A Self–Routing Protocol for Distributed Consensus on Logical Information

Adriano Fagiolini, Simone Martini, Davide Di Baccio, and Antonio Bicchi

***Abstract*— In this paper, we address decision making problems, depending on a set of input events, with networks of dynamic agents that have partial visibility of such events. Previous work by the authors proposed so–called logical consensus approach, by which a network of agents, that can exchange binary values representing their local estimates of the events, is able to reach a unique and consistent decision. The approach therein proposed is based on the construction of an iterative map, whose computation is centralized and guaranteed under suitable conditions on the input visibility and graph connectivity. Under the same conditions, we extend the approach in this work by allowing the construction of a logical linear consensus system that is globally stable in a fully distributed way. The effectiveness of the proposed method is showed through the real implementation of a wireless sensor network as a framework for the surveillance of an urban area.**

I. Introduction

Recent years have witnessed a constant migration of interests toward applications where the use of robotic multi–agent systems offers many advantages w.r.t. the traditional solution of single controlled system. Thanks to their flexibility, multi–agent systems are now perceived as crucial technology not only for effectively exploiting the increasing availability of diverse, heterogeneous, and distributed online information sources, but also as a framework for building complex, robust, and distributed control systems, which exploit the efficiencies of an organized behavior. Examples include physiological systems and gene networks [1], large scale energy systems, and aerial or land vehicles [2]–[4]. It is envisioned that, in a near future, groups of autonomous robots will be able to collaborate by exchanging information through a wireless connection, which will enable applications such as large–scale sensing of environmental monitoring and provide on–demand communication in rescue operations.

For most of these problems, solutions have been proposed that can be ultimately formulated as consensus systems over *continuous domains*, where local agents exchange and combine data consisting of real vectors or scalars. Falling into this linear framework are most of the key papers on consensus [5]–[7]. Notwithstanding the richness and boldness of this literature, the above mentioned applications and indeed many others would benefit from availability of more general forms of consensus, where agents are *de facto* able to reach an agreement on non–scalar quantities. In this respect, the work of Cortes *et al.* for achieving consensus on general functions is an interesting result toward this direction [8]. Very recently, [9] has addressed the sensing coverage problem with agents that are allowed to move in a discrete, network–like environment. The problem of detecting misbehaving agents within a collection of robots that are supposed to plan their motions according to a share set of rules was considered in [10]. Therein, the objective is attained through use of a *set–valued consensus algorithm*, where local agents exchange data representing free and occupied regions of the environment. Whereas these problems have been separately addressed in different manners, we proposed in [11] the notion of Boolean consensus systems as a unifying framework for achieving consensus on Boolean information (not only including binary data). In fact, what really prevents, in our opinion, a wide exploitation of the multi–agent paradigm is the lack of a systematic approach to the design of a generic consensus algorithm that is applicable in a vast number of scenarios. This is well–known to computer scientists that have studied consensus on generic data and provided efficient solutions that can tolerate even the presence of misbehaving or simply faulty agents (see e.g. Lynch's book [12]).

In this vein, it is worth noting that, at a suitable abstract level, every multi–agent system requires that agents consent on a centralized logical decision. This decision depends on a set of input events that have to be estimated by the agents themselves. In [13], so–called logical consensus approach was proposed, by which a network of agents that are able to exchange binary values representing their local estimates of the events, is able to reach a unique and consistent decision. The approach is based on the construction of an iterative map, whose computation is centralized and guaranteed under suitable conditions on the input visibility and graph connectivity. Under the same conditions, we extend it in this work by allowing the construction of a logical linear consensus system in a fully distributed way. Such a distributed synthesis is instrumental in mobile networked robots, where the presence of a centralized supervisor is impossible or the hypothesis of a fixed communication topology is unrealistic. The solution consists of so–called Self Routing Network Protocol (SRNP) allowing a set of (possibly mobile) robots randomly deployed, to self configure their communication network in such a way that a unique and consistent decision can be reached. The effectiveness of the proposed method is showed through an experimental setup consisting of a wireless sensor network for surveillance of an urban area.

II. Problem Formulation

We consider application scenarios requiring computation of a set of p *decisions*, $y_1, \ldots, y_p$, that depend on m logical

Authors are with the Interdepartmental Research Center "E.Piaggio" Faculty of Engineering - University of Pisa, e-mail: {a.fagiolini, s.martini, d.dibaccio, bicchi}@centropiaggio.unipi.it.

events, $u_1, \dots, u_m$. Such events may represent e.g. the presence of an intruder or of a fire within an indoor environment. More precisely, for any given combination of input events, we consider a *decision task* that requires computation of the following system of logical functions:

$$\begin{cases} y_1 = f_1(u_1, \dots, u_m)\,, \\ \cdots \\ y_p = f_p(u_1, \dots, u_m)\,, \end{cases} \tag{1}$$

where each $f_i : \mathbb{B}^m \to \mathbb{B}$ consists of a logical condition on the inputs. Let us denote with $u = (u_1, \dots, u_m)^T \in \mathbb{B}^m$ the input event vector, and with $y = (y_1, \dots, y_p)^T \in \mathbb{B}^p$ the output decision vector. Then, we will write $y = f(u)$ as a compact form of Eq. 1, where $f = (f_1, \dots, f_p)^T$, with $f : \mathbb{B}^m \to \mathbb{B}^p$, is a logical vector function. It is worth noting that computation of f is *centralized* in the sense that it may require knowledge of the entire input vector u to determine the output vector y.

