Distributed Intermittent Communication Control of Mobile Robot Networks under Time-Critical Dynamic Tasks

Yiannis Kantaros, and Michael M. Zavlanos

***Abstract*— In this paper, we develop a distributed intermittent communication framework for teams of mobile robots that are responsible for accomplishing time-critical dynamic tasks and sharing the collected information with all other robots and possibly also with a user. Specifically, we consider situations where the robot communication capabilities are not sufficient to maintain reliable and connected networks while the robots move to accomplish their tasks. In this case, intermittent communication protocols are necessary that allow the robots to temporarily disconnect from the network in order to accomplish their tasks free of communication constraints. We assume that the robots can only communicate with each other when they meet at common locations in space. Our proposed distributed control framework determines offline schedules of communication events and integrates them online with task planning. The resulting paths ensure task accomplishment and exchange of information among robots infinitely often at locations that minimize a user-specified metric. Simulation results corroborate the proposed distributed control framework.**

I. INTRODUCTION

Recently, there has been a large amount of work focused on designing controllers that ensure point-to-point or end-to-end network connectivity for all time. Such controllers either rely on graph theory to model robot communication [1]–[3] or employ more realistic communication models that take into account path loss, shadowing, and multi-path fading as well as optimal routing decisions for desired information rates [4]–[6]. Nevertheless, due to the uncertainty in the wireless channel, that affects signal strength in an unpredictable way, it is often impossible to ensure all-time connectivity in practice. Moreover, maintaining all-time connectivity can severely restrict the robots from accomplishing their tasks, as motion planning is always constrained by network connectivity constraints. Therefore, a much preferred solution is to allow robots to communicate in an intermittent fashion and operate in disconnect mode the rest of the time.

In this paper, we consider robots that are responsible for accomplishing high-level tasks that are (i) *dynamic*, i.e., the task specifications can change over time, and (ii) *time-critical* in the sense that the robots should not hold onto the information they collect as they navigate the workspace for a long time; instead, they need to communicate frequently enough, according to desired specifications. We assume that the robots have limited communication ranges and, therefore, they can only communicate when they are physically close to each other. Motivated by that, we propose a novel distributed task planning and intermittent communication framework that allows robots to temporarily disconnect from the network to accomplish their assigned tasks free of communication constraints, but at the same time ensures that the communication network is connected over time so that information can be propagated in the network intermittently, in a multi-hop fashion. Our proposed distributed control framework constructs offline schedules of communication events that dictate how the robots will communicate and integrates them online with task planning. The resulting paths ensure task accomplishment and exchange of information among robots infinitely often at locations that minimize a user-specified metric, such as traveled distance.

The most relevant works to the one proposed here are recent works by the authors [7]–[11]. Specifically, [7] proposes an asynchronous distributed intermittent communication framework that is a special case of the one proposed here in that every robot belongs to exactly two teams and the robots in every team can only meet at a single predetermined location. This framework is extended in [8], where robots can belong to any number of teams and every team can select among multiple locations to meet, same as in the work considered here. Nevertheless, neither of the approaches in [7], [8] consider concurrent task planning. Intermittent communication control and task planning is considered in [9]. Nevertheless, this approach is centralized and does not scale well with the number of robots. A distributed offline approach for this problem is presented in [10] that can only handle periodic tasks captured by Linear Temporal Logic (LTL) formulas. Also, optimality guarantees are provided in [10] by exploiting the periodic structure of the designed paths, which do not exist in this work. Nevertheless, our proposed algorithm can handle arbitrary dynamic tasks and also allows the robots in every team to arrive at the selected communication point simultaneously, avoiding in this way waiting delays, which is not the case in [10]. A distributed online approach to this problem for LTL tasks is proposed in [11]. Nevertheless, the method proposed here, is more general in that it can handle the data gathering tasks and the two-hop star communication topology in [11] that considers information flow only to the root/user. In fact, in the proposed method, information can flow intermittently between any pair of robots and possibly a user in a multi-hop fashion. Other relevant methods are presented in [12], [13]. However, these methods either impose strong restrictions on the communication pattern that can be achieved or they do not consider concurrent task planning. We provide theoret-

Yiannis Kantaros and Michael M. Zavlanos are with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA. {yiannis.kantaros, michael.zavlanos}@duke.edu. This work is supported in part by NSF under grants CNS #1261828 and CNS #1302284.

ical guarantees and numerical simulations that support the proposed framework. To the best of our knowledge, this is the first distributed and online intermittent communication framework that can handle arbitrary time-critical dynamic tasksx. Also, the proposed framework scales well with the size of the network.

The rest of this paper is organized as follows. The problem formulation is described in Section II. In Section III sequences of communication events are defined that ensure that the communication network is connected over time infinitely often. The proposed distributed integrated task planning and intermittent communication framework is presented in Section IV. Simulation results are included in Section VI.

II. Problem Formulation

Consider $N \geq 1$ mobile robots operating in a workspace $\mathcal{W} \subset \mathbb{R}^d$, $d = 2, 3$, and let $\dot{\mathbf{x}}_i(t) = f_i(\mathbf{x}_i(t), \mathbf{u}_i(t))$ denote the equations of motion of robot i, where $\mathbf{x}_i(t) \in \mathbb{R}^d$ and $\mathbf{u}_i(t) \in \mathbb{R}^d$ are the position and control input of robot i, respectively, at time $t \geq 0$. Let $\mathcal{N} = \{1, \dots, N\}$ denote the set of all robots.

We assume that the robots have to accomplish a time-critical dynamic task, defined as $\mathcal{H}_i = \{\mathbf{p}_i^1, \dots, \mathbf{p}_i^{h_i}, \dots, \mathbf{p}_i^{H_i}\}$, where $\mathbf{p}_i^{h_i} \in \mathcal{W}$, are waypoints associated with locations in space where the tasks take place, $h_i \in \{1, \dots, H_i\}$, and $H_i \in \mathbb{N}_+$. Note that we impose no restriction on the structure of the sequence $\mathcal{H}_i$, i.e., it can be periodic or aperiodic, and H_i can be finite or infinite. Moreover, we assume that the tasks $\mathcal{H}_i$ are not *a priori* known to the robots and instead, they are revealed over time. Specifically, at any time t, every robot i has access to a part of the task $\mathcal{H}_i$, defined as $\mathcal{H}_i^{\text{cur}}(t) = \{\mathbf{p}_i^{\xi_i^0(t)}, \dots, \mathbf{p}_i^{\xi_i^f(t)}\} \subseteq \mathcal{H}_i$, where $\xi_i^0(t)$ through $\xi_i^f(t)$ are consecutive indices to $\{1, \dots, H_i\}$ and point to the entries in $\mathcal{H}_i$ that are the first and the last entries in $\mathcal{H}_i^{\text{cur}}(t)$; see also Figure 1(a). The current tasks $\mathcal{H}_i^{\text{cur}}(t)$ can be updated as the robots navigate the workspace, by adding to them additional waypoints from $\mathcal{H}_i$. Specifically, the current task $\mathcal{H}_i^{\text{cur}}(t^+)$ of robot i at time t^+, right *after* an update at time t, is constructed as $\mathcal{H}_i^{\text{cur}}(t^+) = \mathcal{H}_i^{\text{cur}}(t) \cup \{\mathbf{p}_i^{\xi_i^f(t)+1}, \dots, \mathbf{p}_i^{\xi_i^f(t^+)}\} \subseteq \mathcal{H}_i$. The time instants t when the current tasks $\mathcal{H}_i^{\text{cur}}(t)$ are updated, as well as, and the corresponding the new task specifications/waypoints $\{\mathbf{p}_i^{\xi_i^f(t)+1}, \dots, \mathbf{p}_i^{\xi_i^f(t^+)}\}$ are determined on-line and are not known *a priori*. Also, to ensure that $\mathcal{H}_i^{\text{cur}}(t)$ are always finite, every robot i deletes from $\mathcal{H}_i^{\text{cur}}(t)$ all waypoints that they have already visited.

