Distributed Intermittent Connectivity Control of Mobile Robot Networks

Yiannis Kantaros, *Student Member, IEEE,* and Michael M. Zavlanos, *Member, IEEE*

***Abstract*—In this paper we develop an intermittent communication framework for teams of mobile robots. 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. We design distributed controllers for the robots that determine meeting times at the nodes 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 meeting points. To generate motion plans that satisfy the LTL expression, we propose a novel technique that approximately decomposes the global LTL formula into local LTL formulas and assigns them to the robots. Since the approximate decomposition of the LTL formula can result in conflicting robot behaviors, we develop a distributed conflict resolution scheme that generates conflict-free motion plans that satisfy the global LTL expression. By appropriately introducing delays in the execution of the generated motion plans we show that the proposed controllers can be executed asynchronously.**

***Index Terms*—Multi-robot networks, intermittent communication, distributed LTL-based planning.**

I. Introduction

MANY coordinated tasks performed by teams of mobile robots e.g., area coverage and exploration [1], environmental monitoring [2], and synchronization [3], critically depend on the ability of the robots to exchange information with each other through a communication network that is connected either for all time, or intermittently but infinitely often. This requirement for network connectivity has recently led to a broad range of techniques to control it. Typically these methods focus on all-time connectivity, while intermittent connectivity is still widely considered an assumption. The reason is that connectivity is a global network property that requires coordination between the robots to control, which is unavailable when the robots operate in disconnect mode. In this paper, we address this challenge by presenting a distributed intermittent communication framework that schedules communication events between robots so that connectivity is ensured over time, infinitely often. The advantage of intermittent communication is that it provides more flexibility to the robots to accomplish their tasks as they are not constrained by all-time communication requirements.

Communication among robots has been typically modeled using proximity graphs and the communication problem is often treated as preservation of graph connectivity. For example, in [4] a function that measures the local connectedness of a network is introduced, which under mild assumptions can provide conditions for global network connectivity. Alternative methods to control graph connectivity rely on controlling the Fiedler value of the underlying graph either in a centralized [5] or distributed [6]–[10] fashion. Also, potential fields that model loss of connectivity as an obstacle in the free space can be employed for connectivity maintenance, as shown in [11]. A distributed hybrid approach to connectivity control is presented in [12] that decouples control of the discrete communication graph from continuous robot mobility via an efficient manipulation of communication links. Further distributed controllers for graph connectivity maintenance have been implemented in [13], [14]. A recent survey on graph theoretic methods for connectivity control can be found in [15]. In practice, the above graph-based communication models turn out to be rather conservative, since proximity does not necessarily imply tangible and reliable communication. Therefore, more realistic communication models have recently been proposed in [16]–[22] that take into account path loss, shadowing, and multi-path fading as well as optimal routing decisions for desired information rates.

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, or the effect of the ambient environment on wireless communications may prevent robots from operating independently, as dictated by their assigned tasks, in favor of behaviors that maintain a reliable communication network that is required for coordination. 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. Intermittent communication in multi-agent systems has been studied in consensus problems [23], [24], coverage problems [25], and in delay-tolerant networks [26]–[30]. The common assumption in these works is that the communication network is intermittently connected infinitely often, over time. Relevant is also work on event-based network control [31]–[35] where, although the network is assumed to be connected for all time, messages between the agents are exchanged intermittently when certain events take place.

In this paper, we lift all connectivity assumptions and, instead, control the communication network itself so that it is guaranteed to be intermittently connected over time, infinitely often. In particular, we consider robots that 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

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

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 [36], [37] 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 bottom-up approaches when independent LTL expressions are assigned to robots [38]–[44] or top-down approaches when a global LTL describing a collaborative task is assigned to a team of robots [45], [46], 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. Then robot mobility along the generated discrete motion plans is performed by a continuous controller giving rise to a hybrid robot system. By appropriately introducing delays in the execution of the generated motion plans we show the proposed controllers can also be executed in an asynchronous fashion while ensuring that there are no deadlocks and the global LTL is satisfied. In contrast, most relevant literature assumes that robot control is performed in a synchronous way [45], [46]. To the best of our knowledge, although specific to the problem under consideration, this is the first distributed, scalable, and asynchronous LTL-based framework for the coordination of teams of multiple robots.

The most relevant works to the one proposed here are presented in [47]–[51]. In particular, [47] presents an intermittent communication control scheme that ensures communication among robots infinitely often, however, this method is centralized and does not scale well with the number of robots. In [48] a distributed intermittent communication control scheme is proposed that requires synchronization among robots, unlike the approach developed here that is fully asynchronous. The work in [49] proposes a receding horizon framework for periodic connectivity that ensures recovery of connectivity within a given time horizon. This work is experimentally validated in [50]. 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. Moreover, when connectivity is recovered in [49] the whole network needs to be connected. Instead, our method does not require that the communication network is ever connected at once, but it ensures connectivity over time, infinitely often. Finally, a distributed synchronization scheme is presented in [51] 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.

The rest of this paper is organized as follows. In Section II we present some basic definitions and preliminaries in LTL and model checking theory. The problem formulation is described in Section III and the proposed distributed algorithm is presented in Section IV. Simulation studies are included in Section VI.

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 [36].

Linear temporal logic is a type of formal logic whose basic ingredients are a set of atomic propositions $\mathcal{AP}$, the boolean operators, i.e., 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* $\Box$, *eventually* $\Diamond$, *implication* $\Rightarrow$ which can be found in [36].

An infinite *word* σ 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 ω denotes infinite repetition and $\pi_k \in 2^{\mathcal{AP}}$, $\forall k$. The language $\texttt{Words}(\phi) = \{\sigma \in (2^{\mathcal{AP}})^\omega | \sigma \models \phi\}$, where $\models \subseteq (2^{\mathcal{AP}}) \times \phi$ is the satisfaction relation, is defined as the set of words that satisfy the LTL ϕ.

Any LTL formula ϕ can be translated into a Nondeterministic Büchi Automaton (NBA) over $2^{\mathcal{AP}}$ denoted by B [52], which is defined as follows:

Definition 2.1: A *Nondeterministic Büchi Automaton* (NBA) B over $2^{\mathcal{AP}}$ is defined as a tuple $B = (\mathcal{Q}_B, \mathcal{Q}_B^0, \Sigma, \rightarrow_B, \mathcal{F}_B)$ where

- $\mathcal{Q}_B$ is the set of states,
- $\mathcal{Q}_B^0 \subseteq \mathcal{Q}_B$ is a set of initial states,
- $\Sigma = 2^{\mathcal{AP}}$ is an alphabet,
- $\rightarrow_B \subseteq \mathcal{Q}_B \times \Sigma \times \mathcal{Q}_B$ is the transition relation,
- $\mathcal{F}_B \subseteq \mathcal{Q}_B$ is a set of accepting/final states.

An *infinite run* ρ_B of B over an infinite word $\sigma = \pi_0\pi_1\pi_2\ldots$, $\pi_k \in \Sigma = 2^{\mathcal{AP}}$ is a sequence $\rho_B = q_B^0 q_B^1 q_B^2 \ldots$ 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 ρ_B is called *accepting run* if $\texttt{Inf}(\rho_B) \cap \mathcal{F}_B \neq \varnothing$, where $\texttt{Inf}(\rho_B)$ represents the set of states that appear in ρ_B infinitely often. The words σ that result in an accepting run of B constitute the accepted language of B, denoted by $\mathcal{L}_B$, i.e., $\mathcal{L}_B = \texttt{Words}(\phi)$.

III. Problem Formulation

Assume R locations in space positioned at $\mathbf{v}_i \in \mathbb{R}^n$ and paths $\gamma_{ij} : [0,1] \rightarrow \mathbb{R}^n$ that connect two locations i and j such that $\gamma_{ij}(0) = \mathbf{v}_i$ and $\gamma_i(1) = \mathbf{v}_j$. The union of locations

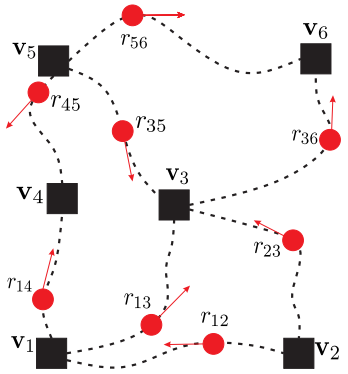


Fig. 1. A graphical illustration of the problem formulation. Black squares represent communication points and red circles stand for robots r_{ij} that move along paths γ_{ij} that are depicted by black dashed curves.

$\mathbf{v}_i$ 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). For example, while traveling along their assigned paths robots can monitor and collect sensing data in a region of interest. 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 model the environment in which r_{ij} resides by a transition system (TS) denoted by TS_{ij} that is defined as follows[1]

Definition 3.1 (Transition System): A *transition system* TS_{ij} is a tuple $\left(\mathcal{Q}_{ij}, q_{ij}^0, \mathcal{A}_{ij}, \rightarrow_{ij}, \mathcal{AP}, L_{ij}\right)$ 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 $q_{ij}^{\mathbf{v}_i}$'.
- $\rightarrow_{ij} \subseteq \mathcal{Q}_{ij} \times \mathcal{A}_{ij} \times \mathcal{Q}_{ij}$ is the transition relation,
- $\mathcal{AP}$ is the set of atomic propositions, and
- $L_{ij} : \mathcal{Q}_{ij} \rightarrow 2^{\mathcal{AP}}$ is an observation/output relation giving the set of atomic propositions that are satisfied at a state.

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

Definition 3.2 (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 $\tau_{ij}(1) = q_{ij}^0$, $\tau_{ij}(m) \in \mathcal{Q}_{ij}$, and $(\tau_{ij}(m), a_{ij}^m, \tau_{ij}(m+1)) \in \rightarrow_{ij}$, for some $a_{ij}^m \in \mathcal{A}_{ij}$, $\forall m \in \mathbb{N}_+$, where m is an index that points to the m-th entry of τ_{ij} denoted by $\tau_{ij}(m)$.

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}.

Definition 3.3 (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 3.4 (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 $\texttt{trace}(\tau_{ij}) \in \texttt{Words}(\phi)$, which is equivalently denoted by $\tau_{ij} \models \phi$.

Definition 3.5 (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 3.6 (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 3.7 (Trace of TS): The trace of a transition system TS_{ij} is defined as $\texttt{trace}(\text{TS}_{ij}) = \bigcup_{\tau_{ij} \in \mathcal{P}} \texttt{trace}(\tau_{ij})$, where $\mathcal{P}$ is the set of all infinite paths τ_{ij} in TS_{ij}.

B. Intermittent Communication

Due to limited communication capabilities we assume that robots can communicate only if they are physically close to each other; specifically, if they meet at common rendezvous locations, defined by the vertices of the mobility graph $\mathcal{G}$. 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. When meeting at these rendezvous locations, the robots can communicate the sensing data they have collected while traveling along their assigned paths to a user; see Figure 1. The presence of a network allows the robots to communicate data to a user in a multi-hop fashion, so that they do not have to leave their assigned regions. Applications of this framework involve distributed coverage, estimation, and surveillance.

