Using a linear gain to accelerate average consensus over unreliable networks

Francesco Acciani, Paolo Frasca, Geert Heijenk, Anton Stoorvogel

HAL Id: hal-01614110

https://hal.archives-ouvertes.fr/hal-01614110

Submitted on 21 Oct 2017

Using a linear gain to accelerate average consensus over unreliable networks

Francesco Acciani, Paolo Frasca, Geert Heijenk and Anton Stoorvogel

Abstract— Packet loss is a serious issue in wireless consensus networks, as even few failures might prevent a network to converge to the correct value. However, it is possible to compensate for the errors caused by packet collisions, by modifying the updating weights. Such a modification compensates for the loss of information in an unreliable network, but results in a reduced convergence speed. In this paper, we propose a faster method – based on a suitable gain in the consensus dynamics – to solve the unreliable average consensus problem. We find a sufficient condition for the gain to preserve stability of the network. Simulations are used to discuss the choice of the gain, and to compare our method with the literature.

I. INTRODUCTION

Achieving average consensus over a wireless-connected network is a problem that arises in a variety of scenarios, where a set of agents needs to agree over the value of a network-shared variable in a distributed manner. Consensus algorithms form the basis for a large number of distributed algorithms, such as distributed hypothesis testing [1], distributed maximum likelihood estimation [2], and distributed Kalman filtering [3]. Such consensus algorithms have been studied under a wide variety of conditions including networks that have undirected or directed links, time varying topology [3], [4], and noisy channels [5], [6]. Most average consensus algorithms, however, are not robust to packet losses [7], which can be expected to happen when a wireless communication mechanism is employed.

Several methods exist to deal with packet loss, employing different strategies: retransmissions [8], additional variables to be transmitted [9], [10], or memory mechanisms [11], [12]. It is also possible to preserve consensus using a local compensation mechanism, which modifies the weights of the links between the nodes to accomodate the change in the network topology, as in [7]. In [13] we proposed a method to compensate the loss of information and guarantee average consensus and we estimated its convergence speed. In this paper, we propose a weight-modifying compensation strategy over networks with unreliable bidirectional links, to deal specifically with packet collisions. This compensation

F. Acciani is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands f.acciani@utwente.nl

P. Frasca is with Univ. Grenoble Alpes, CNRS, Inria, GIPSA-lab, F-38000 Grenoble, France paolo.frasca@gipsa-lab.grenoble-inp.fr

A. Stoorvogel is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands a.a.stoorvogel@utwente.nl,

G. Heijenk is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands geert.heijenk@utwente.nl

method drives the network to converge to the correct value despite failures in the communication between nodes. The compensation mechanism is enhanced with a gain factor: this allows for a faster network converge.

The rest of the paper is organised as follows. The second section recalls the consensus problem and discusses the issues that arise in the presence of packet loss. Section 3 presents our accelerated convergence method and a result to choose the corresponding gain in order to preserve convergence, whereas the results from simulation, supporting our theoretical claims, are in Section 4. In the final section our conclusions are presented.

II. CONSENSUS AND COMPENSATION

A network of n agents can be represented by a graph, where each vertex represents one agent, and each edge represents a wireless link between two nodes. The bidirectional nature of the wireless communication – hypothesis that arises naturally when the nodes are homogeneous – results in an undirected (symmetric) graph.

Assume that each agent i is endowed with a state x_i and can communicate with a subset of agents, namely its neighbourhood $\mathcal{N}_i$. Moreover, each agent updates itself, computing its state at time $k+1$ according to an *update rule* $g(\cdot)$:

$$x_i(k+1) = g_i\left(x_j(k)|j \in \mathcal{N}_i\right),$$

We assume that every node can access its own state, so $i \in \mathcal{N}_i$. The consensus problem is solved if the network converges – asymptotically – to the same value, so if:

$$\lim_{k \to \infty} x_i(k) - x_j(k) = 0 \quad \forall i,j \in \mathcal{N} \tag{1}$$

where $\mathcal{N}$ is the set of agents of the network. Moreover, if

$$\lim_{k \to \infty} x_i(k) = \frac{1}{n} \sum_{i=1}^{n} x_i(0) \quad \forall i \in \mathcal{N} \tag{2}$$

then we say that the system achieves *average* consensus. When the updates are synchronous and the communication links are reliable and fixed, i.e. the topology is time invariant, an update rule to achieve consensus is:

