

# Intermittent Connectivity Control in Mobile Robot Networks

Yiannis Kantaros and Michael M. Zavlanos

**Abstract**—In this paper, we consider networks of mobile robots responsible for accomplishing tasks, captured by Linear Temporal Logic (LTL) formulas, while ensuring communication with all other robots in the network. The robots operate in complex environments represented by appropriate transition systems (TS). We propose an intermittent communication framework, which is based on a LTL statement that enforces the robots to meet and communicate at pre-determined points in the environment infinitely often. Our approach combines an existing model checking method with a novel technique that aims to reduce the state-space of the TS satisfying at the same time the LTL statement.

## I. INTRODUCTION

Wireless communication in mobile robot networks plays a pivotal role in successful accomplishment of robot missions, as it allows robots to exchange gathered information in order to achieve a globally assigned task. Typically, communication is modeled using proximity graphs and the communication problem is treated as preservation of graph connectivity. Methods to control graph connectivity range from centralized [1], [2] to distributed ones [3], [4]. A recent survey on graph theoretic methods for connectivity control can be found in [5]. More realistic communication models have recently been proposed in [6], [7].

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 an *intermittent communication* protocol for mobile robot networks that allows robots to communicate in an intermittent fashion when they are within a communication range and operate in disconnect mode the rest of the time. In this way, we provide more flexibility to robots to accomplish their tasks as robot mobility is not always bound by communication constraints. Intermittent communication is due to an LTL statement that enforces the robots to meet and exchange information at pre-determined points in the environment infinitely often. Our approach combines an existing model checking method [8], [9] with a novel technique that aims to reduce the state-space of the TS satisfying at the same time the LTL statement.

LTL-based control synthesis and task specification for mobile robots build upon either a top-down approach when

independent LTL expressions are assigned to robots [10]–[12] or bottom-up approaches when a global LTL describing a collaborative task is assigned to a team of robots [13], [14], as in our work. Bottom-up 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 combine an existing model checking method with a novel technique that aims to reduce the state-space of the TS satisfying at the same time the LTL statement.

The most relevant works to the one proposed here are presented in [15], [16]. In particular, [15] 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, [16] proposes a receding horizon framework for periodic connectivity that ensures recovery of connectivity within a given time horizon. To the contrary, in our proposed method the network is never required to be fully connected, but only connected over time. In our proposed framework, robots communicate infinitely often due to a LTL expression at time instants and locations that minimize the total distance traveled by them.

## II. PRELIMINARIES

In this section we formally describe Linear Temporal Logic (LTL) by presenting its syntax and semantics. Also, we briefly review preliminaries of automata-based LTL model checking, while a detailed overview of this theory can be found in [8].

Linear temporal logic is a type of formal logic whose basic ingredients are a set of atomic propositions  $\mathcal{AP}$ , the boolean operators, conjunction  $\wedge$  and negation  $\neg$ , and two temporal operators, next  $\bigcirc$  and until  $\mathcal{U}$ . LTL formulas over a set  $\mathcal{AP}$  can be constructed based on the following grammar:  $\phi ::= \text{true} \mid \pi \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \bigcirc\phi \mid \phi_1 \mathcal{U} \phi_2$ , where  $\pi \in \mathcal{AP}$ . For the sake of brevity we abstain from presenting the derivations of other Boolean and temporal operators, e.g., *always*  $\square$ , *eventually*  $\diamond$ , *implication*  $\Rightarrow$ , which can be found in [8].

An infinite word  $\sigma$  over the alphabet  $2^{\mathcal{AP}}$  is defined as an infinite sequence  $\sigma = \pi_0\pi_1\pi_2\cdots \in (2^{\mathcal{AP}})^\omega$ , where  $\omega$  denotes infinite repetition and  $\pi_k \in 2^{\mathcal{AP}}$ ,  $\forall k$ . The language  $\text{Words}(\phi) = \{\sigma \in (2^{\mathcal{AP}})^\omega \mid \sigma \models \phi\}$ , where  $\models \subseteq (2^{\mathcal{AP}}) \times \phi$  is the satisfaction relation, is defined as the set of words that satisfy the LTL  $\phi$ .

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

Any LTL formula  $\phi$  can be translated into a Nondeterministic Büchi Automaton (NBA) over  $2^{\mathcal{AP}}$  denoted by  $B$  [17], which is defined as follows:

*Definition 2.1:* A *Nondeterministic Büchi Automaton* (NBA)  $B$  over  $2^{\mathcal{AP}}$  is defined as a tuple  $B = (\mathcal{Q}_B, \mathcal{Q}_B^0, 2^{\mathcal{AP}}, \rightarrow_B, \mathcal{F}_B)$  where: (a)  $\mathcal{Q}_B$  is the set of states; (b)  $\mathcal{Q}_B^0 \subseteq \mathcal{Q}_B$  is a set of initial states; (c)  $\Sigma = 2^{\mathcal{AP}}$  is an alphabet; (d)  $\rightarrow_B \subseteq \mathcal{Q}_B \times \Sigma \times \mathcal{Q}_B$  is the transition relation; and (e)  $\mathcal{F}_B \subseteq \mathcal{Q}_B$  is a set of accepting/final states.

