On the Convergence of the Max-Consensus Protocol with Asynchronous Updates

Silvia Giannini, Donato Di Paola, Antonio Petitti, Alessandro Rizzo

Abstract—In this paper, we present new theoretical results on the convergence of max-consensus protocols for asynchronous networks. The analysis is carried out exploiting well-established concepts in the field of partially asynchronous iterative algorithms and of analytic synchronization. As a main result, we propose a theoretical setting to prove the convergence of the asynchronous max-consensus protocol. Moreover, we provide an upper bound on the convergence time of the max-consensus protocol in asynchronous networks.

I. INTRODUCTION

The research on consensus problems in distributed systems aims at establishing protocols and conditions under which a number of components agree upon a common value, through only local interactions and computations [1]. The increasing interest in the field of consensus problems is due to their wide range of applications in distributed control and coordination of networked multi-agent systems, such as of autonomous vehicles [2], ad hoc wireless networks [3], and sensor networks [4]. Among consensus protocols, max-consensus allows the network nodes to converge to the maximum of their initial values. Max-consensus has wide application in distributed decision making problems [5], [6]. In the aforementioned works, the max-consensus protocol is defined in a synchronous framework, that is, agents’ operations must be synchronized to a common clock shared by all the agents in the network. However, asynchronous scenarios are prevalent in real distributed systems (e.g., in a companion paper [7], we propose a solution to the consensus-based distributed target tracking problem [6], in presence of asynchronous operations). Asynchronous settings allow the user to loose assumptions on the agents’ timing properties, that is, on the processing speeds of nodes and on the communication delays over the network.

In this paper, we deal with the convergence properties of the discrete-time asynchronous max-consensus protocol. Here, we formulate the hypothesis of partial asynchronism, usually satisfied by real message-passing systems [8]. Each agent updates its state in discrete time, by using an individual clock period. Agents’ state updates are performed on the base of the most recent information received by neighboring agents. No further assumptions are made about the relative frequency of agent’s clocks. Moreover, we derive an upper bound on the convergence time in the asynchronous setting. The convergence analysis is carried out by putting in correspondence the asynchronous system under exam with a suitably defined synchronous discrete-time system with switching topology. To this aim, we make use of the analytic synchronization formalism, introduced in [8], [9], and frequently used in similar contexts [10], [11]. The equivalence with a synchronous model allows us to exploit results obtained for synchronous settings, obtained by making use of the max-plus algebra [5], [12]. The main result in this paper is a theorem proving that, if the static graph which describes the asynchronous communication network is connected, then the consensus of all the network agents to the maximum of the initial information states will be reached in finite time.

The rest of the paper is organized as follows. In Section II, the problem formulation is given. A practical application of the formulated problem (i.e., the distributed task-assignment problem) is shown in Section III. The synchronous equivalent model is introduced in Section IV. Section V presents the convergence analysis of the asynchronous max-consensus protocol. Conclusions are drawn in Section VI.

II. PROBLEM FORMULATION

A. Agent and Network Model

Consider a network of $n$ dynamical systems, $n > 1$. The inter-agent communication is modeled by a static undirected graph $G = (\mathcal{I}, \mathcal{E})$, where $\mathcal{I} = \{1, 2, \ldots, n\}$ is the node set representing the $n$ agents, and $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{I}$ models the set of communication links between them.

Assumptions 2.1: We assume that the graph $G$ is connected, i.e., there is a multi-hop undirected path between any two nodes in $\mathcal{I}$. The adjacency matrix $A = [a_{ij}]$ associated with $G$, is defined as $a_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$. Here, we allow self-loops, i.e. $(i, i) \in \mathcal{E} \forall i \in \mathcal{I}$. Therefore, $a_{ii} = 1, \forall i \in \mathcal{I}$. The set of neighbors of node $i$ is defined by $\mathcal{N}(i) = \{j \in \mathcal{I} | a_{ij} = 1\}$.

