mathbf{x}(t_0)$ and is required to complete a given task by time t_f . This task is assumed to be a convex function that maps formations to real values, $\Gamma(\mathbf{x}) : \mathbb{R}^{2N} \rightarrow \mathbb{R}$, and has a minimum value for the desired final formation, \mathbf{x}^* . In order to complete the task the team must follow a safe trajectory, such that $\mathbf{x}(t_f) = \mathbf{x}^*$. A safe trajectory is one in which, for all $t \in [t_0, t_f]$ the formation, $\mathbf{x}(t)$, avoids robot-robot collisions, as well as robot-obstacle collisions. To ensure no robot-robot collisions, we define the minimum safe distance for a robot, δ_r , and require $\|x_i(t) - x_j(t)\|^2 > \delta_r, \forall i, j$. Likewise, to ensure no robot-obstacle collisions, we define the set of physical obstacles $\mathcal{O} = \{o_s\}_{s=1}^S$ that must be avoided and require $\|x_i(t) - o_s\|^2 > \delta_r, \forall i, s$. Using these two requirements we define the set of safe formations \mathcal{F} and require $\mathbf{x}(t) \in \mathcal{F}, \forall t \in [t_0, t_f]$.

While the goal of the team is to complete the given task, this must not be done at the expense of K end-to-end data rates between designated pairs. These end-to-end rates, referred to as flows, consist of the source-destination pair, $(S_k, \mathcal{D}_k) \in \{N \times N\}$, and the desired minimum rate for each robot, $a_{i,min}^k \in [0, 1]$, where, $a_{i,min}^k > 0$ for $i = S_k$ and $a_{i,min}^k = 0$ for all $i \neq S_k$. To understand how data flows over the ad-hoc network, we begin by modeling the rate at which robot i , located as position $x_i(t)$, can communicate with robot j , located at position $x_j(t)$, as $R(x_i(t), x_j(t)) = R_{ij}(t) : \mathbb{R}^4 \rightarrow [0, 1]$. The amount of data for flow k that robot i sends to robot j over this link is proportional to the time the link is active, which is captured by the routing variable $\alpha_{ij}^k(t) \in [0, 1]$. Therefore, the product of the channel rate and the routing variable, $\alpha_{ij}^k(t)R_{ij}(t)$, is the data rate between i and j . To simplify notation, the rates and the routing variables are consolidated into a rate matrix, $\mathbf{R}(t) \in \mathbb{R}^{N \times N}$, with entries $R_{ij}(t)$, and a routing solution $\alpha(t) \in \mathbb{R}^{N \times N \times K}$, with entries $\alpha_{ij}^k(t)$. To maintain feasibility, we require that the proportion of time that a given node is actively transmitting is bounded from above by 1, $\sum_{j,k} \alpha_{ij}^k(t) \leq 1$ for all i . Using these definitions, we define the difference between the outbound data rate and the inbound rate of data not destined for the robot i as the communication margin for robot i ,

$$a_i^k(\alpha(t), \mathbf{x}(t)) = \sum_{j=1}^N \alpha_{ij}^k(t)R_{ij}(t) - \sum_{j=1, i \notin \mathcal{D}_k}^N \alpha_{ji}^k(t)R_{ji}(t). \quad (1)$$

To prevent unbounded queue growth we require (1) to be non-negative, which allows for $a_i^k(\alpha(t), \mathbf{x}(t))$ to be interpreted as the amount of data that robot i can add to the network without compromising stability. Thus, network integrity is achieved when,

$$a_i^k(\alpha(t), \mathbf{x}(t)) \geq a_{i,min}^k, \forall i, k, \quad \sum_{j,k} \alpha_{ij}^k(t) \leq 1, \forall i. \quad (2)$$

Turning our attention to the motion of the team, we begin by modeling the kinematics of a robot as a single input control system, $\dot{x}_i(t) =$

Supported by the ARL MAST-CTA under Grant W911NF-08-2-0004. An extended version of this paper has been accepted for publication at the IEEE International Conference on Robotics and Automation, 2016 [1].

