A Distributed LTL-based Approach for Intermittent Communication in Mobile Robot Networks

Yiannis Kantaros and Michael M. Zavlanos

Abstract— **In this paper we develop an intermittent communication framework for mobile robot networks. Intermittent communication provides significantly more flexibility to the robots to accomplish their tasks compared to approaches that enforce communication constraints for all time. We consider robots that move along the edges of a mobility graph and communicate only when they meet at the nodes of that graph giving rise to a dynamic communication network. Assuming that the mobility graph is connected, we design distributed controllers for the robots that determine meeting times at the vertices of the mobility graph so that connectivity of the communication network is ensured over time, infinitely often. We show that this requirement can be captured by a global Linear Temporal Logic (LTL) formula that forces robots to meet infinitely often at the rendezvous points. To generate discrete high-level motion plans for all robots that satisfy the LTL expression, we propose a novel technique that performs an approximate decomposition of the global LTL expression into local LTL expressions and assigns them to the robots. Since the approximate decomposition of the global LTL formula can result in conflicting robot behaviors, we develop a distributed conflict resolution scheme that generates discrete motion plans for every robot, based on the assigned local LTL expressions, whose composition satisfies the global LTL formula. Computer simulations are provided that verify the efficacy of the proposed distributed control scheme.**

I. INTRODUCTION

Communication among robots has been typically modeled using proximity graphs and the communication problem is often treated as preservation of graph connectivity. Graph theoretic methods for connectivity control range from centralized [1], [2] to distributed ones [3]–[6], while a recent survey can be found in [7]. In practice, the above graph-based communication models turn out to be rather conservative, since proximity does not necessarily imply tangible and reliable communication and, therefore, more realistic communication models have recently been proposed [8]–[10].

Common in the above works is that point-to-point or end-to-end network connectivity is required to be preserved for all time. However, this requirement is often very conservative, since limited resources, e.g., transmission power or number of wireless robots, may hinder robots from accomplishing their assigned goals. Motivated by this fact, in this paper we propose a *distributed intermittent communication protocol* for mobile robot networks. In particular, we consider that robots move along the edges of a mobility graph and communicate only when they meet at the vertices of this graph giving rise to a dynamic communication network. Assuming that the mobility graph is connected, we design distributed controllers for the robots that determine meeting times at the vertices of the mobility graph so that connectivity of the communication network is ensured over time, infinitely often. We show that intermittent connectivity of the communication network can be captured by a global Linear Temporal Logic (LTL) formula that forces robots to meet infinitely often at the rendezvous points. Given such a LTL expression, existing model checking techniques [11], [12] can be employed in order to implement correct by construction controllers for all robots.

LTL-based control synthesis and task specification for mobile robots build upon either a bottom-up approach when independent LTL expressions are assigned to robots [13]–[16] or top-down approaches when a global LTL describing a collaborative task is assigned to a team of robots [17], [18], as in our work. Top-down approaches generate a discrete high-level motion plan for all robots using a discretized abstraction of the environment and constructing a synchronous product automaton among the agents and, therefore, they are resource demanding and scale poorly with the number of robots. To mitigate these issues, we propose a novel technique that approximately decomposes the global LTL formula into local ones and assigns them to robots. Since the approximate decomposition of the global LTL formula can result in conflicting robot behaviors we develop a distributed conflict resolution scheme that generates discrete motion plans for every robot based on the assigned local LTL expressions. To the best of our knowledge, although specific to the problem under consideration, this is the first distributed and scalable LTL-based framework for the coordination of teams of multiple robots.

The most relevant works to the one proposed here are presented in [19]–[21]. Specifically, in [19] an intermittent communication control scheme is proposed. This approach ensures communication among robots infinitely often, however, this method is centralized and does not scale well with the number of robots. [20] proposes a distributed synchronization scheme that allows robots that move along the edges of a bipartite mobility graph to meet periodically at the vertices of this graph. Instead, here we make no assumptions on the graph structure on which robots reside or on the communication pattern to be achieved. On the other hand, [21] proposes a receding horizon framework for periodic connectivity that ensures recovery of end-to-end

This work is supported in part by the NSF awards CNS #1261828 and CNS #1302284.

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

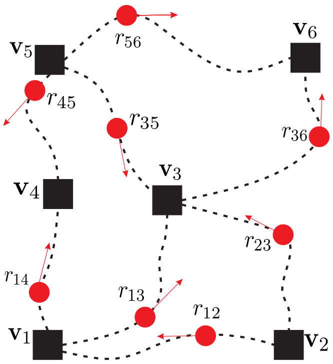

Fig. 1. A graphical illustration of the problem formulation. Black squares represent communication points and red circles stand for robots r_{ij} that move along paths γ_{ij} that are depicted by black dashed curves.

connectivity within a given time horizon. As the number of robots or the size of the time horizon grows, this approach can become computationally expensive. To the contrary, our proposed method scales very well to large numbers of robots and can handle situations where the whole network can not be connected at once, by ensuring connectivity over time, infinitely often.

II. Problem Formulation

Assume R locations in space positioned at $\mathbf{v}_i \in \mathbb{R}^n$ and paths $\gamma_{ij} : [0,1] \to \mathbb{R}^n$ that connect two locations i and j such that $\gamma_{ij}(0) = \mathbf{v}_i$ and $\gamma_{ij}(1) = \mathbf{v}_j$. The union of locations $\mathbf{v}_i$ and paths γ_{ij} gives rise to an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the set of nodes $\mathcal{V} = \{1, 2, \ldots, R\}$ is indexed by the set of locations $\mathbf{v}_i$ and the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is determined by the paths γ_{ij} such that an edge $(i,j) \in \mathcal{E}$ exists if and only if a path γ_{ij} exists. Two nodes i, j are called neighbors in $\mathcal{G}$ if and only if there exists an edge $(i,j) \in \mathcal{E}$ and, thus, we can define the set of neighbors of node i by $\mathcal{N}_i = \{j \in \mathcal{V} | (i,j) \in \mathcal{E}\}$. In what follows, we assume that the graph $\mathcal{G}$ is connected.