We denote with $t_{k}^{(i)} \in \mathbb{R}, k \in \mathbb{N}$, the $k$-th update time for the $i$-th agent. An update time is a time instant in which the $i$-th agent updates its information state. In the most general...
Given the system the only information needed by agent
mean the past history of the state evolution up to time
case, the sequence of local update times, here specified for
agent \(i \in \mathcal{I}\), can be formalized as
\[
I_{k+1}^i = I_k^i + \chi_i^k, \quad k \in \mathbb{N},
\]
where \(\chi_i^k\) is a uniformly distributed random variable, with mean \(c_i^k \in \mathbb{Q}\) (the mean clock period) and variance \(\frac{c_i^k}{12}\) [13]. The set \(T_i^k = \{t_0^i, t_1^i, \ldots\}\) denotes the sequence of update times for agent \(i \in \mathcal{I}\). Note that, in the previous description, it is not required that \(t_0^i = t_0^j\) for any couple of agents \(i, j \subseteq \mathcal{I}\), that is, the local discrete time base of every agent \(i\) is referred to its own start time, \(t_0^i\). In other words, it is not possible to set a partial order among different agents’ update times based only on the knowledge of their clock mean periods, without mapping every single update time to a global time base. The information state of agent \(i\) at time \(t_k^i\) is denoted by \(x(t_k^i)\). The presence of different local mean clock periods \(c_i^k\), as well as different start times \(t_0^i\), constitutes the source of misalignment among agents’ local time bases, which consequently causes the agents’ state updates to be asynchronous.

B. Asynchronous Max-Consensus

The asynchronous max-consensus protocol \(P\) on the graph \(\mathcal{G}\) is defined by the local state-update rule:
\[
x(t_{k+1}^i) = \max_{j \in N[i]} \{x(t_k^j) + \theta_j^k\}, \quad i = 1, \ldots, n, \quad k \in \mathbb{N},
\]
(2)

The values \(x(t_k^j) + \theta_j^k\) are the information states of agents \(j \in N[i]\) available to agent \(i\) at the update time \(t_{k+1}^i\). The time variable \(\theta_j^k \in \mathbb{R}\), evaluated with respect to the update time \(t_k^i\), is used here to express asynchronous state updates occurring at the neighbors of agent \(i\), between two consecutive updates of the state of agent \(i\). It results \(0 \leq \theta_j^k < t_{k+1}^i - t_k^i\), \(\forall j \in N[i]\). In particular, \(\theta_j^k = 0\), \(\forall i \in \mathcal{I}\) and \(\forall k \in \mathbb{N}\). Moreover, it results \(\theta_j^k = 0\) for each \(j \in N[i]\) that does not produce an update in \([t_k^i, t_{k+1}^i)\). Note that, for any agent \(j \in N[i], j \neq i\), it is possible to have more than one update time in the interval \([t_k^i, t_{k+1}^i)\). Hence, we assume that the variable \(\theta_j^k\) always refers to the last update occurring for agent \(j\) in \([t_k^i, t_{k+1}^i)\). This implies that, in the update rule (2), agent \(i\) always uses the most recent information received asynchronously by its neighbors, while the value of the information state \(x(t_k^j) + \theta_j^k\) condenses the past history of the state evolution up to time \(t_k^i\). The values \(x(t_k^j)\) and \(x(t_k^j) + \theta_j^k\), with \(j \neq i\), are therefore the only information needed by agent \(i\) to evolve to the new state \(x(t_{k+1}^i)\). We denote with
\[
\mathcal{S} = \langle \mathcal{G}, P, T \rangle,
\]
the multi-agent system described so far, where \(\mathcal{G}\) models the network topology, \(P\) is the asynchronous max-consensus protocol introduced in (2), and \(T = \bigcup_{i=1}^{n} T_i^k\) represents the unordered set of all the agents’ update times.

\textbf{Problem 2.1 (Asynchronous max-consensus problem)}: Given the system \(\mathcal{S}\), protocol \(P\) solves the asynchronous max-consensus problem if \(\forall i \in \mathcal{I}\) there exists a time index \(\nu_i^k \in \mathbb{N}\), s.t. \(\forall k \geq \nu_i^k, i \in \mathbb{N}\), the relation \(x(t_k^i) = \max_{j \in \mathcal{I}} \{x(t_j^i)\}\) is verified.

III. A Motivational Example

In this section, we sketch a motivational example showing the importance of asynchronous max-consensus protocols. In particular, we cope with a simplified version of the task assignment problem [14], in an asynchronous setting.