$$x_i(k+1) = \sum_{j=1}^{n} \overline{w}_{ij} x_j(k), \tag{3}$$

where $\overline{w}_{ij} \neq 0 \iff i$ and j are connected, which can be written, assuming $x(k) = [x_1(k)\, x_2(k) \cdots x_n(k)]^T$, as:

$$x(k+1) = \overline{W} x(k). \tag{4}$$

The $\overline{W}$ matrix is the weighted adjacency matrix, associated with a graph where the nodes are the vertex of the graph. If the graph associated with the $\overline{W}$ matrix is connected and aperiodic – i.e. a path exists between each pair of nodes, and the lengths of cycles in the graph are coprime integers – then the network converges in the sense of (1) if $\overline{W}$ is stochastic. In the rest of this paper, the graph associated to the network – in its *nominal* state, i.e. without losses – is always assumed to be connected and aperiodic, and the $\overline{W}$ matrix is assumed to be symmetric and stochastic, i.e.: $\overline{W}\mathbb{1} = \mathbb{1}, \overline{W} = \overline{W}^T$, and $\overline{w}_{ij} \geq 0$ for all i, j, where $\mathbb{1}$ denotes a vector of ones. This double stochasticity assumption implies that the network converges to the *average* of the states: if $\overline{W}$ is only row-stochastic, the network converges to a weighted average of the initial conditions instead.

When the links between nodes are not reliable, the dynamics (4) changes to a time-dependent one:

$$x(k+1) = W(k)x(k), \tag{5}$$

because if the communication between nodes i and j drops for some instant $\hat{k}$, then $w_{ij}(\hat{k}) = 0$. Consequently, the matrix $W(\hat{k})$ is not stochastic anymore, and thus either consensus is lost or, even if consensus is preserved, we might no longer achieve average consensus. To achieve consensus in the sense of (2), a compensation mechanism is needed, modifying the $W(k)$ matrix, as we will do in (6).

Let us consider the case when a packet collision causes a communication loss. This is the case when two nodes try to communicate at the same time, creating interference in the wireless medium, thus making communication impossible in their neighbourhood: this phenomenon suggests a node-based failure model, instead of a link-based one. If such a collision arises, the message broadcasted from one node is lost for every neighbour, and then it can be modelled as the transmitting node failing its communication entirely. The communication loss can then be described by a *failure vector* $f(k)$ such that $f_i(k) = 1$ if the communication from node i to its neighbours is successful during the kth consensus iteration, 0 otherwise. We assume that the failure probability is the same for each agent, and that the failures are independent between each other. The homogeneity in the failure probability does not modify the structure of the compensation method, but it makes it easier to discuss. We denote the failure probability for each node as:

$$\mathbb{P}[f_i(k) = 0] = p \quad \forall i.$$

In the aforementioned failures scenario, it is possible to achieve average consensus if the node i uses its own value, in place of the missing value from a neighbour j that failed its transmission, to update its state, and it does not update its state when it fails its own transmission. The update rule to implement the compensation is then:

$$x_i(k+1) = f_i \left(\sum_{j=1}^n f_j \overline{w}_{ij} x_j(k) + \sum_{j=1}^n (1-f_j)\overline{w}_{ij} x_i(k) \right) + (1-f_i)x_i(k)$$

which reads, after some manipulation:

$$x_i(k+1) = x_i(k) + \sum_{j\in\mathcal{N}_i} f_i(k)f_j(k)\overline{w}_{ij}\left(x_j(k)-x_i(k)\right). \tag{6}$$

In (6), the f_j term accounts for drop in the incoming messages, while the f_i term represents the compensation: the node i will neglect incoming messages when it knows that its last communication was not successful.

When the update rule (6) is used, the dynamics (4) becomes:

$$x(k+1) = W(k)\ x(k), \tag{7}$$

where the matrix $W(k)$ is:

$$W(k) = I + F(k)\overline{W}F(k) - \mathrm{diag}(F(k)\overline{W}F(k)\mathbb{1}), \tag{8}$$

and $\mathrm{diag}(v)$ is a diagonal matrix which entries are the elements of the vector v while $F(k) = \mathrm{diag}(f(k))$.