Consider also a team of $N = |\mathcal{E}|$ robots so that robot r_{ij} moves back and forth between the nodes i and j in $\mathcal{G}$, along the path γ_{ij}, to possibly accomplish some high-level task (see also Figure 1). We call $\mathcal{G}$ a mobility graph and assume the robots move along the edges of $\mathcal{G}$ according to the following kinematics:

$$\dot{\mathbf{x}}_{ij}(t) = \mathbf{u}_{ij}(t), \tag{1}$$

where $\mathbf{x}_{ij}(t) \in \mathbb{R}^n$ is the position of robot r_{ij} at time t and $\mathbf{u}_{ij}(t) \in \mathbb{R}^n$ is a control action that drives that robot between nodes $\mathbf{v}_i$ and $\mathbf{v}_j$ along the path γ_{ij}.

A. Discretized Abstraction of the Workspace

Since robot r_{ij} moves back and forth between the nodes i and j along the path γ_{ij}, we can construct a transition system (TS) denoted by TS_{ij} to abstract the motion of every robot r_{ij}, which is defined as follows

Definition 2.1 (Transition System): A *transition system* TS_{ij} is a tuple $\left(\mathcal{Q}_{ij}, q_{ij}^0, \mathcal{A}_{ij}, \to_{ij}, \mathcal{AP}, L_{ij}\right)$ where

- $\mathcal{Q}_{ij} = \left\{q_{ij}^{\mathbf{v}_i}, q_{ij}^{\mathbf{v}_j}\right\}$ is the set of states, where the states $q_{ij}^{\mathbf{v}_i}$ and $q_{ij}^{\mathbf{v}_j}$ indicate that robot r_{ij} is at node i and j, respectively,
- $q_{ij}^0 \in \mathcal{Q}_{ij}$ is the initial state,
- $\mathcal{A}_{ij}$ is a set of actions. The available actions in state $q_{ij}^{\mathbf{v}_i}$ are 'go to state $q_{ij}^{\mathbf{v}_j}$' and 'wait in state $q_{ij}^{\mathbf{v}_i}$'.
- $\to_{ij} \subseteq \mathcal{Q}_{ij} \times \mathcal{A}_{ij} \times \mathcal{Q}_{ij}$ is the transition relation,
- $\mathcal{AP}$ is the set of atomic propositions, and
- $L_{ij} : \mathcal{Q}_{ij} \to 2^{\mathcal{AP}}$ is an observation/output relation giving the set of atomic propositions that are satisfied in a state.

In what follows we give definitions related to TS_{ij}, that we will use throughout the rest of the paper.

Definition 2.2 (Infinite Path): An *infinite path* τ_{ij} of TS_{ij} is an infinite sequence of states, $\tau_{ij} = \tau_{ij}(1)\tau_{ij}(2)\tau_{ij}(3)\ldots$ such that $\tau_{ij}(1) = q_{ij}^0$, $\tau_{ij}(k) \in \mathcal{Q}_{ij}$, and $(\tau_{ij}(k), a_{ij}^k, \tau_{ij}(k+1)) \in \to_{ij}$, for some $a_{ij}^k \in \mathcal{A}_{ij}$, $\forall k$.[1]

Definition 2.3 (Composition): Composition of M infinite paths $\tau_m = \tau_m(1)\tau_m(2)\tau_m(3)\ldots$, where $m \in \{1, \ldots, M\}$, denoted by $\tau = \otimes_{\forall m} \tau_m$ is an infinite sequence of states defined as $\tau = \tau(1)\tau(2)\cdots = [\tau(k)]_{k=1}^{\infty}$, where $\tau(k) = (\tau_1(k), \tau_2(k), \ldots, \tau_M(k))$.

Definition 2.4 (Projection): For an infinite path $\tau = \tau(1)\tau(2)\tau(3)\ldots$, we denote by $\Pi|_{\text{TS}_{ij}}\tau$ its projection onto TS_{ij}, which is obtained by erasing all states in τ that do not belong to $\mathcal{Q}_{ij}$.

Definition 2.5 (Trace of infinite path): The trace of an infinite path $\tau_{ij} = \tau_{ij}(1)\tau_{ij}(2)\tau_{ij}(3)\ldots$ of a transition system TS_{ij}, denoted by $\texttt{trace}(\tau_{ij})$, is an infinite word that is determined by the sequence of atomic propositions that are true in the states along τ_{ij}, i.e., $\texttt{trace}(\tau_{ij}) = L_{ij}(\tau_{ij}(1))L_{ij}(\tau_{ij}(2))\ldots$.

Definition 2.6 (Motion Plan): Given a LTL formula ϕ, a transition system TS_{ij} both defined over the set of atomic propositions $\mathcal{AP}$, an infinite path τ_{ij} of TS_{ij} is called *motion plan* if and only if $\texttt{trace}(\tau_{ij}) \in \texttt{Words}(\phi)$, where $\texttt{Words}(\phi) = \left\{\sigma \in (2^{\mathcal{AP}})^\omega | \sigma \models \phi\right\}$ is defined as the set of words $\sigma \in (2^{\mathcal{AP}})^\omega$ that satisfy the LTL ϕ and $\models \subseteq (2^{\mathcal{AP}}) \times \phi$ is the satisfaction relation. The relation $\texttt{trace}(\tau_{ij}) \in \texttt{Words}(\phi)$ is equivalently denoted by $\tau_{ij} \models \phi$.