Consider \(n\) agents, with index set \(\mathcal{I} = \{1, \ldots, n\}\), connected by a static undirected communication network \(\mathcal{G}\), as described in Section II-A. A single task has to be assigned to one of the \(n\) agents (conflict free assignment). To this aim, a nonnegative reward (or bid) \(v_i^k \geq 0\) is assigned by each agent \(i \in \mathcal{I}\) to that task. A feasible assignment is indicated by the variable \(\bar{z} \in \mathcal{I}\), that represents the index of the agent assigned to the task at the end of the task-assignment process.

In the following example, we apply a decentralized solution to this single-task assignment problem, based on single-item auctions and on the max-consensus protocol. The solution strategy is implemented so that agents assume that their operations are synchronous, whereas in reality they are not. In this way, we highlight the issues related to the application of synchronous protocols to asynchronous environments.

Consider a network of \(n = 3\) agents, connected via a line topology (see Fig. 1). Suppose that each agent \(i \in \mathcal{I}\) has asynchronously bid on the task at \(t_0^i\), and assigned it to itself, i.e., it results \(z[i](t_0^i) = i, \forall i \in \mathcal{I}\), where \(z[i](t_0^i)\) indicates the local assignment for agent \(i\) at time \(t_0^i\). We assume that the agents’ bids on the task are the following: \(r_1[0](t_0^1) > r_2[0](t_0^2) > r_3[0](t_0^3)\). Thus, the feasible assignment is \(\bar{z} = 1\), i.e., the final local assignment should be \(z[i] = 1, \forall i \in \{1, 2, 3\}\).

In order to reach a feasible final assignment, a distributed max-consensus protocol is adopted by agents in the network. In each iteration of the consensus phase, each agent \(i \in \mathcal{I}\) updates a winning bid variable \(w_i[i](t_{k+1}^i)\) to the maximum value of the bids received by neighboring agents, and sends this value to each agent \(j \in N[i]\). At the same time, the ID of the current winner of the task is stored in \(z[i](t_{k+1}^i)\).

At the beginning, the winning bid is set to \(w_1[0](t_0^1) = r_1[0](t_0^1)\) at each iteration \(q\) of the consensus phase occurs in correspondence to a time instant \(t_{k+1}^i, k \in \mathbb{N}_0\), of the local time scale of the generic agent \(i\). Therefore, there is no simultaneous communication among agents.

It is well known that, in synchronous settings, the number of iterations required to converge to a feasible assignment is less or equal than the network diameter of \(\mathcal{G}\), here and henceforth indicated with \(D\) [15]. Thus, if agents assume that the setting is synchronous, they can rely on the upper bound \(D\), which in our example is easily computed as \(D = 2\), as the stop criterion for the execution of the consensus phase. We assume the following relation among agents’ clock periods, \(c_1^k > c_2^k > c_3^k\). Thus, agent 1 updates its variables less frequently than agent 3 (see Fig. 2). We will now show that this time misalignment leads to an unfeasible assignment.
In the example illustrated in Fig. 2, we indicate with \( t_1^i \) and \( t_2^i \), respectively, the time instants at which the first and the second iteration of the consensus phase occur, respectively, for agent \( i \).

At first, agent 3 communicates to agent 2 its bid \( r_3^i(t_3^3) \). Then, agent 2 already knows the bid of agent 3, and sends to both agent 1 and 3 its bid, i.e., \( \max\{r_2^i(t_2^2), r_3^i(t_3^3)\} \).

Agent 1, after receiving both bids \( r_2^i(t_2^2) \) and \( r_3^i(t_3^3) \), sends its bid \( \max\{r_1^1(t_1^1), r_2^i(t_2^2), r_3^i(t_3^3)\} \) to agent 2. At iteration 2, occurring at \( t_2^2 \), each agent terminates the auction phase asynchronously. Agent 3 considers agent 2 as being the winner of the task \( z_2^3(t_2^3) = 2 \), because, as a consequence of the time misalignment, it has not yet received the bid of agent 1 at the current time. Agent 2, which has already received all the bids in its second iteration, reaches the correct assignment \( z_2^2(t_2^2) = 1 \). Finally, agent 1, which has obtained the bids from all agents, assigns correctly the task to itself \( z_1^1(t_1^1) = 1 \). As it can be noted, the correct assignment has not been computed correctly by all agents. Thus, the execution in an asynchronous scenario of the algorithm designed for synchronous settings may lead to incorrect outcomes. To cope with this issue, in the following we study the convergence properties of the asynchronous max-consensus protocol.