$f(x_i(t), u_i(t))$, with input $u_i(t)$. For this paper, we limit our consideration to robots with simple dynamics, resulting in full controllability. Thus, we can more simply model the kinematics as $\dot{x}_i(t) = u_i(t)$, resulting in an integral model of motion, where $x_i(t) = \int_{t_0}^t \dot{x}_i(s) ds + x_i(t_0)$. This allows us to write the following problem,

$$\begin{aligned} & \min_{\mathbf{x}(t)} \Gamma(\mathbf{x}(t_f)) \\ \text{s. t. } & \mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^t \dot{\mathbf{x}}(s) ds, \quad \mathbf{x}(t) \in \mathcal{F}, \\ & a_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t)) \geq a_{i,min}^k, \quad \sum_{j,k} \alpha_{ij}^k(t) \leq 1. \end{aligned} \quad (3)$$

The solution to (3), $\dot{\mathbf{x}}(t)$, is a series of control inputs for the team that result in the robots moving from $\mathbf{x}(t_0)$ to $\mathbf{x}(t_f)$ while avoiding collisions and maintaining network integrity from $t \in [t_0, t_f]$.

3. COMMUNICATION-AWARE SYSTEMS

Multiple systems have been proposed to solve the concurrent mobility and communication problem. The previously proposed systems can be classified into two distinct groups, centralized and distributed. In this section, examine one system from each group, highlighting their benefits and drawbacks, and conclude with our hybrid architecture that is able to achieve the benefits without the limitations.

3.1. Centralized System

A centralized system is one in which a single node determines the globally optimal action for each robot. One such system solves the concurrent mobility and communication problem by requiring ample margin for the network integrity constraints and planning trajectories a priori that satisfy these heightened requirements [8]. This is achieved by reformulating the channel rates, R_{ij} as a Gaussian random variable and providing a probabilistic guarantee on the satisfaction of (2). Specifically, this guarantee is written,

$$\mathbb{P}\left[a_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t)) \geq a_{i,min}^k\right] > 1 - \epsilon. \quad (4)$$

This formulation provides a margin of error when computing the end-to-end rates for trajectories that visit locations where channel rates have yet to be measured. Since R_{ij} is a Gaussian random variable, $a_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t))$ is also a Gaussian random variable with mean, $\bar{a}_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t))$ and variance $\tilde{a}_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t))$. This allows (4) to be written more precisely as,

$$\frac{\bar{a}_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t)) - a_{i,min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t))}} \geq \Phi^{-1}(\epsilon), \quad (5)$$

where $\Phi^{-1}(\epsilon)$ is the inverse Gaussian complementary cumulative distribution function. The constraint in (5) has been shown to define a second order cone problem when $\epsilon < 0.5$. Therefore, given the formation $\mathbf{x}(t)$, feasible $\boldsymbol{\alpha}(t)$ can be determined in polynomial time via convex programming techniques [18]. This allows for efficient validation of the feasibility of a specific formation $\mathbf{x}(t)$. Using this, the system solves the following modified version of (3),

$$\begin{aligned} & \min_{\dot{\mathbf{x}}(t), \boldsymbol{\alpha}(t)} \Gamma(\mathbf{x}(t_f)) \\ \text{s. t. } & \mathbf{x}(t) = \mathbf{x}(t_0) + \int_0^t \dot{\mathbf{x}}(s) ds, \quad \mathbf{x}(t) \in \mathcal{F}, \\ & \bar{a}_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t)) \geq a_{i,min}^k + \Phi^{-1}(\epsilon) \sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t))}, \\ & \sum_{j,k} \alpha_{ij}^k(t) \leq 1. \end{aligned} \quad (6)$$

To solve (6), a Rapidly Exploring Random Tree (RRT), [19], is used to determine trajectories for the robots that are free from physical collisions, as well as preserve network integrity. This is achieved by the RRT verifying the existence of feasible $\boldsymbol{\alpha}$ for each candidate formation, in addition to collision free motion between those formations. This modification allows the RRT to determine a feasible trajectory, where by the robots successfully solve (6) without requiring an exhaustive search of the configuration space.