Moreover, the assigned tasks are *time-critical* in the sense that the information collected by the robots as they visit waypoints included in $\mathcal{H}_i^{\text{cur}}(t)$ is time-critical and, as result, robots should not hold onto the gathered data for a long time. Instead, they have to communicate to other robots frequently enough, according to desired specifications. Specifically, we require that every robot i should communicate with other robots before visiting a specified number of waypoints included in $\mathcal{H}_i^{\text{cur}}(t)$ since the last communication event they participated. This allowed number of task waypoints that they can visit without communicating can change with time based, e.g., on the importance of the collected data.

To define a communication network among the robots, we first partition the robot team into $M \geq 1$ robot subgroups, called also teams that are user-specified and fixed with time. Also, we require that every robot belongs to at least one team. The indices i of the robots that belong to the m-th team are collected in a set denoted by $\mathcal{T}_m$, for all $m \in \mathcal{M} := \{1, 2, \dots, M\}$. We define the set that collects the indices of teams that robot i belongs to as $\mathcal{M}_i = \{m | i \in \mathcal{T}_m, \ m \in \mathcal{M}\}$. Given the teams $\mathcal{T}_m$, for all $m \in \mathcal{M}$, we can define the graph over these teams as follows.

Definition 2.1 (Team Membership Graph $\mathcal{G}_\mathcal{T}$): The graph over the teams $\mathcal{T}_m$, $m \in \mathcal{M}$ is defined as $\mathcal{G}_\mathcal{T} = (\mathcal{V}_\mathcal{T}, \mathcal{E}_\mathcal{T})$, where the set of nodes $\mathcal{V}_\mathcal{T}$ is indexed by the teams $\mathcal{T}_m(t)$ and set of edges $\mathcal{E}_\mathcal{T}$ is defined as $\mathcal{E}_\mathcal{T} = \{(m,n) | \mathcal{T}_m \cap \mathcal{T}_n \neq \emptyset, \forall m, n \in \mathcal{V}_\mathcal{T}, m \neq n\}$.

We assume that the robots have limited communication capabilities and, therefore, they can communicate only if they are physically close to each other at a common location in space, hereafter called a communication point. Specifically, we assume that there are $R \geq 1$ available communication points at locations $\mathbf{v}_j \in \mathcal{W}$, for $j = 1, \dots, R$, and we denote by $\mathcal{C} = \{1, \dots, R\}$ the index set of all communication points. The indices j of the communication points $\mathbf{v}_j$ where communication can take place for the robotic team $\mathcal{T}_m$ are collected in a finite and fixed set $\mathcal{C}_m \subseteq \mathcal{C}$, where the sets $\mathcal{C}_m$ are not necessarily disjoint. When all robots in a team $\mathcal{T}_m$ have arrived at a common communication location, we assume that communication happens and the robots leave to accomplish their tasks or communicate with other teams. This way, a dynamic robot communication network is constructed, defined as follows.

Definition 2.2 (Communication Network $\mathcal{G}_c(t)$): The communication network among the robots is defined as a dynamic undirected graph $\mathcal{G}_c(t) = (\mathcal{V}_c, \mathcal{E}_c(t))$, where the set of nodes $\mathcal{V}_c$ is indexed by the robots, i.e., $\mathcal{V}_c = \mathcal{N}$, and $\mathcal{E}_c(t) \subseteq \mathcal{V}_c \times \mathcal{V}_c$ is the set of communication links that emerge among robots in every team $\mathcal{T}_m(t)$, when they all meet at a common communication point $\mathbf{v}_j$, $j \in \mathcal{C}_m$, simultaneously.

To ensure that information is continuously transmitted across the network of robots, we require that the communication graph $\mathcal{G}_c(t)$ is *connected over time infinitely often*, i.e., that all robots in every team $\mathcal{T}_m$ meet infinitely often at a common communication point $\mathbf{v}_j$, $j \in \mathcal{C}_m$, that does not need to be fixed over time. [1] For this, it is necessary to assume that the graph of teams $\mathcal{G}_\mathcal{T}$ is connected. Specifically, if $\mathcal{G}_\mathcal{T}$ is connected, then information can be propagated intermittently across teams through robots that are common to these teams and, in this way, information can reach all robots in the network. Connectivity of $\mathcal{G}_\mathcal{T}$ and the fact that robots can be members of only a few teams means that

[1] More details on the definition of connectivity over time can be found in our previous works [8], [10].

information can be transferred over long distances, possibly to reach a remote user, without requiring that the robots leave their assigned regions of interest defined by their assigned tasks and communication points corresponding to the teams they belong to. Moreover, we assume that the teams are *a priori* known and can be selected arbitrarily as long as the graph of teams $\mathcal{G}_{\mathcal{T}}$ is connected. Moreover, we assume that the communication points $\mathbf{v}_j$, $j \in \mathcal{C}_m$ for the first communication event of all teams $\mathcal{T}_m$ are also user-specified.

The goal in this paper is to design paths $\mathcal{P}_i(t)$ for all robots i so that the assigned tasks $\mathcal{H}_i^{\text{cur}}(t)$ are accomplished, the intermittent connectivity requirement is satisfied, and a user defined cost $\sum_{i\in\mathcal{N}} J(\mathcal{P}_i(t))$ is minimized, where

$$J(\mathcal{P}_i(t)) = \sum_{k_i=1}^{K_i(t)-1} w(\mathcal{P}_i^{k_i}(t), \mathcal{P}_i^{k_i+1}(t)). \tag{1}$$

Specifically, the paths $\mathcal{P}_i(t)$ consist of the waypoints in $\mathcal{H}_i^{\text{cur}}(t)$ in the given order as well as communication points from the sets $\mathcal{C}_m$ associated with the teams $\mathcal{T}_m$ to which robot i belongs. Note that any communication points from these sets $\mathcal{C}_m$ can enter $\mathcal{P}_i(t)$ in any order. Optimization of the cost in (1) ensures that the communication points are selected and placed in $\mathcal{P}_i(t)$ optimally. Moreover, in (1), $K_i(t)$ denotes the number of waypoints in $\mathcal{P}_i(t)$, $\mathcal{P}_i^{k_i}(t)$ stands for the k_i-th waypoint in $\mathcal{P}_i(t)$, and $w(\mathcal{P}_i^{k_i}(t), \mathcal{P}_i^{k_i+1}(t))$ represents the cost to transition from $\mathcal{P}_i^{k_i}(t)$ to $\mathcal{P}_i^{k_i+1}(t)$. Hereafter, we define the transition cost $w(\mathcal{P}_i^{k_i}(t), \mathcal{P}_i^{k_i+1}(t))$ as the distance between $\mathcal{P}_i^{k_i}(t)$ and $\mathcal{P}_i^{k_i+1}(t)$, i.e.,

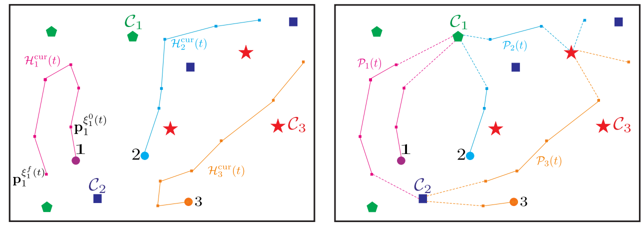
$$w(\mathcal{P}_i^{k_i}(t), \mathcal{P}_i^{k_i+1}(t)) = \left\| \mathcal{P}_i^{k_i}(t) - \mathcal{P}_i^{k_i+1}(t) \right\|. \tag{2}$$

Note that alternative transition costs w can be defined that can capture, e.g., consumed energy or travel time. The problem that is addressed in this paper can be summarized as follows and illustrated in Figure 1.