IV. ASYNCHRONOUS FRAMEWORK

In this section, we apply the analytic synchronization method [9] to derive an equivalent synchronous discrete-time model of the asynchronous multi-agent system described by the tuple (3). It will allow us to exploit the results proposed in literature for synchronous max-consensus problems [12], in the convergence analysis of the asynchronous protocol presented in Sec. V.

A. Virtual Global Time

We firstly introduce a discrete time base \( \tau \), designating in the following an abstract global time base possessed by a hypothetical external observer of the network state evolution.

The global time base is used as a reference to fulfill a total ordering on the sequence of update times of the \( n \) agents, and is only needed for analysis purposes. Thus, from the external observer point of view, it is possible to sort along the temporal dimension \( \tau \) the set \( T = \bigcup_{i=1}^{n} T^i \). In the following, we refer to the sorted set with \( \bar{T} \).

Inspired to the analytic synchronization method [9], in the following we show that it is always possible to derive a correspondence between every agent update time in \( T^i \), \( \forall i \in I \), and a discrete time (i.e., a virtual update time) of the global time base \( \tau \). We indicate with

\[
\Upsilon = \{ \tau_l | \tau_h = h \cdot c^O, h \in \mathbb{N} \}
\]

the discrete set of virtual update times of the abstract global time base \( \tau \), where \( c^O \) is a suitably chosen clock period for \( \tau \). In other words, the definition of \( c^O \) allows us to put in correspondence every occurrence of local discrete event times, \( t_k^i, i \in I \), with a discrete time, \( \tau_{k,i} \), expressed in the global reference time scale \( \tau \). It can be proved that \( c^O \) can always be chosen to deal with any asynchronous setting of interest. In particular, three cases are envisioned.

1) Variable clock periods: In the case of clocks modeled by uniformly distributed random variables as in Eq. (1), it is possible to identify \( c^O \) as the biggest time interval contained an integer number of times in every time interval occurring between two consecutive update times in the sorted set \( \bar{T} \) (see Fig. 3).

2) Constant clock periods: Consider now the case in which each agent has a constant local clock period \( c^i \in \mathbb{Q} \), \( i \in I \), i.e., each agent updates its state at local discrete-time instants \( t_k^i = k \cdot c^i \), \( k = 0, 1, 2, \ldots \). If the start times \( t_0^i \) and \( t_0^j \) (respectively, \( \tau_0, \tau_j \in \bar{T} \)) are not aligned for some \( i, j \in I \) (see Fig. 4), the relation \( t_k^i < t_k^j \), \( k = 1, 2, \ldots \), (i.e., in the global time reference, \( \tau_{k,i} < \tau_{k,j} \)) might not hold.
though it would result $c[i] < c[j]$. Therefore, it is not possible
desire a formal expression of the value of the global clock
period yet, and $c^O$ is defined as in the case of variable

clocks.

3) Constant clock periods with aligned initial instants:

Consider now the case in which each agent has an individual,
yet constant local clock period $c[i] \in \mathbb{Q}$, $i \in \mathcal{I}$, and let us
assume that $t^{i}_0 = v^{j}_0$, $\forall \{i,j\} \subseteq \mathcal{I}$, i.e., all the agents
start times are aligned (see fig. 5). Thus, in this case, it results
$t^{i}_0 = \tau_0$, $\forall i \in \mathcal{I}$, where, without loss of generality, $\tau_0 = 0$
can be considered as the common virtual start time. In this
case, the global clock period $c^O$ can be defined as:

$$
c^O = \text{gcd}(c[i]), \quad i \in \mathcal{I},
$$

where gcd stands for the greatest common divisor operator.

**Assumptions 4.1:** For simplicity of discussion, without
loss of generality, in the following we refer to case 3).

**Remark 4.1:** Assumption 4.1 is not restrictive and does
not invalidate the subsequent analysis, which is valid for
more general settings, since it is always possible to define
$c^O$ as in the general case of variable clocks described above.

Therefore, for each agent $i \in \mathcal{I}$, distinct local updates can
always be put in correspondence with distinct global update
times in the global time scale $\tau$. Moreover, as stated above,
every local update time in $T$ always corresponds to a virtual
update time in $\Upsilon$ (multiple simultaneous local update
times are mapped to the same global update time). Please note
that the contrary does not hold, that is, it is possible to have
virtual update times in $\Upsilon$ that do not correspond to any local
update time in $T$.