Upon successfully finding feasible trajectories, the system then executes them by way of a closed loop control system. To begin, the controller commands the robots to move to their first goal location based on its trajectory. After every robot reaches its goal, the next set of goals are published. This process is repeated until the entire trajectory is executed, at which point the task has been successfully completed. The $\boldsymbol{\alpha}(t)$ are determined during the planning process and provided to the robots along with the trajectories. This approach results in a system that is able to successfully operate in a complex environments, but requires a large amount of coordination to guarantee the $\boldsymbol{\alpha}(t)$ and $\mathbf{x}(t)$ are synchronized across the team. Also, the tight control exhibited over the motion of the robots introduces a rigidity to the system that can result in poor performance in the presence of unexpected events.

3.2. Distributed System

In contrast to the centralized system, a distributed system is one in which each robot determines its optimal action based on locally available information to complete the task. To achieve this, these systems formulate the problem of task completion such that distributed optimization techniques can be applied. One such system allows for the team to reach designated locations in the environment using only local information. The system utilizes a continuous-time motion-gradient controller and a discrete-time network routing system to solve the concurrent mobility and communication problem [7].

The motion-controller employs a modified navigation function that allows each robot to reach their designated goal location, $x_{i,0}$ while avoiding obstacles, [20, 21]. The navigation function relies on the creation of virtual communication obstacles, $\beta_i(\mathbf{x}(t))$, which are derived from the current $\boldsymbol{\alpha}(t)$. To compute $\beta_i(\mathbf{x}(t))$, the network integrity constraints from (2) are used along with an additional tolerance, $e > 0$, as follows,

$$\beta_i(\mathbf{x}(t)) = \min_{k=1, \dots, K} \{a_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t)) - a_{i,min}^k + e\}. \quad (7)$$

The definition in (7) is non-negative when all of the network integrity requirements at node i are satisfied, within the prescribed tolerance e , and is negative when one of the requirements is violated. This indicates that the zero-point of $\beta_i(\mathbf{x}(t))$ can be treated as an obstacle in a standard navigation function,

$$\phi_i(\mathbf{x}(t)) = \frac{\rho_i(\mathbf{x}(t))}{(\rho_i(\mathbf{x}(t))^\kappa + \beta_i(\mathbf{x}(t))^2)^{1/\kappa}}, \quad (8)$$

where $\rho_i(\mathbf{x}(t)) = \|x_i(t) - x_{i,0}\|^2$. When κ is sufficiently large, the effects of these communication obstacles are confined to regions where network integrity is at risk of being violated. Therefore, using,

$$\dot{x}_i(t) = -\nabla_{x_i} \phi_i(\mathbf{x}(t)) \quad (9)$$

will result in the robot reaching its goal while preserving network integrity.

The successful operation of the motion controller is predicated on the proper computation of $\boldsymbol{\alpha}(t)$. To compute $\boldsymbol{\alpha}(t)$ such that they are not only feasible but provide maximum margin for the network integrity

constraints the following optimization problem is used,

$$\begin{aligned} \boldsymbol{\alpha}(t) = \operatorname{argmax}_{a_i^k, \alpha_{ij}^k} & \sum_{k=1}^K \sum_{i=1}^N \left[U_i^k(a_i^k) + \sum_{j=1}^N V_{ij}^k(\alpha_{ij}^k) \right] \\ \text{s. t.} & a_i^k(\boldsymbol{\alpha}, \mathbf{x}(t)) = a_i^k \geq a_{i,min}^k, \quad \sum_{j,k} \alpha_{ij}^k(t) \leq 1. \end{aligned} \quad (10)$$

The goal of (10) is to compute the $\boldsymbol{\alpha}(t)$ that not only preserve network integrity, but also maximize the objective function to disambiguate among the feasible $\boldsymbol{\alpha}(t)$, given the spatial configuration $\mathbf{x}(t)$. The objective function is composed of a $U_i^k(a_i^k)$, which captures the benefit of achieving a_i^k , and $V_{ij}^k(\alpha_{ij}^k)$, which disincentivizes a heavy reliance on single links. The resulting $\boldsymbol{\alpha}(t)$ provides the required end-to-end rates, when feasible, while providing a balance between robustness and rate maximization.

