Distributed averaging on digital noisy networks

Ruggero Carli*, Giacomo Como†, Paolo Frasca‡, and Federica Garin§

*DEI, Università di Padova, Padova, Italy. Email: carlirug@dei.unipd.it
†LIDS, Massachusetts Institute of Technology, Cambridge, MA. Email: giacomo@mit.edu
‡Dip. di Matematica, Politecnico di Torino, Torino, Italy. Email: paolo.frasca@polito.it
§NeCS, INRIA Rhône-Alpes, Grenoble, France. Email: federica.garin@inrialpes.fr

***Abstract*—We consider a class of distributed algorithms for computing arithmetic averages (average consensus) over networks of agents connected through digital noisy broadcast channels. Our algorithms combine error-correcting codes with the classical linear consensus iterative algorithm, and do not require the agents to have knowledge of the global network structure. We improve the performance by introducing in the state-upadate a compensation for the quantization error, avoiding its accumulation. We prove almost sure convergence to state agreement, and we discuss the speed of convergence and the distance between the asymptotic value and the average of the initial values.**

I. Introduction

The problem of distributed computation of averages has been attracting much interest in the last decades. Indeed, this problem is widely regarded as a significant case study, in order to understand the role of communication constraints in more general problems of distributed computation, in which a network of communicating processors/agents is asked to collectively approximate a certain function of their initial states. In the present paper, we shall consider the adaptation of the well-known linear averaging algorithms to a network of digital noisy links.

Related works

The implementation of the classical linear averaging algorithms relies on the ability to communicate real numbers between agents in an instantaneous and reliable way. As this assumption is clearly not met by digital communication, the community has been studying how to adapt such algorithms to realistic digital networks. A few papers have considered the issue of quantization, i.e., limited precision in communications, and provided results on the precision which can be achieved in approximating the average when using static uniform quantizers [9], [1], or designed effective quantization schemes to achieve arbitrary precision (based on adaptive [14] or logarithmic [3] quantizers). On the other hand, in the literature we find mainly two approaches to the issue of noisy communication. First, researchers have considered the possibility of packet drops or link failures, providing robustness results; e.g. [13], [8]. Second, noisy communication has been modeled as the superposition of an additive noise to the communicated value. This dynamics, first studied in [19], clearly implies the accumulation of errors, which forbids to approximate the average with arbitrary precision. For this reason, several papers have proposed strategies to overcome this drawback. The proposed solutions usually require the dynamics to be dynamically adjusted, by decreasing the so-called "consensus gains" [7], [10], [11], [12], [15], [17], [18]. This choice leads to a dynamics which can be studied by stochastic approximation techniques [2]. However, the communication complexity of such algorithms is polynomial in the desired precision.

Starting from this background literature, in [4] we have studied the problem of computing averages by a network connected by noisy digital channels, and we have proposed an algorithm whose communication complexity is only polylogarithmic in the desired precision. In the proposed algorithm, the agents' states are communicated using a joint source and channel coding [5] during communication phases of increasing length. The latter feature allows the scheme, thus named "Increasing Precision Algorithm", to avoid the accumulation of errors.

Contribution

In this note we consider the problem of averaging on a network of noisy digital communication channels, under the mathematical formalization of [4]. While the algorithm in the latter paper is based on joint channel and source coding, in the present work we decompose quantization and communication errors, in such a way to exploit the inherent feedback information which is available about the quantization error. As such error can be effectively compensated at the network level, we are able to present a modification of the Increasing Precision Algorithm, with improved performance in terms of communication complexity.

Paper structure

The paper is orgnized as follows. After this introduction, we describe the problem setting in Sect. II-A, and we briefly review the Increasing Precision Algorithm (IPA) in Sect. II-B and the relevant criteria for performance analysis in Sect. II-C, referring the reader to [4] for more details.

In Sect. III we show how the performance of IPA algorithm can be improved by compensating the error due to quantization; in particular, in Sect III-A we introduce the Increasing Precision Algorithm with Partial Compensation (PC-IPA) and in Sect III-B we theoretically analyze its performance.