It is possible to study the convergence properties of the network, subjected to failures, examining the matrix $W(k)$: from the properties of a connected and aperiodic graph, and the stochasticity of the matrix $W(k)$, defined in (8), it is possible to conclude that the network converges almost surely to the average of its agents' initial values, and the convergence speed is related to the eigenvalues of $W(k)$. This compensation method has been studied in [13].

III. A FASTER CONVERGENCE METHOD

Using the update rule (6) a node that experienced a failure will not update itself: such a strategy prevents the node from introducing errors in the network, but when the probability of packet loss is high, several nodes might be in an idle-status, slowing down the network convergence. While those nodes are idle, however, the nodes that are updating might speed up the process, e.g. they could update *twice*, using the information received during the last successful round, hopefully going closer to the average. Using this empiric method leads to a weighted adjacency matrix as follows:

$$W(k) = I + 2F(k)\overline{W}F(k) - 2\,\mathrm{diag}(F(k)\overline{W}F(k)\mathbb{1})$$

Simulations show that this approach actually increases the speed of converge of the network, for high probability of packet loss, but would lead to instability if the probability of packet loss is low. Generalising, it is possible to increase the speed of convergence, using a gain α, modifying the update rule to:

$$x_i(k+1) = x_i(k) + \alpha \sum_{j\in\mathcal{N}_i} f_i(k)f_j(k)\overline{w}_{ij}\left(x_j(k)-x_i(k)\right).$$

and then the associated matrix is:

$$W_\alpha(k) = I + \alpha F(k)\overline{W}F(k) - \alpha\,\mathrm{diag}(F(k)\overline{W}F(k)\mathbb{1}) \tag{9}$$

with $\alpha > 1$.

Modifying the compensation algorithm using the gain α leads to increase in performance but the new $W_\alpha(k)$ matrix is not necessarily stochastic anymore: some of its entries might be negative. However, even if the matrix $W_\alpha(k)$ is not

stochastic for every k anymore, the system converges, faster than the case when $\alpha = 1$.

We will now study the relation between the choice of α and the convergence of the network. To find a condition for α such that the convergence is guaranteed, it is possible to study the error dynamics, defined as:

$$y(k) = x(k) - \frac{1}{n}\mathbb{1}\mathbb{1}^T x(0), \tag{10}$$

where n denotes the number of agents in the network, i.e. the number of vertex in the graph.

From the update rule of $x(k)$ follows that, remembering that $W_\alpha(k)\mathbb{1} = \mathbb{1}$:

$$y(k+1) = x(k+1) - \frac{1}{n}\mathbb{1}\mathbb{1}^T x(0)$$
$$= W_\alpha(k)x(k) - \frac{1}{n}W_\alpha(k)\mathbb{1}\mathbb{1}^T x(0)$$
$$= W_\alpha(k)\left(x(k) - \frac{1}{n}\mathbb{1}\mathbb{1}^T x(0)\right) = W_\alpha(k)y(k).$$

The analysis of the latter equation allows us to derive the following theorem:

Theorem 3.1: A system described by $x(k+1) = W_\alpha(k)x(k)$, where

$$W_\alpha(k) = I + \alpha F(k)\overline{W}F(k) - \alpha \operatorname{diag}(F(k)\overline{W}F(k)\mathbb{1})$$

converges in mean square sense, i.e.: $\lim_{k\to\infty}\mathbb{E}[(y^T(k)y(k)] = 0$, where $y(k)$ is defined in (10), if:

$$\alpha \leq \frac{2\overline{w}_{ij}}{\Xi_{ij}}$$

for each i, j such that $\Xi_{ij} > 0$, where:

$$\Xi_{ij} = 2p\overline{w}_{ij}^2 + 2(1-p)\overline{w}_{ij} - (1-p)[\overline{W}^T\overline{W}]_{ij}. \tag{11}$$
Proof: The expected value of the squared error is

$$\mathbb{E}[y(k+1)^T y(k+1)|y(k)] = y(k)^T \mathbb{E}[W_\alpha(k)^T W_\alpha(k)]y(k),$$

which leads to:

$$\mathbb{E}[y(k+1)^T y(k+1)|y(k)] \leq \lambda_1\left(\mathbb{E}[W_\alpha^T W_\alpha]\right)y^T(k)y(k),$$

where λ_1 denotes the biggest eigenvalue of $\mathbb{E}[W_\alpha^T W_\alpha]$. Recursively it is possible to get:

$$\mathbb{E}[y(k+1)^T y(k+1)] \leq \lambda_1^{(k+1)}\left(\mathbb{E}[W_\alpha^T W_\alpha]\right)\mathbb{E}[y(0)^T y(0)].$$

