Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 601–606

Average Consensus with Asynchronous Updates and Unreliable Communication

Nicoletta Bof * Ruggero Carli * Luca Schenato *

* *All authors are with the Department of Information Engineering, University of Padova, Via Gradenigo 6/a,35131 Padova, Italy.* (e-mails:{bofnicol,carlirug,schenato}@dei.unipd.it)

Abstract: In this work we introduce an algorithm for distributed average consensus which is able to deal with asynchronous and unreliable communication systems. It is inspired by two algorithms for average consensus already present in the literature, one which deals with asynchronous but reliable communication and the other which deals with unreliable but synchronous communication. We show that the proposed algorithm is exponentially convergent under mild assumptions regarding the nodes update frequency and the link failures. The theoretical results are complemented with numerical simulations.

Keywords: Multi-agent systems, Distributed control and estimation, Control and estimation with data loss, Asynchronous algorithms, Average consensus

1. INTRODUCTION

In recent years, substantial effort has been dedicated towards the design of distributed algorithms for large-scale systems. The main driver for this is that, nowadays, the availability of small and cheap computational units is becoming widespread. As a consequence, it is affordable to develop extended systems to monitor and control many different environments. However, due to the size of this systems, it is not always possible to collect all the information in a single computational unit and sometimes it is neither advisable, since some of the information collected by the system could be sensible. Moreover, each unit is endowed with some computational power which will not be fully exploited otherwise. Another additional advantage of using a distributed approach is that the whole system is in a way safer, since it does not rely on a single unit but is assigned to many different ones. Distributed algorithms present two major drawbacks: memory and computational constraints and need of a reliable communication system. The first is due to the fact that the units which compose the system can be quite limited in their computational and storage capabilities, the second is intrinsic to the distributed setting, since each unit can exchange information only with its neighbours in order to perform a global task.

It is therefore fundamental to always consider the properties of the communication system adopted. One very important characteristic is the communication protocol, which can be synchronous or asynchronous. Synchronous protocols require substantial coordination between the nodes, and when the number of agents in the system increases, this coordination can become difficult to achieve. An asynchronous communication protocol, on the other hand, has no coordination requirements, but an algorithm which uses such a protocol could in general require more iterations because in each iteration of the algorithm only a subset of the nodes in the networks are activated. Another important feature of the communication system is its reliability due to the possibility that packets are lost during the transmission. Obviously, the system should not lose packets but perfect reliability could be difficult or too expensive to enforce; to deal with possible packet losses, either an acknowledgement scheme is developed or the algorithm is implemented in such a way that the loss of a packet is not detrimental for the convergence.

In this paper we describe and study a distributed algorithm for average consensus. Basically, each unit has a given scalar quantity and the aim for each node is to compute the average of all these quantities. Sensor networks represent a remarkable domain where the evaluation of the average of the measured quantities is required in several applications Xiao et al. (2005), Carron et al. (2014), Carli et al. (2011), Garin and Schenato (2010). However, differently from the rich literature on this topic, we adopt an asynchronous and unreliable communication system, and we allow the communication not to be bi-directional, that is if a unit communicates with another one, the converse is not assured. In a synchronous and reliable communication scenario, important works are Boyd et al. (2004), Olshevsky and Tsitsiklis (2009), Oreshkin et al. (2010), Domínguez-García and Hadjicostis (2011). When unreliability in the communication is introduced, some works have adopted the acknowledgement scheme Chen et al. (2010) or assumed that each unit can determine whether the communication works Patterson et al. (2007), Xiao et al. (2005). However, an acknowledgement scheme requires additional secondary transmissions, which slow down the entire algorithm and consume extra energy. Therefore, in context where the energy consumption is constrained, the latter scheme is not adoptable and the transmission has to be reduced only to essential informa-

⋆ This work is partially supported by Progetto di Ateneo CPDA147754/14 - New statistical learning approach for multi-agents adaptive estimation and coverage control.

10.1016/j.ifacol.2017.08.093

tion. In an asynchronous setting, Bénézit et al. (2010) introduce an algorithm that reaches average consensus using the so-called ratio consensus. A very interesting idea is introduced in Dominguez-Garcia et al. (2011) and Vaidya et al. (2011) where the adopted communication is synchronous and unreliable. In these works a robust and synchronous algorithm inspired by Bénézit et al. (2010) is introduced.

Adopting the idea of mass transfer given in Vaidya et al. (2011), but developing an asynchronous algorithm as done in Bénézit et al. (2010), we describe an algorithm for average consensus which is provably exponentially convergent to the average in an asynchronous and unreliable communication scenario. The convergence proof relies on the introduction of two assumptions concerning the communication scheme, one regarding the frequency of waking up of each node and the other regarding how many consecutive times a given link can fail. Our interest in this algorithm is justified by its possible execution in more complex algorithms. In fact, even though it is an interesting stand-alone algorithm, there are some algorithms which need average consensus algorithm as a building block, e.g. the Newton-Raphson algorithm for convex optimization (see Varagnolo et al. (2016)), some distributed versions of the Kalman filter (see Carli et al. (2008)) or some algorithms for energy resources distribution in power grids (see Dominguez-Garcia and Hadjicostis (2010)). However, to be used in such algorithms, the average consensus has to be exponentially convergent. The aforementioned works by Bénézit et al. (2010) and Vaidya et al. (2011), for example, do not show whether exponential convergence is guaranteed in their algorithms. By proving this kind of convergence for our algorithm, we make possible to use it in more advanced algorithms which could then be applied in a realistic communication scenarios (i.e. asynchronous and unreliable channels).

