Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Robust Network Agreement on Logical Information

Simone Martini * **Davide Di Baccio** * **Adriano Fagiolini** *
Antonio Bicchi *

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

Abstract: Logical consensus is an approach to distributed decision making which is based on the availability of a network of agents with incomplete system knowledge. The method requires the construction of a Boolean map which defines a dynamic system allowing the entire network to consent on a unique, global decision. Previous work by the authors proved the method to be viable for applications such as intrusion detection within a structured environment, when the agent's communication topology is known in advance. The current work aims at providing a fully distributed protocol, requiring no a priori knowledge of each agent's communication neighbors. The protocol allows the construction of a robust Boolean map that is able to tolerate incorrect information spread by some agents, due to spontaneous failure or malicious intents. Effectiveness of the proposed method is shown through an implementation on a low–cost platform.

Keywords: Intrusion detection, security, robust logical consensus, networked and distributed systems.

1. INTRODUCTION

The increasing need of integration of robotic agents in human environments has given a boost to the research for new control techniques involving forms of cooperation among these agents. Applications in which heterogeneous agents differing in size, perception capacity, computation, and actuation will coexist and collaborate to achieve a common goal are becoming a reality. The functions and structures of these multi–agent systems may differ for their organization, e.g. agents can work together as teams, flocks or swarms, so as to more effectively and robustly pursue a goal which is common to all members Parker (2008). The paradigm of "intentional cooperation" is evoked when complex agents combine their specific capabilities to achieve a shared goal Parker (2008), whereas the paradigm of "emergent behaviors" is used for large–scale systems composed of simple agents with limited individual capabilities which still are able to accomplish complex tasks (see e.g. Tanner et al. (2007); Olfati-Saber (2006); Oh et al. (2007); Pavone et al. (2009)). Forms of cooperation have been proposed to solve applications such as target tracking with agents having limited communication capacity Arrichiello et al. (2009), or surveillance with mobile rovers having limited visibility Ganguli et al. (2008).

Most of these applications require that agents *consent* on a task–related decision depending on a set of input events. Available solutions to achieve this involve suitable distributed algorithms, where agents are able to reach an agreement by exchanging data via communication and merging their own information with the one received from neighbors (see e.g. Olfati-Saber et al. (2007); Jadbabaie et al. (2003); Fax and Murray (2004)). However the related literature has mainly focused on linear forms of consensus systems,

$$\dot{x}(t) = A\,x(t) + B\,u(t)\,,$$

where $A \in \mathbb{R}^{n\times n}$ is a strongly connected doubly–stochastic matrix, $B \in \mathbb{R}^{n\times m}$ is the input matrix, and $u \in \mathbb{R}^m$ is an input. Only very recently, alternative forms of consensus have been considered: Frasca et al. (2008) addressed the problem of reaching consensus in networks with limited communication capacities, by using a logarithmic quantizer which is used by the agents to convert their data into symbols that can be sent through the network; Cortés et al. (2004) proposed another nonlinear consensus algorithm, based on the centroidal Voronoi tessellation, to deploy a collection of mobile agents so as to maximize the ability of the network to accomplish a sensing task; Fagiolini et al. introduced the *set–valued consensus* for distributed misbehavior detection and clock synchronization (Fagiolini et al. (2008a), Fagiolini et al. (2009a)), where exchanged data is represented by *sets* or *intervals*.

In this vein, so–called *logical consensus* was proposed by Fagiolini et al. (2008b) to enable a collection of agents, that share binary values representing local estimates of input events, to reach a unique and consistent decision. The approach is based on the synthesis of an iterative linear logical map converging to consensus under suitable conditions that involve the inputs' visibility and the communication graph's connectivity. The limitations of the approach are that the map synthesis is centralized, i.e. it requires a priori knowledge of each agent's neighbors, and that the obtained map only converges to the correct value if every agent process and send correct information.

The current work extends further the approach by describing a fully distributed synthesis procedure that generates a logical nonlinear consensus map that is able to tolerate misbehaving agents that may send incorrect information, due to spontaneous failure or even tampering. The procedure is formalized as a distributed protocol, called $\mathrm{SR}^2\mathrm{NP}$, which is guaranteed to converge under similar visibility and communication conditions. The protocol is based on the known result of Lamport et al. (1982) ensuring that redundant minimum–length paths from a generic input u_j to every agent of the network can be found if the number of faults is bounded by $\gamma \in \mathbb{N}$ and the communication graph is at least $(2\,\gamma + 1)$–connected. The final contribution of the paper is the implementation of the proposed solution in a real networked multi–agent system, that uses low–cost communication and computation devices, which allows us to show the effectiveness of the new design procedure and of the obtained robust maps.