An *infinite run*  $\rho_B$  of  $B$  over an infinite word  $\sigma = \pi_0\pi_1\pi_2\dots, \pi_k \in \Sigma = 2^{\mathcal{AP}}$  is a sequence  $\rho_B = q_B^0 q_B^1 q_B^2 \dots$  such that  $q_B^0 \in \mathcal{Q}_B^0$  and  $(q_B^k, \pi_k, q_B^{k+1}) \in \rightarrow_B$ , for some  $\pi_k \in \Sigma = 2^{\mathcal{AP}}, \forall k$ , which can alternatively be denoted by  $q_B^k \xrightarrow{\pi_k} q_B^{k+1}, \forall k$ . An infinite run  $\rho_B$  is called *accepting run* if  $\text{Inf}(\rho_B) \cap \mathcal{F}_B \neq \emptyset$ , where  $\text{Inf}(\rho_B)$  represents the set of states that appear in  $\rho_B$  infinitely often. The words  $\sigma$  that result in an accepting run of  $B$  constitute the accepted language of  $B$ , denoted by  $\mathcal{L}_B$ , i.e.,  $\mathcal{L}_B = \text{Words}(\phi)$ .

### III. PROBLEM FORMULATION

Consider a team of  $N$  robots residing in a workspace  $\mathcal{W} \subset \mathbb{R}^n$  that are responsible for accomplishing a collaborative task captured by a LTL expression denoted by  $\phi_{\text{task}}$ . The robots are assumed to move in  $\mathcal{W}$  according to the following first-order differential equation  $\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t)$ , where  $\mathbf{x}_i(t) \in \mathbb{R}^n$  is the position of robot  $i$  at time  $t$  and  $\mathbf{u}_i(t) \in \mathbb{R}^n$  is a control input.

We assume that the workspace  $\mathcal{W}$  is partitioned into  $R$  disjoint regions of arbitrary shape. The  $k$ -th region is denoted by  $\ell_k$ . The task  $\phi_{\text{task}}$  can model collaborative patrolling, investigation, or avoidance of some of those regions. Let  $\pi_i^{\ell_k}$  denote atomic propositions which are true if robot  $i$  is located in region  $\ell_k$ . Then, we can define a discrete abstraction of the workspace captured by a weighted Transition System (wTS) associated with robot  $i$  defined, as follows:

*Definition 3.1 (weighted Transition System):* A *weighted Transition System*  $\text{wTS}_i$  associated with robot  $i$  is a tuple  $(\mathcal{Q}_i, \mathcal{Q}_i^0, \mathcal{A}_i, \rightarrow_i, w_i, \mathcal{AP}_i, L_i)$  where: (a)  $\mathcal{Q}_i = \{\ell_k\}_{k=1}^R$  is the set of states. A state  $q_i \in \mathcal{Q}_i$  is associated with the presence of robot  $i$  in a location  $\ell_k$ ; (b)  $\mathcal{Q}_i^0 \subseteq \mathcal{Q}_i$  is the initial state of robot  $i$ ; (c)  $\mathcal{A}_i$  is a set of actions; (d)  $\rightarrow_i \subseteq \mathcal{Q}_i \times \mathcal{A}_i \times \mathcal{Q}_i$  is the transition relation; (e)  $w_i : \mathcal{Q}_i \times \mathcal{Q}_i \rightarrow \mathbb{R}_+$  is a cost function that assigns weights/cost to each possible transition in wTS. These costs can be associated with the distance between two states  $q_i$  and  $q'_i$ ; (f)  $\mathcal{AP}_i = \{\pi_i^{\ell_k}\}_{k=1}^R$  is the set of atomic propositions; and (g)  $L_i : \mathcal{Q}_i \rightarrow 2^{\mathcal{AP}_i}$  is an observation/output relation giving the set of atomic propositions that are satisfied at a state.

In what follows we provide definitions related to  $\text{wTS}_i$ , that we will use in the rest of this section. For more details on LTL terminology and semantics we refer the reader to [8], [9].

*Definition 3.2 (Infinite Path):* An *infinite path*  $\tau_i$  of  $\text{wTS}_i$  is an infinite sequence of states,  $\tau_i = \tau_i(1)\tau_i(2)\tau_i(3)\dots$  such that  $\tau_i(1) = q_i^0$ ,  $\tau_i(k) \in \mathcal{Q}_i$ , and  $(\tau_i(k), a_i^k, \tau_i(k+1)) \in \rightarrow_i$ , for some  $a_i^k \in \mathcal{A}_i, \forall k$ .