B. Communication Network

We assume that robots can communicate only if they are physically located at a common rendezvous point. This way, a dynamic robot communication graph $\mathcal{G}_c = \{\mathcal{V}_c, \mathcal{E}_c\}$ is constructed where the set of nodes $\mathcal{V}_c$ is indexed by robots, i.e., $\mathcal{V}_c = \{1, 2, \ldots, N\}$ and $\mathcal{E}_c \subseteq \mathcal{V}_c \times \mathcal{V}_c$ is the set of communication links that emerge among robots when they are located at the same rendezvous point. At every rendezvous point i communication takes place when all robots in the set $\mathcal{R}_i = \{r_{ij} | j \in \mathcal{N}_i\}$ are present at node i, simultaneously. Hence, every robot r_{ij} can directly communicate with all robots that belong to the set $\mathcal{N}_{ij} = \mathcal{R}_i \cup \mathcal{R}_j \setminus \{r_{ij}\}$. Then the communication graph $\mathcal{G}_c$ is defined to be *connected over*

[1] A *finite path* of TS_{ij} can be defined accordingly. The only difference with the infinite path is that a finite path is defined as a finite sequence of states of TS_{ij}.

time if all robots in $\mathcal{R}_i$ meet at the rendezvous point i infinitely often, for all nodes $i \in \mathcal{V}$. Such a requirement can be captured by the following global LTL expression:

$$\phi = \bigwedge_{i \in \mathcal{V}} \left(\Box \Diamond \bigwedge_{j \in \mathcal{N}_i} \pi_{ij}^{\mathbf{v}_i} \right), \tag{2}$$

where $\pi_{ij}^{\mathbf{v}_i}$ is an atomic proposition defined as

$$\pi_{ij}^{\mathbf{v}_i} = \begin{cases} 1 & \text{if } \|\mathbf{x}_{ij} - \mathbf{v}_i\| \leq \epsilon \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

for a sufficiently small $\epsilon > 0$, $\bigwedge$ is the conjuction operator, while $\Box$ and $\Diamond$ stand for the temporal operators 'always' and 'eventually', respectively. For more details on LTL, we refer the reader to [11], [12].

Assuming that all robots r_{ij} make transitions synchronously by picking their next state in their respective transition systems, the problem that is addressed in this paper can be stated as:

Problem 1: Given any initial configuration of the robots in the mobility graph $\mathcal{G}$ determine motion plans τ_{ij} for all robots r_{ij} such that the global LTL expression given in (2) is satisfied, i.e., connectivity of the communication graph $\mathcal{G}_c$ is guaranteed over time, infinitely often.

III. Intermittent Communication Control

To solve Problem 1, known centralized model checking techniques can be employed, that typically rely on a discretized abstraction of the environment captured by a TS and the construction of a synchronized product system among all robots of the network. As a result, such approaches are resource demanding and scale poorly with the size of the network. Therefore, a distributed solution is preferred whereby discrete high-level motion plans for every robot can be computed locally across the network. For this purpose, notice first that although the global LTL formula (2) is not decomposable with respect to robots, it can be decomposed in local LTL formulas $\phi_{\mathbf{v}_i}$ associated with the rendezvous nodes $i \in \mathcal{V}$, which are coupled with each other by the conjunction operator $\wedge$. Specifically, we can write $\phi = \bigwedge_{i \in \mathcal{V}} \phi_{\mathbf{v}_i}$, where $\phi_{\mathbf{v}_i}$ is defined as

$$\phi_{\mathbf{v}_i} = \Box \Diamond \left(\bigwedge_{j \in \mathcal{N}_i} \pi_{ij}^{\mathbf{v}_i} \right), \tag{4}$$

and forces all robots $r_{ij} \in \mathcal{R}_i$ to meet infinitely often at the rendezvous point located at $\mathbf{v}_i$.

Given the decomposition of ϕ into local LTL formulas $\phi_{\mathbf{v}_i}$, every robot r_{ij} needs to develop motion plans τ_{ij} so that the composition of plans τ_{im}, $\forall r_{im} \in \mathcal{R}_i$ denoted by $\tau_{\mathbf{v}_i} = \otimes_{r_{im} \in \mathcal{R}_i} \tau_{im}$ and the composition of plans τ_{jn}, $\forall r_{jn} \in \mathcal{R}_j$, denoted by $\tau_{\mathbf{v}_j} = \otimes_{r_{jn} \in \mathcal{R}_j} \tau_{jn}$ satisfy the local LTL expressions $\phi_{\mathbf{v}_i}$ and $\phi_{\mathbf{v}_j}$, respectively. In this way, we can ensure that the composition of τ_{ij}, $\forall r_{ij}$, satisfies the global LTL expression (2), since all local LTL expressions $\phi_{\mathbf{v}_i}$ are satisfied.

Motion plans $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$, $\forall i$, can be constructed using existing tools from model checking theory [11], [12]. However, notice that constructing plans $\tau_{\mathbf{v}_i}$ and $\tau_{\mathbf{v}_j}$, $j \in \mathcal{N}_i$ independently cannot ensure that the robots' behavior in the workspace will satisfy the global LTL formula (2). The reason is that the local LTL formulas $\phi_{\mathbf{v}_i}$ in (4) are not independent from each other, since they are coupled by robots' state in their respective transition systems. In other words, since every robot r_{ij} is responsible for communicating with other robots at vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, this implies that the LTL expressions $\phi_{\mathbf{v}_i}$ and $\phi_{\mathbf{v}_j}$ are coupled with each other by robot r_{ij} through the atomic propositions $\pi_{ij}^{\mathbf{v}_i}$ and $\pi_{ij}^{\mathbf{v}_j}$. Consequently, generating plans $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$ that ignore the LTL expressions $\phi_{\mathbf{v}_{j \in \mathcal{N}_i}}$ may result in conflicting robot behaviors, since the projection of the motion plans $\tau_{\mathbf{v}_i}$ and $\tau_{\mathbf{v}_j}$ onto TS_{ij} may result in two different motion plans τ_{ij}. This means that cases where a robot r_{ij} needs to be located at two different positions in TS_{ij} simultaneously may occur.