Since (10) is a convex problem given $\mathbf{x}(t)$, it can be solved by dual gradient decent with the introduction of non-negative dual variables, $\lambda_i^k(t_n)$. Each of the $\lambda_i^k(t_n)$ are associated with a constraint in (10), and use t_n to track the current iteration. Grouping these variables into a matrix, $\boldsymbol{\lambda}(t_n) \in \mathbb{R}^{N \times K}$, allows for the construction of a Lagrangian, $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{x})$, which can be decomposed into local Lagrangians,

$$\begin{aligned} \mathcal{L}_i(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{x}) = & \sum_{k=1}^K U_i^k(a_i^k) - \lambda_i^k a_i^k \\ & + \sum_{j=1}^N \left[V_{ij}^k(\alpha_{ij}^k) + \alpha_{ij}^k R_{ij}(\lambda_i^k - \lambda_j^k) \right], \end{aligned} \quad (11)$$

where $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{x}) = \sum_{i=1}^N \mathcal{L}_i(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{x})$. Since $\mathcal{L}_i(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{x})$ only depends on a_i^k , λ_i^k , and α_{ij}^k , as well as only the λ_j^k 's for which $R_{ij} > 0$, it can be computed using locally available information. Therefore, by exchanging λ_i^k among immediate neighbors, each robot can compute their optimal rate and portion of the routing solution at iteration t_n by solving,

$$\begin{aligned} a_i^k(t_n), \left\{ \alpha_{ij}^k(t_n) \right\}_{j=1}^N = & \operatorname{argmax} \mathcal{L}_i(\boldsymbol{\lambda}(t_n), \boldsymbol{\alpha}(t_n), \mathbf{x}(t_n)). \\ \text{s. t.} & a_i^k \geq a_{i,min}^k, \\ & \sum_{j,k} \alpha_{ij}^k(t) \leq 1. \end{aligned} \quad (12)$$

After solving (12), the next step is to update the value of λ_i^k . To maintain the non-negative requirement for λ_i^k , a non-negative projection $\mathbb{P}[y]$, which returns y is $y \geq 0$ and 0 if $y < 0$ is used. Thus, by following $\nabla_{\lambda_i^k} \mathcal{L}_i(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{x})$, the $\lambda_i^k(t_n)$ are updated by, $\lambda_i^k(t_{n+1}) = \mathbb{P} \left[\lambda_i^k(t_n) - \epsilon \left(\sum_{j=1}^N \alpha_{ij}^k(t_n) R_{ij} - \sum_{j=1}^N \alpha_{ji}^k(t_n) R_{ji} - a_i^k(t_n) \right) \right]$. These updated values are then shared with all the robots within communication range so they can be used in the next iteration of (12). This process is repeated and converges to the optimal routing solution when the formation is static. If the formation is changing the resulting solutions will be near optimal, and the deviation from optimality is dependent on the frequency of the iterations and the allowable velocity of the robots. This system has demonstrated the ability to navigate a simple environment and achieve desired locations for specific robots with no global coordination. This allows the robots to react to other members of the team dynamically, while locally optimizing the path traveled. Due to operating with only local information, when the environment becomes more complex this system is unable to avoid local minima and the result is the inability to achieve the desired formation.

3.3. Hybrid System

To overcome the limitations of both the centralized and distributed systems, we propose a hybrid architecture that draws on the strengths of

each system while avoiding their pitfalls. This is accomplished by constructing a two-stage feedback system where an outer centralized loop is responsible for infrequent global path planning and an inner distributed loop is used to control the motion and network routing.

For the outer loop, the centralized path planner from Section 3.1 is used. The process begins with the user providing the global task, $\Gamma(\mathbf{x})$, that is to be completed. The global planner then determines trajectories, $\tilde{\mathbf{x}}(t) = \{\tilde{x}_i(t)\}_{i=1}^N$, that minimize $\Gamma(\mathbf{x})$, while providing sufficient margin for the network integrity constraints. Instead of executing the trajectories exactly, they are fed into a system that generates a series of waypoints that are disseminated to the robots according to, $\mathcal{X}_i = \{\tilde{x}_i(\tau_w)\}_{w=1}^W$. The values chosen for the τ_w are critical to decomposing (3) into a series of subproblems that are free of local minima, in the region where the robots are currently operating. This allows a distributed controller to successfully solve the sequential subproblems and thus complete the task.