Our approach to solve the decision task consists of employing a collection of n agents, $\mathcal{A}_1, \dots, \mathcal{A}_n$, that are supposed to cooperate and possibly exchange locally available information. We assume that each agent is described by a triple $\mathcal{A}_i = (\mathcal{S}_i, \mathcal{P}_i, \mathcal{C}_i)$, where $\mathcal{S}_i$ is a collection of sensors, $\mathcal{P}_i$ is a processor that is able to perform elementary logical operations such as $\{\texttt{and}, \texttt{or}, \texttt{not}\}$, and $\mathcal{C}_i$ is a collection of communication devices allowing transmission of only sequences of binary digits, 0 and 1, namely strings of bits. Although we assume that every agent has the same processing capability, i.e. $\mathcal{P}_i = \mathcal{P}$ for all i, we consider situations where agents may be *heterogeneous* in terms of sensors and communication devices. Due to this diversity as well as the fact that agents are placed at different locations, a generic agent i may or may not be able to measure a given input event u_j, for $j \in 1, \dots, m$. Therefore, we can conveniently introduce a *visibility matrix* $V \in \mathbb{B}^{n \times m}$ such that we have $V_{i,j} = 1$ if, and only if, agent $\mathcal{A}_i$ is able to measure input event u_j, or, in other words, if the i–th agent is directly *reachable* from the j–th input. Moreover, for similar reasons of diversity and for reducing battery consumption, each agent is able to communicate only with a subset of other agents. This fact is captured by introducing a *communication matrix* $C \in \mathbb{B}^{n \times n}$, where $C_{i,k} = 1$ if, and only if, agent $\mathcal{A}_i$ is able to receive a data from agent $\mathcal{A}_k$. Hence, agents specified by row $C_{i,:}$ will be referred to as C–neighbors of the i–th agent. The introduction of visibility relations between inputs and agents immediately implies that, at any instant t, only a subset of agents is able to measure the state of each input u_j, for all j. Therefore, to effectively accomplish the given decision task, we need that such an information *flows* from one agent to another, consistently with available communication paths. We require all agents reach an agreement on the centralized decision $y = f(u)$, so that any agent can be *polled* and provide consistent complete information. In this perspective, we pose the problem of reaching a *consensus on logical values*.

In this view, we can imagine that each agent $\mathcal{A}_i$ has a local *state* vector, $X_i = (X_{i,1}, \dots, X_{i,q}) \in \mathbb{B}^q$, that is a *string of bits*. Denote with $X(t) = (X_1^T(t), \dots, X_n^T(t))^T \in \mathbb{B}^{n \times q}$ a matrix representing the network state at a discrete time t. Hence, we assume that each agent $\mathcal{A}_i$ is a *dynamic node* that updates its local state X_i through a *distributed* logical update function F that depends on its state, on the state of its C–neighbors, and on the reachable inputs, i.e. $X_i(t+1) = F_i(X(t), u(t))$. Moreover, we assume that each agent $\mathcal{A}_i$ is able to produce a logical output decision vector $Y_i = (y_{i,1}, \dots, y_{i,p}) \in \mathbb{B}^p$ through a suitable distributed logical output function G depending on the local state X_i and on the reachable inputs u, i.e. $Y_i(t) = G_i(X_i(t), u(t))$. Let us denote with $Y(t) = (Y_1^T(t), \dots, Y_p^T(t))^T \in \mathbb{B}^{p \times q}$ a matrix representing the network output at a discrete time t. Therefore, the network evolution can be modeled as the *distributed finite–state iterative system*

$$\begin{cases} X(t+1) &=& F(X(t), u(t))\,, \\ Y(t) &=& G(X(t), u(t))\,, \end{cases} \tag{2}$$

where we have $F = (F_1^T, \dots, F_n^T)^T$, with $F_i : \mathbb{B}^q \times \mathbb{B}^m \to \mathbb{B}^q$, and $G = (G_1^T, \dots, G_n^T)^T$, with $G_i : \mathbb{B}^q \times \mathbb{B}^m \to \mathbb{B}^p$.