2. PROBLEM FORMULATION

We consider problems involving computation of a set of p decisions, $y_1, ..., y_p$, that depend on m *logical events*, $u_1, ..., u_m$. 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. Having denoted with $u = (u_1, \dots, u_m)^T \in \mathbb{B}^m$ and $y = (y_1, \dots, y_p)^T \in \mathbb{B}^p$ the input event vector and output decision vector, respectively, Eq. 1 can be written more compactly as $y = f(u)$. 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 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, 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. 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.

In this view, we can imagine that each agent $\mathcal{A}_i$ has a local *state* vector, $X_i = (X_{i,1}, \dots, X_{i,q}) \in \mathbb{B}^{1\times q}$, where q is a suitable dimension. 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 $Y_i = (y_{i,1}, \dots, y_{i,p}) \in \mathbb{B}^{1\times p}$ 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 $Y(t) = (Y_1^T(t), \dots, Y_n^T(t))^T \in \mathbb{B}^{n\times p}$ 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*

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

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 $v_i \stackrel{\text{def}}{=} \{v_{i,1}, v_{i,2}, \dots\} \subseteq \{1, \dots, m\}$ of the events that it can "see" and the index list $c_i \stackrel{\text{def}}{=} \{c_{i,1}, c_{i,2} \dots\} \subseteq \{1, \dots, n\}$ of its neighbors. In this case, the above mentioned centralized visibility and communication matrices, $V = \{V_{i,j}\}$ and $C = \{C_{i,j}\}$, can be reconstructed according to the rules

$$V_{i,j} = \begin{cases} 0 & \text{if } v_{i,j} \notin v_i \\ 1 & \text{otherwise}\,, \end{cases} \quad C_{i,j} = \begin{cases} 0 & \text{if } c_{i,j} \notin c_i \\ 1 & \text{otherwise}\,. \end{cases}$$

Furthermore, another important property, is the system's ability to cope with possible agents' failures during execution, and to operate properly despite of incorrectness in input given by possible faulty or misbehaving agents. In this perspective we want to solve the following:

Problem 1. [Robust Design] Given a decision system as in Eq. (1), a visibility matrix V, a communication matrix C, and a maximum number γ of faulty agents, design a robust logical consensus system allowing all correct agents to consent on the centralized decision $\tilde{y} = f(u)$, i.e.

$$Y_i(t) = (\tilde{y})^T\,,\ \forall i \notin \Gamma, t \geq \bar{N}\,,$$

where Γ is the set of faulty agents, and $\bar{N}$ is a sufficiently step number.

3. ROBUST LOGICAL INFORMATION FLOW

Failure of an agent can lead the linear logical consensus system to an incorrect and disconnected decision. In this section, we firstly discuss how to generate robust logical consensus maps solving Problem 1 in a centralized way. Consider that temporary faults, occurring when e.g. the measures of some agents are corrupted by noise, can be modeled as variations in the initial state $x(0)$. These faults pose no problem even in the case of linear logical consensus maps, which were shown to possess a unique and globally stable equilibrium $\mathbf{1}_n u_j$, where $\mathbf{1}_n$ is a vector with every elements to 1 (Fagiolini et al. (2008b)). This implies that temporary false alarms are canceled out by the system itself.

The problem becomes more difficult and interesting in case of permanent faults that occur whenever one or more agents are damaged, due to e.g. a spontaneous failure, or even tampering, and do not correctly execute the consensus algorithm. In this case, the convergence of the

algorithm to the correct value may not be reached and the system may not be able to consent on the global decision. This motivates the development of techniques for synthesizing robust logical consensus systems. Suppose that a maximum number of $\gamma \in \mathbb{N}$ faulty agents have to be tolerated. The key to solve such a problem is in redundancy of input measurement and communication. Intuitively, a minimum number r of such sensors must be able to measure the $j-th$ input u_j and/or confirm any transmitted data x on u_j. In particular, to tolerate up to γ faults it is sufficient to chose $r = 2\gamma + 1$ (Lamport et al. (1982)). Therefore, we are concerned with the following:

Definition 1. (Reachability with redundancy).
An agent $\mathcal{A}_i$ is said to be reachable from input u_j with redundancy $r \in \mathbb{N}$, or shortly r–reachable, if, and only if, it can receive at least r measurements of u_j between a direct measurement of u_j and messages of other agents that are r–reachable from u_j.

Moreover, *redundant minimum–length* paths are to be found such that information on u_j can robustly flow through the network, but such paths cannot be found by considering successive powers of $C^k V_j$, because only r–reachable agent are permitted to propagate information over the network. Thus, we need a procedure for finding to which agents the value of input u_j can be *robustly* propagated. Given a pair (C, V_j), we can conveniently introduce the *Robust Reachability matrix* ($R_j^2 \in \mathbb{B}^{n\times l}$), assigned with input u_j, whose k–th column represents the nodes which are r–reachable from input u_j at the k–th step. Algorithm 1 shows an iterative procedure allowing the evaluation of the Robust Reachability matrix, which is explained below.