Then a sufficient condition for $\lim_{k\to\infty}\mathbb{E}[y^T(k)y(k)] = 0$ is that the absolute values of all the eigenvalues of $\mathbb{E}[W_\alpha^T W_\alpha]$ are smaller than one, except one equal to one: this happens if $\mathbb{E}[W_\alpha^T W_\alpha]$ is stochastic and if the graph associated with $\mathbb{E}[W_\alpha^T W_\alpha]$ is connected and aperiodic. The time dependence is dropped, because the matrices $W(k)$ are time independent, according to the failure independence property described in the previous section.

The matrix $\mathbb{E}[W_\alpha^T W_\alpha]$ can be explicitly computed, as in [13]:

$$\mathbb{E}[W_\alpha^T W_\alpha] = \alpha^2(1-p)^3\overline{W}^T\overline{W} -$$
$$2\alpha^2 p(1-p)^2 \overline{W} \odot \overline{W} +$$
$$2\alpha(1-p)^2\overline{W} - 2\alpha^2(1-p)^3\overline{W} +$$
$$(\alpha^2(1-p)^3 - 2\alpha(1-p)^2 + 1)I +$$
$$2\alpha^2 p(1-p)^2 \operatorname{diag}(\overline{W}^T\overline{W}),$$

where $\odot$ denotes the Hadamard – i.e. element-wise – product. From the structure of $\mathbb{E}[W_\alpha^T W_\alpha]$, we can conclude that it is connected and aperiodic if $\overline{W}$ is connected and aperiodic, which is assumed true. By construction, $\mathbb{E}[W_\alpha^T W_\alpha]\mathbb{1} = \mathbb{1}$, and then it is stochastic if $\mathbb{E}[W_\alpha^T W_\alpha]_{ij} \geq 0 \forall i, j$. The diagonal elements of $\mathbb{E}[W^T W]$ are nonnegative:

$$\mathbb{E}[W_\alpha^T W_\alpha]_{i,i} = \mathbb{E}[\sum_{k=1}^{n} w_{\alpha ki}w_{\alpha ki}] = \mathbb{E}[\sum_{k=1}^{n} w_{\alpha ki}^2] \geq 0$$

and then it is sufficient to find α such that the elements outside the diagonal of $\mathbb{E}[W_\alpha^T W_\alpha]$ are nonnegative.

It is possible to evaluate $\mathbb{E}[W_\alpha^T W_\alpha]$ element-wise:

$$\mathbb{E}[W_\alpha^T W_\alpha]_{ij} = \alpha^2(1-p)^3[\overline{W}^T\overline{W}]_{ij} - 2\alpha^2 p(1-p)^2\overline{w}_{ij}^2 +$$
$$2\alpha(1-p)^2\overline{w}_{ij} - 2\alpha^2(1-p)^3\overline{w}_{ij}$$
$$= (1-p)^2\alpha\left(-\alpha\Xi_{ij} + 2\overline{w}_{ij}\right).$$

where Ξ_{ij} is defined in (11).

The condition for the network to converge is then

$$\alpha\left(-\alpha\Xi_{ij} + 2\overline{w}_{ij}\right) \geq 0. \tag{12}$$

If $\Xi_{ij} \leq 0$, then (12) is satisfied, while when $\Xi_{ij} > 0$ then the condition for $\mathbb{E}[W_\alpha^T W_\alpha]_{ij} \geq 0$ becomes:

$$\alpha \leq \frac{2\overline{w}_{ij}}{\Xi_{ij}}, \tag{13}$$

and the proof is completesd. ∎

It should be noticed that Theorem 3.1 can indeed be used to speed up the convergence in the sense that it is possible to find an α bigger than one which satisfies the equation (12), while a value $\alpha < 1$ would result in a lower convergence speed. Proving the existence of $\alpha > 1$ is equivalent to prove the following:

Lemma 3.2: If $0 < \overline{w}_{ij} < 1$ and $\Xi_{ij} > 0$ then:

$$\frac{2\overline{w}_{ij}}{2p\overline{w}_{ij}^2 + 2(1-p)\overline{w}_{ij} - (1-p)[\overline{W}^T\overline{W}]_{ij}} > 1. \tag{14}$$
Proof: Let us assume that

$$\alpha\left(-2p\overline{w}_{ij}^2 - 2(1-p)\overline{w}_{ij} + (1-p)[\overline{W}^T\overline{W}]_{ij}\right) + 2\overline{w}_{ij} > 0$$

and $0 < \overline{w}_{ij} < 1$. When those assumptions are not true, there is no need to study the sign of $\mathbb{E}[\overline{W}^T\overline{W}]$.