Problem 1: Given dynamic task specifications $\mathcal{H}_i^{\text{cur}}(t)$ and fixed teams $\mathcal{T}_m$, $m \in \{1, \dots, M\}$, select respective communication points $\mathbf{v}_j$, $j \in \mathcal{C}_m$ so that the robot paths $\mathcal{P}_i(t)$ for all $i \in N$ satisfy: (i) the assigned tasks are accomplished, i.e., all robots i go through all waypoints of $\mathcal{H}_i^{\text{cur}}(t)$ in the order they appear in $\mathcal{H}_i^{\text{cur}}(t)$; (ii) the communication graph $\mathcal{G}_c(t)$ is connected over time infinitely often; (iii) all robots $i \in \mathcal{N}$ share the collected time-critical information frequently enough with all robots in teams $\mathcal{T}_m$, for all $m \in \mathcal{M}_i$, according to desired specifications; and (iv) the total cost function $\sum_{i\in\mathcal{N}} J(\mathcal{P}_i(t))$ is minimized.

III. Intermittent Connectivity Control

In this section, we define infinite sequences of communication events (also called communication schedules) that ensure that $\mathcal{G}_c(t)$ is connected over time infinitely often. The communication schedules are constructed offline and require that the robots are connected so that they can share information with each other. Due to space limitations the detailed construction of these schedules is omitted and can be found in Section V in [10]. The constructed communication schedules are used in Section IV to design a distributed integrated task planning and intermittent connectivity control framework.



(a) (b)

Fig. 1. Graphical illustration of the problem formulation. A network of $N = 3$ robots (colored dots) divided into $M = 3$ teams is depicted. The robot teams are selected to be: $\mathcal{T}_1 = \{1,2\}$, $\mathcal{T}_2 = \{1,3\}$, and $\mathcal{T}_3 = \{3,2\}$. The green polygons, the blue square, and the red star stand for the communication points in the sets $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$, respectively. Figure 1(a) illustrates the sequences $\mathcal{H}_i^{\text{cur}}(t)$ and Figure 1(b) depicts the paths $\mathcal{P}_i(t)$ that include the task waypoints of $\mathcal{H}_i^{\text{cur}}(t)$ and the communication points for all teams $\mathcal{T}_m$, $m \in \mathcal{M}_i$. The communication schedules are $\texttt{sched}_1 = [1,2,X]^\omega$, $\texttt{sched}_2 = [1,X,3]^\omega$, and $\texttt{sched}_3 = [X,2,3]^\omega$.

In what follows, we define the communication schedules that determine the order in which the robots in every team $\mathcal{T}_m$ should communicate with each other.

Definition 3.1 (Schedule of Communication Events): The schedule of communication events of robot i, denoted by $\texttt{sched}_i$, is defined as an infinite repetition of the finite sequence $s_i = X, \dots, X, \mathcal{M}_i(1), X, \dots, X, \mathcal{M}_i(2), X, \dots, X, \mathcal{M}_i(|\mathcal{M}_i|)$, $X, \dots, X$, i.e., $\texttt{sched}_i = s_i, s_i, \dots = s_i^\omega$, where ω stands for the infinite repetition of s_i.

In Definition 3.1, $\mathcal{M}_i(e)$, $e \in \{1, \dots, |\mathcal{M}_i|\}$ stands for the e-th entry of $\mathcal{M}_i$ and represents a communication event for team with index $\mathcal{M}_i(e)$, and the discrete states X indicate that there is no communication event for robot i. The length of sequence s_i is $\ell = \max\{d_{\mathcal{T}_m}\}_{m=1}^{M} + 1$ for all $i \in \mathcal{N}$, where $d_{\mathcal{T}_m}$ is the degree of node $m \in \mathcal{V}_{\mathcal{T}}$ [10]. The schedule $\texttt{sched}_i$ defines the order in which robot i participates in communication events for the teams $\mathcal{T}_m$, $m \in \mathcal{M}_i$, for all robots $i \in \mathcal{N}$. Specifically, robot i either has to communicate with all robots that belong to team $\mathcal{T}_m$, $m \in \mathcal{M}_i$, if $\texttt{sched}_i(n_i) = m$, or does not need to participate in any communication event if $\texttt{sched}_i(n_i) = X$, where $\texttt{sched}_i(n_i)$ stands for the n_i-th entry of $\texttt{sched}_i$ and $n_i \in \mathbb{N}_+$.

Remark 3.2 (Discrete states X): In $\texttt{sched}_i$, defined in Definition 3.1, the states X indicate that no communication event for robot i. These states are used to ensure that the communication event for a team $\mathcal{T}_m$ is placed at an entry of s_i, $m \in \mathcal{M}_i$, with index that is common for all robots in team $\mathcal{T}_m$; see, e.g., the communication schedules for a network of $N = 3$ robots in Figure 1. Nevertheless, as it will be shown in Proposition 5.1 in Section V, it is the order of communication events in $\texttt{sched}_i$ that is critical to ensure intermittent communication, not the indices of entries in s_i where the communication events are placed. This is due to a control policy applied to robots that are present in communication points; see Section IV-D.

IV. Integrated Task Planning and Intermittent Communication Control

In this section, we synthesize paths $\mathcal{P}_i(t)$ that satisfy the assigned tasks $\mathcal{H}_i^{\text{cur}}(t)$, the intermittent connectivity requirement, and minimize the total cost $\sum_{i\in\mathcal{N}} J(\mathcal{P}_i(t))$. To achieve this, we select communication points that are introduced in the dynamic tasks $\mathcal{H}_i^{\text{cur}}(t)$ so that the total cost $\sum_{i\in\mathcal{N}} J(\mathcal{P}_i(t))$ is minimized.