*Definition 3.3 (Motion Plan):* Given an LTL formula  $\phi$  and a transition system  $\text{wTS}_i$ , both defined over the set of atomic propositions  $\mathcal{AP}_i$ , an infinite path  $\tau_i$  of  $\text{wTS}_i$  is called a *motion plan* if and only if  $\text{trace}(\tau_i) \in \text{Words}(\phi)$ , which is equivalently denoted by  $\tau_i \models \phi$ .

*Definition 3.4 (Trace of infinite path):* The trace of an infinite path  $\tau_i = \tau_i(1)\tau_i(2)\tau_i(3)\dots$  of a transition system  $\text{wTS}_i$ , denoted by  $\text{trace}(\tau_i)$ , is an infinite word that is determined by the sequence of atomic propositions that are true in the states along  $\tau_i$ , i.e.,  $\text{trace}(\tau_i) = L_i(\tau_i(1))L_i(\tau_i(2))\dots$ .

*Definition 3.5 (Trace of wTS):* The trace of a transition system  $\text{wTS}_i$  is defined as  $\text{trace}(\text{wTS}_i) = \bigcup_{\tau_i \in \mathcal{P}} \text{trace}(\tau_i)$ , where  $\mathcal{P}$  is the set of all infinite paths  $\tau_i$  in  $\text{wTS}_i$ .

The total cost associated with an infinite path  $\tau_i$  of  $\text{wTS}_i$  is  $\sum_{k=0}^{\infty} w_i(\tau_i(k), \tau_i(k+1))$ . Consequently the cost incurred by mobility of  $N$  robots in order to accomplish a collaborative task is given by

$$\begin{aligned} J(\tau_1, \dots, \tau_N) &= \sum_{i=1}^N \sum_{k=0}^{\infty} w_i(\tau_i(k), \tau_i(k+1)) \\ &= \sum_{k=0}^{\infty} w(\tau(k), \tau(k+1)), \end{aligned} \quad (1)$$

where  $w(\tau(k), \tau(k+1)) = \sum_{i=1}^N w_i(\tau_i(k), \tau_i(k+1))$ , and  $\tau$  stands for the composition of the infinite paths  $\tau_i$  of all robots. The composition of infinite paths is defined as:

*Definition 3.6 (Composition):* Composition of  $N$  infinite paths  $\tau_i = \tau_i(1)\tau_i(2)\tau_i(3)\dots$ , where  $i \in \{1, \dots, N\}$ , denoted by  $\tau = \otimes_{\forall i} \tau_i$  is an infinite sequence of states defined as  $\tau = \tau(1)\tau(2)\dots = [\tau(k)]_{k=1}^{\infty}$ , where  $\tau(k) = (\tau_1(k), \tau_2(k), \dots, \tau_N(k))$ .

Apart from accomplishing a high-level task captured by  $\phi_{\text{task}}$ , robots need to communicate and exchange gathered information. Particularly, we assume that the robotic team is divided into  $M$  subgroups  $\mathcal{T}_m, m \in \{1, 2, \dots, M\}$  and that every robot can belong to more than one subgroup. All robots in a subgroup  $\mathcal{T}_m$  can communicate only when they are present simultaneously at a common region  $\ell_k$ , giving rise to an underlying communication network. The regions  $\ell_k$  where communication can take place for the robotic team  $\mathcal{T}_m$  are collected in a set  $\mathcal{C}_m$ . We also assume that the graph whose set of nodes is indexed by the teams  $\mathcal{T}_m$  and set of edges consists of links between nodes  $m$  and  $n$  if  $\mathcal{T}_m \cap \mathcal{T}_n \neq \emptyset$  is connected in order to ensure dissemination of information to the network. Assuming also that the locations  $\ell_k \in \bigcup_{m=1}^M \mathcal{C}_m$  and the robot paths that connect them form a connected graph, the communication network is said to be connected over time if the robots in all subgroups  $\mathcal{T}_m$  meet infinitely often at the regions  $\ell_k \in \mathcal{C}_m$ . Such a requirement can be captured by the following LTL expression:  $\phi_{\text{com}} = \bigwedge_{\forall m \in \{1, 2, \dots, M\}} \left( \bigvee_{\ell_k \in \mathcal{C}_m} (\square \diamond \bigwedge_{\forall i \in \mathcal{T}_m} \pi_i^{\ell_k}) \right)$ .

Assuming that all robots make transitions synchronously by picking their next state in their respective transition systems, the problem addressed in this paper becomes:

*Problem 1:* Determine a motion plan  $\tau$  for the team of robots such that the global LTL expression  $\phi = \phi_{\text{task}} \wedge \phi_{\text{com}}$

is satisfied, i.e., the assigned global task is accomplished and intermittent communication among robots is ensured, while minimizing the total cost (1).