In a fully decentralized setting, every agent is unaware of all inputs and all other agents' existence, and it only knows the index list $v_i \overset{\text{def}}{=} \{v_{i,1}, v_{i,2}, \dots\} \subseteq \{1, \dots, m\}$ of the events that it can "see" and the index list $c_i \overset{\text{def}}{=} \{c_{i,1}, c_{i,2} \dots\} \subseteq \{1, \dots, n\}$ of its neighbors. In this case, the above mentioned centralized visibility and communication matrices, $V = \{V_j(i)\}$ and $C = \{C_{i,j}\}$, can be reconstructed according to the rules

$$V_j(i) = \begin{cases} 0 & \text{if } i \notin v_j \\ 1 & \text{otherwise}\,, \end{cases} \qquad C_{i,j} = \begin{cases} 0 & \text{if } i \notin c_j \\ 1 & \text{otherwise}\,. \end{cases}$$

Therefore, we are interested in solving the following

Problem 1 (Distributed Synthesis of Consensus Maps): Given a decision system as in Eq. 1, the visibility and communication lists, v_i and c_i, for all i, design a distributed logical consensus system of the form in Eq. 2, that is distributed, i.e. every agent directly uses only information compatible with its own v_i and c_i, and that converges to the centralized decision $y^* = f(u)$, i.e. $Y(t) = \mathbf{1}_n\,(y^*)^T$, for all initial network state $X(0)$ and inputs u.

III. Centralized Consensus Map Synthesis

A solution to the centralized version of the synthesis problem was presented in [13], where the complete visibility and communication matrices are known and available for the consensus system's designer. The proposed solution is recalled in this section for the reader's convenience. It consists of an algorithm that, under suitable conditions on the input visibility and graph connectivity, generates a logical linear consensus system that is distributed, i.e. compliant with given C and V, and optimal, i.e. it minimizes the number of messages to be exchanged and the time needed to reach the consensus (a.k.a. *rounds*).

First consider vectors $C^k V_j$, for $k = 0, 1, \dots,$ each containing 1 in all entries corresponding to agents that are reachable from input u_j after *exactly* k steps. The i–th element of $C^k V_j$ is 1 if, and only if, there exists a *path* of length k from any agent directly reached by u_j to agent $\mathcal{A}_i$. Recall that, by definition of graph diameter, all agents that are reachable from an initial set of agents are indeed reached in at most $\operatorname{diam}(G)$ steps, with $\operatorname{diam}(G) \leq n-1$. Let us denote with κ the *visibility diameter* of the pair (C, V_j) being the number of steps after which the sequence $\{C^k V_j\}$ does not reach new agents. Agents that can be reached by the j–input are specified by non–null elements of the Boolean vector $I_j = \sum_{k=0}^{n-1} C^k V_j$, that contains 1 for all agents for which there exists at least one path originating from an agent that is able to measure u_j.

Suppose, for simplicity, that only agent $\mathcal{A}_1$ is able to measure u_j. Then, a straightforward and yet optimal *strategy to allow the information on u_j flowing* through the network is obtained if agent $\mathcal{A}_1$ communicates its measurement to all its C–neighbors, which in turn will communicate it to all their C–neighbors without overlapping, and so on. In this way, we have that every agent $\mathcal{A}_i$ receives u_j from exactly one minimum–length path originating from agent $\mathcal{A}_1$. The vector sequence $\{C^k V_j\}$ can be exploited to this aim. Indeed, it trivially holds that $C^k V_j = C(C^{k-1} V_j)$, meaning that agents reached after k steps have received the input value from agents that were reached after exactly $k-1$ steps. Then, any consecutive sequence of agents that is extracted from non–null elements of vectors in $\{C^k V_j\}$ are (C, V_j)–compliant by construction. A consensus strategy would minimize the number of rounds if, and only if, at the k–th step, all agents specified by non–null elements of vector $C^k V_j$ receives the value of u_j from the agents specified by non–null elements of vector $C^{k-1} V_j$. Nevertheless, to minimize also the number of messages, only agents specified by non–null elements of vector $C^k V_j$ and that have not been reached yet must receive u_j. If vector $I_j = \sum_{i=0}^{i=k} C^i V_j$ is iteratively updated during the design phase, then the set of all agents that must receive a message on u_j are specified by non–null elements of vector $C^k V_j \wedge \neg I_j$. By doing this, an optimal pair (C^*, V_j^*) allowing a consensus to be established over the reachable subgraph is obtained. Observe that is $C^* = S\,C \leq C$, where S is a suitable selection matrix.