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 $\mathcal{N}_{ij} =$

[1]Throughout the paper, we assume that the transition systems TS_{ij} do not have a terminal state.

$\mathcal{R}_i \cup \mathcal{R}_j \setminus \{r_{ij}\}$. Then the communication graph $\mathcal{G}_c$ is defined to be *connected over 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:

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

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 $\epsilon > 0$.

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.

Remark 3.8: Although the proposed framework is developed and motivated with the objective of ensuring intermittent communication within teams of robots in a distributed way, other applications are also possible. For example, instead of collecting sensor data, the robots can collect and transfer supplies from designated depots that fall along their motion paths. Then, during meeting events at the rendezvous locations, the robots can deliver supplies from one another, giving rise to a distributed transportation system.

IV. 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

$$\phi = \bigwedge\nolimits_{i\in\mathcal{V}} \phi_{\mathbf{v}_i}, \tag{4}$$

where $\phi_{\mathbf{v}_i}$ is defined as

$$\phi_{\mathbf{v}_i} = \square\Diamond \left(\bigwedge\nolimits_{j\in\mathcal{N}_i} \pi_{ij}^{\mathbf{v}_i} \right), \tag{5}$$

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 $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$, $\forall i \in \mathcal{V}$, can be constructed using existing tools from model checking theory [36], [37]. 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 briefly present how motion plans $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$ can be generated using existing automata-based model checking algorithms. 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. Automata-based LTL Model Checking

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 implemented using existing automata-based model checking methods [36], [37]. First the *Product Transition System* (PTS) $\text{TS}_{\mathbf{v}_i}$ is constructed, which essentially captures all the possible combinations of robots' states in their respective TS_{ij}, $\forall r_{ij} \in \mathcal{R}_i$ and is defined as follows:

Definition 4.1 (Product Transition System): Given $|\mathcal{N}_i|$ transition systems $\text{TS}_{ij_k} = \left(\mathcal{Q}_{ij_k}, q^0_{ij_k}, \mathcal{A}_{ij_k}, \rightarrow_{ij_k}, \mathcal{AP}, L_{ij_k}\right)$, where $j_k \in \mathcal{N}_i$, $k = 1, 2, \ldots, |\mathcal{N}_i|$, the *product transition system* $\text{TS}_{\mathbf{v}_i} = \text{TS}_{ij_1} \otimes \text{TS}_{ij_2} \otimes \cdots \otimes \text{TS}_{ij_{|\mathcal{N}_i|}}$ is a tuple $\left(\mathcal{Q}_{\text{TS}_{\mathbf{v}_i}}, q^0_{\text{TS}_{\mathbf{v}_i}}, \mathcal{A}_{\text{TS}_{\mathbf{v}_i}}, \longrightarrow_{\text{TS}_{\mathbf{v}_i}}, \mathcal{AP}_{\text{TS}_{\mathbf{v}_i}}, L_{\text{TS}_{\mathbf{v}_i}}\right)$ where

- $\mathcal{Q}_{\text{TS}_{\mathbf{v}_i}} = \mathcal{Q}_{ij_1} \times \mathcal{Q}_{ij_2} \times \cdots \times \mathcal{Q}_{ij_{|\mathcal{N}_i|}}$ is the set of states,
- $q^0_{\text{TS}_{\mathbf{v}_i}} = (q^0_{ij_1}, q^0_{ij_2}, \ldots, q^0_{ij_{|\mathcal{N}_i|}}) \in \mathcal{Q}_{\text{TS}_{\mathbf{v}_i}}$ is the initial state,
- $\mathcal{A}_{\text{TS}_{\mathbf{v}_i}} = \mathcal{A}_{ij_1} \times \mathcal{A}_{ij_2} \times \cdots \times \mathcal{A}_{ij_{|\mathcal{N}_i|}}$ is a set of actions,
- $\longrightarrow_{\text{TS}_{\mathbf{v}_i}} \subseteq \mathcal{Q}_{\mathbf{v}_i} \times \mathcal{A}_{\mathbf{v}_i} \times \mathcal{Q}_{\mathbf{v}_i}$ is the transition relation

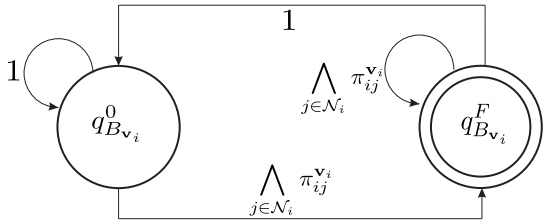


Fig. 2. Graphical depiction of the NBA $B_{\mathbf{v}_i}$ that corresponds to $\phi_{\mathbf{v}_i} = \square\Diamond\left(\bigwedge_{j\in\mathcal{N}_i}\pi^{\mathbf{v}_i}_{ij}\right)$. The set of states of $B_{\mathbf{v}_i}$ consists of an initial state $q^0_{B_{\mathbf{v}_i}}$ and a final state denoted by $q^F_{B_{\mathbf{v}_i}}$.

defined by the rule[2] $\dfrac{\bigwedge_{\forall j_k}\left(q_{ij_k}\xrightarrow{a_{ij_k}}{}_{ij_k}q'_{ij_k}\right)}{q_{\mathrm{TS}_{\mathbf{v}_i}}\xrightarrow{a_{\mathrm{TS}_{\mathbf{v}_i}}=\left(a_{ij_1},\ldots,a_{ij_{|\mathcal{N}_i|}}\right)}{}_{\mathrm{TS}_{\mathbf{v}_i}}q'_{\mathrm{TS}_{\mathbf{v}_i}}}$,[3]

- $\mathcal{AP}$ is the set of atomic propositions, and,
- $L_{\mathrm{TS}_{\mathbf{v}_i}} = \bigcup_{r_{ij_k}\in\mathcal{R}_i} L_{ij_k}$ is an observation/output relation giving the set of atomic propositions that are satisfied at a state.

Once the PTS $\mathrm{TS}_{\mathbf{v}_i}$ and the NBA $B_{\mathbf{v}_i}$ that corresponds to the LTL $\phi_{\mathbf{v}_i}$(see Figure 2) are constructed, a motion plan $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$ can be found by checking the non-emptiness of the language of the *Product Büchi Automaton* (PBA) $P_{\mathbf{v}_i} = \mathrm{TS}_{\mathbf{v}_i} \otimes B_{\mathbf{v}_i}$ [36], which is defined as follows:

Definition 4.2 (Product Büchi Automaton): Given the product transition system $\mathrm{TS}_{\mathbf{v}_i} = \left(\mathcal{Q}_{\mathrm{TS}_{\mathbf{v}_i}}, q^0_{\mathrm{TS}_{\mathbf{v}_i}}, \mathcal{A}_{\mathrm{TS}_{\mathbf{v}_i}}, \longrightarrow_{\mathrm{TS}_{\mathbf{v}_i}}, \mathcal{AP}, L_{\mathrm{TS}_{\mathbf{v}_i}}\right)$ and the NBA $B_{\mathbf{v}_i} = \left(\mathcal{Q}_{B_{\mathbf{v}_i}}, \mathcal{Q}^0_{B_{\mathbf{v}_i}}, 2^{\mathcal{AP}}, \rightarrow_{B_{\mathbf{v}_i}}, \mathcal{F}_{B_{\mathbf{v}_i}}\right)$, the *Product Büchi Automaton* $P_{\mathbf{v}_i} = \mathrm{TS}_{\mathbf{v}_i} \otimes B_{\mathbf{v}_i}$ is a tuple $\left(\mathcal{Q}_{P_{\mathbf{v}_i}}, \mathcal{Q}^0_{P_{\mathbf{v}_i}}, \longrightarrow_{P_{\mathbf{v}_i}}, \mathcal{F}_{P_{\mathbf{v}_i}}\right)$ where

- $\mathcal{Q}_{P_{\mathbf{v}_i}} = \mathcal{Q}_{\mathrm{TS}_{\mathbf{v}_i}} \times \mathcal{Q}_{B_{\mathbf{v}_i}}$ is the set of states,
- $\mathcal{Q}^0_{P_{\mathbf{v}_i}} = q^0_{\mathrm{TS}_{\mathbf{v}_i}} \times \mathcal{Q}^0_{B_{\mathbf{v}_i}}$ is a set of initial states,
- $\longrightarrow_{P_{\mathbf{v}_i}} \subseteq \mathcal{Q}_{P_{\mathbf{v}_i}} \times \mathcal{A}_{\mathrm{TS}_{\mathbf{v}_i}} \times 2^{\mathcal{AP}} \times \mathcal{Q}_{P_{\mathbf{v}_i}}$ is the transition relation defined by the rule: $\dfrac{\left(q_{\mathrm{TS}_{\mathbf{v}_i}}\xrightarrow{a_{\mathrm{TS}_{\mathbf{v}_i}}}{}_{\mathrm{TS}_{\mathbf{v}_i}}q'_{\mathrm{TS}_{\mathbf{v}_i}}\right)\wedge\left(q_{B_{\mathbf{v}_i}}\xrightarrow{L_{\mathrm{TS}_{\mathbf{v}_i}}\left(q'_{\mathrm{TS}_{\mathbf{v}_i}}\right)}{}_{B_{\mathbf{v}_i}}q'_{B_{\mathbf{v}_i}}\right)}{q_{P_{\mathbf{v}_i}}=\left(q_{\mathrm{TS}_{\mathbf{v}_i}},q_{B_{\mathbf{v}_i}}\right)\xrightarrow{a_{\mathrm{TS}_{\mathbf{v}_i}}}{}_{P_{\mathbf{v}_i}}q'_{P_{\mathbf{v}_i}}=\left(q'_{\mathrm{TS}_{\mathbf{v}_i}},q'_{B_{\mathbf{v}_i}}\right)}$,
- $\mathcal{F}_{P_{\mathbf{v}_i}} = \mathcal{Q}_{\mathrm{TS}_{\mathbf{v}_i}} \times \mathcal{F}_{B_{\mathbf{v}_i}}$ is a set of accepting/final states.