Finally, in Sect. IV we present some simulation results and a discussion on how the complexity of the PC-IPA algorithm depends on the topology of the network.

Definitions and notation

The following definitions and notation will be used throughout the paper. We denote by $\mathbb{N}$, $\mathbb{Z}^+$, and $\mathbb{R}$, respectively, the sets of naturals, nonnegative integers, and real numbers. The transpose of a vector $\boldsymbol{v} \in \mathbb{R}^n$ and a matrix $M \in \mathbb{R}^{n\times n}$, are denoted by $\boldsymbol{v}^*$ and M^*, respectively. We will denote by $\mathrm{tr}(M)$ the trace of M and by $\|\cdot\|_{\mathrm{F}}$ the Frobenius norm, i.e., $\|M\|_{\mathrm{F}} = \sqrt{\mathrm{tr}(M^*M)}$. Given two matrices M, M', we denote by $M \odot M'$ their entrywise (Hadamard) product. With the symbol $\mathbf{1}$ we denote the n-dimensional vector all of whose entries equal 1. A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the pair of a finite vertex set $\mathcal{V}$ and a set $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ of directed edges. For a vertex $v \in \mathcal{V}$, we denote by $\mathcal{N}_v^+ := \{w \in \mathcal{V} : (v, w) \in \mathcal{E}, w \neq v\}$, and $\mathcal{N}_v^- := \{w \in \mathcal{V} : (w, v) \in \mathcal{E}, w \neq v\}$, respectively, the sets of its out- and in-neighbors. A matrix $P \in \mathbb{R}^{n\times n}$ is *doubly stochastic* if it has non-negative entries, and all its rows and columns sum to one. Its induced graph $\mathcal{G}_P$ has n vertices, and there is an edge (u, v) if and only if $P_{uv} > 0$. P is adapted to a graph $\mathcal{G}$ if $\mathcal{G}_P$ is a subgraph of $\mathcal{G}$. P is primitive if $\mathcal{G}_P$ is strongly connected (i.e., any two nodes u, v are connected by a directed path) and aperiodic (i.e., the greatest common divisor of cycle lengths is 1).

II. Problem setting and proposed algorithm

A. Problem statement

We consider a finite set of agents $\mathcal{V}$ of cardinality n and assume that each agent $v \in \mathcal{V}$ has initially access to some partial information consisting in the observation of a scalar value θ_v, which may for instance represent a noisy measurement of some physical quantity of interest. The full vector of observations is denoted by $\boldsymbol{\theta} = (\theta_v)_{v\in\mathcal{V}}$. We consider the case when all θ_v's take values in the same bounded interval $\Theta \subseteq \mathbb{R}$; without loss of generality, we shall assume $\Theta = [0, 1]$. For the network, the goal is to compute the average of such values,

$$y := \frac{1}{n}\sum\nolimits_{v\in\mathcal{V}} \theta_v$$

through repeated exchanges of information among the agents and without a centralized computing system.

Communication among the agents takes place as follows. At each time instant $t = 1, 2, \ldots$, every agent v broadcasts a bit $a_v(t) \in \{0, 1\}$ to a subset of agents, which will be denoted by $\mathcal{N}_v^+ \subset V$ and will be called the set of out-neighbors of v. We assume that, for all $w \in \mathcal{N}_v^+$, between v and w there is a binary-input memoryless channel (with non-zero capacity) with output alphabet $\mathscr{A}$: agent w will receive a possibly corrupted version $b_{v\to w}(t) \in \mathscr{A}$ of the bit $a_v(t)$ which was sent by v. Notice that we do not need any assumption of mutual independence of the received signals $b_{v\to w}(t)$ for the different neighbors w. However, we do need the assumption that the channels are memoryless with respect to time, in other words, given $a_v(t)$, the received signals $b_{v\to w}(t)$ are conditionally independent from $\boldsymbol{\theta}$ and from $\{a_v(s), b_{v\to w}(s)\, \forall\, 1 \le s < t, v, w \in \mathcal{V}\}$.

For simplicity, we shall assume that all the channels (for all $v \in \mathcal{V}$, $w \in \mathcal{N}_v^+$ and $t \in \mathbb{N}$) have the same output alphabet $\mathscr{A}$ and the same transition probabilities.