A. Construction of paths $\mathcal{P}_i(t)$

1) Initialization: Using the schedules $\texttt{sched}_i$, we design the initial paths $\mathcal{P}_i(t_0)$, where t_0 stands for the initial time instant, that include (i) all waypoints in $\mathcal{H}_i^{\text{cur}}(t_0)$ in the order they appear in $\mathcal{H}_i^{\text{cur}}(t_0)$, and (ii) the user-specified communication points $\mathbf{v}_j$, $j\in\mathcal{C}_m$, for all teams $\mathcal{T}_m$, $m\in\mathcal{M}_i$. Specifically, first the paths $\mathcal{P}_i(t_0)$ are initialized as $\mathcal{P}_i(t_0)=\mathcal{H}_i^{\text{cur}}(t_0)$. If the task specifications $\mathcal{H}_i^{\text{cur}}(t_0)$ are not available, then the paths $\mathcal{P}_i(t_0)$ are initialized as $\mathcal{P}_i(t_0)=\emptyset$. Then, the paths $\mathcal{P}_i(t_0)$ are updated by incorporating into them the user-specified communication points $\mathbf{v}_j$, $j\in\mathcal{C}_m$, for all teams $\mathcal{T}_m$, $m\in\mathcal{M}_i$. The index $k_i^m\in\{1,\dots,K_i(t_0)\}$ of the entry in $\mathcal{P}_i(t_0)$ where the communication point $\mathbf{v}_j$, $j\in\mathcal{C}_m$ for team $\mathcal{T}_m$ will be placed can be selected either arbitrarily or optimally so that the cost function $J(\mathcal{P}_i(t_0))$ is minimized. The only requirement is that the communication points are introduced in $\mathcal{P}_i(t_0)$ in the order the respective communication events appear in $\texttt{sched}_i$, for all $i\in\mathcal{N}$. In this way, we ensure that the communication events during the execution of the paths $\mathcal{P}_i(t_0)$ will occur in the order determined by the $\texttt{sched}_i$. Note that during the initialization phase, the X's that appear in $\texttt{sched}_i$ are ignored and are not introduced in the paths $\mathcal{P}_i(t_0)$.

2) Online Construction: At any time t every robot i can update the current task $\mathcal{H}_i^{\text{cur}}(t)$ by appending additional waypoints from $\mathcal{H}_i$, as discussed in Section II. The additional waypoints are appended to the paths $\mathcal{P}_i(t)$, as well. Moreover, when the robots $i\in\mathcal{T}_m$ meet at the respective communication point that appears in their paths $\mathcal{P}_i(t)$, they communicate and coordinate to select the next communication point for team $\mathcal{T}_m$, the time instant when they will communicate again, and design their corresponding paths $\mathcal{P}_i(t^+)$ that they will have at the time instant t^+, i.e., right after leaving this communication point. This coordination process is described in Sections IV-B-IV-C.

B. Selection of Next Communication Point

To select the next communication point $\mathbf{v}_j$, $j\in\mathcal{C}_m$ for team $\mathcal{T}_m$ and incorporate it into $\mathcal{P}_i(t)$ giving rise to the paths $\mathcal{P}_i(t^+)$, the robots $i\in\mathcal{T}_m(t)$ solve the following integer program.

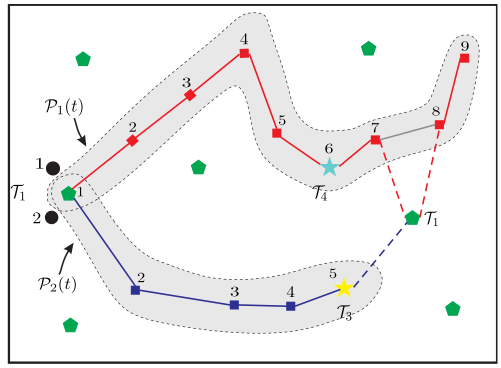


Fig. 2. Graphical illustration of optimization problem (3). Robots 1 and 2 (black dots) in team $\mathcal{T}_1$ meet at the selected communication point $\mathcal{P}_1^1(t)=\mathcal{P}_2^1(t)$ (green polygon) and coordinate to select the next communication point for team $\mathcal{T}_1$. Red and blue squares stand for the waypoints that robots 1 and 2 have to visit to accomplish their tasks, respectively. The communication points for teams $\mathcal{T}_2$ and $\mathcal{T}_4$ are represented by colored stars. The resulting paths $\mathcal{P}_1(t^+)$ and $\mathcal{P}_2(t^+)$ comprise the red and blue, both solid and dashed, lines. The gray line stands for an edge in the path $\mathcal{P}_1(t)$ that does not exist in the path $\mathcal{P}_1(t^+)$ due to the introduction of the communication point for team $\mathcal{T}_1$. The schedules of robots 1 and 2 are $\texttt{sched}_1=[1,X,4]$ and $\texttt{sched}_2=[1,X,3]$. Observe that the communication points appear in $\mathcal{P}_1(t^+)$ and $\mathcal{P}_2(t^+)$ respect the corresponding schedules for both robots.

$$\underset{\mathbf{v}_{j\in\mathcal{C}_m},\{k_i^m\}_{\forall i\in\mathcal{T}_m}}{\text{minimize}} \quad \sum_{i\in\mathcal{T}_m} J(\mathcal{P}_i(t^+)) \tag{3a}$$

subject to

$$\mathcal{P}_i^{k_i^m}(t^+)=\mathbf{v}_j, \tag{3b}$$
$$k_i^m > k_i^{\text{LC}}(t), \tag{3c}$$
$$k_i^m \geq k_i^a(t), \text{where}, \tag{3d}$$
$$K_i(t^+)\geq k_i^a(t)\geq \min(k_i^{\text{LC}}(t)+2, K_i(t^+)),$$
$$k_i^m \leq k_i^b(t), \text{where } K_i(t^+)\geq k_i^b(t)\geq k_i^a(t), \tag{3e}$$

In the optimization problem (3) the paths $\mathcal{P}_i(t^+)$ are initialized as $\mathcal{P}_i(t^+)=\mathcal{P}_i(t)$. In the objective function (3a), $J(\mathcal{P}_i(t^+))$ stands for the cost of the path $\mathcal{P}_i(t^+)$ defined in (1). Also, $K_i(t^+)$ stands for the number of waypoints in $\mathcal{P}_i(t^+)$. Note that $K_i(t^+)=K_i(t)+1$ since $\mathcal{P}_i(t^+)$ includes all waypoints of $\mathcal{P}_i(t)$ and the next communication point for team $\mathcal{T}_m$ that does not exist in $\mathcal{P}_i(t)$. Moreover, k_i^m represents the index of the entry in $\mathcal{P}_i(t^+)$ where the selected communication point $\mathbf{v}_j$, $j\in\mathcal{C}_m$ will be placed, i.e., $\mathcal{P}_i^{k_i^m}(t^+)=\mathbf{v}_j$, $j\in\mathcal{C}_m$.

The first constraint (3b) requires that all robots $i\in\mathcal{T}_m$ will select the same communication point $\mathbf{v}_j$, $j\in\mathcal{C}_m$ for the next communication event associated with team $\mathcal{T}_m$ and incorporate it into the entry of $\mathcal{P}_i(t^+)$ with index k_i^m. The second constraint (3c) ensures that all communication points $\mathbf{v}_j$, $j\in\mathcal{C}_m$, for every team $\mathcal{T}_m$, $m\in\mathcal{M}_i$ are introduced in $\mathcal{P}_i(t^+)$ in the order that the respective communication events appear in s_i. In particular, in the second constraint, the index $k_i^{\text{LC}}(t)$ is defined as the index of the entry in $\mathcal{P}_i(t)$ where the last communication point has been introduced, i.e., none of the waypoints $\mathcal{P}_i^{k_i}(t)$, for all $k_i\in\{k_i^{\text{LC}}(t)+1,\dots,K_i(t)\}$

is a communication point. This constraint requires that robot i will participate at the next communication event for team $\mathcal{T}_m$ only after it has visited all other communication points that already exist in $\mathcal{P}_i(t^+)$, for all robots $i \in \mathcal{T}_m$. This combined with the fact that the communication points appear in the path $\mathcal{P}_i(t_0)$ in the order determined by $\texttt{sched}_i$, for all $i \in \mathcal{N}$, entails that the communication points are introduced into all subsequent paths $\mathcal{P}_i(t)$, for all $t > t_0$ in the order that is determined by $\texttt{sched}_i = s_i^{\omega}$, as well, for all $i \in \mathcal{N}$; see also Example 4.1. As discussed in Remark 3.2, and as it will be shown in Proposition 5.1, this constraint ensures that the network never reaches a deadlock configuration and guarantees intermittent communication infinitely often. Notice that the symbols X that appear in the schedules $\texttt{sched}_i = s_i^{\omega}$ are ignored and are not introduced in $\mathcal{P}_i(t^+)$.