To check the non-emptiness of the language of $P_{\mathbf{v}_i}$, it suffices to check if there is a non-empty intersection between the words that can be generated by $\mathrm{TS}_{\mathbf{v}_i}$ and the words that satisfy the LTL $\phi_{\mathbf{v}_i}$. Thus, since $\mathtt{Words}(\phi) = \mathcal{L}_{B_{\mathbf{v}_i}}$, this is equivalent to checking that $\mathtt{trace}(\mathrm{TS}_{\mathbf{v}_i}) \cap \mathcal{L}_{B_{\mathbf{v}_i}} \neq \varnothing$, where $\mathtt{trace}(\mathrm{TS}_{\mathbf{v}_i}) \cap \mathcal{L}_{B_{\mathbf{v}_i}}$ is the language of $P_{\mathbf{v}_i}$. If this is the case, then to find a motion plan $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$, it suffices to find an infinite path $\tau_{\mathbf{v}_i}$ of $\mathrm{TS}_{\mathbf{v}_i}$ such that $\mathtt{trace}(\tau_{\mathbf{v}_i}) \in \mathtt{trace}(\mathrm{TS}_{\mathbf{v}_i}) \cap \mathcal{L}_{B_{\mathbf{v}_i}}$. To achieve that, an accepting run $\rho_{P_{\mathbf{v}_i}}$ of $P_{\mathbf{v}_i}$, i.e., a run that satisfies $\mathtt{Inf}(\rho_{P_{\mathbf{v}_i}}) \cap \mathcal{F}_{P_{\mathbf{v}_i}} \neq \varnothing$ needs to be computed. Typically, to derive such motion plans, the product automaton is viewed as a graph $\mathcal{G}_{P_{\mathbf{v}_i}} = \left\{\mathcal{V}_{P_{\mathbf{v}_i}}, \mathcal{E}_{P_{\mathbf{v}_i}}\right\}$, where $\mathcal{V}_{P_{\mathbf{v}_i}} = \mathcal{Q}_{P_{\mathbf{v}_i}}$ and the set of edges $\mathcal{E}_{P_{\mathbf{v}_i}}$ is determined by the transition relation $\longrightarrow_{P_{\mathbf{v}_i}}$. Then finding a path from an initial state to an accepting state in $\mathcal{G}_{P_{\mathbf{v}_i}}$ (prefix) followed by a cycle around this accepting state (suffix), which is repeated infinitely, results in an accepting run in a prefix-suffix structure as follows:

$$\begin{aligned}\rho_{P_{\mathbf{v}_i}} =& \rho^{\mathrm{pre}}_{P_{\mathbf{v}_i}}\left[\rho^{\mathrm{suf}}_{P_{\mathbf{v}_i}}\right]^\omega \\ =& \Big[\underbrace{(q^0_{\mathrm{TS}_{\mathbf{v}_i}}, q^0_{B_{\mathbf{v}_i}})}_{\in\mathcal{Q}^0_{P_{\mathbf{v}_i}}}(q^1_{\mathrm{TS}_{\mathbf{v}_i}}, q^1_{B_{\mathbf{v}_i}})\ldots\underbrace{(q^k_{\mathrm{TS}_{\mathbf{v}_i}}, q^k_{B_{\mathbf{v}_i}})}_{\in\mathcal{F}_{P_{\mathbf{v}_i}}}\Big] \\ & \Big[(q^k_{\mathrm{TS}_{\mathbf{v}_i}}, q^k_{B_{\mathbf{v}_i}})\ldots(q^m_{\mathrm{TS}_{\mathbf{v}_i}}, q^m_{B_{\mathbf{v}_i}})\Big]^\omega. \end{aligned} \tag{6}$$

Then, the projection of $\rho_{P_{\mathbf{v}_i}}$ onto $\mathrm{TS}_{\mathbf{v}_i}$, denoted by $\Pi|_{\mathrm{TS}_{\mathbf{v}_i}}\rho_{P_{\mathbf{v}_i}}$ results in a motion plan that has the following prefix-suffix structure:

$$\begin{aligned}\tau_{\mathbf{v}_i} = \Pi|_{\mathrm{TS}_{\mathbf{v}_i}}\rho_{P_{\mathbf{v}_i}} &= \tau^{\mathrm{pre}}_{\mathbf{v}_i}\left[\tau^{\mathrm{suf}}_{\mathbf{v}_i}\right]^\omega \\ &= \left[q^0_{\mathrm{TS}_{\mathbf{v}_i}}\ldots q^k_{\mathrm{TS}_{\mathbf{v}_i}}\right]\left[q^k_{\mathrm{TS}_{\mathbf{v}_i}}\ldots q^m_{\mathrm{TS}_{\mathbf{v}_i}}\right]^\omega, \end{aligned} \tag{7}$$

that satisfies $\phi_{\mathbf{v}_i}$ [52]. Notice that every robot r_{ij} needs to know only the initial states of robots that are in $\mathcal{N}_{ij}$ in order to construct the plan $\tau_{\mathbf{v}_i}$ using the above procedure.

In this work, we pick the shortest path and cycle to compute the prefix and suffix structures by assigning weights to the edges of $\mathcal{G}_{P_{\mathbf{v}_i}}$ which are related to the distance traveled by robots in their respective TSs if such a transition is picked.[4] In case there are multiple shortest paths and cycles, i.e., multiple possible motion plans $\tau_{\mathbf{v}_i}$, then we pick the motion plan with the shortest prefix and suffix part. Since every TS_{ij} can be thought of as a graph consisting of two nodes i and j and an edge connecting them, we conclude that the generated motion plan $\tau_{\mathbf{v}_i}$ has a common structure for all nodes $i \in \mathcal{V}$; therefore, all robots $r_{ij} \in \mathcal{R}_i$ compute the same motion plan $\tau_{\mathbf{v}_i}$. In particular, the motion plans $\tau_{\mathbf{v}_i}$ have the following form:

$$\tau_{\mathbf{v}_i} = \left[q^0_{\mathrm{TS}_{\mathbf{v}_i}} q^1_{\mathrm{TS}_{\mathbf{v}_i}}\right]\left[q^1_{\mathrm{TS}_{\mathbf{v}_i}}\right]^\omega, \tag{8}$$

where $q^0_{\mathrm{TS}_{\mathbf{v}_i}}$ is the initial state of robots in $\mathrm{TS}_{\mathbf{v}_i}$ and $q^1_{\mathrm{TS}_{\mathbf{v}_i}}$ is a state in $\mathcal{Q}_{\mathrm{TS}_{\mathbf{v}_i}}$ where all robots $r_{ij} \in \mathcal{R}_i$ are present at node i. In case $q^0_{\mathrm{TS}_{\mathbf{v}_i}} \equiv q^1_{\mathrm{TS}_{\mathbf{v}_i}}$, then the motion plan $\tau_{\mathbf{v}_i}$ has the following form:

$$\tau_{\mathbf{v}_i} = \left[q^0_{\mathrm{TS}_{\mathbf{v}_i}}\right]\left[q^0_{\mathrm{TS}_{\mathbf{v}_i}}\right]^\omega, \tag{9}$$

i.e., the robots are required to stay at their initial state indefinitely.

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

[2]The notation of this rule is along the lines of the notation used in [36]. In particular, it means that if the proposition above the solid line is true, then so does the proposition below the solid line.

[3]The state $q_{\mathrm{TS}_{\mathbf{v}_i}}$ stands for the state $\left(q_{ij_1},\ldots,q_{ij_{|\mathcal{N}_i|}}\right) \in \mathcal{Q}_{\mathrm{TS}_{\mathbf{v}_i}}$, where with slight abuse of notation $q_{ij_k} \in \mathcal{Q}_{ij_k}$. The state $q'_{\mathrm{TS}_{\mathbf{v}_i}}$ is defined accordingly.

[4]Similar techniques for the construction of motion plans have been used in the existing literature, see, e.g., [42], [53].

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: $p_{ij}^k(n_{ij}^{\mathbf{v}_i}) := \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$, $n_{ij}^{\mathbf{v}_i} = n_{im}^{\mathbf{v}_i}$, $\forall r_{im} \in \mathcal{N}_{ij}$;

2: **if** there exist constructed paths p_{mj}^k **then**

3: $\quad p_{ij}^k(n_{ij}^{\mathbf{v}_j}) := \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$, $n_{ij}^{\mathbf{v}_j} = n_{mj}^{\mathbf{v}_j}$, $\forall r_{mj} \in \mathcal{N}_{ij}$;

4: **else**

5: $\quad p_{ij}^k(n_{ij}^{\mathbf{v}_j}) := \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ provided either $p_{im}^k(n_{ij}^{\mathbf{v}_j}) = X$, or $p_{im}^k(n_{ij}^{\mathbf{v}_j}) = \Pi|_{\text{TS}_{im}}\tau_{\mathbf{v}_m}(k)$ 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;

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[5]

$$\begin{aligned}\tau_{ij} &= \tau_{ij}(1)\tau_{ij}(2)\ldots = [\tau_{ij}(m)]_{m=1}^{\infty}\\ &= \Bigg[\underbrace{X\ldots X\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k)X\ldots X\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k)X\ldots X}_{\ell}\Bigg]_{k=1}^{\infty}\\ &= [p_{ij}^k]_{k=1}^{\infty}, \end{aligned} \tag{10}$$

such that $\tau_{ij}(1) = q_{ij}^0$. In words, τ_{ij} can be written as the concatenation of the finite paths p_{ij}^k of TS_{ij}, $\forall k \in \mathbb{N}_+$. In (10) ℓ is the length of the path p_{ij}^k and is initially 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 4.5. Also, in (10), $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ 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}, i.e., in the X states a robot can decide to wait at its current state or move to the other 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 $\Pi|_{\text{TS}_{im}}\tau_{\mathbf{v}_i}(k)$, i.e., $p_{im}^k(n_{im}^{\mathbf{v}_i}) = \Pi|_{\text{TS}_{im}}\tau_{\mathbf{v}_i}(k)$. Then robot r_{ij} constructs the path p_{ij}^k based on three rules. The first rule determines the index $n_{ij}^{\mathbf{v}_i}$ and the other two rules determine the index $n_{ij}^{\mathbf{v}_j}$.

According to the first rule, the state $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ will be placed at the $n_{ij}^{\mathbf{v}_i}$-th entry, which is selected to be equal to $n_{im}^{\mathbf{v}_i}$, for all robots $r_{im} \in \mathcal{N}_{ij}$ [line 1, Alg. 1]. Due to this rule, all robots $r_{im} \in \mathcal{N}_{ij}$ will have a common index $n_{im}^{\mathbf{v}_i}$. This ensures that robot r_{ij} and all other robots $r_{im} \in \mathcal{R}_i$ will meet at the same time at $\mathbf{v}_i$, as it will be shown in Proposition 4.7. After placing the state $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ at the $n_{ij}^{\mathbf{v}_i}$-th entry of p_{ij}^k, what remains is to place the state $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ at the $n_{ij}^{\mathbf{v}_j}$-th entry of p_{ij}^k. Once this is done, the construction of the path p_{ij}^k is finalized and so is the construction of the motion plan τ_{ij}, as per (10). The index $n_{ij}^{\mathbf{v}_j}$ is determined by one of the two following rules. The objective of these two rules is to ensure that all robots $r_{mj} \in \mathcal{N}_{ij}$ have selected a common index $n_{mj}^{\mathbf{v}_j}$. This will ensure that robot r_{ij} and all other robots $r_{mj} \in \mathcal{R}_j$ will meet at the same time at $\mathbf{v}_j$, as it will be shown in Proposition 4.7.