This procedure actually gives us only a suggestion on how to build a consensus system that solves the centralized version of Problem 1. Theorem 1, stated below in this section, allows us to say that a simple logical linear consensus algorithm of the form

$$x(t+1) = F_j\, x(t) + B_j\, u_j(t)\,, \tag{3}$$

where $F_j = C^*$, $B_j = V_j^*$, and $x \in \mathbb{B}^n$, allows consensus to be reached over the entire reachable subgraph. In all cases where a unique generic agent $\mathcal{A}_i$ is directly reachable from input u_j, an optimal communication matrix C^* for a linear consensus of the form of Eq. 3 can be iteratively found as the incidence matrix of a *input–propagating spanning tree* having $\mathcal{A}_i$ as the root. Then, an optimal pair (C^*, V_j^*) can be written as $C^* = P^T\,(S\,C)\,P$, and $V_j^* = P^T\,V_j$, where S is a selection matrix, and P is a permutation matrix. Furthermore, C^* has the following lower–block triangular form:

$$C^* = \left(\begin{array}{cccc|c} 0 & 0 & \cdots & 0 & 0 \\ \tilde{C}_{i,1} & 0 & \cdots & 0 & 0 \\ \vdots & & & \vdots & \vdots \\ 0 & \cdots & \tilde{C}_{i,\kappa_i} & 0 & 0 \\ \hline 0 & \cdots & 0 & 0 & 0 \end{array}\right)\,, \tag{4}$$

and $V_j^* = P^T\,V_j = (1, 0, \ldots, 0)^T$. It is worth noting that the optimal pair (C^*, V_j^*) preserves the reachability property of the original pair (C, V_j). This can be shown by direct computation of the reachability matrix R_j^*, but it is omitted for the sake of space.

In the general case with ν, $1 \leq \nu \leq n$ agents $A = \{i_1, \ldots, i_\nu\}$ that are reachable from input u_j, the optimal strategy for propagating input u_j consists of having each of the other agents receive the input measurement through a path originating from the nearest reachable agent in A. This naturally induces a network partition into ν disjoint subgraphs or spanning trees, each directly reached by the input through a different agent. Let us extract ν independent vectors $V_j(i_1), \ldots, V_j(i_\nu)$ from vector V_j having a 1 in position i_h. Then, the sequences $\{C^k V_j(i_h)\}$ are to be considered to compute the optimal partition. Let us denote with κ_i, for all $i \in A$ the number k of steps for the sequence $\{C^k V_j(i)\}$ to become stationary. Therefore, we have that the visibility diameter of the pair (C, V_j) is $\texttt{vis-diam}(C, V_j) = \max_i \{\kappa_i\}$. Without loss of generality, we can image that $\kappa_1 \geq \kappa_2 \geq \cdots \geq \kappa_\nu$. Therefore, for the generic case, there exist a permutation matrix P and a selection matrix S such that an optimal pair (C^*, V_j^*) can be obtained as $C^* = P^T\,(S\,C)\,P$, $V_j^* = P^T\,V_j$, where

$$C^* = \operatorname{diag}(C_1, \ldots, C_\nu)\,, V_j^* = (V_{j,1}^T, \ldots, V_{j,\nu}^T)^T\,, \tag{5}$$

and where each C_i and $V_{j,i}$ have the form of the Eq. 4. The actual optimal linear consensus algorithm is obtained choosing $F_j = P\,C^*$, and $B_j = P\,V_j^*$. The procedure is summarized in Algorithm 1. Finally, recall from [13], the following

Theorem 1 (Global Stability of Linear Consensus): A logical linear consensus system of the form $x(t+1) = C^*\,x(t) + V_j^*\,u_j(t)$, where C^* and V_j^* are obtained as in Eq. 5 from a reachable pair (C, V_j), converges to a unique network agreement given by $\mathbf{1}_n\,u_j$ in at most $\texttt{vis-diam}(C, V_j)$ rounds.

Example 3.1: Consider a network of $n = 5$ agents and the following pair (C, V_j) with $\nu = 2$:

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \quad V_j = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6}$$

An optimal pair (C^*, V_j^*) allowing a consensus to be estab-

Algorithm 1 Centralized Synthesis of the Optimal Linear Consensus System

Inputs: C, V_j
Outputs: Minimal pair (C^*, V_j^*), permutation P.

Set $A \leftarrow \{i \,|V_j(i) = 1\}$ ◁ nodes directly reachable from u_j
Set $I(i) \leftarrow 1$ for all $i \in A$ ◁ nodes reached from $i \in A$
Set $N \leftarrow \{1, \ldots, n\} \setminus I$ ◁ nodes not yet reached
Set $k \leftarrow 0$
repeat
  for all nodes $i \in A$ **do**
    Set $Adj(i) \leftarrow C^k V_j(i) \wedge \neg I(i) \wedge N$ ◁ new nodes
    Set $I(i) \leftarrow I(i) \vee Adj(i)$
    Set $N \leftarrow N \wedge \neg Adj(i)$
    Compute $\mathcal{I} \leftarrow \{h : Adj(i)(h) = 1\}$ ◁ index list
    for all new nodes $h \in \mathcal{I}$ **do**
      Set $\tilde{C}(h,:) \leftarrow C(h,:) \wedge Adj(i)^T$ ◁ every new node must communicate with one reach at $k-1$
    end for
  end for
  Set $k \leftarrow k+1$
until $\exists\, i \in A \,|\, Adj(i) \neq 0$
Compute $\kappa_i \leftarrow \text{card}(I(i))$ for all $i \in A$
Find $P \mid C^* \leftarrow P^T \, \tilde{C} \, P$ has $\kappa_1 \geq \cdots \geq \kappa_\nu$ ◁ re-order blocks ◁ re-order
Set $V_j^* \leftarrow P^T \, V_j$

lished over the complete graph G is given by

$$C^* = \left(\begin{array}{ccc|cc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array}\right), \; V_j^* = \left(\begin{array}{c} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{array}\right).$$