The last two constraints (3d)-(3e) are additional constraints for k_i^m and determine how frequently communication events should occur. Specifically, the second constraint requires that the index k_i^m is greater than $k_i^a(t)$ which is also an index of entries of the path $\mathcal{P}_i(t)$ and can change with time. The index $k_i^a(t)$ is selected under the following two requirements. First, to ensure feasibility of the optimization problem (3a), $k_i^a(t)$ should satisfy $K_i(t^+) \geq k_i^a(t)$, for all $n_i \geq 1$, since there are only $K_i(t^+)$ possible entries for the communication point $\mathbf{v}_j$, $j \in \mathcal{C}_m$ in the path $\mathcal{P}_i(t^+)$. Second, we require that $k_i^a(t)$ is selected so that $k_i^a(t) \geq \min(k_i^{\mathrm{LC}}(t) + 2, K_i(t^+))$. Essentially, this requirement motivates robot i to visit $k_i^a(t) - k_i^{\mathrm{LC}}(t) - 1$ waypoints in the path $\mathcal{P}_i(t^+)$ that are not communication points, if there are such waypoints, before communicating with another team; see also Figure 2. As it will be discussed in Theorem 5.2, this ensures that robot i will accomplish its assigned task, i.e., it will eventually visit all waypoints associated with the assigned task in the path $\mathcal{P}_i(t^+)$. The last constraint is similar to the second one and it requires that k_i^m is smaller than $k_i^b(t)$, which is also an index of entries in the path $\mathcal{P}_i(t)$. The index $k_i^b(t)$ is selected so that the inequality $K_i(t^+) \geq k_i^b(t) \geq k_i^a(t)$ is satisfied to ensure feasibility of (3). In other words, the last two constraints require that the index of the next communication point for team $\mathcal{T}_m$ in the path $\mathcal{P}_i(t^+)$ should belong to $[k_i^a(t), k_i^b(t)] \subseteq \mathbb{N}$. Finally, notice that the indices k_i^m are not required to be the same for all robots in team $\mathcal{T}_m$.

Example 4.1 (Optimization Problem (3)*):* Assume that communication within team $\mathcal{T}_1 = \{1, 2\}$ happens, as shown in Figure 2, and its members coordinate to select their next communication point. Observe in Figure 2 that $K_1(t) = 9$, $K_2(t) = 5$, $K_1(t^+) = 10$, and $K_2(t^+) = 6$. Also, observe that $k_1^{\mathrm{LC}}(t) = 6$ and $k_2^{\mathrm{LC}}(t) = 5$. The parameter $k_1^a(t)$ should satisfy $10 \geq k_1^a(t) \geq \min\{8, 10\} = 8$. In this example, we select $k_1^a(t) = 8$, which means that robot i has to visit at least $k_1^a(t) - k_1^{\mathrm{LC}}(t) - 1 = 1$ waypoint before communicating again with team $\mathcal{T}_1$, after the last communication event at $\mathcal{P}_i^{k_i^{\mathrm{LC}}(t)}(t)$ for team $\mathcal{T}_4$. The parameter $k_1^b(t)$ should satisfy $k_1^b(t) \geq k_1^a(t) = 8$ and we select $k_1^b(t) = 9$. As for robot 2, we have that $k_2^a(t)$ should satisfy $6 \geq k_2^a(t) \geq \min\{6, 7\} = 6$ and, therefore, we select $k_2^a(t) = 6$. Also, $k_2^b(t)$ should satisfy $6 \geq k_2^b(t) \geq 6$ and, thus, we select $k_2^b(t) = 6$. Observe also that $k_2^a(t) - k_2^{\mathrm{LC}}(t) - 1 = 0$, i.e., robot 2 will not visit any waypoints associated with the assigned task after the communication event for team $\mathcal{T}_3$, since there are no such waypoints.

Remark 4.2 ($k_i^a(t)$ and $k_i^b(t)$): In practice, $k_i^a(t)$ can be selected so that robot i collects a sufficiently large amount of information before communicating with another team. On the other hand, $k_i^b(t)$ controls the amount of information that robot i is allowed to hold onto before sharing it with other robots. For example, if robot i is expected to collect highly critical information at the next waypoints, then $k_i^b(t)$ should be selected small, so that the collected data can be propagated to the network, as soon as possible. Thus, $k_i^a(t)$ and $k_i^b(t)$ can control the frequency at which communication events occur. Moreover, $k_i^b(t)$ can also capture buffer constraints as, e.g., in [11].

C. Selection of Next Meeting Time Instant

In Section II we assumed that communication between robots in team $\mathcal{T}_m(t)$ happens only when all robots in that team are simultaneously present a common communication location $\mathbf{v}_j$, $j \in \mathcal{C}_m$. Nevertheless, this control policy can lead to large waiting times which can be undesirable in case of time-critical missions, or it can even be infeasible if the robots cannot stay stationary, e.g., due to their dynamics. To avoid waiting delays, once the robots $i \in \mathcal{T}_m(t)$ select their next communication point, they also compute the time instant t^m at which they will communicate again, so that the waiting time at that next communication point is zero.

The computation of the time instant t^m, called also *meeting time instant*, is described in Algorithm 1. First, every robot $i \in \mathcal{T}_m$ computes the last time instant $L_i \in \mathbb{R}_+$ that it needs to participate in a communication event during the execution of the path $\mathcal{P}_i(t)$, i.e., $L_i = \max_{e \in \mathcal{M}_i}(t^e)$ [line 2, Alg. 1]. This communication event takes place at the communication point $\mathcal{P}_i^{k_i^{\mathrm{LC}}(t)}(t) = \mathcal{P}_i^{k_i^{\mathrm{LC}}(t)}(t^+)$, where $k_i^{\mathrm{LC}}(t)$ was defined in Section IV-B. Second, given the next communication point $\mathbf{v}_j$, $j \in \mathcal{C}_m$, for team $\mathcal{T}_m$, determined by the solution of (3), every robot $i \in \mathcal{T}_m(t)$ computes the minimum time required to travel from the location $\mathcal{P}_i^{k_i^{\mathrm{LC}}(t)}(t^+)$ to $\mathbf{v}_j = \mathcal{P}_i^{\texttt{index}_i^m}(t^+)$, denoted by t_i^{tr} [line 3, Alg. 1]. Then, any time instant $t^m \geq \max_{i \in \mathcal{T}_m(t)}(L_i + t_i^{\mathrm{tr}})$, is a *feasible* time instant.[2] Here, feasibility of t^m means that there exists a controller which given the robot dynamics can drive robot i from the communication point $\mathcal{P}_i^{k_i^{\mathrm{LC}}(t)}(t^+)$ to $\mathbf{v}_j$ within $t^m - L_i$ time units, for all robots $i \in \mathcal{T}_m$. Design of such a control input for arbitrary robot dynamics is out of the scope of this paper. In this work, we select $t^m = \max_{i \in \mathcal{T}_m(t)}(L_i + t_i^{\mathrm{tr}})$ [line 4, Alg. 1].