Denote with S_k the set of agents that are able to "securely" estimate the value of input u_j. First observe that we have $S_0 = \{i_1, i_2, \cdots i_c\}$, whose elements are the indices of the non–null elements of the vector V_j. Moreover, we have $S_1 = S_0 \cup \{i_{c+1}, \cdots, i_{c+p}\}$, where the new indices corresponds to agents that can receive a message containing the value of u_j from at least r agents in S_0. These new indices are the non–null elements of the vector

$$\tilde{W}_j^1 = T_r(C * V_j)\,,$$

where $*$ is the integer product between two matrices and T_r is the threshold map

$$T_r : \mathbb{N}^n \to \mathbb{B}^n$$
$$w_i \mapsto \begin{cases} 1 \text{ if } w_i \geq r \\ 0 \text{ otherwise} \end{cases} .$$

The set of agents that are r– reachable after 1 step are then given by the non–null elements of the vector $W_j^1 = \tilde{W}_j^1 + V_j$. In general, S_k is obtained as the set of agents that are specified by non–null elements of the vector

$$W_j^k = T_r(C * W_j^{k-1}) + W_j^{k-1}\,,$$

with $W_j^0 = V_j$. Note that W_j^k is by definition the $(k+1)$–th column of R_j^2. Computation of R_j^2 can stop as soon as the sequence becomes stationary, i.e., $W_j^k = W_j^{k-1}$.

Therefore we can prove the following:

Theorem 1. A pair (C, V_j) is r–reachable if, and only if the logical product of the elements of the last column of the matrix R_j^2 is equal to 1, i.e. $\wedge_{p=1}^n (r_{p,l}) = 1$, where $[r_{1,l}, r_{2,l}, \cdots, r_{n,l}]^T$ is the last column of R_j^2.

Algorithm 1: Algorithm for Robust Reachability matrix.

Input: C,V_j,γ
Output: R_j^2

$r = 2\gamma + 1$;
$k = 1$;
$W_j^k = V_j$;
do
  $W_j^{k+1} = T_r(C * W_j^k) \vee W_j^k$; ◁ `r--reachable nodes after k--steps`
  $k = k + 1$;
while $W_j^k \neq W_j^{k-1}$◁ `no cycle condition`;
$R_j^2 = [W_j^1, \cdots, W_j^{k-1}]$;

Proof. The proof of sufficiency straightforwardly follows from the procedure for the construction of the R_j^2 matrix. Indeed, since the k–th column of that matrix represents nodes that are r–reachable at the k–th step, the non–null element of the last column indicates which nodes are eventually r–reachable. For this reason if all the elements of the last column of R_j^2 are non-null all nodes are r-reachable, i.e. the pair (C, V_j) is r–reachable.

To show that the condition is also necessary, let us proceed by absurd. Suppose that a node is r–reachable, while the corresponding element of the last column of R_j^2 equals 0. This means that such a node cannot be reached by at least r messages containing the value of u_j, which contradicts the hypothesis of r–reachability of the network.

Example 1. Consider the network of $n = 5$ agents depicted in Fig. 1a with

$$C = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix} , \text{ and } V_j = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} .$$

By applying Algorithm 1, we can verify whether the pair (C,V_j) is 3–reachable or not, i.e. if it is feasible to construct a map that is robust to $\gamma = 1$ faulty agents. First we have $W_j^0 = V_j = (1, 1, 0, 1, 0)^T$. At the first step ($k = 1$), we have $W_j^1 = \tilde{W}_j^1 + W_j^0$, i.e.

$$W_j^1 = T_r \begin{pmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} .$$

Analogously at the second step ($k = 2$) we have $W_j^2 = (1, 1, 1, 1, 1)^T$. As $W_j^2 \neq W_j^1$, the procedure proceeds to the next step. At $k = 3$ we obtain $W_j^3 = W_j^2$ and thus the procedure can stop. The resulting Robust Reachability Matrix is

$$R_j^2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} .$$

Remark 1. Once the robust reachability matrix R_j^2 of a pair (C, V_j) has been computed by using Algorithm 1, a logical consensus map tolerating up to γ faults can be built by choosing $r = 2\gamma + 1$ (Lamport et al. (1982)). The explicit construction of the map will be discussed below for the distributed design setting.