#### IV. INTERMITTENT COMMUNICATION CONTROL

To solve Problem 1, known model checking techniques can be employed, that typically rely on a discretized abstraction of the environment captured by a wTS and the construction of a synchronized product system among all robots in the network. In IV-A, we discuss how to solve Problem 1 using such an existing automated-based model checking algorithm. Then, in Section IV-B, we develop a modification to this algorithm that allows us to construct wTSs with smaller state-spaces that generate words that satisfy the LTL formula  $\phi = \phi_{\text{task}} \wedge \phi_{\text{com}}$ .

##### A. Automata-based Model Checking

First the global LTL formula  $\phi$  is translated in a *Nondeterministic Büchi Automaton* (NBA)  $B$  as defined in Definition 2.1 whose alphabet is  $\Sigma = 2^{\mathcal{AP}}$  with  $\mathcal{AP} = \bigcup_{i=1}^N \mathcal{AP}_i$ . Next, given wTS $_i$  for all robots, the *Product Transition System* (PTS) PTS is constructed, which captures all the possible combinations of robots' states in their respective wTS $_i$  and is defined as:

*Definition 4.1 (Product Transition System):*

Given  $N$  transition systems wTS $_i = (Q_i, q_i^0, \mathcal{A}_i, \rightarrow_i, w_i, \mathcal{AP}_i, L_i)$ , the *weighted Product Transition System* PTS = wTS $_1 \otimes$  wTS $_2 \otimes \dots \otimes$  wTS $_N$  is a tuple  $(Q_{\text{PTS}}, q_{\text{PTS}}^0, \mathcal{A}_{\text{PTS}}, \rightarrow_{\text{PTS}}, \mathcal{AP}, L_{\text{PTS}})$  where: (a)  $Q_{\text{PTS}} = Q_1 \times Q_2 \times \dots \times Q_N$  is the set of states; (b)  $q_{\text{PTS}}^0 = (q_1^0, q_2^0, \dots, q_N^0) \in Q_{\text{PTS}}$  is the initial state; (c)  $\mathcal{A}_{\text{PTS}} = \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_N$  is a set of actions; (d)  $\rightarrow_{\text{PTS}} \subseteq Q_{\text{PTS}} \times \mathcal{A}_{\text{PTS}} \times Q_{\text{PTS}}$  is the transition relation defined by the rule<sup>1</sup>  $\frac{\bigwedge_{i=1}^N (q_i \xrightarrow{a_i} q'_i)}{q_{\text{PTS}} = (a_1, \dots, a_{i,j_N}) \rightarrow_{\text{PTS}} q'_{\text{PTS}}}$ ; (e)  $w_{\text{PTS}}(q_{\text{PTS}}, q'_{\text{PTS}}) = w(q_{\text{PTS}}, q'_{\text{PTS}}) = \sum_{i=1}^N w_i(q_i, q'_i)$ ; (f)  $\mathcal{AP} = \bigcup_{i=1}^N \mathcal{AP}_i$  is the set of atomic propositions; and (g)  $L_{\text{PTS}} = \bigcup_i L_i$  is an observation/output relation giving the set of atomic propositions that are satisfied at a state.

Once the PTS and the NBA  $B$  that corresponds to the LTL  $\phi = \phi_{\text{task}} \wedge \phi_{\text{com}}$  are constructed, a motion plan  $\tau$  that satisfies  $\phi$  and at the same time minimizes the cost (1) can be found by checking the non-emptiness of the language of the *Product Büchi Automaton* (PBA)  $P = \text{PTS} \otimes B$  [8], which is defined as follows:

*Definition 4.2 (Product Büchi Automaton):* Given the product transition system PTS =  $(Q_{\text{PTS}}, q_{\text{PTS}}^0, \mathcal{A}_{\text{PTS}}, \rightarrow_{\text{PTS}}, \mathcal{AP}, L_{\text{PTS}})$  and the NBA  $B = (Q_B, q_B^0, 2^{\mathcal{AP}}, \rightarrow_B, \mathcal{F}_B)$ , the *Product Büchi Automaton*  $P = \text{PTS} \otimes B$  is a tuple  $(Q_P, q_P^0, \rightarrow_P, w_P \mathcal{F}_P)$  where: (a)  $Q_P = Q_{\text{PTS}} \times Q_B$  is the set of states; (b)  $Q_P^0 = q_{\text{PTS}}^0 \times q_B^0$  is a set of initial states; (c)  $\rightarrow_P \subseteq Q_P \times \mathcal{A}_{\text{PTS}} \times 2^{\mathcal{AP}} \times Q_P$  is the transition relation