[2]Note that here we assume that if robot i can arrive at a location $\mathbf{v}_j$ at time instant $L_i + t_i^{\mathrm{tr}}$, then it can also arrive at $\mathbf{v}_j$ at $t^m \geq L_i + t_i^{\mathrm{tr}}$ which is a reasonable assumption as long as the the robot velocities are not bounded below or have sufficiently small lower bounds.

Algorithm 1: Computation of meeting time instant t^m

Input: Paths $\mathcal{P}_i(t^+)$, robot dynamics $\dot{\mathbf{x}}_i(t) = \mathbf{f}_i(\mathbf{x}_i(t), \mathbf{u}_i(t))$, for all robots $i \in \mathcal{T}_m$, $\mathbf{v}_{j \in \mathcal{C}_m}$

Output: Meeting time instant t^m

1 **for** $i \in \mathcal{T}_m$ **do**

2 Compute time at which the latest communication event for robot i will occur: $L_i = \max_{e \in \mathcal{M}_i}(t^e)$;

3 Given dynamics $\dot{\mathbf{x}}_i(t) = \mathbf{f}_i(\mathbf{x}_i(t), \mathbf{u}_i(t))$ and the next communication point $\mathbf{v}_j$, $j \in \mathcal{T}_m$, compute the minimum required travel time, denoted by t_i^{tr}, from $\mathcal{P}_i^{k_i^{\text{LC}}(t)}(t^+)$ to $\mathbf{v}_j$;

4 $\mathcal{T}_m$: $t^m = \max_{i \in \mathcal{T}_m(t)}(L_i + t_i^{\text{tr}})$;

Remark 4.3 (Initial Meeting Time Instants): Given the initial paths $\mathcal{P}_i(t_0)$, feasible initial meeting time instants t^m can be designed if every robot $i \in \mathcal{N}$ runs Algorithm 1 for each team $\mathcal{T}_m$, $m \in \mathcal{M}_i$ in the order determined by $\texttt{sched}_i$. Notice that all robots in team $\mathcal{T}_m$ will run Algorithm 1 for team $\mathcal{T}_m$ at the same time, by construction of $\texttt{sched}_i$, and will collectively compute t^m. In this initialization process, (i) in line 2, we set $t^e = t_0$ for all t^e, $e \in \mathcal{M}_i(t_0)$, that have not been computed at previous iterations of Algorithm 1, and (ii) in line 3, we compute the travel time from the communication point that appears right before the communication point of team $\mathcal{T}_m$ in $\mathcal{P}_i(t_0)$ (or $\mathbf{x}_i(t_0)$ if this communication point is not defined) to the communication point of $\mathcal{T}_m$. We show via simulations that even if the initial meeting time instants are selected arbitrarily and, as a result, they are not necessarily feasible, resulting in non-zero waiting times, the waiting time will eventually become zero for all teams $\mathcal{T}_m$, $m \in \mathcal{M}$. By construction of Algorithm 1, if the initial meeting time instants are feasible, then the waiting time will always be zero for all teams $\mathcal{T}_m$, $m \in \mathcal{M}$.

D. Online Execution of Paths $\mathcal{P}_i(t)$

In this section we discuss the online execution of the paths $\mathcal{P}_i(t)$, for all $t \geq t_0$. Given the paths $\mathcal{P}_i(t)$, for any $t \geq t_0$, robots start moving towards the next unvisited waypoint in the path $\mathcal{P}_i(t)$, i.e., the first waypoint in $\mathcal{P}_i(t)$, denoted by $\mathcal{P}_i^1(t)$, since visited waypoints are deleted from $\mathcal{P}_i(t)$ [line 2, Alg. 2]. When robot i reaches the waypoint $\mathcal{P}_i^1(t)$, it checks if this location corresponds to a communication point associated with a team $\mathcal{T}_m$, $m \in \mathcal{M}_i$. If this is not the case, then robot i deletes the waypoint $\mathcal{P}_i^1$ from the path $\mathcal{P}_i(t)$, as there is no need to store it anymore, and moves towards the next waypoint $\mathcal{P}_i^1(t)$ [line 7, Alg. 2]. Otherwise, if $\mathcal{P}_i^1(t) = \mathbf{v}_j$, $j \in \mathcal{C}_m$, $m \in \mathcal{M}_i$ robot i communicates and coordinates with all other robots in team $\mathcal{T}_m$ to select next communication point, next meeting time instant, and design new paths $\mathcal{P}_i(t^+)$ [line 5, Alg. 2]. Note that uncertainty and exogenous disturbances may affect the arrival times of the robots at the communication points. Thus, if the robots $i \in \mathcal{T}_m$ are not able to arrive at $\mathbf{v}_j$, $j \in \mathcal{C}_m$ at the meeting time

Algorithm 2: Online Execution of $\mathcal{P}_i(t)$, $t \geq t_0$

Input: Sets $\mathcal{C}_m$, $\forall m \in \mathcal{M}$, $\mathcal{P}_i(t_0)$

1 **while** $\texttt{Terminate}$*=0* **do**

2 Move towards $\mathcal{P}_i^1(t)$;

3 **if** $(\mathbf{x}_i(t) = \mathcal{P}_i^1(t)) \wedge (\mathcal{P}_i^1(t) = \mathbf{v}_j,\ j \in \mathcal{C}_m)$, $m \in \mathcal{M}_i(t)$ **then**

4 Wait until all robots of team $\mathcal{T}_m(t)$ arrive;

5 Select next communcation point, next meeting time, and design new paths ;

6 **if** $(\mathbf{x}_i(t) = \mathcal{P}_i^1(t))$ **then**

7 Delete the visited waypoint $\mathcal{P}_i^1(t)$ from $\mathcal{P}_i(t)$;

instants t^m computed in Section IV-C, then they *wait* for each other until all of them arrive at $\mathbf{v}_j$, $j \in \mathcal{C}_m$ [line 4, Alg. 2].

V. Correctness

In this section, we present results pertaining to correctness of the proposed control scheme. Specifically, in Theorems 5.2 and 5.3, we show that when the robots follow the paths $\mathcal{P}_i(t)$ both the assigned task and the intermittent communication requirement are satisfied. To show these results, we first need to show that the system is *deadlock-free* when the paths $\mathcal{P}_i(t)$ are executed as discussed in Section IV-D. Specifically, we assume that there is a *deadlock*, if there are robots in any team $\mathcal{T}_m$ that are waiting forever at a communication point for the arrival of all other robots in team $\mathcal{T}_m$. The proof of Proposition 5.1 is the same as the proof of Proposition 7.3 in [10] and, therefore, is omitted.

Proposition 5.1 (Deadlock): The mobile robot network is deadlock-free when the paths $\mathcal{P}_i(t)$ are executed as in Section IV-D.