The corresponding optimal linear consensus algorithm is:

$$\left\{\begin{array}{l} x_1(t+1) = u(t)\,, \\ x_2(t+1) = u(t)\,, \\ x_3(t+1) = x_2(t)\,, \\ x_4(t+1) = x_2(t)\,, \\ x_5(t+1) = x_1(t)\,. \end{array}\right. \tag{7}$$

IV. Distributed Consensus Map Synthesis – The Self Routing Network Protocol (SRNP)

A complete exploitation of the logical consensus approach requires the optimal communication matrix C^* be computed in a fully distributed way. To this aim, we assume that agents are able to exchange messages through a synchronous communication scheme. The input–propagation spanning tree strategy, described in the previous section, can be reproduced by requiring that every agent that is able to see the j–th input event u_j send a *supply message* offering its connection to all its neighbors. Agents sending this supply message are those specified by non–null elements of V_j, whereas agents receiving the message are specified by non–null elements of CV_j. Upon receiving a supply message, every agent send back a *confirmation message* to the agent with the lower index. After k steps, agents sending supply messages are those specified by non–null elements of $C^{k-1}V_j$ and those sending confirmation messages are specified by non–null elements of C^kV_j. Therefore, having denoted with $C^*_{i,:}$ the i–th row vector of the optimal communication matrix, the agents that are able to set their communication row vector after k steps are specified by non–null elements of $\neg\left(C^{k-1}V_j\right) \wedge C^kV_j$. Denote with $m_{i,j}(k)$, for $i, j = 1, \ldots, n$, a Boolean variable taking the value 1 if, and only if, agent $\mathcal{A}_i$ receives a supply message from agent $\mathcal{A}_j$ at time k. It is straightforward to show that, for the matrix $M = \{m_{i,j}\}$, it holds

$$M(k) = \left(C^kV_j\right)^T \wedge C = Adj(i)^T \wedge C\,.$$

By construction, the row vector

$$w_i(k) \stackrel{\text{def}}{=} (m_{i,1}, m_{i,2} \wedge \neg m_{i,1}, \ldots, m_{i,n} \wedge \neg m_{i,n-1})\,,$$

is either 0 or it contains only an entry set to 1 representing the agent with lowest index from which $\mathcal{A}_i$ must receive the value of the j–th input. Then, a generic agent $\mathcal{A}_i$ can set its communication row vector as $C_{i,:} \leftarrow w_i$. After having set its communication row vector, i.e. as soon as $C_{i,:} \neq 0$, or equivalently $\Pi_{h=1}^n \neg(C^*_{i,h}(k)) = 0$, an agent forwards the supply message to its neighbors and avoids any further modification of its communication row vector. Then, we can prove the following

Theorem 2 (Self–Routing Protocol): A network of n agents running the distributed protocol

$$\left\{\begin{array}{l} C^*_{i,:}(k+1) = C^*_{i,:}(k) \vee \left(\Pi_{h=1}^n \neg(C^*_{i,h}(k)\right) \neg V_j(i) \wedge w_i(k)\,, \\ C^*_{i,:}(0) = 0\,, \end{array}\right.$$

consents in finite time on the centralized communication matrix C^*.

Proof: The proof follows straightforwardly from the above description. In fact, if the generic agent $\mathcal{A}_i$ is able to "see" the j–th input, i.e. $V_j(i) = 1$, the iterative rule reduces to $C^*_{i,:}(k+1) = C^*_{i,:}(k)$ and the initial null value is maintained. Otherwise, as soon as a supply message is received, the element in $w_i(k)$ corresponding to the message sender becomes 1, and the communication row vector is accordingly set. Finally, the term $\Pi_{h=1}^n \neg(C^*_{i,h}(k))$ prevents any further modification. ■

Example 4.1: Consider again Example 3.1. By applying the iterative rule of Theorem 2, we have

$$M(0) = \left(\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{array}\right), \; W(0) = 0\,,$$

$$C^*_{i,:}(0) = 0 \;\; \forall i\,,$$

Algorithm 2 Distributed Synthesis of the Linear Consensus System

Inputs: $V_j(i)$
Outputs: $C^*_{i,:}$

Set $\tilde{C}_i \leftarrow 0$
while $\Pi_h \neg C^*_{i,h} \wedge \neg V_{ji}$ **do**
  receive(buf, src) ◁ waiting for a supply message from neighbors of i
  Set $\tilde{C}_i \leftarrow$ src ◁ index of the supply message sender
  Set $C^*_{i,j} \leftarrow 1\,,\ \forall j \in \tilde{C}_i$
end while
send(buf, broadcast) ◁ sends a supply message in broadcast