To circumvent these issues, we propose a distributed algorithm (Algorithm 1) that implements free-of-conflict discrete motion plans τ_{ij}, $\forall r_{ij}$, using the motion plans $\tau_{\mathbf{v}_i}$ and $\tau_{\mathbf{v}_j}$ constructed by existing model checking algorithms so that the global LTL expression ϕ is satisfied. In what follows, first we construct motion plans $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$. Then our proposed algorithm will be described that constructs non-conflicting robot motion plans τ_{ij} using the motion plans $\tau_{\mathbf{v}_i}$ and $\tau_{\mathbf{v}_j}$ so that the global LTL (2) is satisfied.

A. Construction of motion plans $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$

Given a LTL formula $\phi_{\mathbf{v}_i}$ and the transition systems TS_{ij} of all robots $r_{ij} \in \mathcal{R}_i$ a motion plan $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$ can be constructed using existing automata-based model checking methods [11], [12]. Such methods typically rely on constructing a *Product Büchi Automaton* and checking the non-emptiness of its accepting language. Instead of following such an approach, we take advantage of the simple structure of the transition systems TS_{ij} to construct motion plans $\tau_{\mathbf{v}_i}$ that satisfy $\phi_{\mathbf{v}_i}$. In particular, we consider the infinite path $\tau_{\mathbf{v}_i}$ defined as follows

$$\tau_{\mathbf{v}_i} = \begin{cases} \left[q_{\mathbf{v}_i}^0 q_{\mathbf{v}_i}^1\right] \left[q_{\mathbf{v}_i}^1\right]^\omega, & \text{if } q_{\mathbf{v}_i}^0 \neq q_{\mathbf{v}_i}^1. \\ \left[q_{\mathbf{v}_i}^0\right] \left[q_{\mathbf{v}_i}^0\right]^\omega, & \text{otherwise,} \end{cases} \tag{5}$$

where $q_{\mathbf{v}_i}^0$ is the initial state of robots $r_{ij} \in \mathcal{R}_i$, i.e., $q_{\mathbf{v}_i}^0 = ((q_{ij_1}^0, q_{ij_2}^0, \ldots, q_{ij_{|\mathcal{N}_i|}}^0))$ and $q_{\mathbf{v}_i}^1$ is a state where all robots $r_{ij} \in \mathcal{R}_i$ are present at node i, i.e., $q_{\mathbf{v}_i}^1 = (q_{ij_1}^{\mathbf{v}_i}, q_{ij_2}^{\mathbf{v}_i}, \ldots, q_{ij_{|\mathcal{N}_i|}}^{\mathbf{v}_i})$. The infinite path $\Pi_{\text{TS}_{ij_k}} \tau_{\mathbf{v}_i}$ defined in (5) satisfies the transition rule $\longrightarrow_{\text{TS}_{ij_k}}$ for all $r_{ij_k} \in \mathcal{R}_i$, since all TS_{ij_k} have only two states and there are actions that allow transitions among those states. Also, $\tau_{\mathbf{v}_i}$ satisfies $\phi_{\mathbf{v}_i}$, since it forces all robots in $\mathcal{R}_i$ to visit node i and stay there forever. Therefore, we conclude $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$.

B. Conflict Resolution Coordination

Given the motion plans $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$ and $\tau_{\mathbf{v}_j} \models \phi_{\mathbf{v}_j}$, all robots r_{ij} need to construct discrete motion plans τ_{ij} whose composition satisfies ϕ. To achieve that, we propose a distributed algorithm that resolves any conflicts in the robot behavior introduced by the motion plans $\tau_{\mathbf{v}_i}$ and $\tau_{\mathbf{v}_j}$ and constructs motion plans τ_{ij} which have the following general form

$$\begin{aligned}\tau_{ij} &= \tau_{ij}(1)\tau_{ij}(2)\ldots \\ &= \Big[\underbrace{X\ldots X\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_i}(k)X\ldots X\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_j}(k)X\ldots X}_{\ell}\Big]_{k=1}^{\infty} \\ &= \left[p_{ij}^k\right]_{k=1}^{\infty},\end{aligned} \tag{6}$$

such that $\tau_{ij}(1) = q_{ij}^0$. I.e., τ_{ij} can be written as the concatenation of the finite paths p_{ij}^k of TS_{ij}, $\forall k \in \mathbb{N}_+$. In (6) ℓ is the length of the path p_{ij}^k and is *a priori* selected to be $\ell = \max\{d_{\mathbf{v}_i}\}_{i=1}^R + 1$ for all robots, where $d_{\mathbf{v}_i}$ denotes the degree of vertex i in graph $\mathcal{G}$. This particular choice for the parameter ℓ ensures the construction of free-of-conflict motion plans, as it will shown in Proposition 3.2. Also, in (6), $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ denotes the k-th state of robot r_{ij} in TS_{ij} according to the motion plan $\tau_{\mathbf{v}_i}$. Furthermore, the state X denotes that robot r_{ij} can be in any state of its TS_{ij}; hereafter, we assume that in the X states every robot decides to wait at its current state in TS_{ij}.