<sup>1</sup>The notation here is adopted from [8]. It means that if the proposition above the solid line is true, then so is the proposition below the solid line. The state  $q_{\text{PTS}}$  stands for  $(q_1, \dots, q_N) \in Q_{\text{PTS}}$ . Also,  $q'_{\text{PTS}}$  is defined accordingly.

defined by the rule  $\frac{\left( q_{\text{PTS}} \xrightarrow{a_{\text{PTS}}} q'_{\text{PTS}} \right) \wedge \left( q_B \xrightarrow{L_{\text{PTS}}(q'_{\text{PTS}})} q'_B \right)}{q_P = (q_{\text{PTS}}, q_B) \xrightarrow{a_{\text{PTS}}} q'_P = (q'_{\text{PTS}}, q'_B)}$ ; (d)  $w_P(q_P, q'_P) = w_{\text{PTS}}(q_{\text{PTS}}, q'_{\text{PTS}})$ , where  $q_P = (q_{\text{PTS}}, q_B)$  and  $q'_P = (q'_{\text{PTS}}, q'_B)$ ; and (e)  $\mathcal{F}_P = Q_{\text{PTS}} \times \mathcal{F}_B$  is a set of accepting/final states.

To check the non-emptiness of the language of  $P$  denoted by  $\mathcal{L}_P = \text{trace}(\text{PTS}) \cap \mathcal{L}_B$  and to find the motion plan that both satisfies  $\phi$  and at the same time minimizes the cost function (1) we can employ existing model checking methods that are based on graph search algorithms; see, e.g., [18], [19]. Such motion plans can be written in a prefix-suffix structure  $\tau = \tau^{\text{pre}}[\tau^{\text{suf}}]^\omega$ , where  $\tau^{\text{suf}}$  is repeated infinitely.

##### B. LTL-based Intermittent Communication

The model checking algorithm discussed in Section IV-A relies on the construction of a synchronous product transition system among all robots in the network. As a result, it suffers from the state explosion problem and, therefore, it is resource demanding and scales poorly with the number of robots. To mitigate these issues we have developed a novel method that aims to reduce the state-space of the product transition system by taking into account the atomic propositions that appear in the LTL expression  $\phi$ . Specifically, the algorithm checks at which regions these atomic propositions are satisfied and then constructs paths towards those regions. So, different LTL expressions will result in different wTSs. To achieve this, we construct weighted Transitions Systems wTS $_i$  that are trace-included by wTS $_i$  for every robot  $i$ , defined as follows:

*Definition 4.3 (Trace-included wTS):* Consider weighted Transitions Systems wTS $_i = (Q_i, q_i^0, \mathcal{A}_i, \rightarrow_i, w_i, \mathcal{AP}, L_i)$  and  $\widetilde{\text{wTS}}_i = (\widetilde{Q}_i, \widetilde{q}_i^0, \widetilde{\mathcal{A}}_i, \rightarrow_{\widetilde{i}}, \widetilde{w}_i, \mathcal{AP}, L_i)$  both defined over a set of atomic propositions  $\mathcal{AP}$ .  $\widetilde{\text{wTS}}_i$  is trace-included by wTS $_i$  if  $\text{trace}(\widetilde{\text{wTS}}_i) \subseteq \text{trace}(\text{wTS}_i)$ .

Our goal is to construct  $\widetilde{\text{wTS}}_i$  that may not be as “expressive” as wTS $_i$ , since  $\text{trace}(\widetilde{\text{wTS}}_i) \subseteq \text{trace}(\text{wTS}_i)$ , but have smaller state-spaces and are able to generate motion plans that satisfy the LTL formula  $\phi$ . The construction of  $\widetilde{\text{wTS}}_i$  depends on the global LTL expression  $\phi$ . Given an LTL formula  $\phi$  we define the following sets of atomic propositions. Let  $\Pi_i$  be an ordered set that collects all atomic propositions  $\pi_i^{\ell_k}$  associated with robot  $i$  that appear in  $\phi$  without the negation operator  $\neg$  in front of them, including the atomic proposition  $\pi_i^{\ell_0}$  that is true at  $q_i^0$ . Also, let  $\overline{\Pi}_i$  be a set that collects all atomic propositions  $\pi_i^{\ell_k}$  associated with robot  $i$  that appear in  $\phi$  with the negation operator  $\neg$  in front of them. If an atomic proposition appears in  $\phi$  more than once, both with and without the negation operator, then it is included in both sets. For example consider the following  $\phi$ :