Then assume there exists $\left(\overline{w}_{ij}, [\overline{W}^T\overline{W}]_{ij}\right)$ such that:

$$\frac{2\overline{w}_{ij}}{2p\overline{w}_{ij}^2 + 2(1-p)\overline{w}_{ij} - (1-p)[\overline{W}^T\overline{W}]_{ij}} \leq 1.$$

The latter assumption implies

$$\frac{2\overline{w}_{ij}}{2p\overline{w}_{ij}^2 + 2(1-p)\overline{w}_{ij} - (1-p)[\overline{W}^T\overline{W}]_{ij}} \leq 1 \Rightarrow$$

$$\frac{2\overline{w}_{ij} - 2p\overline{w}_{ij}^2 - 2(1-p)\overline{w}_{ij} + (1-p)[\overline{W}^T\overline{W}]_{ij}}{2p\overline{w}_{ij}^2 + 2(1-p)\overline{w}_{ij} - (1-p)[\overline{W}^T\overline{W}]_{ij}} \leq 0 \Rightarrow$$

$$2p(\overline{w}_{ij} - \overline{w}_{ij}^2) + (1-p)[\overline{W}^T\overline{W}]_{ij} \leq 0$$

which is a contradiction, because $0 < \overline{w}_{ij} < 1$, and therefore (14) is true. ∎

Getting a bound for α allows for an *a priori* tuning of the parameter, to ensure convergence when a stochastic modelling of the network is known. It is possible to pick a *safe* α_s, which satisfies the inequalities (12), i.e.:

$$\alpha_s = \min_{ij \text{ s.t.} \Xi_{ij} > 0} \left(\frac{2\overline{w}_{ij}}{2p\overline{w}_{ij}^2 + 2(1-p)\overline{w}_{ij} - (1-p)[\overline{W}^T\overline{W}]_{ij}} \right).$$

An α satisfying the aforementioned set of inequalities, will secure convergence of the network; moreover the network will converge to the average of the nodes' initial conditions: this can be derived from the double stochasticity of the $\mathbb{E}[W_\alpha^T W_\alpha]$ matrix. However, the choice of the parameter is not unique: it is possible to pick bigger values for α, which might make the network to converge faster, but the network might instead reach instability, if α exceeds certain – unknown – threshold. Intuitively, the gain should be one when the probability of packet loss is equal to 0, and it should increase when the probability of packet loss increases. Moreover, it might depend on the size of the network: small network will experience a lower amount of packet losses, simply because the number of links in the network is lower, thus a smaller gain should be employed, because the number of idled nodes will be lower. From those intuitive assumptions, it is possible to choose α empirically as:

$$\alpha_h = \frac{1}{(1-p) + \frac{p}{n}}. \tag{15}$$

Simulations – presented in the next section – suggest that this value makes the network converge and that convergence – in some scenarios – is faster than using α_s, even though convergence is not guaranteed by our theoretical results.

IV. SIMULATIONS

In the previous section a bound for α has been found, to assure network convergence. Our intuition is that the use of a gain will increase the network convergence's speed. While it is possible to find a value for α which assures convergence, the number of iteration required to converge can not be computed beforehand, unless we restrict to some special case: the speed of converge is related to the eigenvalues of the matrix $\mathbb{E}[W_\alpha^T W_\alpha]$, which are generally not known in a closed-form. Therefore, to evaluate the performance in different scenarios, we run a set of simulations.

As mentioned in the previous section, the choice of α is not unique: it is interesting then to check to which degree the convergence speed can be increased, varying the choice

of α. To provide a meaningful set of results, the simulations are performed varying several factors:

- the probability of packet loss, p;
- the size of the network, n;
- the "connectivity probability", m, defined below.