Theorem 5.2 (Task): Construction and execution of paths $\mathcal{P}_i(t)$ as per the proposed algorithm ensures that all robots will accomplish the assigned task.

Proof: First, note that by construction of $\mathcal{P}_i(t)$, the paths $\mathcal{P}_i(t)$ preserve the order in which task waypoints appear in $\mathcal{H}_i^{\text{cur}}(t)$. Therefore, to show this result, it suffices to show that there exists a time instant $t_i' \geq t$ when robot i will visit all task waypoints that appear in the path $\mathcal{P}_i(t)$, for all $t \geq t_0$ and for all $i \in \mathcal{N}$. This is shown by contradiction. Specifically, assume that robot i will never visit any of the task waypoints that appear in $\mathcal{P}_i(t)$. This can happen in two cases. First, this may occur if the network reaches a deadlock configuration which cannot happen due to Proposition 5.1. Second, this may happen if robot i always introduces the communication points for all teams $\mathcal{T}_m$, $m \in \mathcal{M}_i$, in consecutive entries of its path $\mathcal{P}_i(t)$ and, therefore, never visits any task waypoints between any two consecutive communication events. Nevertheless, this cannot happen, if there are task waypoints in $\mathcal{P}_i(t)$, due to the second constraint in (3) completing the proof. ■

Theorem 5.3 (Intermittent Connectivity): Construction and execution of paths $\mathcal{P}_i(t)$ as per the proposed algorithm ensures that that the dynamic communication graph $\mathcal{G}_c(t)$ is connected over time infinitely often.

Proof: To show this result, it suffices to show that the time interval between two consecutive communication events for all teams $\mathcal{T}_m$ is finite. This is because every robot belongs to at least one team $\mathcal{G}_{\mathcal{T}}$ is connected. First recall, at any time instant $t \geq t_0$, there exists a communication point for all teams $\mathcal{T}_m$, $m \in \mathcal{M}_i$, in the paths $\mathcal{P}_i(t)$, for all $i \in \mathcal{N}$. Second, recall that due to Proposition 5.1, it holds that the waiting times of robots at the communication points are bounded and, therefore, the network is deadlock-free. Therefore, we conclude that robots in team $\mathcal{T}_m$ will eventually communicate again, for all $m \in \mathcal{M}$ completing the proof. ■

Remark 5.4 (Comparison with [10]): Algorithm 2 in [10] incorporates optimally communication points in periodic paths, which are known *a priori*, by exploiting their periodicity. Nevertheless, here, we consider arbitrary tasks that are not necessarily satisfied by periodic paths. As a result, (i) the optimality guarantees provided in Proposition 7.1 in [10] do not hold here, and (ii) if we apply both algorithms to periodic tasks the resulting paths will be different.

VI. Simulation Studies

In this section, a simulation study is provided that illustrates our approach for a network of $N = 15$ robots that reside in a 10×10 square workspace free of obstacles. Robots are categorized into $M = 12$ teams as follows: $\mathcal{T}_1 = \{1, 2, 9\}$, $\mathcal{T}_2 = \{3, 4, 5\}$, $\mathcal{T}_3 = \{3, 6, 13\}$, $\mathcal{T}_4 = \{1, 3, 14\}$, $\mathcal{T}_5 = \{2, 5, 6, 11\}$, $\mathcal{T}_6 = \{4, 12, 14\}$, $\mathcal{T}_7 = \{5, 9, 15\}$, $\mathcal{T}_8 = \{4, 9, 12\}$, $\mathcal{T}_9 = \{6, 7, 10, 15\}$, $\mathcal{T}_{10} = \{7, 8, 11\}$, $\mathcal{T}_{11} = \{8, 10, 11, 12\}$, and $\mathcal{T}_{12} = \{7, 10, 13\}$ resulting in a connected graph $\mathcal{G}_{\mathcal{T}}$. In the workspace, there are $R = 60$ communication points that are randomly located in $\mathcal{W}$, where we select $|\mathcal{C}_m| = 5$, for all $m \in \mathcal{M}$ and $\mathcal{C}_m \cap \mathcal{C}_n = \varnothing$, for all $m, n \in \mathcal{M}$. Also, we assume that the robot dynamics are given by $\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t)$, $\|\mathbf{u}_i(t)\| \leq u^i_{\text{max}}$.

Robot 1 has to follow a finite path with $H_1 = 4$, which is randomly generated at the beginning capturing point-to-point navigation tasks [14] or co-safe LTL tasks [15]. For all the other robots we select $H_i = \infty$. Specifically, we assume that robots 2 and N have to follow periodic paths forever to accomplish their assigned tasks. These paths are randomly generated at the beginning resembling in this way surveillance [16], estimation [17], or LTL tasks [11].[3] Also, the periodic path $\mathcal{P}_N$ goes through a user that receives the collected information. All the other robots have to follow infinite and aperiodic paths. These robots initially construct finite paths which are randomly generated and then they extend those paths at random time instants by a number of waypoints that is randomly selected from $[1, 10]$ resembling tasks in unknown or dynamic environments as, e.g., in [18], [19], or receding horizon planning approaches [13], [20].

[3]For periodic paths, $K_i(t)$ can be selected arbitrarily large, since it can be viewed as an infinite and known path. As a result, if $K_i(t)$ is greater than the period of a periodic path, then during a single execution of this periodic path, robot i may not necessarily communicate with all teams $\mathcal{T}_m$, $m \in \mathcal{M}_i$, which is not the case in [10].

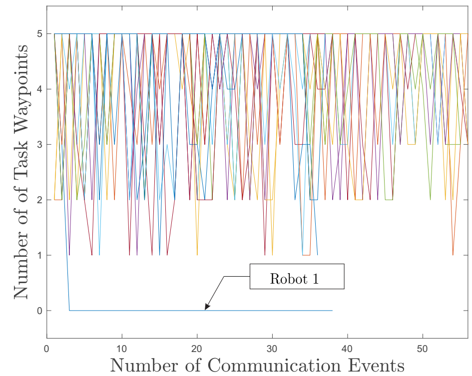


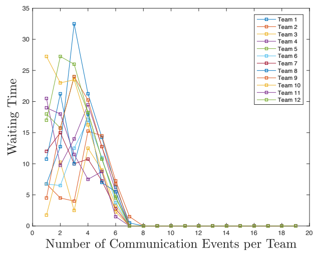
Fig. 3. Graphical depiction of number of waypoints (solid lines) associated with the assigned task that were visited by all robots between consecutive communication events.

The schedules of communication events have the following form.