4. DISTRIBUTED SYNTHESIS OF A ROBUST CONSENSUS MAP: THE SELF–ROUTING ROBUST NETWORK PROTOCOL (SR^{2}NP)

A practical exploitation of the logical consensus approach requires that an optimal r–reachable communication pair (C^*,V_j) is computed in a fully distributed way. To this aim, we assume that agents are able to exchange messages through a synchronous communication scheme. The correct multi–input–propagation spanning tree strategy, can be reproduced by requiring that every agent that is able to "robustly see" the j–th input event u_j sends a *supply message* offering its connection to all its neighbors. A distributed procedure, that can be used by the network agents to correctly propagate the information through the network, i.e. to find the C^* matrix, is described in Algorithm 2. Starting from C_i^* vectors, each agent is able to build a suitable distributed nonlinear iteration map F_i to combine its information with the ones of its neighbors. An agent $\mathcal{A}_i$ that is able to measure u_j can update its state x_i via the local rule $F_i = u_j$. Any other agent needs to rely on information received from its C–neighbors, including possible compromised data. Denote with $\rho = \{k : C_i^*(k) = 1\}$ the set of C^*–neighbors of agent $\mathcal{A}_i$. Denote also with $S_i \in \mathbb{N}^{\sigma\times(\gamma+1)}$ an integer matrix containing the $\sigma = \binom{r}{\gamma+1}$ combinations of $\gamma + 1$ elements extracted from a subset with cardinality r of the set of $\mathcal{A}_i$'s C^*–neighbors, i.e. $S_i = \mathcal{C}_{\gamma+1}^r(\rho)$. A robust update can be obtained by computing the (logical) sum of the $\sigma = \binom{r}{\gamma+1}$ terms, with $r = 2\gamma + 1$, that are composed of the logical product of $\gamma + 1$ states x_s extracted from its C–neighbors, i.e.

$$x_i(t+1) = F_i(x(t), u_j)$$

with

$$F_i : \mathbb{B}^n \times \mathbb{B} \to \mathbb{B}$$
$$(x, u_j) \mapsto \begin{cases} u_j & \text{if } V_{i,j} = 1, \\ \sum_{h=1}^{\sigma} \prod_{k=1}^{\gamma+1} x_{s_{h,k}} & \text{otherwise} \end{cases} \quad (3)$$

where $s_{h,k}$ is the k–th element of the h–th row of the S_i.

In the remainder of the paper, we need the following:

Lemma 1. The local update rule $x_i(t+1) = F_i(x(t))$ where F_i is constructed as in Eq. (3) converges to the consensus value $\bar{x}_i(t) = \alpha,\ t \geq \bar{N}$, if the at least $\gamma + 1$ components of the state vector $x(t) \in \mathbb{B}^n$ equals $x_k = \alpha$.

Proof. Let us proceed by absurd. Suppose that $x(t)$ is composed of γ values equal to $\neg\alpha$, whereas the remaining $\gamma + 1$ values are equal to α, and suppose that $\bar{x}_i(t) = \neg\alpha,\ t \geq \bar{N}$.

Let us consider two different cases: $\alpha = 1$ and $\alpha = 0$. If $\alpha = 1$, we have $\bar{x}_i(t) = 0$. Indeed, as F_i is the logical sum of σ terms consisting of the logical product of $\gamma + 1$ states $x(t)$, in order to have $\bar{x}_i(t) = 0$, all the σ logical terms must be equal to 0. It trivially holds that this is not possible because one of the σ terms is the logical product of $\gamma + 1$ values $x_k = \alpha$, and so in this case the lemma follows.

If $\alpha = 0$ we have $\bar{x}_i(t) = 1$. In order to have this, at least one of the σ terms of the logical sum must be equal to 1. Since all the σ terms are composed of $\gamma+1$ values, it turns out that at least one of these values in the logical product is equal to 0. Therefore, it trivially holds that all the σ terms are equal to zero. This contradicts the hypothesis and proves the thesis. ■

Algorithm 2: Distributed Algorithm for SR^{2}NP

Input: $V_{i,j}$,γ
Output: C_i^*, F_i

$r \leftarrow 2\gamma + 1$;
$\sigma \leftarrow \binom{r}{\gamma+1}$;
$C_i^* = 0^{1\times n}$;
while $(\texttt{SUM}(C_i^*) < r) \wedge (\neg V_{i,j})$ **do** ◁ `verify if the update of` C_i^* `is necessary`
  `receive(`id`);`
  $C_i^*(id) = 1$; ◁ `update Connecting Neighbors Set`
end
$\rho \leftarrow \{k : C_i^*(k) = 1\}$;
$S_i \leftarrow \mathcal{C}_{\gamma+1}^r(\rho)$;
$F_i \leftarrow \sum_{h=1}^{\sigma} \prod_{k=1}^{\gamma+1} x_{s_{h,k}} + V_{i,j}\, u_j$;
`send(`i`);`

Note that that, by using Algorithm 2, a nonlinear logical consensus system as in Eq. (2) can be obtained, which is able to tolerate up to γ permanent faults. In this perspective the following theorem is a solution to Problem 1:

Theorem 2. Given an r–reachable pair (C, V_j), the protocol described in Algorithm 2 produces a robust logical nonlinear consensus system of the form $x(t + 1) = F(x(t), u_j(t))$, which has a unique equilibrium. If the number of permanent faults is upper bounded by γ, the system's equilibrium is given by $\mathbf{1}_n u_j$.