For the distributed controller, the system in Section 3.2 is leveraged with the addition of a waypoint curator. The waypoint curator is responsible for ingesting \mathcal{X}_i from the outer loop and systematically updating $x_{i,0}$ from $\tilde{x}_i(\tau_w)$ to $\tilde{x}_i(\tau_{w+1})$, when certain criteria are met. These two criteria are $\|x_i(t) - x_{i,0}\| \leq \delta$ for a specified value of $\delta > 0$ and $t - t_w > \tau_{w+1} - \tau_w$, where t_w is the time at which the $x_{i,0}$ was updated to $\tilde{x}_i(\tau_w)$. Thus, the waypoint curator is able provide confidence that an individual robot will follow its prescribed trajectory, within some tolerance, and the evolution of the entire formation will be similar. but not identical to, the centralized plan. This combined with the decomposition into sequential subproblems allows the distributed controllers to operate independently after the initial planning stage.

This hybrid architecture is capable of operating in complex environments, similar to the centralized system, while minimizing the necessary global coordination during execution, similar to the distributed system. Thus, this system exhibits the benefits of the two different approaches while avoiding their limitations. A more detailed description of the hybrid system can be found in [1].

4. RESULTS

For this paper, we use a team of *Scarabs* [22], a custom built robot designed at the University of Pennsylvania, as our robotic platform. The ad-hoc network is composed of Digi International XBee transceivers mounted on the *Scarabs*. The environment used is the Graduate Research Wing of the Levine building (Levine-GRW) at the University of Pennsylvania. For the channel estimation, we are using a function that is a polynomial fitting of experimental curves found in the literature [23].

4.1. System Comparison

In the initial set of experiments, we compare the successful packet transmission of our hybrid system to the centralized system developed in [8]. The required task was to drive the sensing robot to the blue square, in the upper right corner of the environment shown in Fig. 1a, from the initial formation, in the lower left corner, while transmitting a video stream from the sensing robot back to the access point, $a_{4,min}^1 = 0.5$. While this task is straightforward, the minimal set up allows for repeated, nearly identical, trials and it is sufficient to demonstrate the capabilities of both the hybrid and the centralized systems. A single set of waypoints from the waypoint generator is used by both the centralized and hybrid systems to remove any bias incurred by different input waypoints. For both trial sets the routing solution was determined at a rate of 10 Hz, thus global coordination is required 10 times a second for the centralized system, while only once at the start of the trial for the hybrid system followed by only local communication. The results of ten trials are plotted in Fig. 1c, where the solid line represents the average over all the trials and the dotted envelope shows the one σ bounds. There are a few

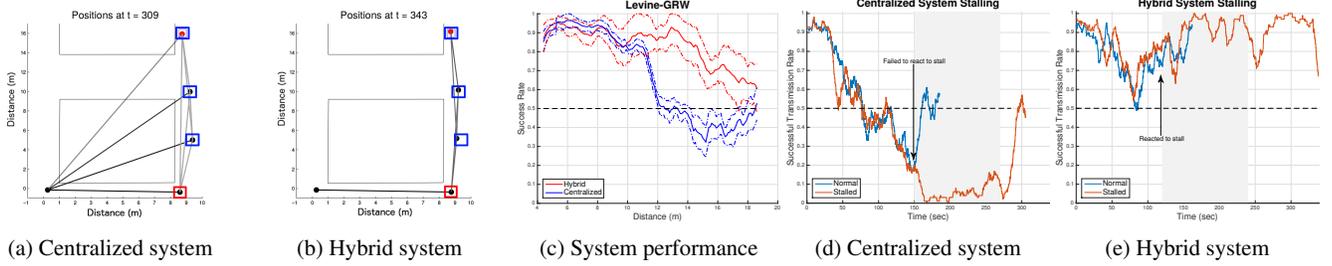


Fig. 1: Figs. (a) and (b) show the final formation for both systems with their routing solutions indicated by the lines connecting the robots. In Fig. (c) the solid line is the average performance and the dotted line indicates the σ bounds of the two systems. The black dashed line is the minimum required data rate. In Figs. (d) and (e) the blue line is from the experiment under normal conditions and the red line is from an experiment where there is a motor failure. The shaded region indicates the time the motor failed for the stalled experiment and the dashed line is the minimum rate.