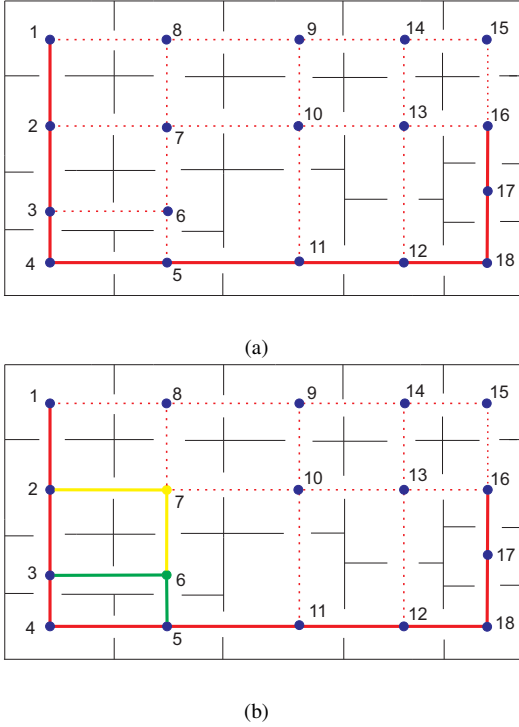


Fig. 1. Fig. 1(a): An environment  $\mathcal{W}$  partitioned into 18 regions. Blue dots and red (solid and dashed) edges represent the states and all possible transitions among these states in  $\text{wTS}_1$ , respectively. Red solid edges represent the transitions among the states of  $\text{wTS}_1$  when the global LTL formula is described by (2). Fig. 1(b): Green (yellow) dots and edges represent the states and transitions that are included in  $\text{wTS}_1$  after the first (second) iteration of Algorithm 1, respectively. Given the LTL expression (2),  $\bar{\Pi}_1 \cap L_1(\tilde{Q}_1) = \{\pi_1^{\ell_3}\}$  for robot 1 and, therefore,  $\mathcal{S}_1 = \{\ell_3\}$ .

$$\begin{aligned}
 \phi = & \overbrace{(\Box\Diamond\pi_1^{\ell_{12}}) \wedge (\Box\Diamond\pi_1^{\ell_3}) \wedge (\Box\neg\pi_1^{\ell_{10}})}^{\text{assigned task}} \\
 & \wedge (\neg\pi_1^{\ell_3} \mathcal{U} \pi_1^{\ell_5}) \wedge (\Box\Diamond\pi_2^{\ell_9}) \\
 & \wedge \overbrace{(\Box\Diamond(\pi_1^{\ell_{17}} \wedge \pi_2^{\ell_{17}}) \vee \Box\Diamond(\pi_1^{\ell_{16}} \wedge \pi_2^{\ell_{16}}))}^{\text{intermittent communication}}. \quad (2)
 \end{aligned}$$

Then,  $\Pi_1$  and  $\bar{\Pi}_1$  become  $\Pi_1 = \{\pi_1^{\ell_1}, \pi_1^{\ell_3}, \pi_1^{\ell_5}, \pi_1^{\ell_{12}}, \pi_1^{\ell_{16}}, \pi_1^{\ell_{17}}\}$  and  $\bar{\Pi}_1 = \{\pi_1^{\ell_3}, \pi_1^{\ell_{10}}\}$  assuming that robot 1 is initially located at  $\ell_1$ .

To construct the state-space  $\tilde{Q}_i$  we view the transition system  $\text{wTS}_i$  as a weighted directed graph  $G_i = \{\mathcal{V}_i, \mathcal{E}_i\}$ , where the set of vertices  $\mathcal{V}_i$  and the set of indices  $\mathcal{E}_i$  are determined by  $Q_i$  and  $\rightarrow_i$ , respectively. Weights on edges are assigned by the function  $w_i$ . Then, we compute the shortest paths from the location at which  $\Pi_i(n)$  is true to the location at which  $\Pi_i(n+1)$  is true avoiding all locations that must never be visited, i.e., locations  $\ell_k$  that appear in  $\phi$  as  $\Box(\neg\pi_i^{\ell_k})$ ; see Figure 1(a). All states of  $\text{wTS}_i$  that appear in these shortest paths comprise the set  $\tilde{Q}_i$ .<sup>2</sup> The set of actions  $\tilde{A}_i$ , the transition rule  $\rightarrow_{\tilde{i}}$ , and the function  $\tilde{w}_i$  are inherited from  $\text{wTS}_i$ . Note that the already constructed transition systems  $\text{wTS}_i$  may or may not satisfy

<sup>2</sup>The order in the sets  $\Pi_i$  can be selected either randomly or by checking which order results in the smaller state-space  $\tilde{Q}_i$ .