Proof. Firstly, we need to prove that the robust logical nonlinear consensus system constructed according to Algorithm 2, has a unique equilibrium point. Recalling the procedure described in Algorithm 2, we have that agents directly reachable from u_j are specified by non–null elements of vector V_j. Let us denote with $S_0 = \{i_1, i_2, \cdots i_c\}$ the index set of these agents. Then, the strategy to allow the information on u_j flowing through the network is obtained if agents in S_0 communicates their measurement to all their neighbors, which in turn, if r–reachable, will communicate it to all their neighbors, and so on. In this way, we have that every agent $\mathcal{A}_i$ receives u_j from a path originating from agents in S_0. Indeed, it trivially holds that agents reached after k steps have received the input value from agents that were "secure" reached in the previous $k - 1$ steps. By reordering the agents according to the order that they are reached by the input propagation, the pair (C^*, V_j^*) can be rewritten as $(\tilde{C}^*, \tilde{V}_j^*)$ where $\tilde{C}^* = P^T (S\, C)\, P$ is a lower triangular block matrix, and $\tilde{V}_j^* = P^T V_j$, where S is a selection matrix and P is a permutation matrix. More precisely, we have:

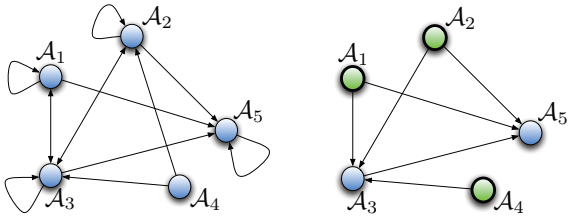


Fig. 1. Example of network with $n = 5$ agents (1a), and its robust communication graph (1b), where green nodes are able to directly measure input u_j.

$$\tilde{C}^* = \left(\begin{array}{cccc|c} 0 & 0 & \cdots & \cdots & 0 \\ \tilde{C}_{1,1} & 0 & \cdots & \ldots & 0 \\ \tilde{C}_{2,1} & \tilde{C}_{2,2} & \ddots & \vdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \tilde{C}_{l,1} & \cdots & \cdots & \tilde{C}_{l,l} & 0 \end{array}\right), \; \tilde{V}_j^* = \begin{pmatrix} 1_\nu \\ 0 \\ 0 \\ \vdots \\ \\ \vdots \\ 0 \end{pmatrix} \tag{4}$$

where $\nu \geq r$ are the agents which directly measure the input u_j. Since the robust nonlinear consensus map is constructed as in Eq. (3) by using the pair (C^*, V_j^*), the incidence matrix of the iteration map, has a strictly lower–triangular form. Thus, from Fagiolini et al. (2009b), follows that the network reaches an agreement on a unique equilibrium in a finite number of steps.

Finally, we need to prove that if the number of permanent faults is upper bounded by γ, then the equilibrium point of the system is $\mathbf{1}_n\, u_j$. Observe that, referring to the blocks in Eq. (4), according to the first elements of V_j^*, the first row block gives the simple relation $x_{i,0}(t) = u(t)$. Then, the second block corresponds to a set of agents that are updated after 1 step. In particular, as they are receiving the value from the agents in the first block, by Lemma 1 we have: $x_{i,1}(t) = x_{i,0}(t) = u_j$. At the generic iteration k, a block of variables $x_{i,k}$ are updated through the k–th matrix $\tilde{C}_{i,k}$. Hence, as they are receiving the value from the agents in the previous blocks, by Lemma 1, we have $x_{i,k}(t) = x_{i,k-1}(t) = u_j$. By repeating this procedure for all blocks, and since all agents that are directly reachable from input u_j read the same value u_j, by Lemma 1 we can prove that the entire network reaches an agreement on the unique global equilibrium $x^* = \mathbf{1}_n\, u_j$ in a finite number of steps. ■