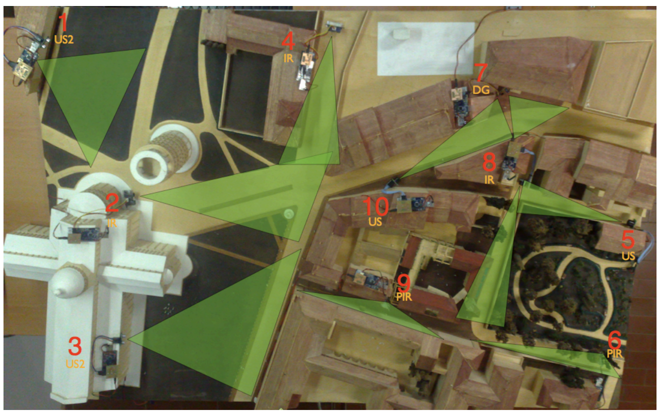


Fig. 1. Scale model of the area nearby Pisa's Leaning Tower. Green cones represent the visibility areas of every agents.

$$M(1) = C\,,\ W(1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$C^*_{i,:}(1) = W(1)\ \forall i\,,$$

and $C^*_{i,:}(k) = C^*_{i,:}(1)$ for all $k \geq 1$, which gives the same logical linear consensus system as in Eq. 7.

Algorithm 2 reports the above described procedure, that is later referred to as the Self Routing Network Protocol (SRNP). Its asymptotic *computational complexity* is in the very worst case $O(n^2)$, where n is the number of agents, and its *space complexity* in terms of memory required for its execution is $\Omega(n)$. However, its implementation can be very efficient since it is based on Boolean operations on bit strings. Finally, *communication complexity* of a run of the consensus protocol in terms of the number of rounds is $\Theta(\texttt{vis-diam}(C, V_j))$.

V. Application

A. Experimental Setup

The effectiveness of SRNP has been shown through the following experimental setup related to the scenarios presented in [13], [14]. Consider a scale model representing the urban area $\mathcal{W}$ nearby Pisa's Leaning Tower, where 10 agents A_i, each represented by a Sentilla Tmote–Sky [15], have been deployed to detect a possible intruder, represented by a radio controlled mini car. Agents are equipped with different sensors and are supposed to monitor fixed safety areas $\mathcal{W}_i, i = 1, \cdots, 10$ (Fig. 1). As in [13], the presence or absence of an intruder in region $\mathcal{W}_j$ can be modeled as a logical input u_j and every agent is required to estimate the $p = m$ decisions $y_i(t) = u_i(t), i = 1, \cdots, 10$. Due to limited sensing range, every agent is able to detect the presence of intruders only within its visibility areas, and thus a visibility matrix $V \in \mathbb{B}^{10\times 10}$ can be defined with $V_{i,j} = 1$ if, and only if, agent $\mathcal{A}_i$ can "see" in the area $\mathcal{W}_j$. The alarm state of the system is $X \in \mathbb{B}^{10\times 10}$, with $X_{i,j} = 1$ if agent $\mathcal{A}_i$ reports an alarm about the presence of an intruder in the area $\mathcal{W}_j$. The alarm can be set because an intruder is actually detected by an agent, or because of communications with neighboring monitors. The communication graph is assumed to be connected as required from theory (see Section IV).