---

### Algorithm 1 Update of $\text{wTS}_i$

---

```

1: if  $\bar{\Pi}_i \cap L_i(\tilde{Q}_i) \neq \emptyset$  then
2:    $\mathcal{S}_i = \{q_i \in \tilde{Q}_i \mid L_i(q_i) \in \bar{\Pi}_i\}$ ;
3: else
4:    $\mathcal{S}_i = \tilde{Q}_i$ ;
5: end if
6:  $n = 1, k = 1$ ;
7:  $\text{PTS} = \otimes_{\forall i} \text{wTS}_i, \text{P} = \widetilde{\text{PTS}} \otimes B$ ;
8: while  $\text{trace}(\widetilde{\text{PTS}}) \cap \mathcal{L}_B = \emptyset$  do
9:   Update all  $\text{wTS}_i$  by adding a state from  $\mathcal{N}_{\mathcal{S}_i(k)}^{n-\text{hops}}$ ;
10:   $\widetilde{\text{PTS}} = \otimes_{\forall i} \widetilde{\text{wTS}}_i$ ;
11:  Update PBA;
12:   $k = k + 1$ ;
13:  if  $k > |\mathcal{S}_i|$  then
14:     $k = 1, n = n + 1$ ;
15:  end if
16: end while

```

---

the LTL expression  $\phi$ . In the latter case all robots update their previously constructed  $\text{wTS}_i$  by adding states to  $\tilde{Q}_i$ , as shown in Algorithm 1. The candidate states to be added to  $\tilde{Q}_i$  are selected from the neighborhood of the states that belong to an ordered set  $\mathcal{S}_i$ . In case  $\bar{\Pi}_i \cap L_i(\tilde{Q}_i) \neq \emptyset$  then we define  $\mathcal{S}_i = \{q_i \in \tilde{Q}_i \mid L_i(q_i) \in \bar{\Pi}_i\}$ , since, intuitively, visitation of these states may lead to violation of  $\phi$  [line 2]. Otherwise,  $\mathcal{S}_i = \tilde{Q}_i$  [line 4], where  $Q_i$  is the set of states of the initially constructed  $\text{wTS}_i$ . As long as the constructed  $\text{wTS}_i$  cannot generate motion plans that satisfy the assigned LTL expression [lines 8] a new state is added to  $\tilde{Q}_i$  from the set  $\mathcal{N}_{\mathcal{S}_i(k)}^{n-\text{hops}}$  that contains the  $n$ -hops connected neighbors of the  $k$ -th state in  $\mathcal{S}_i$ , denoted by  $\mathcal{S}_i(k)$ , in the graph  $G_i = \{\mathcal{V}_i, \mathcal{E}_i\}$  excluding the states that already belong to the state space of  $\tilde{Q}_i$ ; see Figure 1(b). The state that will be included in  $\tilde{Q}_i$  can be selected from the set  $\mathcal{N}_{\mathcal{S}_i(k)}^{n-\text{hops}}$  based on various criteria. For example it can be selected randomly, or based on the degree of that state in the graph  $G_i$ , or based on the sum of weights of the transitions that will be included in  $\rightarrow_{\tilde{i}}$ . Once new states are included, the product transition system is updated [line 10] and so does the PBA [line 11]. The PBA can be reconstructed from scratch, or alternatively can be updated using existing on-the-fly algorithms [8]. If for some  $n$ , all the states from  $\mathcal{N}_{\mathcal{S}_i(k)}^{n-\text{hops}}$  have been included then the index  $n$  is increased [line 14]. We then have the following result:

*Proposition 4.4 (Correctness):* Assume that  $\text{trace}(\text{PTS}) \cap \text{Words}(\phi) \neq \emptyset$ , i.e., that the initial transition systems  $\text{wTS}_i$  can generate motions plans that satisfy  $\phi$ . Then, Algorithm 1 will eventually find a motion plan  $\tau$  that satisfies  $\phi$ .

*Proof:* The proof is based on the fact that at the worst case scenario, Algorithm 1 will incorporate all states from  $Q_i$  into  $\tilde{Q}_i$ , i.e.,  $Q_i = \tilde{Q}_i$  and, therefore,  $\text{trace}(\widetilde{\text{PTS}}) = \text{trace}(\text{PTS})$ . In such a case, we have  $\text{trace}(\widetilde{\text{PTS}}) \cap \text{Words}(\phi) \neq \emptyset$  completing the proof. ■

*Remark 4.5 (Optimality):* Let  $J_{\text{PTS}}^*$  denote the optimal

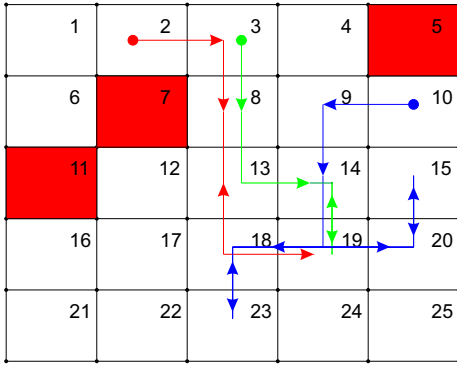


Fig. 2. Illustration of the constructed motion plan  $\tau \models \phi$ . Red, green, and blue dots represent the initial positions of robots in the workspace while the solid lines represent their respective trajectories.