The finite paths p_{ij}^k are constructed sequentially across the nodes $i \in \mathcal{V}$, as follows. Let $\mathcal{S} = \{\mathbf{v}_1, \ldots, \mathbf{v}_i, \ldots\}$ be an ordered sequence of the nodes in the mobility graph $\mathcal{G}$, so that consecutive nodes in $\mathcal{S}$ are connected by an edge in $\mathcal{G}$. We assume that $\mathcal{S}$ is known by all robots and that every robot r_{ij} is initially located at either node $\mathbf{v}_i$ or $\mathbf{v}_j$, whichever appears first in $\mathcal{S}$. Assume that paths have been constructed for all nodes in $\mathcal{S}$ that precede $\mathbf{v}_i$ and that currently all robots $r_{ij} \in \mathcal{R}_i$ are located at node i and coordinate to construct the paths p_{ij}^k. Since consecutive nodes in $\mathcal{S}$ are connected by an edge in $\mathcal{G}$, this means that there is at least one robot $r_{im} \in \mathcal{N}_{ij}$ which previously constructed its path p_{im}^k by placing at its $n_{im}^{\mathbf{v}_i}$-th entry the state $\Pi|_{\mathrm{TS}_{im}}\tau_{\mathbf{v}_i}(k)$, i.e., $p_{im}^k(n_{im}^{\mathbf{v}_i}) = \Pi|_{\mathrm{TS}_{im}}\tau_{\mathbf{v}_i}(k)$. Then robot r_{ij} constructs the path p_{ij}^k based on three rules. According to the first rule, the state $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ will be placed at the $n_{ij}^{\mathbf{v}_i}$-th entry, which is selected to be equal to $n_{im}^{\mathbf{v}_i}$, which is common for all robots $r_{im} \in \mathcal{N}_{ij}$ [line 1, Alg. 1]. This ensures that robot r_{ij} and all other robots r_{im} will meet at the same time at $\mathbf{v}_i$, as it will be shown in Proposition 3.4. The next step is to place the state $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ at the $n_{ij}^{\mathbf{v}_j}$-th entry of p_{ij}^k. The index $n_{ij}^{\mathbf{v}_j}$ will be determined by one of the two following rules. If there exist robots $r_{mj} \in \mathcal{N}_{ij}$ that have already constructed the paths p_{mj}^k, then the index $n_{ij}^{\mathbf{v}_j}$ is selected to be equal to $n_{mj}^{\mathbf{v}_j}$, which is common for all $r_{mj} \in \mathcal{N}_{ij}$ [line 3, Alg. 1]. Otherwise, the state $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ can be placed at any free entry of p_{ij}^k indexed by $n_{ij}^{\mathbf{v}_j}$, provided that the $n_{ij}^{\mathbf{v}_j}$-th entry of all paths p_{im}^k of the robots $r_{im} \in \mathcal{N}_{ij}$ that have already been constructed does not contain states $\Pi|_{\mathrm{TS}_{im}}\tau_{\mathbf{v}_m}(k)$ with $m \in \mathcal{N}_j$ [line 5, Alg. 1]. Note that without the third rule [line 5], at a subsequent iteration of this procedure, robot $r_{jm} \in \mathcal{N}_{ij}$ would have to place the states $\Pi|_{\mathrm{TS}_{jm}}\tau_{\mathbf{v}_j}(k)$ and $\Pi|_{\mathrm{TS}_{jm}}\tau_{\mathbf{v}_m}(k)$ at a common entry of p_{jm}^k, i.e., $n_{jm}^{\mathbf{v}_j} = n_{jm}^{\mathbf{v}_m}$, due to the two previous rules and, therefore, a conflicting behavior for robot r_{jm} would occur. In all the remaining entries of p_{ij}^k, Xs are placed [line 7, Alg. 1].[2] This procedure is repeated until all robots $r_{ij} \in \mathcal{R}_i$ have constructed their respective paths p_{ij}^k. Once this happens, all robots r_{ij} depart from node $\mathbf{v}_i$ and travel to the node $\mathbf{v}_j$ [line 8, Alg. 1]. At that point, all robots connected to the next node in $\mathcal{S}$ are present at that node, and can coordinate to compute their respective paths, as before. The procedure is repeated sequentially over the nodes in $\mathcal{S}$ until all robots have computed their paths. When all robots have constructed their finite paths, they exchange a set of indices denoted by $\mathcal{X}_{ij}$ that collects the indices n_{ij}^X at which $p_{ij}^k(n_{ij}^X) = X$. If there exist states $p_{ij}^k(n_{ij}^X) = X$, for some $n_{ij}^X \in \bigcap_{\forall r_{mn}} \mathcal{X}_{mn}$, they are discarded, since in these states all robots r_{ij} wait at their current states. Communication between the robots in this last stage of the algorithm can happen in the order defined by $\mathcal{S}$, as before.

Algorithm 1 Construction of motion plans $\tau_{ij} = [p_{ij}^k]_{k=1}^{\infty}$ at node i

---

Require: Already constructed finite paths p_{im}^k and p_{mj}^k of robots in $\mathcal{N}_{ij}$;
Require: All robots in $\mathcal{R}_i$ are located at node i;
1: $p_{ij}^k(n_{ij}^{\mathbf{v}_i}) := \Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$, $n_{ij}^{\mathbf{v}_i} = n_{im}^{\mathbf{v}_i}$, $\forall r_{im} \in \mathcal{N}_{ij}$;
2: **if** there exist constructed paths p_{mj}^k **then**
3: $\quad p_{ij}^k(n_{ij}^{\mathbf{v}_j}) := \Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$, $n_{ij}^{\mathbf{v}_j} = n_{mj}^{\mathbf{v}_j}$, $\forall r_{mj} \in \mathcal{N}_{ij}$;
4: **else**
5: $\quad p_{ij}^k(n_{ij}^{\mathbf{v}_j}) := \Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ provided either $p_{im}^k(n_{ij}^{\mathbf{v}_j}) = X$, or $p_{im}^k(n_{ij}^{\mathbf{v}_j}) = \Pi|_{\mathrm{TS}_{im}}\tau_{\mathbf{v}_m}(k)$ with $m \notin \mathcal{N}_j$, $\forall r_{im} \in \mathcal{N}_{ij}$;
6: **end if**
7: Put Xs in the remaining entries;
8: Transmit path p_{ij}^k to a robot in $\mathcal{R}_i$ that has not constructed its motion plan. If there are not such robots, all robots $r_{ij} \in \mathcal{R}_i$ depart from node i;

Remark 3.1: Note that communication according to $\mathcal{S}$ is very predictable and inefficient as it, e.g., does not allow for simultaneous meetings at the nodes of $\mathcal{G}$. For these reasons it is only used to construct conflict-free motion plans that allow for much more efficient intermittent communication between robots.