The networks are generated randomly, without any given structure. This is done to keep generality in the network: if there is some structure in the network, it might be possible to exploit it, to have a better understanding of the eigenvalues – and thus of the convergence – of the network. The parameter n is the number of nodes – representing the agents – of the network. The parameter m defines the connectivity of the network: for each pair of nodes, there is a connection between them with probability m. A higher value for m implies a larger number of connections in the network. Lastly, during each consensus round, a node transmits a packet to its neighbours with a probability equal to $1 - p$.
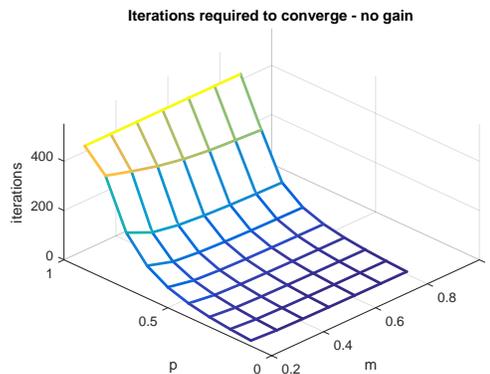

Fig. 1. Iterations required to converge, for a network of 81 nodes, without gain, i.e. $\alpha = 1$.

We are primarily interested in the performance difference when a gain is employed, and secondarily we compare different choices of the gain α. Therefore, the simulations are conducted using three different values for α:

- $\alpha = 1$, representing the case when the gain is not used;
- α_s, which – according to the Theorem 3.1 – assures convergence;
- α_h, an heuristic value for α, which we expect to increase the velocity of converge, without proof over the stability of the system – defined in (15).

To increase readability, the number of iteration required to converge is shown over only two parameters: p and m, because the dependence of the number of iterations required to converge on the size of the network is negligible in comparison with the dependence on the probability of packet loss and connectivity of the network. More specifically, for each triple (p, m, n), 145 simulations are performed and only the averages are displayed, but all the 145 simulation results lie in an interval not bigger than 10% of the displayed values.

The number of iterations required to converge, when $\alpha = 1$, are depicted in Figure 1. For high probability of
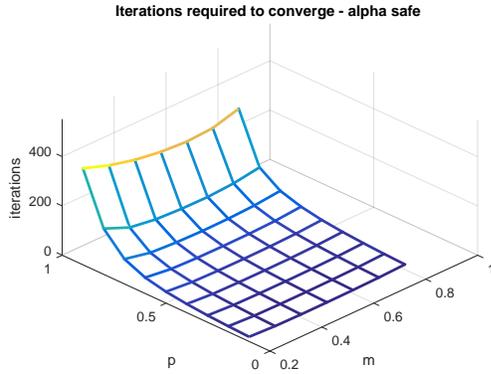
Fig. 2. Iterations required to converge, for a network of 81 nodes, using the safe gain α_s

packet loss, the number of iterations required to converge hits the limit of 500 – which was the upper limit imposed in the simulation – but we can infer that the actual number of iteration required would be bigger. We can notice two different trends: the number of iterations required decreases as the probability of packet loss decreases, and as the connectivity of the network increases. This result is expected and intuitive. Figure 2 displays the number of iterations required to converge when the gain is employed.
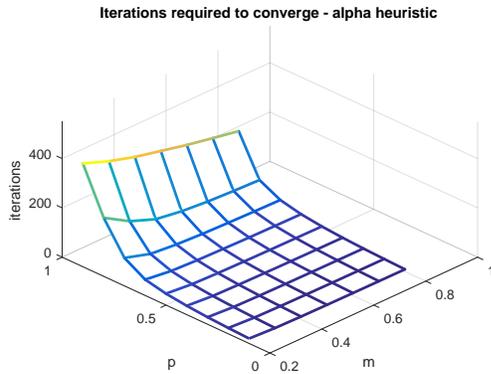

Fig. 3. Iterations required to converge, for a network of 81 nodes, using the heuristic gain α_h

First, we can notice how even for high probability of packet loss, the algorithm converges in less than 500 iterations when a gain is employed. Moreover, the curve is always below the one in Figure 1: using the gain does actually increase the convergence speed. However, from Figure 3, we can notice a unexpected behaviour in the number of iterations required for the two gain method to converge: when the gain is α_s, for high failure probability and high matrix connectivity the number of iterations increases, while this phenomenon is not present when the gain is α_h. This phenomenon may arise from the fact that α_s must

satisfy more conditions as m rises, namely $\mathcal{O}\left(m\frac{n(n-1)}{2}\right)$, which makes it more conservative and might lead to worse performance when the network connectivity is high.