The communication setting outlined above can be conveniently described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (the communication graph), whose vertices are the agents, and such that an ordered pair (u, v) with $u \neq v$ belongs to $\mathcal{E}$ if and only if $v \in \mathcal{N}_u^+$, i.e., if u transmits to v through a discrete memoryless channel with non-zero capacity. Throughout the paper, we shall assume that the graph $\mathcal{G}$ is strongly connected. We will also assume that $\mathcal{G}$ has self-loops on each vertex; this represents the fact that each node has access to its own information, which is equivalent to assume a noiseless channel available from u to itself. The presence of self-loops ensures that $\mathcal{G}$ is aperiodic.

In this setting, an algorithm is *distributed* if, for all transmission time t and for all agent v, the bit transmitted by v at time t is a function only of the information available to agent v at time $t-1$, and the estimate $\hat{y}_v(t)$ of y that agent v computes at time t is a function only of the information available to agent v at time t. In the setting where the communication channel does not provide a feedback on the transmission, the information which is available to v at time t consists of all the bits received so far from its in-neighbors, and of course v's initial condition θ_v.

B. Increasing precision algorithm (IPA)

The idea we introduced in [4] is to use a traditional linear average-consensus algorithm, combined with suitable coding schemes.

The ingredients of our algorithm are

- a consensus matrix P, i.e., a doubly-stochastic primitive matrix adapted to the communication graph $\mathcal{G}$, with non-zero diagonal entries;
- an increasing sequence of positive integers $\{\ell(k)\}_{k\in\mathbb{N}}$, such that $\lim_{k\to\infty} \ell(k) = +\infty$; $\ell(k)$ represents the number of bits that each node transmits at k-th iteration of the consensus algorithm;
- a sequence of encoders, i.e., maps
$$\phi^{(k)} : [0, 1] \to \{0, 1\}^{\ell(k)} ;$$
- a sequence of decoders, i.e., maps
$$\psi^{(k)} : \mathscr{A}^{\ell(k)} \to [0, 1] .$$

Let us recall the classical iterative linear average-consensus algorithm:

$$\boldsymbol{x}(0) = \boldsymbol{\theta}\,, \qquad \boldsymbol{x}(k) = P\boldsymbol{x}(k-1)\,, \quad k \ge 1\,,$$

i.e., at k-th iteration, node u receives from its in-neighbors the numbers $x_v(k)$, $v \in \mathcal{N}_u^-$, and updates its state:

$$x_u(k) = P_{uu}x_u(k-1) + \sum_{v\in\mathcal{N}_u^-} P_{uv}x_v(k-1)\,.$$

It is well-known (Perron-Frobenius theory) that, for primitive doubly-stochastic P, from any initial condition $\boldsymbol{\theta}$, each entry of $\boldsymbol{x}(k)$ converges to y.

We propose to adapt this algorithm, in a way that takes into account the necessity to transmit the real values $x_u(k)$ along digital noisy channels. The initialization of the algorithm is unchanged: $\boldsymbol{x}(0) = \boldsymbol{\theta}$. Between iterations $k-1$ and k of our consensus-like algorithm, we allow each node v to use $\ell(k)$ bits to encode its state $x_v(k-1)$ by the use of a (joint) source and channel encoder $\phi^{(k)}$; the codewords will be broadcasted to neighbors, and then the decoded messages will be used as estimates of the neighbors' states for the consensus-like state-update.