If Assumption 4.1 holds, the relation

$$
k_i = \frac{k \cdot c[i]}{c^O}
$$

maps the occurrence of every local discrete instant $t^{i}_k \in \mathcal{T}^{[i]}$, $i \in \mathcal{I}$, to a global virtual update time $\tau_k \in \Upsilon$.

**B. Equivalent Synchronous Model**

After introducing the set $\Upsilon$, let us assume the external
observer point of view, and define the interaction topology
as seen from that point of observation. In that way, we will
observe that the asynchronous protocol acting on a fixed
topology is equivalent to a synchronous one, acting instead
on a switching topology. Let $h \in \mathbb{N}$ be the index variable
of the virtual discrete times $\tau_h \in \Upsilon$ defined by Eqs. (4–
5). It is straightforward to verify that direct communication
from agent $i$ to agent $j$ can take place through the network
if and only if $j \in N^{[i]}$ and $\tau_h \in \mathcal{T}^{[i]}$ (please note that
the expression $\tau_h \in \mathcal{T}^{[i]}$ corresponds to the local update
time $t^{i}_h \in \mathcal{T}^{[i]}$ of agent $i \in \mathcal{I}$, where the value $h^*$
is obtained from $h$ by applying the inverse formula of Eq. (6)).

Therefore, it can be easily verified that the multi-agent
interaction topology, seen from the external observer, can be
modeled by a directed switching graph $\mathcal{G}_h = (\mathcal{I}, \mathcal{E}(h))$. In
particular, $\mathcal{G}_h$ has the same set of vertices $\mathcal{I}$ of graph $\mathcal{G}$
in the tuple (3), while $\mathcal{E}(h)$ represents the set of (possibly
unidirectional) data-exchange flows among network agents
at time $\tau_h$. A directed edge $(i, j) \in \mathcal{E}(h)$ if and only if
$\tau_h$ corresponds to an update time for agent $i$, that is, the
time when $i$-th node updates its information state value and
sends it to all the neighboring agents $j \in N^{[i]}$. Moreover,
Assumption 2.1 implies that $(i, i) \in \mathcal{E}(h) \forall \tau_h \in \Upsilon$, i.e.,
every node can always access its own state. The switching
topology of the directed graph $\mathcal{G}_h$ is described by the
adjacency matrix $A_h \in \{0, 1\}^{n \times n}$. In every time interval
$[\tau_h, \tau_{h+1})$, the generic element $a^{ij}_h \in \{0, 1\}$ of $A_h$ is defined
as follows:

$$
a^{ij}_h = \begin{cases} a^{ij} & \text{if } \tau_h \in \mathcal{T}^{[i]}, j \neq i \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases}
$$

We identify the dynamic set of neighbors of node $i$ at time
$\tau_h \in \Upsilon$ as

$$
\mathcal{N}^{[i]}_h = \{j \in \mathcal{I} | a_h^{ij} = 1\}.
$$

It obviously results $\mathcal{N}^{[i]}_h \subseteq N^{[i]}$, $\forall h \in \mathbb{N}$. In particular,
$\mathcal{N}^{[i]}_h = N^{[i]}$ if $\tau_h \in \mathcal{T}^{[i]}$; otherwise, $\mathcal{N}^{[i]}_h = \{i\}$.

A synchronous max-consensus protocol $\mathcal{P}$ over the switching
topology represented by graphs $\mathcal{G}_h$, is defined by the local
state-update rule:

$$
\tilde{x}^{[i]}(\tau_{h+1}) = \max_{j \in \mathcal{N}^{[i]}_h} \{\tilde{x}^{[j]}(\tau_h)\}, \quad i = 1, \ldots, n
$$

The update rule is analogous to the one in Eq. (2), which
holds for the asynchronous case. Note that here, differently
from Eq. (2), every agent $i \in \mathcal{I}$ updates its state at the same
time instant $\tau_h \in \Upsilon$. The only functional difference between
the two cases is the set of neighbors $\mathcal{N}^{[i]}_h$ in particular, the state evolution rule in Eq. (9) can be decoupled in the two rules

$$
\begin{align*}
\tilde{x}^{[i]}(\tau_{h+1}) &= \max_{j \in \mathcal{N}^{[i]}_h} \{\tilde{x}^{[j]}(\tau_h)\} \quad \forall \tau_{h+1} \in \mathcal{T}^{[i]} \\
\tilde{x}^{[i]}(\tau_{h+1}) &= \tilde{x}^{[i]}(\tau_h) \quad \forall \tau_{h+1} \notin \mathcal{T}^{[i]}
\end{align*}
$$