In Figure 4 it is possible to appreciate a better comparison between the choice of α_s or α_h: this plot is obtained averaging all the simulations, to show only the dependence of the number of iterations on the probability of packet loss – which is the principal cause of slower convergence. From the graph, it is evident that the two choices are substantially equivalent: the heuristic value of α leads to slightly better performance than the safe one for lower probability of packet loss.
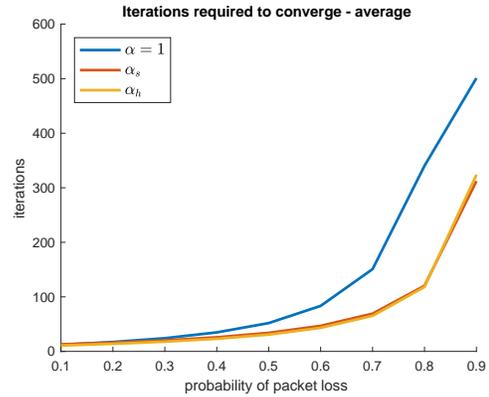

Fig. 4. Iterations required to converge for different values of α, average over different network sizes n and connectivity degrees m
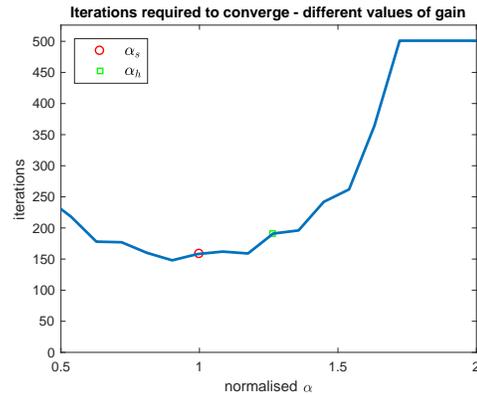

Fig. 5. Iterations required to converge for different values of $\frac{\alpha}{\alpha_s}$.

The α which satisfies the bound in (12) is not the optimal one: it is possible to find different values of gain to speed up the convergence of the network. However, it is interesting to evaluate how good the value of α_s is, and to check wether this choice is limiting or not. In figure 5 is depicted the number of iteration required for a network to converge, for different values of α. The values are normalised over α_s, in a range from $0.5\alpha_s$ to $3\alpha_s$, to make the plot clearer. Choosing the safe value – circled in red – is indeed not a limiting

factor: it would be possible to pick a different gain value to converge faster, but the increase in performance is not dramatic. Moreover, the value of α that achieves the fastest convergence is not known a priori, and it can be only be obtained through trial-and-error, while α_s can be computed beforehand.

Lastly, it is interesting to compare our novel algorithm with another from the literature, the Hadjicostis algorithm presented in [11], chosen due to its simplicity of implementation and its good performance. To simplify the latter algorithm implementation, a circulant topology has been chosen: the nodes are organised in a ring, and each node can communicate with a symmetric set of neighbours, $\frac{k-1}{2}$ in the clockwise direction and $\frac{k-1}{2}$ in the counter-clockwise direction in the ring. This ensures a topology where each node has the same connectivity, and the knowledge for each node of its connectivity degree is required by the Hadjicostis algorithm. Those results are obtained for a network of 80 elements, with the aforementioned circulant topology: a node i is connected with k nodes, from $i - \frac{k}{2}$ to $i + \frac{k}{2}$. The matrix representing this topology has as first row:

$$\overline{W}_{1,j} = \begin{cases} \dfrac{1}{m} & \text{if } j \leq \frac{m-1}{2} \vee j > n - \frac{k-1}{2} \\ 0 & \text{otherwise} \end{cases}, \qquad (16)$$

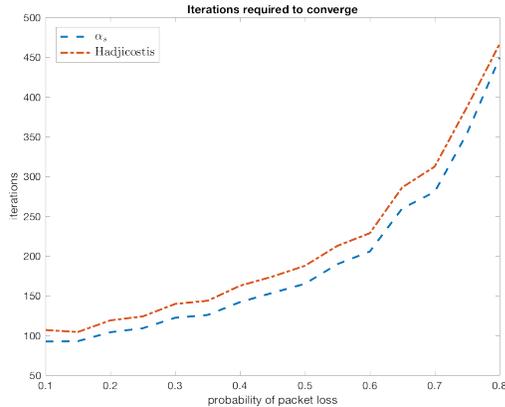and all the other rows are obtained shifting the first one.