2. NOTATION & COMMUNICATION PROTOCOLS

Given a scalar $x \in \mathbb{R}$, $|x|$ denotes its absolute value. Given a matrix $A \in \mathbb{R}^{N \times N}$, $[A]_{ij}$ denotes its $(i,j)-$th element, and $A^\top$ indicate its transpose. A vector $\mathbf{x}$ is strictly positive if $x_i > 0$, $\forall i \in \{1,\dots,N\}$. Given two vectors $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$, with x_i or $[\mathbf{x}]_i$ we denote its $i-$th element and with $\mathbf{x}/\mathbf{z}$ the Hadamard division of the two vectors. I_N indicates the $N \times N$ identity matrix. A graph $\mathcal{G}$ is represented by the couple $(\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ the set of nodes $\{1,\dots,N\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ the set of edges. The number of edges in the graph is E. We consider directed, strongly connected and static graphs, with all the nodes having a self loop. Given a node $i \in \mathcal{V}$, the set $\mathcal{N}_{\text{in}}^i$ contains all the neighbours which communicate to i, that is $\mathcal{N}_{\text{in}}^i = \{j \,|\, j \in \mathcal{V}, i \neq j, (j,i) \in \mathcal{E}\}$, while the set $\mathcal{N}_{\text{out}}^i$ contains all the neighbours to which i communicates, that is $\mathcal{N}_{\text{out}}^i = \{j \,|\, j \in \mathcal{V}, i \neq j, (i,j) \in \mathcal{E}\}$. For a set $\mathcal{A} \in \mathcal{V}$, $|\mathcal{A}|$ denotes the cardinality of the set. A matrix $P \in \mathbb{R}^{N \times N}$ is row stochastic if $P\mathbb{1}_N = \mathbb{1}_N$, where $\mathbb{1}_N$ is the all-ones vector of dimension N. Finally, if $a, b \in \mathbb{R}$, $a < b$, $[a,b]$ indicates the interval between a and b, extremes included. In the following we briefly describe some of the communication protocols which are usually adopted in wireless sensors networks given a pre-assigned communication graph, namely, the *synchronous* protocols, as opposed to the *asynchronous* ones like *broadcast asymmetric*, *coordinated broadcast*, *gossip asymmetric* and *gossip symmetric*.

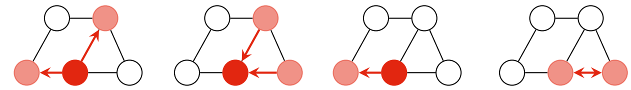

Fig. 1. Asynchronous communication protocols: the opaque red node is the one that wakes up (for the first 3 protocols), the other highlighted nodes are those which exchange information with it. In gossip symmetric there is no hierarchy in the nodes selected.

In a **synchronous** protocol all the nodes activate at the same time instants and perform the updating and communication operations (almost) synchronously. This protocol requires a common notion of time among the nodes; indeed all the agents have to wake up simultaneously and so a perfect coordination is required. If the graph is moderately small, requiring synchronization may not be a great deal, but as the dimension of the network increases, synchronization can become an issue. Instead, in **asynchronous** protocols, during each iteration, only a small subsets of all the nodes in the network perform the communication and updating steps. Specifically, in the **broadcast asymmetric** protocol, at each iteration, there is only one node transmitting information to its out-neighbours, which, based on the received messages, update their internal variables. At a given iteration, the (unique) transmitter node is said to be the one that wakes up (or turns on). The same terminology is applied to the node that performs the first step of the communication in the two protocols we describe next. The **coordinate broadcast** can be considered as the dual protocol of the broadcast asymmetric. Indeed, at each iteration, there is only one node which wakes up, but, instead of sending information, it polls all its in-neighbours in order to receive from them some desired messages. In **asymmetric gossip** again only one node wakes up but it sends information to only one of its out-neighbours, typically randomly chosen. Finally, the **symmetric gossip** is a protocol that requires bidirectional communication, that is the communication graph $\mathcal{G}$ has to be undirected (implying $\mathcal{N}_{\text{in}}^i = \mathcal{N}_{\text{out}}^i$ for all $i \in \mathcal{V}$); during each iteration an edge of the graph is selected and only the two nodes which are pointed by this edge exchange information with each other. Figure 1 gives a pictorial description of the asynchronous protocols just described.

3. PROBLEM FORMULATION

Consider N agents (also called nodes) which can communicate with each other according to a graph $\mathcal{G}$ and throughout some asynchronous communication protocol. We assume the communications to be unreliable, that is, some packet losses might occur during the transmission of the messages. Each node $i \in \{1,\dots,N\}$ has a private scalar quantity[1] $v_i \in \mathbb{R}$, which can be collected in vector $\mathbf{v} \in \mathbb{R}^N$, and the problem to solve is the evaluation of the mean of these v_i, that is of $\bar{v} = \sum_i v_i/N$. The evaluation has to be carried out by each node in a distributed way,

[1] The algorithm can be modified to manage multidimensional quantities.