It can be easily verified that for any given set of agents $\mathcal{I}$, the
value of $\tilde{x}(\tau_h) = [\tilde{x}^{[i]}(\tau_h)]_{i \in \mathcal{I}} \in \mathbb{R}^{n \times 1}$, $\tau_h \in \Upsilon$ and $h > 0$,$\quad$ is uniquely determined by the initial conditions $\tilde{x}(0)$ [8].

We denote with the tuple

$$
\tilde{\mathcal{S}} = (\mathcal{G}, \mathcal{P}, \Upsilon),
$$

}
By applying the inverse formula of Eq. (6) to Eq. (16), we
used in the update rule of Eq. (13), was produced and the
interval to the value is verified. In the following, we prove that the value
chronous protocol (2) is equivalent to protocol (10) of the synchronous system \( \tilde{S} \), and to state that they are equivalently valid for the asynchronous system \( S \), we need to prove the equivalence of the two systems.

**Theorem 4.1:** Given the system \( S \) in (3), the asynchronous protocol (2) is equivalent to protocol (10) of the synchronous system \( \tilde{S} \), derived under the analytic synchronization procedure.

**Proof:** Consider an agent \( i \in I \) and its information state at a generic local update time \( t^i_k \in T^i, k \in \mathbb{N} \). Assume that the subsequent relation

\[
\tilde{x}^i[\tau_k] = x^i[ t^i_k ], \quad k \in \mathbb{N}. \tag{12}
\]

is verified. In the following, we prove that the value

\[
x^i[ t^i_{k+1} ] = \max \{ x^i[ t^i_k ], \max \{ x^j[ t^i_k + \theta^j ] j \in X^i \} \}, \tag{13}
\]
determined by the local state update rule in Eq. (2), is equal to the value \( \tilde{x}^i[\tau_{k+1}] \), obtained through the application of Eq. (9) at every virtual global time occurring in the interval \( [\tau_k, \tau_{k+1}] \).

First, exploiting Eq. (6), we obtain that

\[
(k + 1)\theta^i = k \cdot \frac{c^i}{\epsilon^O} + c^i + \epsilon^i = k_i + \epsilon^i, \tag{14}
\]

where \( \epsilon^i \in \mathbb{N}_0 \). Then, the relation

\[
\tilde{x}^i[\tau_h] = \tilde{x}^i[\tau_k], \quad \forall h \in \mathbb{N} \text{ s.t. } k_i \leq h < k_i + \epsilon^i \tag{15}
\]

follows by equation (10-(ii)).

Suppose now that \( \exists q^j \in T^j, j \in X^i, j \neq i \), s.t.

\[
k_i \leq q_j + m \cdot \frac{c^j}{\epsilon^O} < k_i + \epsilon^i, \tag{16}
\]

where \( q_j \) is the index of the virtual global time instant corresponding to the \( q \)-th update time for agent \( j \), according to \( (6) \), and \( m \in \mathbb{N} \). This means that agent \( j \) updates its information state \( m + 1 \) times in the time interval \( [\tau_k, \tau_{k+1}] \). By applying the inverse formula of Eq. (6) to Eq. (16), we obtain:

\[
k_i \epsilon^O \leq q_j \epsilon^O + m \cdot \frac{c^j}{\epsilon^O} < k_i \epsilon^O + \epsilon^i, \tag{17}
\]

\[
k_i \cdot \epsilon^i \leq (q + m) \frac{c^j}{\epsilon^O} < (k + 1) \epsilon^i. \tag{18}
\]

Then, for every agent \( j \in X^i \), \( j \neq i \), complying with condition (16), the following equivalence derives from (10-(ii)):

\[
\tilde{x}^j[\tau_h] = \tilde{x}^j[\tau_{(q+m)j}], \quad \forall h \in \mathbb{N} \text{ s.t. } q_j + m \frac{c^j}{\epsilon^O} \leq h < k_i + \epsilon^i \tag{21}
\]