Specifically, suppose 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 $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ 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 $\Pi|_{\text{TS}_{im}}\tau_{\mathbf{v}_m}(k)$ 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 $\Pi|_{\text{TS}_{jm}}\tau_{\mathbf{v}_j}(k)$ and $\Pi|_{\text{TS}_{jm}}\tau_{\mathbf{v}_m}(k)$ 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].[6] 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 described by the above three rules 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.

To illustrate Algorithm 1, consider the network of 3 robots shown in Figure 3. Let the sequence $\mathcal{S}$ be $\mathcal{S} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$. Hence, initially the robots in the set $\mathcal{R}_1 = \{r_{12}, r_{13}\}$

[5]Notice in (10) that, in general, the indices m and k are different. Particularly, index m points to the m-th entry of τ_{ij}, denoted by $\tau_{ij}(m)$, while index k points to the k-th finite path p_{ij}^k.

[6]If $i = 1$ then initially, a randomly selected robot r_{1j} creates arbitrarily its path p_{1j}^k by placing the states $\Pi|_{\text{TS}_{1j}}\tau_{\mathbf{v}_1}(k)$ and $\Pi|_{\text{TS}_{1j}}\tau_{\mathbf{v}_j}(k)$ 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.

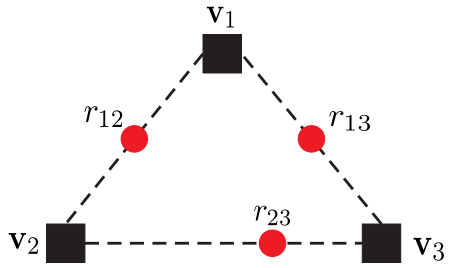


Fig. 3. An illustrative example related to Algorithm 1. The discrete motion plans τ_{ij} generated by Algorithm 1 for all robots r_{ij} are $\tau_{12} = \left[\Pi|_{\text{TS}_{12}}\tau_{\mathbf{v}_1}(k)\Pi|_{\text{TS}_{12}}\tau_{\mathbf{v}_2}(k)X\right]_{k=1}^{\infty}$, $\tau_{23} = \left[X\Pi|_{\text{TS}_{23}}\tau_{\mathbf{v}_2}(k)\Pi|_{\text{TS}_{23}}\tau_{\mathbf{v}_3}(k)\right]_{k=1}^{\infty}$, and $\tau_{13} = \left[\Pi|_{\text{TS}_{13}}\tau_{\mathbf{v}_1}(k)X\Pi|_{\text{TS}_{13}}\tau_{\mathbf{v}_3}(k)\right]_{k=1}^{\infty}$.

will construct their respective finite paths. Assume that initially robot r_{12} constructs the finite path p_{12}^k of length equal to $\ell = \max\{d_{\mathbf{v}_i}\}_{i=1}^3 + 1 = 3$ as follows: $p_{12}^k = \left[\Pi|_{\text{TS}_{12}}\tau_{\mathbf{v}_1}(k)\Pi|_{\text{TS}_{12}}\tau_{\mathbf{v}_2}(k)X\right]$. Then the finite path p_{12}^k is transmitted to the neighboring robot $r_{13} \in \mathcal{N}_{12}$ which is now responsible for constructing the finite path p_{13}^k. To construct p_{13}^k, according to the first rule, the state $\Pi|_{\text{TS}_{13}}\tau_{\mathbf{v}_2}(k)$ is placed at the first entry of p_{13}^k, i.e., $n_{13}^{\mathbf{v}_1} = n_{12}^{\mathbf{v}_1} = 1$. Next, the index $n_{13}^{\mathbf{v}_3}$ is determined by the third rule. Specifically, notice that among the two available entries in p_{13}^k for the state $\Pi|_{\text{TS}_{13}}\tau_{\mathbf{v}_3}(k)$ the entry $n_{13}^{\mathbf{v}_3} = 2$ is invalid, since the neighboring robot r_{12} has already constructed its vector $\mathbf{p}_{12}^k$ so that $n_{12}^{\mathbf{v}_2} = 2$ and for nodes 3 and 2 it holds that $3 \in \mathcal{N}_2$. Therefore, robot r_{13} selects $n_{13}^{\mathbf{v}_3} = 3$ and constructs the finite path $p_{13}^k = \left[\Pi|_{\text{TS}_{13}}\tau_{\mathbf{v}_1}(k)X\Pi|_{\text{TS}_{13}}\tau_{\mathbf{v}_3}(k)\right]$. At the next iteration of Algorithm 1 the robots in the set $\mathcal{R}_2 = \{r_{23}, r_{12}\}$ construct their finite paths. Robot r_{12} has already constructed the finite path p_{12}^k at the previous iteration. Robot r_{23} constructs the finite path $p_{23}^k = \left[X\Pi|_{\text{TS}_{23}}\tau_{\mathbf{v}_2}(k)\Pi|_{\text{TS}_{23}}\tau_{\mathbf{v}_3}(k)\right]$ using the first and the second rule. Finally, the robots in the set $\mathcal{R}_3 = \{r_{23}, r_{13}\}$ have already constructed their finite paths at previous iterations.

Remark 4.3: 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.

Remark 4.4: In the proposed intermittent communication framework, we assume that communication occurs at a meeting point i only if all robots in the set $\mathcal{R}_i$ are physically present at location $\mathbf{v}_i$. Note that such an assumption can be relaxed so that pairwise communication of the robots at the meeting points, or any other more complicated communication protocol can be employed. Specifically, robots in the set $\mathcal{R}_i$ can be divided into T_i teams, where the m-th team is denoted by $\mathcal{T}_i^m$, $m \in \{1, 2, \ldots, T_i\}$. Then communication at node i can occur if all robots of any team $\mathcal{T}_i^m$ are present at location $\mathbf{v}_i$. In this scenario, to ensure dissemination of information in the network, we need to ensure that the graph whose nodes are the teams $\mathcal{T}_i^m$, $m \in \{1, 2, \ldots, T_i\}$ and whose edges are formed between teams $\mathcal{T}_i^m$ and $\mathcal{T}_i^e$, for $m \neq e$, that share a common robot is connected. Notice that to account for such communication protocols the main modification that is required in Algorithm 1 is that robot r_{ij}, instead of constructing motion plans $\tau_{\mathbf{v}_i}$ and $\tau_{\mathbf{v}_j}$, will construct motion plans $\tau_{\mathcal{T}_i^m}$ and $\tau_{\mathcal{T}_j^m}$ for all teams m that robot r_{ij} belongs to. Each motion plan $\tau_{\mathcal{T}_i^m}$ will ensure that all robots in $\mathcal{T}_i^m$ meet infinitely often at node i. Then these motion plans can be ordered in a similar way, as in Algorithm 1, to construct free-of-conflicts motion plans τ_{ij}.

C. Correctness

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.

Proposition 4.5: 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 $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}$ 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 4.5 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 4.6: 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 $\tau_{ij}(m)$ to $\tau_{ij}(m+1)$, for all $m \in \mathbb{N}_+$, 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 of 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 $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ and $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ such that $n_{ij}^{\mathbf{v}_i} \neq n_{ij}^{\mathbf{v}_j}$, as shown in Proposition 4.5. Therefore, robot r_{ij} will never need to be at states $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k)$ and $\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k)$ at the same time, which completes the proof. ■

Proposition 4.7: 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.

Proof: To prove this result, it suffices to show that $\tau_{\mathcal{R}_i} \models \phi_{\mathbf{v}_i}$, for all nodes $i \in \mathcal{V}$, where $\tau_{\mathcal{R}_i}$ represents the composition of non-conflicting motion plans τ_{ij}, for all robots

$r_{ij} \in \mathcal{R}_i$, generated by Algorithm 1. Equivalently, according to Definition 3.4 it suffices to show that

$$\texttt{trace}(\tau_{\mathcal{R}_i}) \in \texttt{Words}(\phi_{\mathbf{v}_i}). \tag{11}$$

First, assume that a motion plan $\tau_{\mathbf{v}_i} \models \phi_{\mathbf{v}_i}$ has been constructed using a standard model checking method as described in Section IV-A and, therefore, we have[7]

$$\texttt{trace}(\tau_{\mathbf{v}_i}) \in \texttt{Words}(\phi_{\mathbf{v}_i}). \tag{12}$$

Assume also that all robots $r_{ij_q} \in \mathcal{R}_i$, $q = 1, 2, \ldots, |\mathcal{R}_i|$ have constructed motion plans τ_{ij_q} as shown in (10). The composition of these plans (see also Definition 3.5) can be written in the following general form

$$\begin{aligned}\tau_{\mathcal{R}_i} &= \otimes_{\forall r_{ij} \in \mathcal{R}_i} \tau_{ij} = \tau_{\mathcal{R}_i}(1)\tau_{\mathcal{R}_i}(2)\cdots = [\tau_{\mathcal{R}_i}(m)]_{m=1}^{\infty} \\ &= \Big[\underbrace{\Big(\tau_{ij_1}(m), \tau_{ij_2}(m), \ldots, \tau_{ij_{|\mathcal{R}_i|}}(m)\Big)}_{\in \mathcal{Q}_{\text{TS}_{\mathbf{v}_i}}}\Big]_{m=1}^{\infty}\end{aligned} \tag{13}$$

Using (10), the motion plan $\tau_{\mathcal{R}_i}$ can be expanded as follows

$$\begin{aligned}\tau_{\mathcal{R}_i} = \Big[&\underbrace{\Big(p^k_{ij_1}(1), p^k_{ij_2}(1), \ldots, p^k_{ij_{|\mathcal{R}_i|}}(1)\Big)}_{\in \mathcal{Q}_{\text{TS}_{\mathbf{v}_i}}} \\ &\Big(p^k_{ij_1}(2), p^k_{ij_2}(2), \ldots, p^k_{ij_{|\mathcal{R}_i|}}(2)\Big) \ldots \\ &\Big(p^k_{ij_1}(\ell), p^k_{ij_2}(\ell), \ldots, p^k_{ij_{|\mathcal{R}_i|}}(\ell)\Big)\Big]_{k=1}^{\infty} = \Big[p^k_{\mathcal{R}_i}\Big]_{k=1}^{\infty},\end{aligned} \tag{14}$$

i.e., the motion plan $\tau_{\mathcal{R}_i}$ can be written as the concatenation of the finite paths $p^k_{\mathcal{R}_i}$ of $\text{TS}_{\mathbf{v}_i}$, for all $k \in \mathbb{N}_+$. [8]