Example 2. Consider the same scenario as in Example 1 and Fig. 1. The robust distributed nonlinear iteration map is obtained by using the procedure described in Algorithm 2. In particular, at step $k = 1$, agents $\mathcal{A}_1, \mathcal{A}_2$, and $\mathcal{A}_4$ set

$$C_1^* = C_2^* = C_4^* = (\,0,\,0,\,0,\,0,\,0\,),$$

and send a supply message to their neighbors. Since at step $k = 2$, the agent $\mathcal{A}_3$ receives three messages from $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_4$, it sets

$$C_3^* = (\,1,\,1,\,0,\,1,\,0\,),$$

and is able to send a supply message to its neighbors. At the same step, $\mathcal{A}_5$ sets

$$C_5^* = (\,1,\,1,\,0,\,0,0\,),$$

and, in order to be able to exit out of the while loop condition, i.e. to be r–reachable, waits to receive an other message. Finally, at step $k = 3$, since $\mathcal{A}_5$ receives a message from $\mathcal{A}_3$, it is able to set

$$C_5^* = (\,1,\,1,\,1,\,0,0\,),$$

and thus leaves the loop. Afterward, each agent $\mathcal{A}_i$ can evaluate its own nonlinear logical map, F_i as in Eq. (3). Therefore we have

$$\begin{cases} x_1(t+1) = u_j(t)\,, \\ x_2(t+1) = u_j(t)\,, \\ x_3(t+1) = x_1(t)\,x_2(t) + x_1(t)\,x_4(t) + x_2(t)\,x_4(t)\,, \\ x_4(t+1) = u_j(t)\,, \\ x_5(t+1) = x_1(t)\,x_2(t) + x_1(t)\,x_3(t) + x_2(t)\,x_3(t)\,. \end{cases}$$

whose communication graph is showed in Fig. 1b.

5. APPLICATION

This section shows the use of the proposed protocol via realization of an intrusion detection system. The reader may refer to the site
`http://www.centropiaggio.unipi.it/~martini/ifac2011/` for the complete simulation run.

5.1 Experimental Setup

The effectiveness of SR^{2}NP is shown through an experimental setup involving a scale model representing the urban area W nearby Pisa's Leaning Tower (see the scenarios presented in Fagiolini et al. (2008b, 2010)). In the model, 14 agents $\mathcal{A}_1, \cdots, \mathcal{A}_{14}$ have been deployed to detect a possible intruder, represented by one or more radio controlled mini car and some agents may fail.

Agents are represented by Sentilla Tmote–Sky nodes (MOTEIV (2006)), which includes a *low-power MSP430* micro-controller and a *ZigBee* radio chip. Every mote run a *Contiki* OS version optimized for embedded systems with limited hardware resource and wireless connectivity, and use the μIP communication protocol developed by Dunkels et al. (2004a,b). Agents are equipped with 3 color LEDs and 2 light sensors, I^{2}C–bus connectors and an A/D converter both allowing additional sensors to be installed. Additional sensors include proximity sensors such as ultrasonic sensors, IR range finder, and PIR–based motion detectors, which are used in the experiments to monitor shaped and limited safety areas $\mathcal{W}_j$, $j = 1, ..., 6$ (Fig. 2a).

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 $V \in \mathbb{B}^{14\times 6}$ can be defined with $V_{i,j} = 1$ if, and only if, agent $\mathcal{A}_i$ is able to monitor the area $\mathcal{W}_j$. The presence or absence of an intruder in region the $\mathcal{W}_j$ is modelled as a logical input u_j and each agent is responsible for running its corresponding row, f_i, of the logical map obtained through the execution of the SR^{2}NP. So the alarm will be set if, and only if an intruder is detected at least by $\gamma + 1$ agents, between those able to directly monitor the area $\mathcal{W}_j$, and those are r–reachable. The protocol presented in the paper is general and is applicable with any collection of heterogeneous agents.

In the current implementation, communication follows a *round-robin* scheme, which requires pre–synchronization of the agents' clocks (via e.g. the solutions in Schenato and Gamba (2007); Fagiolini et al. (2009a)) and allows each agent to send a message to its neighbors during a

pre–allocated time–slot (its duration is $9 \cdot 10^{-2}$ sec on the available hardware). The choice of the round-robin solution is motivated by the fact that Tmote-Sky nodes are unable to avoid effectively multiple radio collision in indoor environments which is due to a known limitation of *CSMA/CA*'s protocol implementation.

5.2 Experimental Results

Snapshots of a typical experiment are reported in Fig. 2 which reveal the ability of SR^{2}NP to: 1) tolerate γ faults due to proximity malfunction sensors, 2) reconfigure upon entrance of a new robot, and 3) realize a distributed intrusion detection system through execution of a logical consensus system.

The following conventions are adopted to represent the different operation phases of the agents: blue LEDs represent that the self–routing phase is in progress; green LEDs indicate that agents are running the monitoring phase and no intruders have been detected yet, while red LEDs turn on whenever an agent is informed of the existence of an intruder. Red and blue LEDs on the same agent indicate that an intruder has been detected in its visibility area.