items to note, first notice how well the hybrid system performs. Even the one σ bound stays above the required data rate. This is mostly due to the robots locally optimizing their trajectory and not moving in straight lines. Another item to note is the spread on the one σ bounds. Since the centralized system is including variance of the channel the spread is much less than the hybrid system which is only using expected value of the channel. Also since the hybrid system allows for deviations to locally optimize, the trajectories taken by the robots is not always the same compared to the tightly controlled trajectories executed by the centralized system. The final item to note is the divergence of the results for the two systems at 12 meters. While the hybrid system continues to exceed the required data rates the centralized system drops off dramatically to marginally meeting the requirements. This is a result of the centralized system’s conservative approach to the routing problem and thus sacrificing performance in favor of reliability. Conversely, the hybrid system is leveraging current information which allows it to forgo a conservative approach and maximize performance.

4.2. Dynamic Response

In this set of experiments, we highlight a major benefits of using a local controller, as opposed a centralized waypoint system, namely dynamic response to unexpected events. In these experiments, as with the previous section, the goal was to drive around the corner in Levine-GRW to a goal location, but during deployment one of the robots has a temporary restriction to its motion. This set of experiments were run just as the previous section was but when a specific support robot reaches middle of the lower hallway in Fig. 1a its motor is disabled for 120 seconds, to simulate a temporary restriction in motion.

To compare the network performance of these tests we re-ran the experiments without the robot stalling. The results of the two experiments for the centralized and hybrid systems are plotted in Figs. 1d and 1e. In these plots the red and blue lines are the data rate of system with and without the stall, which is indicated by the shaded region. It can be seen that prior to the stall the two lines are in agreement for both systems, but when the stall occurs we see that the two lines in Fig. 1d diverge, while they do not in Fig. 1e. The divergence in Fig. 1d is due to the formation deviating greatly from the one that was verified by the centralized planner. After the stall is recovered from we see that the network performance returns to the desired value. In contrast in Fig. 1e we see that the network performance never suffers from the robots being out of position. This is because when the stall occurs the other members of the team react accordingly, specifically the sensing robot halting its motion. These experiments show how the hybrid system is more robust to dynamic changes in the environment and other obstacles that may arise during the execution of a task when compared to the more brittle waypoint synchronization of the centralized approach.

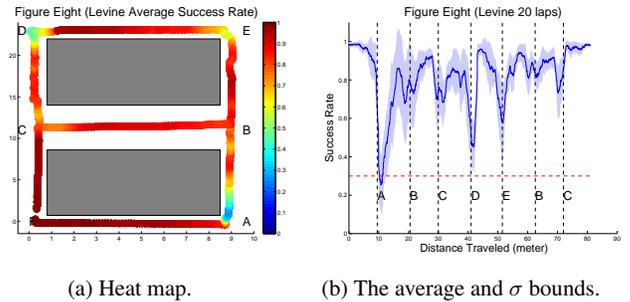


Fig. 2: Data rates while patrolling the hallway.