C. Correctness of the Proposed Algorithm

In this section, we show that the composition of the distributed discrete motion plans τ_{ij} satisfies the global LTL expression (2), i.e., that all robots $r_{ij} \in \mathcal{R}_i$ rendezvous infinitely often at node i, for all nodes $i \in \mathcal{V}$. To prove this result, we need first to show that Algorithm 1 can develop non-conflicting motion plans τ_{ij}, for which we have the following two results.

[2] If $i = 1$ then initially, a randomly selected robot r_{1j} creates arbitrarily its path p_{1j}^k by placing the states $\Pi|_{\mathrm{TS}_{1j}}\tau_{\mathbf{v}_1}(k)$ and $\Pi|_{\mathrm{TS}_{1j}}\tau_{\mathbf{v}_j}(k)$ at the $n_{1j}^{\mathbf{v}_1}$-th and $n_{1j}^{\mathbf{v}_j}$-th entry of p_{1j}^k, respectively, with $n_{1j}^{\mathbf{v}_1} \neq n_{1j}^{\mathbf{v}_j}$. Then the procedure previously described follows. Moreover, depending on the structure of the mobility graph $\mathcal{G}$ it is possible that a node $\mathbf{v}_i$ appears more than once in $\mathcal{S}$. In this case, robots $r_{ij} \in \mathcal{R}_i$ construct the finite paths p_{ij}^k only the first time that $\mathbf{v}_i$ appears in $\mathcal{S}$.

Proposition 3.2: Algorithm 1 can always construct finite paths p_{ij}^k with length at most equal to $\ell = \max\{d_{\mathbf{v}_e}\}_{e=1}^R + 1$.

Proof:

The proof is based on contradiction. Assume that a robot r_{ij} needs a finite path p_{ij}^k of length greater than $\ell = \max\{d_{\mathbf{v}_e}\}_{e=1}^R + 1$ when Algorithm 1 is applied. This means that there is a state $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_i}$ which cannot be placed at any of the first ℓ entries of p_{ij}^k. By construction of Algorithm 1, this means that node i has at least $\ell = \max\{d_{\mathbf{v}_e}\}_{e=1}^R + 1$ neighbors in graph $\mathcal{G}$, i.e., $d_{\mathbf{v}_i} \geq \max\{d_{\mathbf{v}_e}\}_{e=1}^R + 1$, which can never happen. Hence, the length $\ell = \max\{d_{\mathbf{v}_e}\}_{e=1}^R + 1$ of the path p_{ij}^k is sufficiently large for the construction of discrete motion plans τ_{ij} by Algorithm 1, which completes the proof. ■

Proposition 3.2 shows also that the finite paths p_{ij}^k and consequently, the motion plans τ_{ij} are scale free, i.e., they depend on the node degree of graph $\mathcal{G}$, and not on its size.

Proposition 3.3: Algorithm 1 generates *admissible* discrete motion plans τ_{ij}, i.e., motion plans that are free of conflicts and satisfy the transition rule $\rightarrow_{ij}$.

Proof: The discrete motion plans τ_{ij} satisfy the transition rule $\rightarrow_{ij}$ by construction of the transition systems TS_{ij}. In particular, all transitions from $\tau_{ij}(k)$ to $\tau_{ij}(k+1)$, for all k, satisfy the transition rule in TS_{ij}, since all TS_{ij} have only two states and there are actions that allow transitions among those states.

A conflicting behavior for robot r_{ij} can occur if this robot needs to be located at two different states in TS_{ij}, simultaneously. Note that this can not happen, since in p_{ij}^k there are always two available entries $n_{ij}^{\mathbf{v}_i}$ and $n_{ij}^{\mathbf{v}_j}$ for the plans $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ and $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ such that $n_{ij}^{\mathbf{v}_i} \neq n_{ij}^{\mathbf{v}_j}$, as shown in Proposition 3.2. Therefore, robot r_{ij} will never need to be at states $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ and $\Pi|_{\mathrm{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ at the same time, which completes the proof. ■

The following proposition shows that the motion plans generated by Algorithm 1 ensure connectivity of the network infinitely often over time, which we present without proof due to space limitations.

Proposition 3.4: The composition of motion plans τ_{ij} generated by Algorithm 1 satisfies the global LTL expression (2), i.e., connectivity of the robot network is ensured over time, infinitely often.

In general, the motion plans τ_{ij} are infinite paths of TS_{ij} and, therefore, in practice they are hard to implement and manipulate. In the following proposition, we show that the motion plans τ_{ij} have a finite representation and they can be expressed in a prefix-suffix structure, where the prefix part τ_{ij}^{pre} is traversed only once and the suffix part τ_{ij}^{suf} is repeated infinitely.