cost when the motion plan  $\tau$  is computed over  $\text{PTS} = \otimes_{\forall i} \text{wTS}_i$ . Given that  $\tilde{Q}_i \subseteq Q_i$ , for all robots  $i$ , it holds that  $\tilde{Q}_{\text{PTS}} \subseteq Q_{\text{PTS}}$  and, therefore, in general, for the optimal cost computed over  $\tilde{\text{PTS}}$  we have that  $J_{\text{PTS}}^* \geq J_{\tilde{\text{PTS}}}^*$ .

## V. SIMULATION STUDIES

We illustrate our approach on a network of  $N = 3$  robots and  $M = 1$  group of robots. The assigned task and the intermittent communication requirement are captured by the following LTL expression which corresponds to a Non-deterministic Büchi Automaton with  $|Q_B| = 6$  states:

$$\begin{aligned} \phi = & \overbrace{[\Box\Diamond(\pi_r^{13} \wedge \pi_g^{14})] \wedge [\neg(\pi_r^{13} \wedge \pi_g^{14})\mathcal{U}\pi_b^{23}]}^{\text{assigned task}} \\ & \wedge [\Box\Diamond(\pi_b^{15})] \wedge (\neg\pi_b^{15}\mathcal{U}\pi_b^{23}) \\ & \overbrace{\wedge [\Box\Diamond(\pi_r^{17} \wedge \pi_g^{17} \wedge \pi_b^{17})] \vee [\Box\Diamond(\pi_r^{19} \wedge \pi_g^{19} \wedge \pi_b^{19})]}^{\text{intermittent communication}} \end{aligned} \quad (3)$$

The robots reside in a workspace  $\mathcal{W}$  partitioned in 25 subregions as shown in Figure 2 and, therefore, the wTSs for all robots have  $|Q_i| = 25$ ,  $i \in \{r, g, b\}$  states each one associated with a subregion of  $\mathcal{W}$ . The function  $w_i$  is equal to 1 for every possible transition for all robots. Red areas in Figure 2 stand for obstacles in the workspace and, therefore, there are no transitions towards these locations in  $\rightarrow_i$  for all robots  $i$ . Communication among robots can occur only if they are simultaneously located in a subregion in  $\mathcal{C}_1 = \{\ell_{17}, \ell_{19}\}$ , as shown in the second part of  $\phi$  in (3). Given such transition systems and the LTL expression given in (3), the Product Büchi Automaton has 93750 states. Applying Algorithm 1, robots construct wTSs with  $|\tilde{Q}_r| = 7$ ,  $|\tilde{Q}_g| = 6$ , and  $|\tilde{Q}_b| = 9$  and, therefore, the state-space of the PBA reduces to 2268 states. The generated motion plan has the form  $\tau = \tau^{\text{pre}}[\tau^{\text{suf}}]^\omega$ . Figure 2 shows the resulting robot trajectories. The robots communicate infinitely often at state  $\ell_{19}$ .

## VI. CONCLUSION

In this paper, we proposed an intermittent communication framework for mobile robot networks based on an LTL statement that enforces the robots to meet and exchange

information at pre-determined points in the environment infinitely often. The rest of the time robots operate in disconnect mode providing more flexibility to robots to accomplish their tasks. Our approach combined an existing model checking method with a novel technique that aims to reduce the state-space of the transition systems that model the environment satisfying at the same time the LTL statement.

## 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*. IEEE, 2011, pp. 4296–4301.
- [4] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428, 2008.
- [5] 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.
- [6] 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.
- [7] Y. Yan and Y. Mostofi, "Robotic router formation in realistic communication environments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 810–827, 2012.
- [8] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press Cambridge, 2008, vol. 26202649.
- [9] E. M. Clarke, O. Grumberg, and D. Peled, *Model checking*. MIT press, 1999.
- [10] 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.
- [11] 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.
- [12] 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: IEEE, 2014, pp. 75–80.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] M. Y. Vardi and P. Wolper, "An automata-theoretic approach to automatic program verification," in *1st Symposium in Logic in Computer Science (LICS)*. IEEE Computer Society, 1986.
- [18] M. Guo, K. H. Johansson, and D. V. Dimarogonas, "Motion and action planning under ltl specifications using navigation functions and action description language," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo Big Sight, Japan, November 2013, pp. 240–245.
- [19] M. Guo and D. V. Dimarogonas, "Reconfiguration in motion planning of single-and multi-agent systems under infeasible local ltl specifications," in *IEEE 52nd Annual Conference on Decision and Control (CDC)*, Florence, December 2013, pp. 2758–2763.