The experiment starts with all agents running the SR^{2}NP so as to establish a communication matrix C^* that is used to find a suitable distributed nonlinear map that enables consensus on the shared input events $u_1, \cdots, u_6$ (Fig. 2b). After completion of the self–routing phase, agents run the monitoring phase so that they are ready to detect and consent on the presence of an intruder (Fig. 2c). During the monitoring phase, whenever new agents join in the sensor network, a new self–routing phase starts along with a new clock synchronization. In the experiment, as soon as the intruder enters the area $\mathcal{W}_1$, agents $\mathcal{A}_1$ and $\mathcal{A}_{12}$ correctly detect its presence (see the red and blue LEDs on) and inform their neighbors, while a faulty agent $\mathcal{A}_{14}$, does not detect it and sends the incorrect information to the others (Fig. 2d). Despite of this fault, the other agents correctly detect the intruder based on their own robust logical map. When the intruder moves out of region $\mathcal{W}_1$ and enters into region $\mathcal{W}_3$, a new consensus is achieved and all agents become aware of the new intruder's position (Fig. 2e).

6. CONCLUSION

The problem of reaching consensus on binary values in a network of agents by means of distributed Boolean iteration maps was considered. We extended previous work on the topic by defining a map synthesis protocol, so–called Self–Routing Robust Network Protocol (SR^{2}NP), requiring no a priori knowledge of communication neighbors. The procedure is able to generate maps that can tolerate failure of some of the network agents. The applicability of the method to a surveillance task was shown by an experimental setup.

ACKNOWLEDGEMENTS

Authors wish to thank the Department of Civil Engineering of the University of Pisa for kindly providing the scale model used in the experiments. 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", with contract number FP7-2010-257649 PLANET, "PLAtform for the deployment and operation of heterogeneous NETworked cooperating objects", and with contract number FP7-2010-257462 HYCON2, "Highly-complex and networked control systems".

REFERENCES

Arrichiello, F., Das, J., Heidarsson, H., Pereira, A., Chiaverini, S., and Sukhatme, G. (2009). Multi-Robot Collaboration with Range-Limited Communication: Experiments with Two Underactuated ASVs. In *The 7th International Conference on Field and Service Robots, Cambridge, Massachusetts*.

Cortés, J., Martínez, S., Karataş, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2), 243–255.

Dunkels, A., Grönvall, B., and Voigt, T. (2004a). Contiki a lightweight and flexible operating system for tiny networked sensor. In *Proceeding of IEEE Workshop on Embedded Networked Sensor*. IEEE.

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

Fagiolini, A., Martini, S., and Bicchi, A. (2009a). Set–valued consensus for distributed clock synchronization. *Proc. 5rd Annual IEEE Conf. on Automation Science and Engineering*.

Fagiolini, A., Martini, S., Di Baccio, D., and Bicchi, A. (2010). A self-routing protocol for distributed consensus on logical information. In *Intelligent Robots and Systems, 2010. IROS 2010. IEEE/RSJ International Conference on*.

Fagiolini, A., Martini, S., Dubbini, N., and Bicchi, A. (2009b). Distributed consensus on boolean information. In *1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys'09)*, 72 – 77. Venice, Italy.