Proposition 3.5: Algorithm 1 generates discrete motion plans τ_{ij} for all robots r_{ij} in a prefix-suffix structure, i.e.,

$$\tau_{ij} = \tau_{ij}^{\mathrm{pre}} \left[\tau_{ij}^{\mathrm{suf}}\right]^{\omega}, \tag{7}$$

where $\tau_{ij}^{\mathrm{pre}} = \left[p_{ij}^k\right]_{k=1}^{\max(S_i,S_j)}$, $\tau_{ij}^{\mathrm{suf}} = \left[p_{ij}^k\right]_{k=\max(S_i,S_j)+1}^{\max(P_i,P_j)+\max(S_i,S_j)}$, and S_i and P_i refer to the length of the prefix and suffix part of $\tau_{\mathbf{v}_i}$, respectively. [3]

Proof: The result can straightforwardly be derived by expanding the motion plan τ_{ij} given in (6) and observing the repetitive pattern $\tau_{ij}^{\mathrm{suf}} = \left[p_{ij}^k\right]_{k=\max(S_i,S_j)+1}^{\max(P_i,P_j)+\max(S_i,S_j)}$. ■

IV. Simulation Studies

In this section, a simulation study is provided that illustrates our approach for a network of $N = 18$ robots that move along the edges of the mobility graph shown in Figure 2. As discussed in Section III, at the beginning, all robots $r_{ij} \in \mathcal{R}_i$, for all nodes i, construct motion plans $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$. For example, for the three leftmost rendezvous points shown in Figure 2, these motion plans have the following form:

$$\begin{aligned}
\tau_{\mathbf{v}_1} &= [(q_{12}^{\mathbf{v}_1}, q_{13}^{\mathbf{v}_3})\,(q_{12}^{\mathbf{v}_1}, q_{13}^{\mathbf{v}_1})]\,[(q_{12}^{\mathbf{v}_1}, q_{13}^{\mathbf{v}_1})]^{\omega}, & (8)\\
\tau_{\mathbf{v}_2} &= [(q_{12}^{\mathbf{v}_1}, q_{23}^{\mathbf{v}_2})\,(q_{12}^{\mathbf{v}_2}, q_{23}^{\mathbf{v}_2})]\,[(q_{12}^{\mathbf{v}_2} q_{23}^{\mathbf{v}_2})]^{\omega}, & (9)\\
\tau_{\mathbf{v}_3} &= [(q_{13}^{\mathbf{v}_3}, q_{23}^{\mathbf{v}_2}, q_{35}^{\mathbf{v}_5}, q_{34}^{\mathbf{v}_4})\,(q_{13}^{\mathbf{v}_3}, q_{23}^{\mathbf{v}_3}, q_{35}^{\mathbf{v}_3}, q_{34}^{\mathbf{v}_3})]\\
&\quad [(q_{13}^{\mathbf{v}_3}, q_{23}^{\mathbf{v}_3}, q_{35}^{\mathbf{v}_3}, q_{34}^{\mathbf{v}_3})]^{\omega}. & (10)
\end{aligned}$$

Then, motion plans τ_{ij} are constructed by Algorithm 1 which, e.g., for robots r_{12}, r_{13}, r_{23} have the following structure

$$\begin{aligned}
\tau_{12} &= \left[p_{12}^k\right]_{k=1}^{\infty} = [\Pi|_{\mathrm{TS}_{12}}\tau_{\mathbf{v}_1}(k)\Pi|_{\mathrm{TS}_{12}}\tau_{\mathbf{v}_2}(k)X]_{k=1}^{\infty}, & (11)\\
\tau_{13} &= \left[p_{13}^k\right]_{k=1}^{\infty} = [\Pi|_{\mathrm{TS}_{13}}\tau_{\mathbf{v}_1}(k)X\Pi|_{\mathrm{TS}_{13}}\tau_{\mathbf{v}_3}(k)]_{k=1}^{\infty}, & (12)\\
\tau_{23} &= \left[p_{23}^k\right]_{k=1}^{\infty} = [X\Pi|_{\mathrm{TS}_{23}}\tau_{\mathbf{v}_2}(k)\Pi|_{\mathrm{TS}_{23}}\tau_{\mathbf{v}_3}(k)]_{k=1}^{\infty}. & (13)
\end{aligned}$$

Using the result from Proposition 3.5, the above motion plans can be written in the following prefix-suffix form[4]

$$\tau_{12} = \Bigg[\underbrace{q_{12}^{\mathbf{v}_1} q_{12}^{\mathbf{v}_1} q_{12}^{\mathbf{v}_1}}_{=p_{12}^1}\,\underbrace{q_{12}^{\mathbf{v}_1} q_{12}^{\mathbf{v}_2} q_{12}^{\mathbf{v}_2}}_{=p_{12}^2}\Bigg]\Bigg[\underbrace{q_{12}^{\mathbf{v}_1} q_{12}^{\mathbf{v}_2} q_{12}^{\mathbf{v}_2}}_{=p_{12}^3}\Bigg]^{\omega}, \tag{14}$$

$$\tau_{13} = \Bigg[\underbrace{q_{13}^{\mathbf{v}_3} q_{13}^{\mathbf{v}_3} q_{13}^{\mathbf{v}_3}}_{=p_{13}^1}\,\underbrace{q_{13}^{\mathbf{v}_1} q_{13}^{\mathbf{v}_1} q_{13}^{\mathbf{v}_3}}_{=p_{13}^2}\Bigg]\Bigg[\underbrace{q_{13}^{\mathbf{v}_1} q_{13}^{\mathbf{v}_1} q_{13}^{\mathbf{v}_3}}_{=p_{13}^3}\Bigg]^{\omega}, \tag{15}$$

$$\tau_{23} = \Bigg[\underbrace{q_{23}^{\mathbf{v}_2} q_{23}^{\mathbf{v}_2} q_{23}^{\mathbf{v}_2}}_{=p_{23}^1}\,\underbrace{q_{23}^{\mathbf{v}_2} q_{23}^{\mathbf{v}_2} q_{23}^{\mathbf{v}_3}}_{=p_{23}^2}\Bigg]\Bigg[\underbrace{q_{23}^{\mathbf{v}_2} q_{23}^{\mathbf{v}_2} q_{23}^{\mathbf{v}_3}}_{=p_{23}^3}\Bigg]^{\omega}. \tag{16}$$

To illustrate that under the proposed motion plans connectivity is ensured over time, we implement a consensus algorithm over the dynamic network $\mathcal{G}_c$. Specifically, we assume that initially robots generate a random number $v_{ij}(t_0)$ and when all robots $r_{ij} \in \mathcal{R}_i$ meet at a rendezvous point i they perform the following consensus update: $v_{ij}(t) = \frac{1}{|\mathcal{R}_i|}\sum_{r_{im}\in\mathcal{R}_i} v_{im}(t)$.