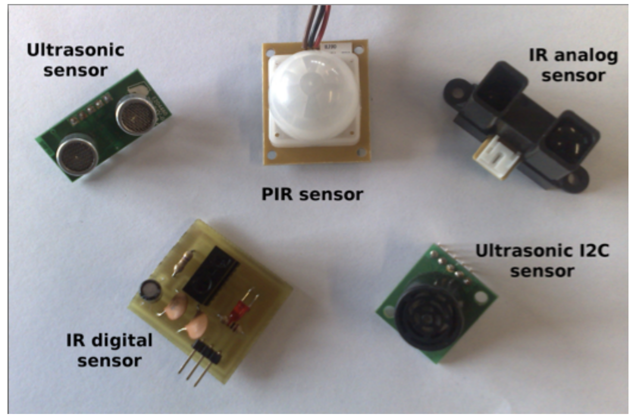
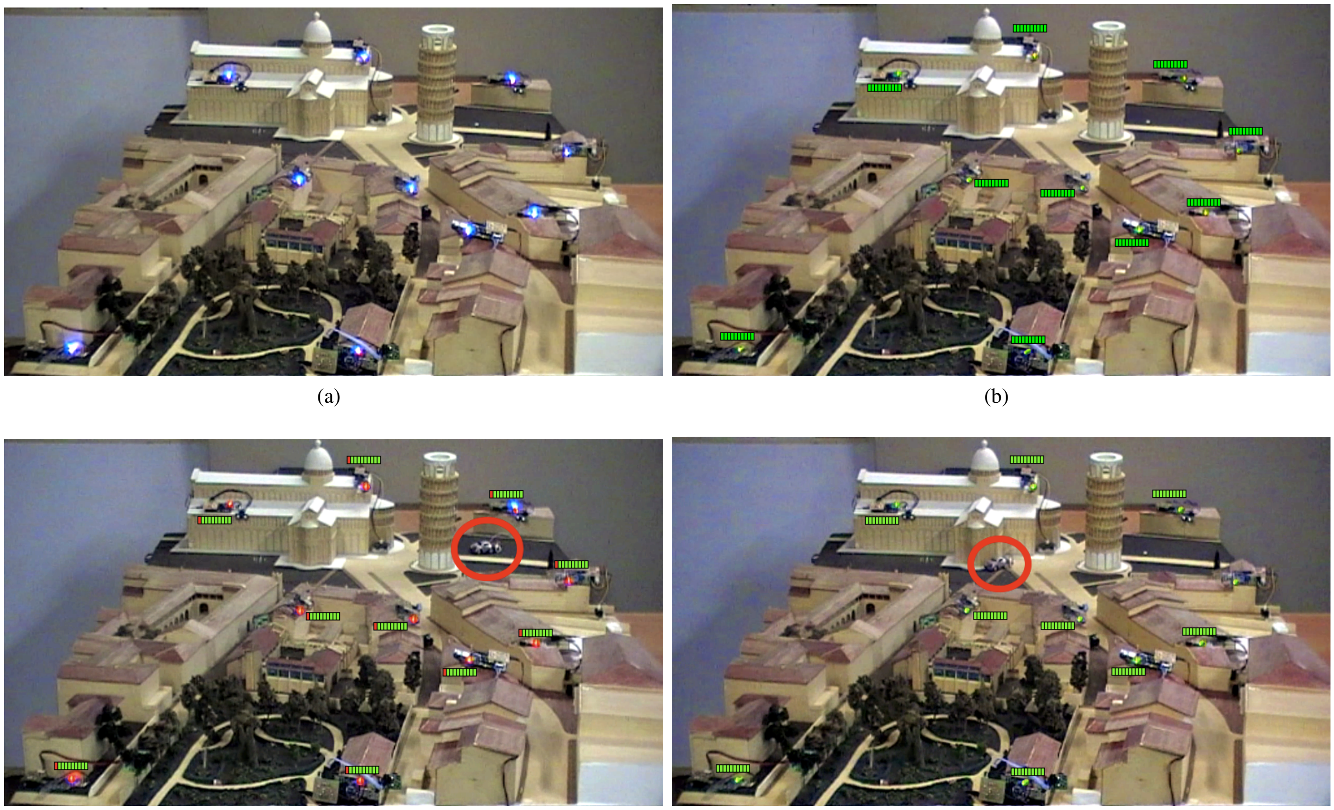


Fig. 2. The different types of sensors that have been used during the experiments.

This type of mote is an MSP430–based battery–powered board, with an 802.15.4–compatible CC2420 radio chip. Every mote runs Contiki 2.0 operating system and uses μIP communication protocol [16], [17]. The Contiki OS is optimized for embedded systems with limited hardware resources and wireless connectivity, such as these motes, and it enables multi-thread programming. A Tmote-Sky represents a natural platform for implementing a logical consensus system. The platform is equipped with three colored LEDs and two light sensors, and it is provided with connectors for installing extra sensors by I^2C bus and analog to digital converter. The set of extra sensors used in the experiments comprises ultrasonic sensors, IR range finder, and PIR–based motion detectors (Fig. 2).

s Every agent is provided with a software implementation of Algorithm 2, presented in Section IV, that it runs in order to construct its row vector $C^*_{i,:}$. As the map synthesis is generated through input propagation, every round may involve several sending messages and thus its length must be chosen so as to allow every agent's communication. On the available hardware a communication time–slot of $9\cdot 10^{-2}$ sec



(a) (b) (c) (d)

Fig. 3. Snapshots of the four main phases of the experiment: (a) execution of the SRNP, (b) intrusion detection mode, (c) detection of the intruder by agent $\mathcal{A}_1$, (d) the intruder has moved in an area that is out of every agents' range. A colored flag of 10 squares represents the alarm state X_i of a generic agent $\mathcal{A}_i$ and shows its knowledge about the presence of an intruder in every visibility areas.

is possible, and thus the length of every round is set to 1 sec. Although the current implementation of the communication scheme requires the agents' pre–synchronization (via e.g. the solutions in [18], [19]), this does not affect the general applicability of SRNP.

B. Experimental Results

A video, enclosed with the paper, shows the experiment described below. The aim of the experiment is to show two main properties of SRNP: 1) its ability to reconfigure upon entrance of a new agent, and 2) its ability to realize a distributed intrusion detection system through execution of a logical consensus system. In the video, the following conventions are adopted to represent the different operation phases of the agents: blue represents the self–routing phase, green means that agents is operating in intrusion detection mode and sees no intruder, red is turned on when the agent is informed of the existence of an intruder, red and blue LEDs simultaneously turned on mean that the agent has seen the intruder in its visibility area.