For simplicity of notation, we omit the dependence of \( m \) on the agents’ indices \( j \in X^i \) that satisfy (16). Consider now \( h = k_i + \epsilon^i - 1 \). Then, in the virtual global time \( \tau_{h+1} \), equation (10-(i)) holds. Jointly applying relations (10-(i)), (12), (15), (19) - (21), it follows that

\[
\tilde{x}^j[\tau_{h+1} = \max \{ \tilde{x}^j[\tau_{h}], \max \{ x^j[\tau_{(q+m)j}] \} \}, \tag{22}
\]

where \( \tilde{x}^j[\tau_{h+1}] = \tilde{x}^j[\tau_{(k+1)j}] \), and the equivalence \( \tilde{x}^j[\tau_{(q+m)j}] = x^j[ t^j_{q+m} ] \) can be proved by induction.

Comparing equations (13) and (22), it is straightforward to note that the values \( x^i[ t^i_{k+1} ] \) and \( \tilde{x}^i[\tau_{(k+1)j}] \) are equivalent under relations (12), (19) and (20). Moreover, due to assumption 4.1, the relation \( x^i[ t^i_{k+1} ] = \tilde{x}^i[\tau_{(k+1)j}] \) is verified, and the hypotheses (12) can be proved by induction. Therefore, the synchronous protocol (10) is an equivalent model for protocol (2).

**V. CONVERGENCE ANALYSIS**

We prove now the convergence of the asynchronous max-consensus protocol (2) in \( S \), relying on its derived synchronous model \( \tilde{S} \) expressed by (11). The necessary and sufficient condition for the convergence of synchronous max-consensus protocols on switching topologies is given in [12], and reported in the following theorem.

**Theorem 5.1 ([12]):** Given a finite sequence of \( f \) adjacency matrices \( \tilde{A}_1, \ldots, \tilde{A}_k, \ldots, \tilde{A}_f \) which defines a switching topology, the synchronous system described by (11) achieves max-consensus in finite time for all initial condition vectors \( \tilde{x}(0) \) if and only if the directed graphs \( \tilde{G}(\tilde{A}_1), \ldots, \tilde{G}(\tilde{A}_k), \ldots, \tilde{G}(\tilde{A}_f) \) are jointly strongly connected (\( \tilde{G}(\tilde{A}_k) \) stands for a directed graph with adjacency matrix \( \tilde{A}_k \).

Before proving that the aforementioned condition is satisfied by the synchronous equivalent model (11), we introduce the following definition.

**Definition 5.1:** (Partially asynchronous algorithm [8]) An asynchronous algorithm is called partially asynchronous if it satisfies two conditions:

1) each node performs an update at least once during any time interval of \( B \) global time units;
2) the information used by any node is outdated by at most \( B \) global time units.

The positive constant \( B \in \mathbb{N} \) is called asynchronism measure. It means that, in protocol (2) we assume local clocks not arbitrarily slow, if compared with reference to a common timebase.

The partial asynchronism requirement, expressed for the
system $S$ by definition 5.1, can be rewritten as follows in terms of the synchronous model $\tilde{S}$.

Assumptions 5.1: There exists a positive integer $B$ s.t.

1) for every $i \in I$ and for every $\tau_i \in \mathbb{T}$, at least one of the elements of the set $\{\tau_i, \tau_i + 1, \ldots, \tau_i + B - 1\}$ belongs to $\mathbb{T}^i$;
2) defining $\tau_{ij} \in \mathbb{T}^j$ as the global time at which agent $j \in N^i$ produces the information used by the $i$-th agent in the state update rule (2) at time $\tau_{ki} \in \mathbb{T}^i$, there holds $\tau_{ki} - B \cdot c^2 < \tau_{ij} < \tau_{ki}$, $\forall j \in N^i$, and $\forall k \in \mathbb{T}^i$, $\forall j \in \mathbb{T}^j$.

Theorem 5.2: Given the partially asynchronous system $S = (G, P, T)$ described in (3), and its equivalent synchronous model $\tilde{S} = (\tilde{G}, \tilde{P}, \tilde{T})$ defined in (11), if the graph $G$ is connected, then $\forall \tau_i \in \mathbb{T}$, $\exists f \in \mathbb{N}$ s.t. the sequence $\{\tilde{G}_f, \tilde{G}_{f+1}, \ldots, \tilde{G}_{f+B} \}$ is jointly strongly connected.