In Figure 2 we observe that there are robots that either wait at the rendezvous points for the arrival of other robots or depart from a meeting point in order to communicate

[3] Note that by construction of motion plans $\tau_{\mathbf{v}_i}$ in (5), we have that $S_i = 1$ or $S_i = 2$, and $P_i = 1$.

[4] Note that lengths of the prefixes and suffixes of the motion plans shown in (8), (9), and (10) are $S_1 = S_2 = S_3 = 2$ and $P_1 = P_2 = P_3 = 1$, respectively.

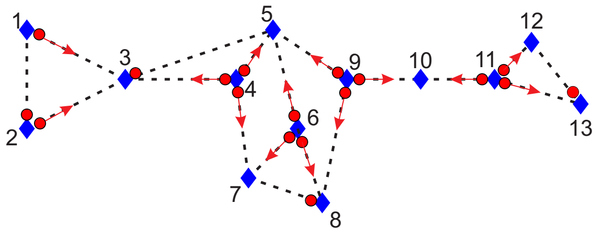

Fig. 2. Intermittent communication of $N = 18$ robots moving along the edges of an underlying mobility graph. Blue rhombuses represent rendezvous points and red circles stand for robots. Robots with red arrows will move to a another rendezvous point at the next iteration, while robots without arrows wait at their current positions for the arrival of robots.

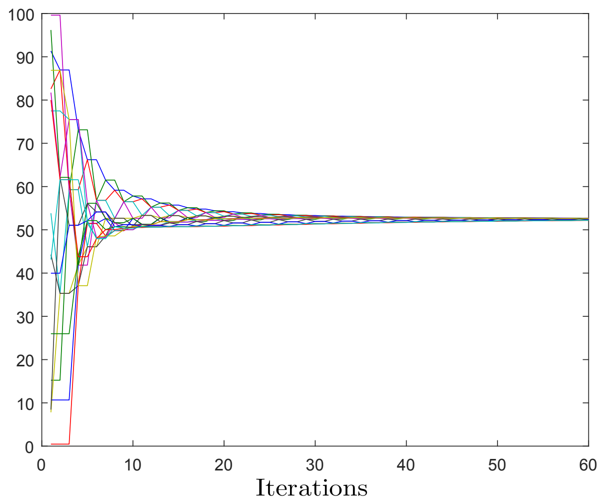

Fig. 3. Graphical depiction of consensus of numbers $u_{ij}(t)$.

with other robots. Figure 3 shows that eventually all robots reach a consensus on the numbers $v_{ij}(t)$, as expected due to Proposition 3.4. In our simulations, robots move with constant velocity between their rendezvous points in $\mathcal{G}$.

V. Conclusion

In this paper we considered the problem of controlling connectivity of mobile networks in an intermittent fashion providing in this way more flexibility to robots to accomplish their tasks as they are not always restricted by communication constraints. In particular, we assumed that robots move along the edges of a mobility graph and they can communicate only when they meet at the nodes of that network, which gave rise to a dynamic communication network. The network was defined to be connected over time if communication takes place at the rendezvous points infinitely often which was encapsulated by a LTL formula. Then to generate discrete high-level motion plans for all robots in a distributed way, we proposed a novel technique that performed an approximate decomposition of the global LTL expression into local LTL expressions and assigned them to robots. To avoid conflicting robot behaviors that could occur due to the approximate decomposition of the global LTL formula, we implemented a distributed conflict resolution scheme that generated discrete motion plans for every robot that ensure connectivity over time, infinitely often, as verified by computer simulations.

References

[1] 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, 2006.

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

[3] E. Montijano, J. Montijano, and C. Sagues, "Adaptive consensus and algebraic connectivity estimation in sensor networks with chebyshev polynomials," in *50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, December 2011, pp. 4296–4301.

[4] M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu, "Decentralized estimation of laplacian eigenvalues in multi-agent systems," *Automatica*, vol. 49, no. 4, pp. 1031–1036, 2013.

[5] D. P. Spanos and R. M. Murray, "Robust connectivity of networked vehicles," in *43rd IEEE Conference on Decision and Control*, vol. 3, The Bahamas, December 2004, pp. 2893–2898.

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

[7] 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.

[8] 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.

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

[10] J. Le Ny, A. Ribeiro, and G. J. Pappas, "Adaptive communication-constrained deployment of unmanned vehicle systems," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 923–934, 2012.

[11] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press Cambridge, 2008, vol. 26202649.

[12] E. M. Clarke, O. Grumberg, and D. Peled, *Model checking*. MIT press, 1999.

[13] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multi-agent motion tasks based on ltl specifications," in *43rd IEEE Conference on Decision and Control (CDC)*, vol. 1, The Bahamas, December 2004, pp. 153–158.

[14] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[15] M. Guo, K. H. Johansson, and D. V. Dimarogonas, "Revising motion planning under linear temporal logic specifications in partially known workspaces," in *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013, pp. 5025–5032.

[16] M. Guo, J. Tumova, and D. V. Dimarogonas, "Cooperative decentralized multi-agent control under local ltl tasks and connectivity constraints," in *53rd Conference on Decision and Control (CDC)*, Los Angeles, CA, USA, December 2014, pp. 75–80.

[17] Y. Chen, X. C. Ding, and C. Belta, "Synthesis of distributed control and communication schemes from global ltl specifications," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, December 2011, pp. 2718–2723.

[18] M. Kloetzer and C. Belta, "Distributed implementations of global temporal logic motion specifications," in *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, May 2008, pp. 393–398.

[19] 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.

[20] 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.

[21] 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.