By construction of Algorithm 1, there is a common index denoted by $n^{\mathbf{v}_i}$, $1 \leq n^{\mathbf{v}_i} \leq \ell$, $\forall r_{ij_q} \in \mathcal{R}_i$ such that $p^k_{ij_q}(n^{\mathbf{v}_i}) = \Pi|_{\text{TS}_{ij_q}} \tau_{\mathbf{v}_i}(n^{\mathbf{v}_i})$, $\forall r_{ij_q} \in \mathcal{R}_i$. Therefore, the $n^{\mathbf{v}_i}$-th state in $\text{TS}_{\mathbf{v}_i}$ contained in $p^k_{\mathcal{R}_i}$ has the following form:

$$\begin{aligned}p^k_{\mathcal{R}_i}(n^{\mathbf{v}_i}) &= \Big(p^k_{ij_1}(n^{\mathbf{v}_i}), p^k_{ij_2}(n^{\mathbf{v}_i}), \ldots, p^k_{ij_{|\mathcal{R}_i|}}(n^{\mathbf{v}_i})\Big) \\ &= \Big(\Pi|_{\text{TS}_{ij_1}} \tau_{\mathbf{v}_i}(k), \ldots, \Pi|_{\text{TS}_{ij_{|\mathcal{R}_i|}}} \tau_{\mathbf{v}_i}(k)\Big) \equiv \tau_{\mathbf{v}_i}(k),\end{aligned} \tag{15}$$

while for any other index $n \neq n^{\mathbf{v}_i}$, $1 \leq n \leq \ell$ there is at least a robot $r_{ij_q} \in \mathcal{R}_i$ whose state inside $p^k_{\mathcal{R}_i}(n) \in \mathcal{Q}_{\text{TS}_{\mathbf{v}_i}}$ is not determined by $\Pi|_{\text{TS}_{ij_q}} \tau_{\mathbf{v}_i}(k)$.

Observe in (14)-(15) that mobility of all robots $r_{ij_q} \in \mathcal{R}_i$ is dictated 'intermittently' by $\tau_{\mathbf{v}_i}$. In other words, this means that all robots $r_{ij_q} \in \mathcal{R}_i$ that are in a state $\tau_{\mathbf{v}_i}(k)$ will traverse through some states of $\text{TS}_{\mathbf{v}_i}$ that are determined by $\tau_{\mathcal{R}_i}$ and, eventually, they will move to the state $\tau_{\mathbf{v}_i}(k+1)$, $\forall k$, i.e., robots do not move from state $\tau_{\mathbf{v}_i}(k)$ to $\tau_{\mathbf{v}_i}(k+1)$, directly. Now, we need to show that these intermediate transitions that robots make in order to move from $\tau_{\mathbf{v}_i}(k)$ to $\tau_{\mathbf{v}_i}(k+1)$ are admissible in $\text{TS}_{\mathbf{v}_i}$ and do not violate $\phi_{\mathbf{v}_i}$. These transitions are admissible as implied by Proposition 4.6. Also, they cannot violate $\phi_{\mathbf{v}_i}$, since the LTL expressions $\phi_{\mathbf{v}_i}$ in (5), for all i, do not include the negation operator $\neg$ in front of the atomic propositions $\pi^{\mathbf{v}_i}_{ij}$ defined in (3). Therefore, robots r_{ij_q} can move freely in their respective TS_{ij_q} without violating the LTL expressions $\phi_{\mathbf{v}_i}$.

Since all robots in $\mathcal{R}_i$ will eventually pass through all states $\tau_{\mathbf{v}_i}(k)$, this means that as robots move according to the motion plan $\tau_{\mathcal{R}_i}$, the generated trace will certainly include the atomic propositions that are satisfied along the path $\tau_{\mathbf{v}_i}$; see also Definition 3.3. In other words, the sequence of atomic propositions $\texttt{trace}(\tau_{\mathbf{v}_i})$ is a subsequence of $\texttt{trace}(\tau_{\mathcal{R}_i})$ while the additional atomic propositions that exist in $\texttt{trace}(\tau_{\mathcal{R}_i})$ cannot violate $\phi_{\mathbf{v}_i}$, as previously discussed. Combining this observation with (12), we conclude that

$$\texttt{trace}(\tau_{\mathcal{R}_i}) \in \texttt{Words}(\phi_{\mathbf{v}_i}). \tag{16}$$

Consequently, we get $\tau_{\mathcal{R}_i} \models \phi_{\mathbf{v}_i}$, for all nodes $i \in \mathcal{V}$, i.e., the global LTL expression (2) is satisfied, which completes the proof. ■

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^{pre}_{ij} is traversed only once and the suffix part τ^{suf}_{ij} is repeated infinitely.

Proposition 4.8: Algorithm 1 generates discrete motion plans τ_{ij} for all robots r_{ij} in a prefix-suffix structure, i.e.,

$$\tau_{ij} = \tau^{\text{pre}}_{ij} \left[\tau^{\text{suf}}_{ij}\right]^{\omega}, \tag{17}$$

where

$$\tau^{\text{pre}}_{ij} = \left[p^k_{ij}\right]_{k=1}^{\max(S_i,S_j)}, \tag{18}$$

$$\tau^{\text{suf}}_{ij} = \left[p^k_{ij}\right]_{k=\max(S_i,S_j)+1}^{\max(P_i,P_j)+\max(S_i,S_j)}, \tag{19}$$

where S_i and P_i refer to the length of the prefix and suffix part of $\tau_{\mathbf{v}_i}$, respectively.

Proof: The result can straightforwardly be derived by expanding the motion plan τ_{ij} given in (10) and observing the repetitive pattern $\tau^{\text{suf}}_{ij} = \left[p^k_{ij}\right]_{k=\max(S_i,S_j)+1}^{\max(P_i,P_j)+\max(S_i,S_j)}$. ■

V. Asynchronous Intermittent Communication

In section IV, we showed that if all robots r_{ij} pick synchronously their next states in TS_{ij} according to the motion plans τ_{ij}, then the global LTL expression (2) is satisfied, as shown in Proposition 4.7. However, assuming that robot motion is performed in a synchronous way is rather conservative due to, e.g., uncertainty and exogenous disturbances that may affect the arrival times of the robots at the meeting locations. Motivated by this fact, in this section, we show that the generated motion plans can be executed asynchronously, as well, by appropriately introducing delays in the continuous-time execution of τ_{ij}. In this way, the proposed controller

[7] Note that, throughout this proof, the only difference between the plans $\tau_{\mathcal{R}_i}$ and $\tau_{\mathbf{v}_i}$ is that the first one has been derived through composing all motion plans τ_{ij}, $\forall r_{ij} \in \mathcal{R}_i$ generated by Algorithm 1, while $\tau_{\mathbf{v}_i}$ is the plan for all robots $r_{ij} \in \mathcal{R}_{ij}$ computed as described in Section III-A. Consequently, motion plans $\tau_{\mathcal{R}_i}$ are the result of non-conflicting robot motion plans τ_{ij}, which is not the case for the motion plans $\tau_{\mathbf{v}_i}$.

[8] Notice in (13)-(14) that the indices k and m are in general different. The index m points to the current state of robots $r_{ij_q} \in \mathcal{R}_i$, which is represented by $\tau_{\mathcal{R}_i}(m)$, while the index k points to the k-th path $p^k_{\mathcal{R}_i}$ of $\text{TS}_{\mathbf{v}_i}$. Also, hereafter, we denote by $p^k_{\mathcal{R}_i}(n)$, the n-the state of $\text{TS}_{\mathbf{v}_i}$ contained in the path $p^k_{\mathcal{R}_i}$, i.e., $p^k_{\mathcal{R}_i}(n) = \Big(p^k_{ij_1}(n), p^k_{ij_2}(n), \ldots, p^k_{ij_{|\mathcal{R}_i|}}(n)\Big) \in \mathcal{Q}_{\text{TS}_{\mathbf{v}_i}}$.

becomes more robust to uncertainty and disturbances. In the rest of this section, we assume that robots pick asynchronously their next states in TS_{ij} and that the travel time from node i to node j, denoted by T_{ij}, is finite and varies across the robots. Note also that in general the travel times T_{ij} can be time dependent and, consequently, the paths γ_{ij} that connect nodes i and j can change over time. Due to the asynchronous execution of the controllers, we rewrite (10) as follows:

$$\begin{aligned}\tau_{ij} &= \tau_{ij}(1)\tau_{ij}(2)\ldots = [\tau_{ij}(m_{ij})]_{m_{ij}=1}^{\infty}\\ &= \Big[\underbrace{X\ldots X\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij})X\ldots X\Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k_{ij})X\ldots X}_{\ell}\Big]_{k_{ij}=1}^{\infty}\\ &= \left[p_{ij}^{k_{ij}}\right]_{k_{ij}=1}^{\infty}, \end{aligned} \tag{20}$$

where the only difference with (10) lies on the indices m_{ij} and k_{ij}. Hereafter, m_{ij} refers to the m_{ij}-th entry of τ_{ij} and k_{ij} refers to the k_{ij}-th finite path $p_{ij}^{k_{ij}}$. The different indices for robots's states in the infinite paths τ_{ij} allow us to model the situation where the robots pick asynchronously their next states in TS_{ij}.

The asynchronous execution of the infinite paths τ_{ij} is described in Algorithm 2. In particular, robot r_{ij} moves from state $\tau_{ij}(m_{ij}-1)$ to $\tau_{ij}(m_{ij})$ according to a continuous-time motion controller $\mathbf{u}_{ij}(t) \in \mathbb{R}^n$ that belongs to the tangent space of γ_{ij} at $\mathbf{x}_{ij}(t)$ [line 1, Alg. 2]. Assume that $\tau_{ij}(m_{ij}) = q_{ij}^{\mathbf{v}_i}$. When robot r_{ij} arrives at node i it checks if the index m_{ij} belongs to the set $\mathcal{M}_{\mathbf{v}_i} \subset \mathbb{N}_+$ [line 2, Alg. 2] defined as follows:

$$\begin{aligned}&\mathcal{M}_{\mathbf{v}_i} =\\ &\left\{m_{ij} \in \mathbb{N}_+ | \tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij}) = q_{ij}^{\mathbf{v}_i},\ \forall r_{ij} \in \mathcal{R}_i\right\}.\end{aligned} \tag{21}$$

In principle, the set $\mathcal{M}_{\mathbf{v}_i}$ is computed by Algorithm 1 and collects indices that point to states of τ_{ij} where robots $r_{ij} \in \mathcal{R}_i$ have to meet at node i. In case $m_{ij} \in \mathcal{M}_{\mathbf{v}_i}$, then robot r_{ij} waits at node i until all other robots $r_{ij} \in \mathcal{R}_i$ are present there [line 3, Alg. 2]. When this happens, or if $m_{ij} \notin \mathcal{M}_{\mathbf{v}_i}$, then robot r_{ij} moves towards the next state $\tau_{ij}(m_{ij}+1)$ [lines 5,13,15 Alg. 2]. The same reasoning holds also if $\tau_{ij}(m_{ij}) = q_{ij}^{\mathbf{v}_j}$ [lines 7-10, Alg 2].