Proof: Consider an agent $i \in I$ and a generic global time $\tau_i \in \mathbb{T}$. From the first assertion of Assumption 5.1, it follows that $\forall j \in N^i$, $\exists \tau_{ij}$, $\tau_{ij} < \tau_{ki} < \tau_i + B - 1$, s.t. $\tau_{ij} = \tau_{ki}$, that is, $N^i = \bigcup_{p=h}^{h+B-1} N^i_{\tau_{ij}}$, $\forall i \in \mathbb{T}$, according to (8). This implies that, in at most $B$ global time instants, every agent $i$ receives the information state by each one-hop neighbor $j \in N^i$, and vice versa. Analogously, it can be derived that, $\forall \tau_i \in \mathbb{T}$, $A = \bigcup_{p=h}^{h+B-1} \tilde{A}_p$ and $\tilde{G} = \bigcup_{p=h}^{h+B-1} \tilde{G}_p$, where $\bigcup$ is the elementwise logical or of the adjacency matrices, and $\tilde{A}_p$ is the adjacency matrix of the dynamic directed graph $\tilde{G}_p$ at time $\tau_{p} \in \mathbb{T}$. Then, the theorem is proved under Assumption 2.1, since it is possible to consider $f = h + B - 1$, $\forall \tau_i \in \mathbb{T}$.

Consequently, due to the equivalence between $S$ and $\tilde{S}$, we conclude that, if the topology $G$ of the partially asynchronous system (3) is connected, the convergence of the asynchronous max-consensus protocol (2) is guaranteed.

Theorem 5.3: The max-consensus protocol in (2) converges for all initial conditions $x(0)$ if and only if the graph $G$ of the partially asynchronous system $S$ is connected.

Proof: The proof follows directly from the results stated in Theorems 5.1 and 5.2.

In the case of max-consensus protocol over static synchronous networks, it is well known that if a consensus occurs, it will be reached in a finite number of steps. In particular, we recall a theorem proved in [12].

Theorem 5.4: Let $G$ be a static directed graph and $\mathcal{P}$ the synchronous max-consensus protocol in (9). If $G$ is strongly connected, then max-consensus is achieved for all initial conditions in $D$ time steps, where $D$ is the diameter of $G$.

We remark that this theorem refers to static graphs, so the strong connectivity condition does not depend on time.

Lemma 5.1: (Convergence time) Let $G$ be a static undirected graph and $\mathcal{P}$ the asynchronous max-consensus protocol expressed by (2). If $G$ is connected, then protocol $\mathcal{P}$ achieves max-consensus for all initial conditions in at most $B \cdot D$ time steps, where $B$ is the asynchronism measure and $D$ the diameter of $G$.

Proof: Consider the synchronous model $\tilde{S}$ and the following sequences of adjacency matrices $S_1 : A_1, \ldots, A_B$; $S_2 : \tilde{A}_{B+1}, \ldots, \tilde{A}_2 B$; $\ldots$; $S_D : \tilde{A}_{(D-1)B+1}, \ldots, \tilde{A}_{DB}$. Each adjacency matrix models the network topology at a specific global time $\tau_h$, $h \in \{1, \ldots, D \}$. From the proof of Theorem 5.2, it follows that each sequence $S_k$, $k \in \{1, \ldots, D \}$, is jointly strongly connected. Moreover, $\mathcal{G}(S_k) = G$, $\forall k \in \{1, \ldots, D \}$. Since the graph $G$ is undirected and connected by hypothesis, it is possible to apply the result of Theorem 5.4. In particular, each of the $D$ steps, required to reach consensus, is formed by $B$ time instants. Therefore, the entire process takes at most $B \cdot D$ instants, referred to the global time scale.

VI. CONCLUSIONS

In this paper, we study the convergence properties of the asynchronous max-consensus protocol, motivated by real-world asynchronous applications. The analysis takes advantage of an equivalence settled between the asynchronous protocol on a static network and a suitable synchronous model on a switching topology. In particular, it is proved that, under the partial asynchronism assumption, the max-consensus converges in finite time. Moreover, the convergence time is bounded by a quantity which is function of the network topology and of the asynchronism measure.

REFERENCES