The simulation starts with all agents running the SRNP so as to establish a communication matrix C^* that enables consensus on the different input events $u_1, \cdots, u_{10}$ (Fig. 3a). At conclusion of this phase, agents are ready to detect and consent on the presence of an intruder (Fig. 3b). Afterward, a new agent is added, which triggers a reconfiguration of the network including first a re–synchronization and then a re–execution of the SRNP. An intruder, represented by the radio controlled car is introduced. As soon as the car becomes close to agent $\mathcal{A}_1$, the agent detects it (its blue and red LEDs are both turned on) and the information is dispatched through the network according to the logical consensus scheme (Fig. 3c). Therefore, every single agent can be polled to know about the presence and the location of the intruder. Whenever the intruder reaches an uncovered area, its presence is not detected (Fig. 3d), which would require the introduction of a new agent in the system.

VI. Conclusion

The problem of generating a distributed logical consensus system was considered in this paper. Logical consensus is an approach to solve decision problems through a network of agents that can elaborate and exchange binary values. Previous work, concerning the centralized design of suitable logical iterative maps, has been extended in the paper so as to also allow the computation of such maps in a fully distributed way. This has led to the definition of the Self Routing Network Protocol (SRNP). An experimental setup was presented, through which exploitation of the technique has been shown in a surveillance task. Future work will concern the study and realization of a self-routing protocol that guarantees a correct execution of the logical consensus also in the presence of faulty agents, i.e. agents spreading false information.

ACKNOWLEDGMENT

This work has been partially supported by the European Commission with contract FP7-IST-2008-224428 “CHAT - Control of Heterogeneous Automation Systems: Technologies for scalability, reconfigurability and security”, and with contract number FP7-2007-2-224053 CONET, the “Cooperating Objects Network of Excellence”.

REFERENCES

[1] C. Fall, E. Marland, J. Wagner, and J. Tyson, *Computational Cell Biology*. Springer, 2005.

[2] T. Balch and R. Arkin, “Behavior-based formation control for multirobot systems,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[3] B. Bamieh, F. Paganini, and M. Dahleh, “Distributed control of spatially-invariant systems,” *IEEE Transactions on Robotics and Automation*, vol. 47, no. 7, pp. 1091–1107, 2002.

[4] M. Mesbahi and F. Y. Hadaegh, “Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 369–377, 2001.

[5] R. Olfati-Saber, J. Fax, and R. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, p. 215, 2007.

[6] A. Jadbabaie, J. Lin, and A. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[7] J. Fax and R. Murray, “Information flow and cooperative control of vehicle formations,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, Sept. 2004.

[8] J. Cortés, S. Martínez, T. Karataş, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[9] L. Greco, M. Gaeta, and B. Piccoli, “Deployment of sensors in a network-like environment,” in *47th IEEE Conference on Decision and Control, 2008. CDC 2008*, 2008, pp. 4257–4262.

[10] A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi, “Consensus based Distributed Intrusion Detection for Multi Robot Systems,” in *Proc. IEEE International Conf. on Robotics and Automation*, 2008, pp. 120–127.

[11] A. Fagiolini, S. Martini, N. Dubbini, and A. Bicchi, “Distributed consensus on boolean information,” in *1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys’09)*, Venice, Italy, September, 24 - 26 2009, pp. 72 – 77.

[12] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers, 1997.

[13] A. Fagiolini, E. Visibelli, and A. Bicchi, “Logical Consensus for Distributed Network Agreement,” in *47th IEEE Conference on Decision and Control, 2008. CDC 2008*, 2008, pp. 5250–5255.

[14] J. O’Rourke, *Art Gallery Theorems and Algorithms*. Oxford, UK: Oxford University Press, 1987.

[15] MOTEIV, “Tmotesky datasheet: Ultra low power ieee 802.15.4 compliant wireless sensor module,” 2006.

[16] A. Dunkels, B. Grönvall, and T. Voigt, “Contiki a lightweight and flexible operating system for tiny networked sensor,” in *Proceeding of IEEE Workshop on Embedded Networked Sensor*. IEEE, 2004.

[17] A. Dunkels, T. Voigt, J. Alonso, H. Ritter, and J. Schiller, “Connecting Wireless Sensornets with TCP/IP Networks,” in *Proceedings of the Second International Conference on Wired/Wireless Internet Communications (WWIC2004)*, Frankfurt (Oder), Germany, FEB 2004.

[18] L. Schenato and G. Gamba, “A distributed consensus protocol for clock synchronization inwireless sensor network,” *46th IEEE Conference on Decision and Control*, Dec 2007.

[19] A. Fagiolini, S. Martini, and A. Bicchi, “Set–valued consensus for distributed clock synchronization,” *Proc. 5rd Annual IEEE Conf. on Automation Science and Engineering*, 2009.