Algorithm 2 Asynchronous Intermittent Communication

Require: Motion plan τ_{ij} and sets of indices $\mathcal{M}_{\mathbf{v}_i}$, $\mathcal{M}_{\mathbf{v}_j}$;
1: Move towards $\tau_{ij}(m_{ij})$ according to (1);
2: **if** $m_{ij} \in \mathcal{M}_{\mathbf{v}_i}$ **then**
3: Wait at node i until all robots $r_{ij} \in \mathcal{R}_i$ are present at node i;
4: **if** all robots $r_{ij} \in \mathcal{R}_i$ are present at node i **then**
5: $m_{ij} = m_{ij} + 1$;
6: **end if**
7: **else if** $m_{ij} \in \mathcal{M}_{\mathbf{v}_j}$ **then**
8: Wait at node j until all robots $r_{ij} \in \mathcal{R}_j$ are present at node j;
9: **if** all robots $r_{ij} \in \mathcal{R}_j$ are present at node j **then**
10: $m_{ij} = m_{ij} + 1$;
11: **end if**
12: **else**
13: $m_{ij} = m_{ij} + 1$;
14: **end if**
15: Goto line 1;

A. Correctness

In this section, we show that the global LTL expression (2) is satisfied even if the robots execute asynchronously the motion plans τ_{ij} as per Algorithm 2. To prove this, we first show that the network will never deadlock if robots move according to Algorithm 2.

Proposition 5.1: The composition of the transition systems TS_{ij} under the discrete motion plans τ_{ij} generated by Algorithm 1 satisfy the deadlock property, i.e., the system will never come to a halt where two or more robots will wait for each other's action indefinitely.

Proof: Let $\mathcal{W}_i \subset \mathcal{R}_i$ denote a set that includes all robots that are waiting at node i. Assume that the robots in $\mathcal{R}_i \backslash \mathcal{W}_i$ never arrive at that node so that a meeting at node i never occurs. This means that the robots in $\mathcal{R}_i \backslash \mathcal{W}_i$ are waiting indefinitely at nodes $j \neq i$ to meet with their neighbors in $\mathcal{R}_j$. Let r_{ij} be any of the robots in $\mathcal{R}_i \backslash \mathcal{W}_i$. The fact that r_{ij} remains indefinitely at node j means that a meeting at that node j never occurs by construction of Algorithm 2. Following an argument similar to the above, we conclude that the robots in $\mathcal{R}_j \backslash \mathcal{W}_j$ are waiting indefinitely at nodes $k \neq j, i$ to meet with their neighbors in $\mathcal{R}_k$. Therefore, if a meeting never occurs at a node i, then all robots need to be waiting at meeting points and, consequently, there is no meeting point where all robots are present. Throughout the rest of the proof we will refer to this network configuration as a stationary configuration.

In what follows we show by contradiction that the network can never get trapped in a stationary configuration. In particular, we will show that if the network gets trapped at a stationary configuration, then this means that some robots $r_{ij} \in \mathcal{R}_i$ missed a meeting at node i at a previous time instant, which cannot happen by construction of Algorithm 2. Recall first that by construction of the set $\mathcal{M}_{\mathbf{v}_i}$ in (21), robots r_{ij} wait for the arrival of all robots in $\mathcal{R}_i$ only in states $\tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij}) = q_{ij}^{\mathbf{v}_i}$, $m_{ij} \in \mathcal{M}_{\mathbf{v}_i}$, while they traverse through the states $\tau_{ij}(m_{ij})$, $m_{ij} \notin \mathcal{M}_{\mathbf{v}_i}$ without waiting. Hereafter, we denote the indices $m_{ij} \in \mathcal{M}_{\mathbf{v}_i}$ by $m_{\mathbf{v}_i}$. Assume now that there is an arbitrary time instant t_0 at which the network is at a stationary configuration. Then, all robots are at states $\tau_{ij}(m_{\mathbf{v}_i}(t_0)) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij}(t_0)) = q_{ij}^{\mathbf{v}_i}$, for some $m_{\mathbf{v}_i}(t_0) \in \mathcal{M}_{\mathbf{v}_i}$, $\forall i$. Define also the set $\mathcal{M}_{\min}(t_0) = \left\{m_{\mathbf{v}_i}(t_0) | m_{\mathbf{v}_i}(t_0) = \min\left\{m_{\mathbf{v}_e}(t_0)\right\}_{e=1}^{N}\right\}$ and let $m_{\mathbf{v}_e}(t_0)$ be an index such that $m_{\mathbf{v}_e}(t_0) \in \mathcal{M}_{\min}(t_0)$. By assumption there are robots $r_{eg} \in \mathcal{W}_e(t_0)$, $g \in \mathcal{N}_e$ and robots $r_{en} \in \mathcal{R}_e \setminus \mathcal{W}_e(t_0)$, $n \in \mathcal{N}_e$ and, therefore, their states are $\tau_{eg}(m_{\mathbf{v}_e}(t_0)) = \Pi|_{\text{TS}_{eg}}\tau_{\mathbf{v}_e}(k_{eg}(t_0)) = q_{eg}^{\mathbf{v}_e}$ and $\tau_{en}(m_{\mathbf{v}_n}) = \Pi|_{\text{TS}_{en}}\tau_{\mathbf{v}_n}(k_n) = q_{en}^{\mathbf{v}_n}$, respectively. Since $m_{\mathbf{v}_e}(t_0) \in \mathcal{M}_{\min}(t_0)$ we have that $m_{\mathbf{v}_n}(t_0) \geq m_{\mathbf{v}_e}(t_0)$, which along with the fact that $n \in \mathcal{N}_e$ results in $m_{\mathbf{v}_n}(t_0) > m_{\mathbf{v}_e}(t_0)$

by construction of Algorithm 1. This leads to the two following scenarios. First, the fact that $m_{\mathbf{v}_n}(t_0) > m_{\mathbf{v}_e}(t_0)$ means that there exists a time instant $t < t_0$ at which robots r_{en} were at states $\tau_{en}(m_{\mathbf{v}_e}(t)) = \Pi|_{\text{TS}_{en}}\tau_{\mathbf{v}_e}(k_{en}(t)) = q_{en}^{\mathbf{v}_e}$ without waiting for the arrival of all robots in $\mathcal{R}_e$, since there are still robots waiting at node e. However, such a scenario is precluded by construction of Algorithm 2. Second, in case robots r_{en} had waited at node e at iteration $m_{\mathbf{v}_e}$(t), as dictated by Algorithm 2, then this leads again to a contradiction, since that would mean that robots r_{em} must be in a state $\tau_{eg}(m_{eg}(t_0))$, where $m_{eg}(t_0) > m_{\mathbf{v}_e}(t_0)$ at the time instant t_0. Therefore, our initial assumption that the system can get trapped at a stationary configuration is invalid, which completes the proof. ∎

In Proposition 4.7 we showed that the composition $\tau_{\mathcal{R}_i}$ of motion plans τ_{ij}, for all robots $r_{ij} \in \mathcal{R}_i$, generated by Algorithm 1, satisfies the local LTL expression $\phi_{\mathbf{v}_i}$ assuming that all robots in $\mathcal{R}_i$ move synchronously. The main idea in that proof was that according to the motion plan $\tau_{\mathcal{R}_i}$, all robots $r_{ij} \in \mathcal{R}_i$ will eventually pass through all states $\tau_{\mathbf{v}_i}(k_{ij})$. This entailed that $\texttt{trace}(\tau_{\mathcal{R}_i}) \in \texttt{Words}(\phi_{\mathbf{v}_i})$ for all nodes i, since $\texttt{trace}(\tau_{\mathbf{v}_i}) \in \texttt{Words}(\phi_{\mathbf{v}_i})$. Therefore, we concluded that the global LTL statement (2) is satisfied. In a similar way, in Proposition 5.2, we show that if the robots move asynchronously, as per Algorithm 2, then all robots $r_{ij} \in \mathcal{R}_i$ will pass through all states of $\tau_{\mathbf{v}_i}$ for all nodes i, and, then, the global LTL expression is satisfied.

Proposition 5.2: Robot mobility dictated by Algorithm 2 satisfies the global LTL statement (2).

Proof: In case robots move asynchronously in their respective transition systems, then according to Algorithm 2 every robot r_{ij} waits at states $\tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij}) = q_{ij}^{\mathbf{v}_i}$, $\forall m_{ij} \in \mathcal{M}_{\mathbf{v}_i} = \left\{m_{ij} \in \mathbb{N}_+ | \tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij}) = q_{ij}^{\mathbf{v}_i},\ \forall r_{ij} \in \mathcal{R}_i\right\}$. Notice that by construction of motion plan $\tau_{\mathbf{v}_i}$ in (8)-(9), every state in $\tau_{\mathbf{v}_i}$, except possibly for the initial state $\tau_{\mathbf{v}_i}(1) = q^0_{\text{TS}_{\mathbf{v}_i}}$, as shown in (8), determines a case where $\tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij}) = q_{ij}^{\mathbf{v}_i}$ for all robots in $\mathcal{R}_i$. This equivalently means that if $\tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij})$ then $m_{ij} \in \mathcal{M}_{\mathbf{v}_i}$ and, consequently, robot r_{ij} needs to wait for the arrival of all other robots in $\mathcal{R}_i$; except if $\tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij}) = q^0_{ij}$, which happens only if $0 \leq m_{ij} \leq \ell$ by construction of the finite paths $p_{ij}^{k_{ij}}$ in (20). Consequently, we conclude that all robots $r_{ij} \in \mathcal{R}_i$ will pass through all states in the motion plan $\tau_{\mathbf{v}_i}$, since the system will never deadlock for any initial configuration under the proposed control scheme, as shown in Proposition 5.1. Then we have that $\texttt{trace}(\tau_{\mathcal{R}_i}) \in \texttt{Words}(\phi_{\mathbf{v}_i})$ for all nodes i due to Proposition 4.7. Consequently, the global LTL statement (2) is satisfied, which completes the proof. ∎

Algorithm 2 requires the sets $\mathcal{M}_{\mathbf{v}_i}$ and $\mathcal{M}_{\mathbf{v}_j}$ that contain an infinite number of indices that point at states in τ_{ij} at which robot r_{ij} has to wait at nodes i and j, respectively. In the following proposition, we show that these sets have a finite representation, as well.