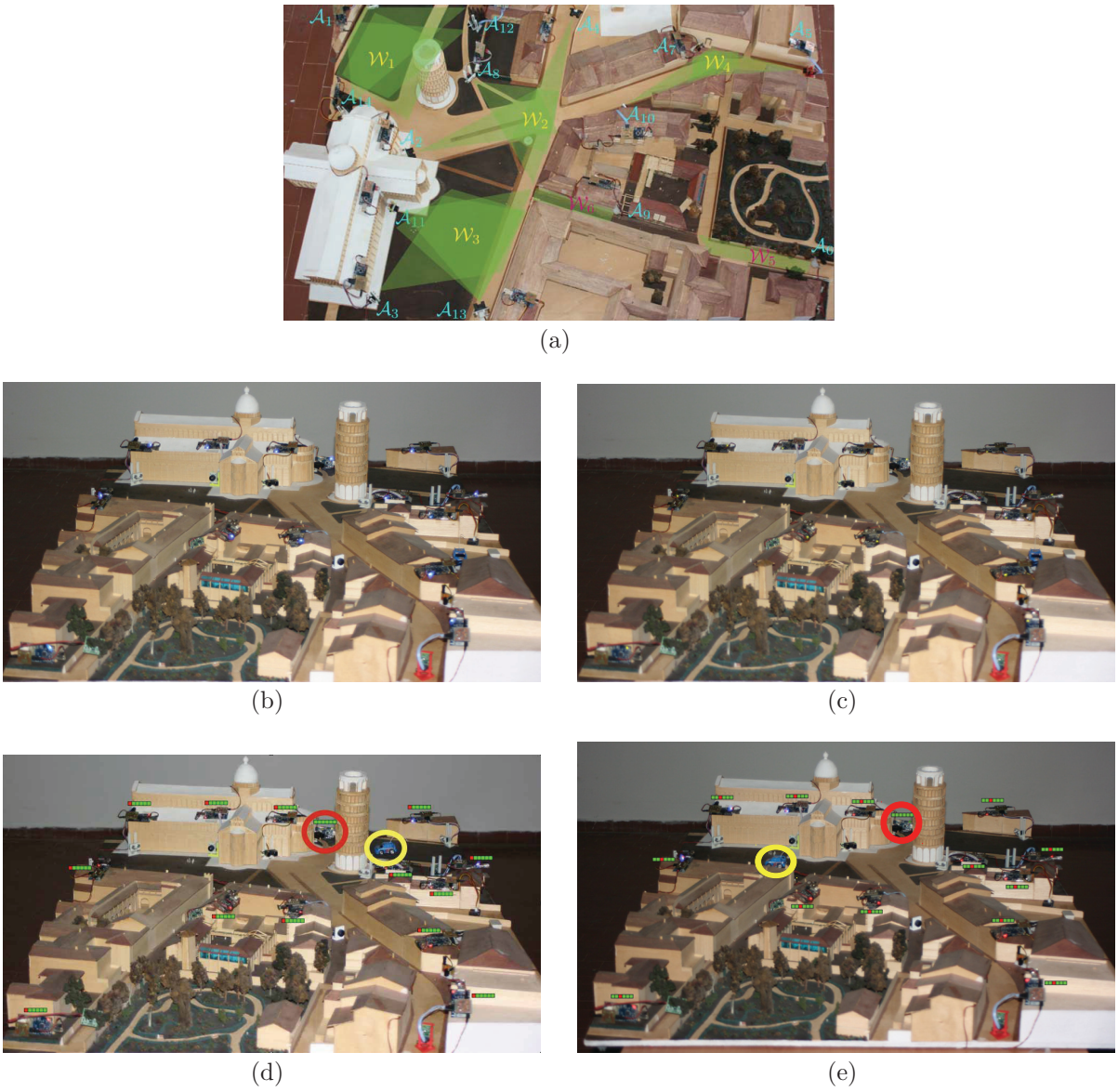
Fagiolini, A., Pellinacci, M., Valenti, G., Dini, G., and Bicchi, A. (2008a). Consensus based Distributed Intrusion Detection for Multi Robot Systems. In *Proc. IEEE International Conf. on Robotics and Automation*, 120–127.

Fagiolini, A., Visibelli, E., and Bicchi, A. (2008b). Logical Consensus for Distributed Network Agreement. In *47th IEEE Conference on Decision and Control, 2008. CDC 2008*, 5250–5255.

Fax, J. and Murray, R. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9), 1465–1476. doi: 10.1109/TAC.2004.834433.

Frasca, P., Carli, R., Fagnani, F., and Zampieri, S. (2008). Average consensus on networks with quantized communication. *International Journal of Robust and Nonlinear Control*.

Ganguli, A., Cortés, J., and Bullo, F. (2008). Distributed coverage of nonconvex environments. *Networked Sensing Information and Control*, 289–305.



(a)

(b)

(c)

(d)

(e)

Fig. 2. Scale model of the urban area nearby Pisa's Leaning Tower: (a) sensor nodes location and shared areas (yellow $\mathcal{W}_i$), (b) execution of SR^{2}NP, (c) intrusion detection mode, (d) intruder (yellow circle) is detected by agents $\mathcal{A}_1$,$\mathcal{A}_{12}$ and the intrusion is correctly dispatched to every agent even though the faulty agent $\mathcal{A}_{14}$ (red circle) transmits an incorrect information, (e) intruder, now in area $\mathcal{W}_3$, is correctly detected by agents $\mathcal{A}_3$,$\mathcal{A}_{11}$,$\mathcal{A}_{13}$.

Jadbabaie, A., Lin, J., and Morse, A. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6), 988–1001.

Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3), 382–401. doi: http://doi.acm.org/10.1145/357172.357176.

MOTEIV (2006). Tmotesky datasheet: Ultra low power ieee 802.15.4 compliant wireless sensor module.

Oh, S., Schenato, L., Chen, P., and Sastry, S. (2007). Tracking and coordination of multiple agents using sensor networks: system design, algorithms and experiments. *Proceedings of the IEEE*, 95(1), 234–254.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3), 401–420.

Olfati-Saber, R., Fax, J., and Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215.

Parker, L. (2008). Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1), 5.

Pavone, M., Frazzoli, E., and Bullo, F. (2009). Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. *IEEE Trans. on Automatic Control*. Provisionally accepted.

Schenato, L. and Gamba, G. (2007). A distributed consensus protocol for clock synchronization inwireless sensor network. *46th IEEE Conference on Decision and Control*.

Tanner, H., Jadbabaie, A., and Pappas, G.J. (2007). Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5), 863–868.