Fig. 6. Iterations required to converge for the algorithm in [11] and the α-gain one.

The results of this comparison are presented in Figure 6. It is interesting to notice how the two curves are similar, but the novel algorithm is faster. However, a more extended set of simulation must be conducted, in order to derive more general conclusions.

V. CONCLUSION AND FUTURE WORK

We have shown that it is possible to converge to the average in presence of packet loss, and the convergence can be accelerated using a gain. However, the presence of such gain in the dynamics of the network, might lead to instability: a value for such a gain has been found, that ensures stability of the network, and increases its performance. The value of α_s is found such that $\mathbb{E}[W_{\alpha_s}^T W_{\alpha_s}]$ is stochastic, which implies that all its eigenvalues are less or equal than one. As future work, it is possible to investigate whether there are other conditions, less restrictive than stochasticity, for $\mathbb{E}[W_\alpha^T W_\alpha]$ to be a contraction. This would allow for a broader range of stabilising α, possibly allowing faster convergence. An interesting question is the rigorous study of the heuristic value α_h, which apparently preserves the stability of the system, and can achieve faster convergence. However, the safe α_s is close to the optimal one – found *a posteriori* for specific instances of the problem.

Moreover, the analysis of performance and the design of the gain are made without taking into account the differences in the nodes: a value of α is found, and it is the same for each node. It would be possible to find a set of different gains, one for each node, stabilising the network. This would relax the Theorem 3.1, allowing for faster convergence with guaranteed stability. Finally we observe that our analysis is done for general networks. The presence of a structure in the network topology might allow for better performance: if the eigenvalues computation is feasible, a tighter bound for α can be found.

REFERENCES

[1] M. Alanyali, S. Venkatesh, O. Savas, and S. Aeron, "Distributed Bayesian hypothesis testing in sensor networks," in *American Control Conference, 2004*, vol. 6, June 2004, pp. 5369–5374.

[2] D. Blatt and A. Hero, "Distributed maximum likelihood estimation for sensor networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004 (ICASSP '04).*, vol. 3, May 2004, pp. 929–932.

[3] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Decision and Control, 2007 46th IEEE Conference on*, Dec 2007, pp. 5492–5498.

[4] S. Kar and J. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *Signal Processing, IEEE Transactions on*, vol. 57, no. 1, pp. 355–369, Jan 2009.

[5] T. Li and J.-F. Zhang, "Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises," *Automatic Control, IEEE Transactions on*, vol. 55, no. 9, pp. 2043–2057, Sept 2010.

[6] U. Schmid and C. Fetzer, "Randomized asynchronous consensus with imperfect communications," in *Reliable Distributed Systems, 2003. Proceedings. 22nd International Symposium on*, Oct 2003, pp. 361–370.

[7] F. Fagnani and S. Zampieri, "Average consensus with packet drop communication," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 102–133, 2009.

[8] Y. Chen, R. Tron, A. Terzis, and R. Vidal, "Corrective consensus: Converging to the exact average," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, Dec 2010, pp. 1221–1228.

[9] K. Cai and H. Ishii, "Average consensus on general strongly connected digraphs," *Automatica*, vol. 48, no. 11, pp. 2750 – 2761, 2012.

[10] N. Vaidya, C. Hadjicostis, and A. Dominguez-Garcia, "Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec 2012, pp. 2761–2766.

[11] C. N. Hadjicostis, N. H. Vaidya, and A. D. Domínguez-García, "Robust distributed average consensus via exchange of running sums," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1492–1507, June 2016.

[12] S. Wang, C. An, X.-X. Sun, and X. Du, "Average consensus over communication channels with uniform packet losses," in *Control and Decision Conference (CCDC), 2010 Chinese*, May 2010, pp. 114–119.

[13] F. Acciani, G. Heijenk, and P. Frasca, "Achieving robust average consensus over wireless networks," in *2016 European Control Conference (ECC)*, June 2016, pp. 555–560.