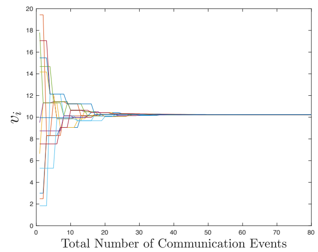
$$
\begin{aligned}
&\texttt{sched}_1 = [1,\ 4,\ X,\ X]^\omega,\ \texttt{sched}_9 = [1,\ 8,\ 7,\ X]^\omega,\\
&\texttt{sched}_2 = [1,\ 5,\ X,\ X]^\omega,\ \texttt{sched}_{10} = [9,\ 12,\ X,\ 11]^\omega,\\
&\texttt{sched}_3 = [2,\ 4,\ 3,\ X]^\omega,\ \texttt{sched}_{11} = [X,\ 5,\ 10,\ 11]^\omega,\\
&\texttt{sched}_4 = [2,\ 8,\ 6,\ X]^\omega,\ \texttt{sched}_{12} = [X,\ 8,\ 6,\ 11]^\omega,\\
&\texttt{sched}_5 = [2,\ 5,\ 7,\ X]^\omega,\ \texttt{sched}_{13} = [X,\ 12,\ 13,\ X]^\omega,\\
&\texttt{sched}_6 = [9,\ 5,\ 3,\ X]^\omega,\ \texttt{sched}_{14} = [X,\ 4,\ 6,\ X]^\omega,\\
&\texttt{sched}_7 = [9,\ 12,\ 10,\ X]^\omega,\ \texttt{sched}_{15} = [9,\ X,\ 7,\ X]^\omega,\\
&\texttt{sched}_8 = [X,\ X,\ 10,\ 11]^\omega.
\end{aligned}
$$

Moreover, we select $k^a_i(t) = \min(K_i(t^+), k^{\text{LC}}_i(t) + 2)$, for all $i \in \mathcal{N}$, which means that robot i has to visit at least one waypoint associated with the assigned task, if there exists such a waypoint in $\mathcal{P}_i(t)$, between consecutive communication events. Also, we assume that every time the robots visit a waypoint related to the assigned task, they collect one packet of information while they should never keep more than five packets that have never been transmitted to other robots. To capture such limitations, we select $k^b_i(t) = \min(K_i(t^+), k^{\text{LC}}_i(t) + 6)$, for all $i \in \mathcal{N}$. Notice that the selected values for $k^a_i(t)$ and $k^b_i(t)$ meet all the requirements described in Section IV-B to guarantee feasibility of the optimization problem (3), for all $t > t_0$. Observe in Figure 3 that all robots visit at least one and at most five waypoints related to the assigned task between consecutive communication events, as required. Note also that in this simulation study, it always holds that $k^a_i(t) = k^{\text{LC}}_i(t) + 2$ and $k^b_i(t) = k^{\text{LC}}_i(t) + 6$, for all $t \geq t_0$ and for all robots $i \neq 1$. Since robot 1 has to follow a finite path to accomplish its task, there exists a time instant t', where all the locations in its path $\mathcal{P}_1(t)$ are only communication points, for all $t \geq t'$. Therefore, there are no waypoints related to the assigned task between communication points.

Also, the initial meeting time instants are selected as $t^m = t_0$, for all teams $\mathcal{T}_m$, $m \in \mathcal{M}$, which are clearly infeasible. The resulting waiting times of all robots at the



(a) Waiting Time



(b) Consensus

Fig. 4. Figure 4(a) shows the waiting time of team $\mathcal{T}_m$, every time the robots $i \in \mathcal{T}_m$ communicate, for all $m \in \mathcal{M}$. The waiting time of team $\mathcal{T}_m$ is defined as the maximum waiting time of all robots $i \in \mathcal{T}_m$ at the selected communication point $\mathbf{v}_j$, $j \in \mathcal{C}_m$. Figure 4(b) depicts the consensus of numbers $v_i(t)$. In Figure 4(b), the number of iterations in the x-axis is increased by one every time any team communicates.

communication points are depicted in Figure 4(a). Observe that eventually the waiting time at the selected communication points is zero for all teams despite the initially infeasible meeting time instants.

To illustrate that the proposed motion plans ensure intermittent communication among the robots infinitely often, we implement a simple consensus algorithm over the dynamic network $\mathcal{G}_c$. Specifically, we assume that initially all robots generate a random number $v_i(t_0)$ and when all robots $i \in \mathcal{T}_m$ meet at $\mathbf{v}_j$, $j \in \mathcal{C}_m$ they perform the following consensus update $v_i(t) = \frac{1}{|\mathcal{T}_m|}\sum_{e\in\mathcal{T}_m} v_e(t)$. Figure 4(b) shows that eventually all robots reach a consensus on the numbers $v_i(t)$, which means that communication among robots takes place infinitely often, as proven in Theorem 5.3. The simulation video along with its description can be found in [21].

VII. Conclusion

In this paper, we proposed a distributed intermittent communication framework for teams of mobile robots with limited communication ranges that are responsible for accomplishing time-critical dynamic tasks. To the best of our knowledge, this is the first distributed and online intermittent communication framework that scales well with the size of the network and can handle time-critical dynamic tasks and arbitrary communication topologies.

References

[1] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428, 2008.

[2] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[3] M. Zavlanos, M. Egerstedt, and G. Pappas, "Graph theoretic connectivity control of mobile robot networks," *Proc. of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[4] Y. Yan and Y. Mostofi, "Robotic router formation in realistic communication environments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 810–827, 2012.

[5] M. M. Zavlanos, A. Ribeiro, and G. J. Pappas, "Network integrity in mobile robotic networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 3–18, 2013.

[6] Y. Kantaros and M. M. Zavlanos, "Global planning for multi-robot communication networks in complex environments," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1045–1061, 2016.

[7] Y. Kantaros and M. M. Zavlanos, "Distributed intermittent connectivity control of mobile robot networks," *Transactions on Automatic Control*, vol. 62, no. 7, pp. 3109–3121, July 2017.

[8] Y. Kantaros and M. M. Zavlanos, "Simultaneous intermittent communication control and path optimization in networks of mobile robots," in *Conference on Decision and Control (CDC)*. Las Vegas, NV: IEEE, December 2016, pp. 1794–1795.

[9] Y. Kantaros and M. M. Zavlanos, "Intermittent connectivity control in mobile robot networks," in *49th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, November, 2015, pp. 1125–1129.

[10] Y. Kantaros, M. Guo, and M. M. Zavlanos, "Temporal task planning and intermittent communication control of mobile robot networks," *IEEE Transactions on Automatic Control (submitted)*, 2017. [Online]. Available: https://arxiv.org/abs/1706.00765

[11] M. Guo and M. M. Zavlanos, "Distributed data gathering with buffer constraints and intermittent communication," in *IEEE International Conference on Robotics and Automation*, Singapore, May 2017, pp. 279–284.

[12] M. M. Zavlanos, "Synchronous rendezvous of very-low-range wireless agents," in *49th IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, USA, December 2010, pp. 4740–4745.

[13] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, May 2010, pp. 4457–4462.

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[15] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *International Conference on Robotics and Automation (ICRA)*, Anchorage, AL, May 2010, pp. 2689–2696.

[16] S. Nardi, T. Fabbri, A. Caiti, and L. Pallottino, "A game theoretic approach for antagonistic-task coordination of underwater autonomous robots in asymmetric threats scenarios," in *OCEANS 2016 MTS/IEEE*, Monterey, CA, September 2016, pp. 1–9.

[17] X. Lan and M. Schwager, "Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016.

[18] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local LTL specifications," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 218–235, 2015.

[19] J. Alonso-Mora, J. A. DeCastro, V. Raman, D. Rus, and H. Kress-Gazit, "Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles," *Autonomous Robots*, pp. 1–24, 2017.

[20] A. Ulusoy and C. Belta, "Receding horizon temporal logic control in dynamic environments," *The International Journal of Robotics Research*, vol. 33, no. 12, pp. 1593–1607, 2014.

[21] SimulationVideo, `https://vimeo.com/255775629`, 2018.