Proposition 5.3: Algorithm 1 constructs sets $\mathcal{M}_{\mathbf{v}_i}$ and $\mathcal{M}_{\mathbf{v}_j}$ that have the following form

$$\mathcal{M}_{\mathbf{v}_i} = \left\{[n_{ij}^{\mathbf{v}_i} + e\ell]_{e=1}^{\infty}\right\} \tag{22}$$

$$\mathcal{M}_{\mathbf{v}_j} = \left\{[n_{ij}^{\mathbf{v}_j} + e\ell]_{e=1}^{\infty}\right\}, \tag{23}$$

where $n_{ij}^{\mathbf{v}_i}$ and $n_{ij}^{\mathbf{v}_j}$ are indices in the finite path p_{ij}^k such that $p_{ij}^{k_{ij}}(n_{ij}^{\mathbf{v}_i}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij})$ and $p_{ij}^{k_{ij}}(n_{ij}^{\mathbf{v}_j}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k_{ij})$.

Proof: In Proposition 5.2, we showed that when $\tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij})$ then $m_{ij} \in \mathcal{M}_{\mathbf{v}_i}$. Hence, it suffices to show that $\tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij})$ and $\tau_{ij}(m_{ij}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k_{ij})$ if $m_{ij} = n_{ij}^{\mathbf{v}_i} + e\ell$ and $m_{ij} = n_{ij}^{\mathbf{v}_j} + e\ell$, respectively, for $e \in \mathbb{N}_+$. This result can straightforwardly be derived by observing that the motion plans τ_{ij} in (20) can be written as a concatenation of the finite paths $p_{ij}^{k_{ij}}$ of length ℓ, i.e., $\tau_{ij} = p_{ij}^1 p_{ij}^2 \cdots = [p_{ij}^{k_{ij}}]_{k_{ij}=1}^{\infty}$, where $p_{ij}^{k_{ij}}(n_{ij}^{\mathbf{v}_i}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij})$ and $p_{ij}^{k_{ij}}(n_{ij}^{\mathbf{v}_j}) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k_{ij})$ for all $k_{ij} \in \mathbb{N}_+$. Consequently, we have that

$$\tau_{ij}(n_{ij}^{\mathbf{v}_i} + e\ell) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_i}(k_{ij}), \tag{24}$$

$$\tau_{ij}(n_{ij}^{\mathbf{v}_j} + e\ell) = \Pi|_{\text{TS}_{ij}}\tau_{\mathbf{v}_j}(k_{ij}), \tag{25}$$

$\forall e \in \mathbb{N}$. Since the case $e = 0$ always refers to the initial states of robots in their respective TS_{ij} by construction of the motion plans τ_{ij} in (20), and robots never wait at their initial states, we have that

$$\mathcal{M}_{\mathbf{v}_i} = \left\{n_{ij}^{\mathbf{v}_i} + \ell, n_{ij}^{\mathbf{v}_i} + 2\ell, \dots\right\} = \left\{[n_{ij}^{\mathbf{v}_i} + e\ell]_{e=1}^{\infty}\right\}$$

$$\mathcal{M}_{\mathbf{v}_j} = \left\{n_{ij}^{\mathbf{v}_j} + \ell, n_{ij}^{\mathbf{v}_j} + 2\ell, \dots\right\} = \left\{[n_{ij}^{\mathbf{v}_j} + e\ell]_{e=1}^{\infty}\right\},$$

which completes the proof. ∎

VI. Numerical Experiments

In this section, a simulation study is provided that illustrates our approach for a network of $N = 42$ robots that move along the edges of a mobility graph with $R = 32$ communication points, as shown in Figure 4. To model uncertainty in robot mobility, we assume that values for the travel times T_{ij} are generated by a uniform distribution $U(1, 4)$ every time robot r_{ij} reaches node j. The same holds for travel times T_{ji}, as well. Also, we assume that robots move back and forth between rendezvous locations along paths γ_{ij} that are straight lines, as shown in Figure 4. Therefore, the controller $\mathbf{u}_{ij}$ that drives robot r_{ij} from node i to node j is given by:

$$\mathbf{u}_{ij} = \frac{1}{T_{ij}} \frac{\mathbf{v}_i - \mathbf{v}_j}{\|\mathbf{v}_i - \mathbf{v}_j\|}. \tag{26}$$

As discussed in Section IV, in 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 4, these motion plans have the following form:

$$\tau_{\mathbf{v}_1} = [(q_{12}^{\mathbf{v}_1}, q_{13}^{\mathbf{v}_3})\,(q_{12}^{\mathbf{v}_1}, q_{13}^{\mathbf{v}_1})]\,[(q_{12}^{\mathbf{v}_1}, q_{13}^{\mathbf{v}_1})]^{\omega}, \tag{27}$$

$$\tau_{\mathbf{v}_2} = [(q_{12}^{\mathbf{v}_1}, q_{23}^{\mathbf{v}_2})\,(q_{12}^{\mathbf{v}_2}, q_{23}^{\mathbf{v}_2})]\,[(q_{12}^{\mathbf{v}_2} q_{23}^{\mathbf{v}_2})]^{\omega}, \tag{28}$$

$$\tau_{\mathbf{v}_3} = [(q_{13}^{\mathbf{v}_3}, q_{23}^{\mathbf{v}_2}, q_{35}^{\mathbf{v}_5}, q_{34}^{\mathbf{v}_4})\,(q_{13}^{\mathbf{v}_3}, q_{23}^{\mathbf{v}_3}, q_{35}^{\mathbf{v}_3}, q_{34}^{\mathbf{v}_3})]\,[(q_{13}^{\mathbf{v}_3}, q_{23}^{\mathbf{v}_3}, q_{35}^{\mathbf{v}_3}, q_{34}^{\mathbf{v}_3})]^{\omega}. \tag{29}$$

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

$$\tau_{12} = [p_{12}^k]_{k=1}^{\infty} = [\Pi|_{\mathrm{TS}_{12}}\tau_{\mathbf{v}_1}(k)\Pi|_{\mathrm{TS}_{12}}\tau_{\mathbf{v}_2}(k)X]_{k=1}^{\infty}, \quad (30)$$
$$\tau_{13} = [p_{13}^k]_{k=1}^{\infty} = [\Pi|_{\mathrm{TS}_{13}}\tau_{\mathbf{v}_1}(k)X\Pi|_{\mathrm{TS}_{12}}\tau_{\mathbf{v}_3}(k)]_{k=1}^{\infty}, \quad (31)$$
$$\tau_{23} = [p_{23}^k]_{k=1}^{\infty} = [X\Pi|_{\mathrm{TS}_{23}}\tau_{\mathbf{v}_2}(k)\Pi|_{\mathrm{TS}_{12}}\tau_{\mathbf{v}_3}(k)]_{k=1}^{\infty}. \quad (32)$$

Using the result from Proposition 4.8, the above motion plans can be written in the following prefix-suffix form[9]

$$\tau_{12} = \Big[\underbrace{q_{12}^{\mathbf{v}_1}q_{12}^{\mathbf{v}_1}X}_{=p_{12}^1}\underbrace{q_{12}^{\mathbf{v}_1}q_{12}^{\mathbf{v}_2}X}_{=p_{12}^2}\Big]\Big[\underbrace{q_{12}^{\mathbf{v}_1}q_{12}^{\mathbf{v}_2}X}_{=p_{12}^3}\Big]^{\omega}, \quad (33)$$

$$\tau_{13} = \Big[\underbrace{q_{13}^{\mathbf{v}_3}Xq_{13}^{\mathbf{v}_3}}_{=p_{13}^1}\underbrace{q_{13}^{\mathbf{v}_1}Xq_{13}^{\mathbf{v}_3}}_{=p_{13}^2}\Big]\Big[\underbrace{q_{13}^{\mathbf{v}_1}Xq_{13}^{\mathbf{v}_3}}_{=p_{13}^3}\Big]^{\omega}, \quad (34)$$

$$\tau_{23} = \Big[\underbrace{q_{23}^{\mathbf{v}_2}q_{23}^{\mathbf{v}_2}q_{23}^{\mathbf{v}_2}}_{=p_{23}^1}\underbrace{Xq_{23}^{\mathbf{v}_2}q_{23}^{\mathbf{v}_3}}_{=p_{23}^2}\Big]\Big[\underbrace{Xq_{23}^{\mathbf{v}_2}q_{23}^{\mathbf{v}_3}}_{=p_{23}^3}\Big]^{\omega}. \quad (35)$$

Moreover, according to Proposition 5.3, the sets $\mathcal{M}_{\mathbf{v}_i}$ for nodes 1, 2, and, 3 have the following form

$$\mathcal{M}_{\mathbf{v}_1} = \{[1+3e]_{e=1}^{\infty}\}, \quad (36)$$
$$\mathcal{M}_{\mathbf{v}_2} = \{[2+3e]_{e=1}^{\infty}\}, \quad (37)$$
$$\mathcal{M}_{\mathbf{v}_3} = \{[3+3e]_{e=1}^{\infty}\}, \quad (38)$$

since $\ell = 3$, $n_{1j}^{\mathbf{v}_1} = 1$, $n_{2j}^{\mathbf{v}_2} = 2$, and $n_{3j}^{\mathbf{v}_3} = 3$.

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:

$$v_{ij}(t) = \frac{1}{|\mathcal{R}_i|}\sum_{r_{im}\in\mathcal{R}_i} v_{im}(t). \quad (39)$$

In Figure 4 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 with other robots. Figure 5 shows that eventually all robots reach a consensus on the numbers $v_{ij}(t)$, which means that communication takes place infinitely often, as expected due to Proposition 5.2. The meeting events over time for rendezvous points 1 and 3 are depicted in Figure 6. Observe also in Figure 6 that the meeting time instances do not depend linearly on time, which means that communication at these nodes is aperiodic.

Note also that it would be impossible to generate motion plans τ_{ij} by applying existing LTL-based planning methods due to the large size of the network. Specifically, these techniques rely on the construction of a product transition system (PTS), whose state space has dimension $|\mathcal{Q}_{\mathrm{PTS}}| = \times_{\forall r_{ij}}|\mathcal{Q}_{ij}| = 2^N$, since $|Q_{ij}| = 2$. This PTS is combined with the Büchi Automaton B that corresponds to the LTL statement (2) to construct a Product Büchi Automaton whose state space has dimension $|\mathcal{Q}_{\mathrm{PBA}}| = |\mathcal{Q}_{\mathrm{PTS}}| \times |\mathcal{Q}_B| = 2^N \times |\mathcal{Q}_B|$. Finally the PBA is represented by a graph with $2^N \times |\mathcal{Q}_B|$ nodes so that motion plans τ_{ij} can be generated using graph search techniques. Notice that as the number of robots increases the state space of this graph increases and, therefore, it cannot be manipulated in practice. Additionally, notice that the distributed intermittent communication framework presented in [51] cannot be applied in this simulation case, since the mobility graph shown in Figure 4 is not bipartite.

[9]Note that lengths of the prefixes and suffixes of the motion plans shown in (27), (28), and (29) are $S_1 = S_2 = S_3 = 2$ and $P_1 = P_2 = P_3 = 1$, respectively.