5. APPLICATION

Having demonstrated the ability to control the motion of the team so that the lead robot is able to reach a desired location while preserving network integrity, we increase the complexity and duration of the task. The task considered in this experiment is the repeated patrolling of a series of hallways while preserving network integrity. For this task to be accomplished, the system must be modified to accept multiple goal locations for the sensing robot. Additionally, it must determine locations for the support robots so that their required motion to support the patrolling robot is minimized during the deployment. This experiment was conducted in Levine-GRW with three support robots, an access point, and a leader transmitting low-quality video, $\alpha_{4,min}^1 = 0.3$. The lead robot is tasked with repeatedly visiting sensing locations (A, B, C, D, E, B, C) shown in Fig. 2a, for a total of 20 laps. Given the series of sensing locations, the global planner determined that the optimal location for the three support robots are B, C, and D. With the support robots in these locations the lead robot has a sufficient link back to the access point for the entirety of the deployment. The results of the 20 laps are shown in both Figs. 2a and 2b, where the individual laps are overlaid and examined. In Fig. 2a the average data rate experienced at every location visited is shown, with the color indicating the rate. In Fig. 2b the average data rate and one σ bounds are plotted as a function of distance travelled from the starting location. The minimum required rate is indicated by the red dotted line and the sensing locations are indicated by the vertical dashed lines. Note that not only is the average data rate above the required rate, but the one σ value is also sufficient. The one location where the minimum rate is not maintained is location A. This drop in performance can be attributed to the time needed for the distributed controller to converge to the new optimal solution after a dramatic change in the underlying communication network. The change is dramatic because prior to A there is a direct link to the access point, but right after A the data must now be routed through the two robots stationed at B and C.

6. REFERENCES

- [1] J. Stephan, J. Fink, V. Kumar, and A. Ribeiro, "Hybrid architecture for communication-aware multi-robot systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [2] M. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *IEEE Conference on Decision and Control*, Dec 2006, pp. 3628–3633.
- [3] M. Schuresko and J. Cortés, "Distributed motion constraints for algebraic connectivity of robotic networks," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1-2, pp. 99–126, 2009. [Online]. Available: <http://dx.doi.org/10.1007/s10846-009-9328-8>
- [4] M. Zavlanos and G. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 812–816, Aug 2007.
- [5] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, Jan 2006.
- [6] E. Stump, A. Jadbabaie, and V. Kumar, "Connectivity management in mobile robot teams," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008, pp. 1525–1530.
- [7] M. Zavlanos, A. Ribeiro, and G. Pappas, "Network integrity in mobile robotic networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 3–18, Jan 2013.
- [8] J. Fink, A. Ribeiro, and V. Kumar, "Robust control of mobility and communications in autonomous robot teams," *IEEE Access*, vol. 1, pp. 290–309, 2013.
- [9] M. Ji and M. Egerstedt, "Distributed coordination control of multi-agent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, Aug 2007.
- [10] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428, Dec 2008.
- [11] D. Spanos and R. Murray, "Motion planning with wireless network constraints," in *Proceedings of the American Control Conference*, June 2005, pp. 87–92.
- [12] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie, "Maintaining limited-range connectivity among second-order agents," in *American Control Conference, 2006*, June 2006, pp. 6 pp.–.
- [13] O. Tekdas, W. Yang, and V. Isler, "Robotic routers: Algorithms and implementation," *The International Journal of Robotics Research*, 2009. [Online]. Available: <http://ijr.sagepub.com/content/early/2009/05/19/0278364909105053.abstract>
- [14] Y. Mostofi, "Communication-aware motion planning in fading environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008, pp. 3169–3174.
- [15] Y. Mostofi, A. Gonzalez-Ruiz, A. Gaffarkhah, and D. Li, "Characterization and modeling of wireless channels for networked robotic and control systems - a comprehensive overview," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2009, pp. 4849–4854.
- [16] Y. Yan and Y. Mostofi, "Robotic router formation in realistic communication environments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 810–827, Aug 2012.
- [17] J. Fink, "Communication for teams of networked robots," Ph.D. dissertation, University of Pennsylvania, 2011.
- [18] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, no. 1–3, pp. 193 – 228, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0024379598100320>
- [19] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2000, pp. 995–1001 vol.2.
- [20] E. Rimon and D. Koditschek, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412 – 442, 1990. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0196885890900175>
- [21] —, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, Oct 1992.
- [22] N. Michael, J. Fink, and V. Kumar, "Experimental testbed for large multirobot teams," *Robotics Automation Magazine, IEEE*, vol. 15, no. 1, pp. 53–61, 2008.
- [23] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 121–132, Aug. 2004. [Online]. Available: <http://doi.acm.org/10.1145/1030194.1015482>