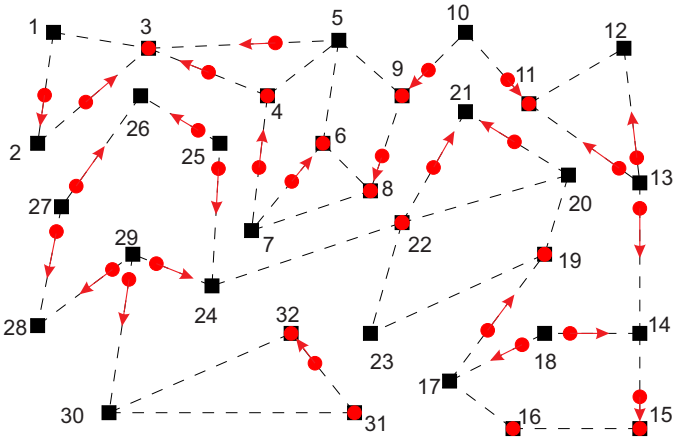


Fig. 4. Intermittent communication of $N = 42$ robots moving along the edges of an underlying mobility graph. Black squares represent rendezvous points and red circles stand for robots. Red arrows show the direction along which robots are moving. Robots without arrows are waiting at the meeting points for the arrival of robots.

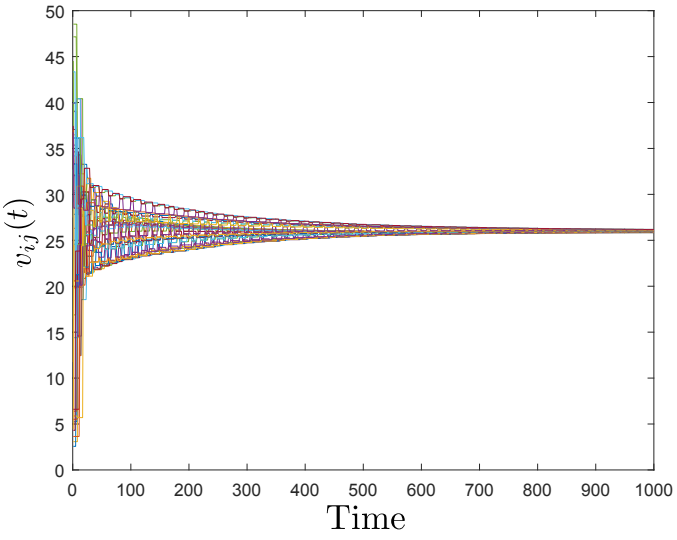
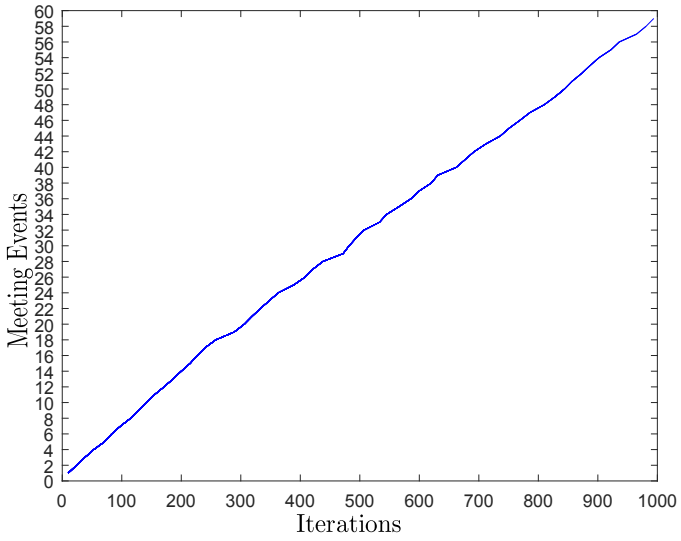


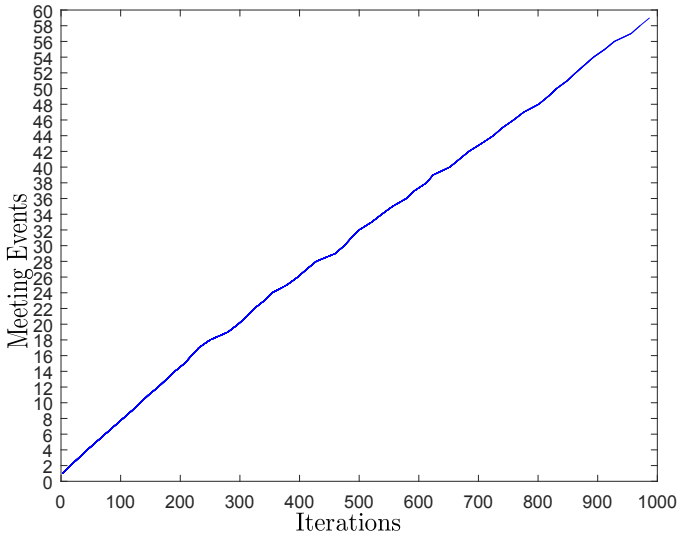
Fig. 5. Graphical depiction of consensus of numbers $v_{ij}(t)$.

VII. 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



(a) Node 1



(b) Node 3

Fig. 6. Graphical depiction of meeting events at nodes 1 (Figure 6(a)) and 3 (Figure 6(b)) with respect to time.

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 meeting 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. By introducing delays in the continuous-time execution of the generated motion plans, we allowed robots to move asynchronously in their respective transition systems while ensuring that the network will never deadlock and the global LTL expression is satisfied. The proposed distributed control algorithm was verified by computer simulations. In our future research work we plan to incorporate tasks described by LTL specifications as well as allow the robots to select not only communication time instances but also communication locations.

REFERENCES

[1] Y. Kantaros and M. M. Zavlanos, "Distributed communication-aware coverage control by mobile sensor networks," *Automatica*, vol. 63, pp. 209–220, 2016.

[2] S. He, J. Chen, and Y. Sun, "Coverage and connectivity in duty-cycled wireless sensor networks for event monitoring," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 475–482, 2012.

[3] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, "Flocking while preserving network connectivity," in *46th IEEE Conference on Decision and Control*. New Orleans, LA: IEEE, 2007, pp. 2919–2924.

[4] 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.

[5] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[6] M. C. DeGennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December 2006, pp. 3628–3633.

[7] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[8] 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.

[9] 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.

[10] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1411–1423, 2013.

[11] 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.

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

[13] M. Ji and M. B. Egerstedt, "Distributed coordination control of multi-agent systems while preserving connectedness." *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, August 2007.

[14] K. Savla, G. Notarstefano, and F. Bullo, "Maintaining limited-range connectivity among second-order agents," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 187–205, 2009.

[15] 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.

[16] M. Lindhé and K. H. Johansson, "Adaptive exploitation of multipath fading for mobile sensors," in *IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010, pp. 1934–1939.

[17] 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.

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

[19] A. Ghaffarkhah and Y. Mostofi, "Communication-aware motion planning in mobile networks," *IEEE Transactions on Automatic Control,*, vol. 56, no. 10, pp. 2478–2485, 2011.

[20] ——, "Channel learning and communication-aware motion planning in mobile networks," in *IEEE American Control Conference*, Baltimore, Maryland, USA, 2010, pp. 5413–5420.

[21] 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.

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

[23] G. Wen, Z. Duan, W. Ren, and G. Chen, "Distributed consensus of multi-agent systems with general linear node dynamics and intermittent communications," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 16, pp. 2438–2457, 2014.

[24] G. Wen, Z. Duan, G. Chen, and W. Yu, "Consensus tracking of multi-agent systems with lipschitz-type node dynamics and switching topologies," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 2, pp. 499–511, 2014.

[25] Y. Wang and I. I. Hussein, "Awareness coverage control over large-scale domains with intermittent communications," *Automatic Control, IEEE Transactions on*, vol. 55, no. 8, pp. 1850–1859, 2010.

[26] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, 2007, pp. 32–40.

[27] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[28] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *Communications Surveys & Tutorials, IEEE*, vol. 8, no. 1, pp. 24–37, 2006.

[29] E. P. Jones, L. Li, J. K. Schmidtke, and P. A. Ward, "Practical routing in delay-tolerant networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 8, pp. 943–959, 2007.

[30] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 5, pp. 748–760, 2008.

[31] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.

[32] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[33] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[34] X. Wang and M. D. Lemmon, "Event-triggered broadcasting across distributed networked control systems," in *American Control Conference, 2008*, Seattle, Washington, USA, June 2008, pp. 3139–3144.

[35] ——, "Event-triggering in distributed networked control systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 586–601, 2011.

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

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

[38] 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.

[39] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Hybrid controllers for path planning: A temporal logic approach," in *44th IEEE Conference on Decision and Control, European Control Conference, (CDC-ECC)*, Seville, Spain, December 2005, pp. 4885–4890.

[40] ——, "Temporal logic motion planning for mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005, pp. 2020–2025.

[41] 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.

[42] 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.

[43] ——, "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.

[44] 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.

[45] 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*, Orlando, FL, USA, December 2011, pp. 2718–2723.

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

[47] 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.

[48] ——, "A distributed LTL-based approach for intermittent communication in mobile robot networks," in *American Control Conference*, Boston, MA, USA, July, 2016, pp. 5557–5562.

[49] 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.

[50] G. Hollinger, S. Singh *et al.*, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, 2012.

[51] 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.

[52] 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.

[53] 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.

Yiannis Kantaros received the Diploma in Electrical and Computer Engineering in 2012 from the University of Patras, Patras, Greece. He is currently working toward the Ph.D. degree in the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA. His current research interests include distributed control, distributed optimization, multi-agent systems and robotics. He received the Best Student Paper Award at the 2nd IEEE Global Conference on Signal and Information Processing in 2014.

Michael M. Zavlanos received the Diploma in Mechanical Engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2002, and the M.S.E. and Ph.D. degrees in Electrical and Systems Engineering from the University of Pennsylvania, Philadelphia, PA, in 2005 and 2008, respectively.

From 2008 to 2009 he was a Post-Doctoral Researcher in the Department of Electrical and Systems Engineering at the University of Pennsylvania, Philadelphia. He then joined the Stevens Institute of Technology, Hoboken, NJ, as an Assistant Professor of Mechanical Engineering, where he remained until 2012. Currently, he is an assistant professor of mechanical engineering and materials science at Duke University, Durham, NC. He also holds a secondary appointment in the department of electrical and computer engineering. His research interests include a wide range of topics in the emerging discipline of networked systems, with applications in robotic, sensor, communication, and biomolecular networks. He is particularly interested in hybrid solution techniques, on the interface of control theory, distributed optimization, estimation, and networking.

He is a recipient of the 2014 Office of Naval Research Young Investigator Program (YIP) Award and the 2011 National Science Foundation Faculty Early Career Development (CAREER) Award. He was also a finalist for the best student paper award at CDC 2006.