More precisely, with the definitions of $h_0 = 0$ and $h_k := \sum_{r\leq k} \ell(r)$, the algorithm is the following. Initialize $\boldsymbol{x}(0) = \boldsymbol{\theta}$ and then iterate for all $k \in \mathbb{N}$ the following steps:

- **Transmission phase.** Each node v broadcasts to its neighbors the encoded version of its state $x_v(k-1)$. Namely, for all $t \in (h_{k-1}, h_k]$, it transmits the bit $a_v^{(t-h_{k-1})}(k)$ where $\left(a_v^{(1)}(k), \ldots, a_v^{(\ell(k))}(k)\right) = \phi^{(k)}(x_v(k-1)) \in \{0,1\}^{\ell(k)}$.
- **Decoding and state update.** At the end of the k-th communication phase, i.e., at time $t = h_k$, each node u decodes the messages it has received from its out-neighbors, finding estimates of their states: for all $v \in \mathcal{N}_u^-$, u receives $\left(b_{v\to u}^{(1)}(k), \ldots, b_{v\to u}^{(\ell(k))}(k)\right) \in \mathscr{A}^{\ell(k)}$ and it computes $\hat{x}_v^{(u)}(k-1) := \psi^{(k)}\left(\left(b_{v\to u}^{(1)}(k), \ldots, b_{v\to u}^{(\ell(k))}(k)\right)\right) \in [0,1]$. Then, u updates its own state according to the following consensus-like step:
$$x_u(k) = P_{uu} x_u(k-1) + \sum_{v\in\mathcal{N}_u^-} P_{uv} \hat{x}_v^{(u)}(k-1)\,. \quad (1)$$

During each phase, the estimate $\hat{\boldsymbol{y}}(t)$ remains constant, and is equal to the current state:
$$\hat{\boldsymbol{y}}(t) = \boldsymbol{x}(k-1)\,, \qquad \forall t \in [h_{k-1}, h_k)\,.$$

Notice that the encoders $\phi^{(k)}$ and $\psi^{(k)}$ are by definition source and channel encoders/decoders (jointly performing the two tasks, or cascading two suitable codings). In principle, they could be different at each transmitting-receiving pair, but we assume that they are all the same, for simplicity and also to ensure anonymity to agents and reconfigurability to the network. Moreover, in principle we need to define a different encoder/decoder pair for each different k, but it is useful to consider encoders and decoders which allow to exploit a common coding scheme at various lengths, see [5].

Another detail of the algorithm that needs to be further specified is the choice of the phase lengths $\{\ell(k)\}_{k\in\mathbb{N}}$. The actual choice will be done in a different way for different choice of the encoders and decoders, in order to optimize the convergence speed. However, an important general remark is that, in order to avoid accumulation of errors and ensure convergence of the algorithm, we need to choose increasingly long phase lengths.

In fact, we can write $\hat{x}_u^{(v)}(k-1) = x_u(k-1) + \delta_{u\to v}(k)$, and we may think of $\delta_{u\to v}(k)$ as a noise, which is in part due to the quantization of the real-valued $x_u(k-1)$ and in part to the channel noise during the $\ell(k)$ transmissions of the k-th transmission phase, where the effect of the channel has already been mitigated by some error-correction. Notice that $\delta_{u\to v}(k)$ in general does not have zero mean, and depends on $x_u(t)$ (and thus depends on all past noises). A suitable choice of the encoder/decoder pairs and of the transmission phases allows one to obtain a noise decreasing with respect to k, with a speed which is discussed in [4]. To this effect, the assumption that the transmission lengths $\ell(k)$ are increasing in k is essential, because the coding gain is asymptotic in the length of codewords. This remark leads us to name our algorithm 'Increasing Precision Algorithm' (IPA).

C. Performance criteria

The performance of an algorithm solving the problem described in Sect. II-A will be evaluated by studying the error
$$\boldsymbol{e}(t) := \hat{\boldsymbol{y}}(t) - y\mathbf{1}\,,$$
i.e., the distance between the current local estimates and the correct average. It is often useful to decompose this error into two orthogonal parts:
$$\boldsymbol{e}(t) = \boldsymbol{z}(t) + \zeta(t)\mathbf{1}\,,$$
where:

- $\boldsymbol{z}(t) := \hat{\boldsymbol{y}}(t) - \frac{1}{n}\mathbf{1}^*\hat{\boldsymbol{y}}(t)\mathbf{1}$ describes how far the various agents are from reaching an agreement. Notice that $\boldsymbol{z}(t)$ is orthogonal to $\mathbf{1}$.
- $\zeta(t) := \frac{1}{n}\mathbf{1}^*\hat{\boldsymbol{y}}(t) - y$ accounts for the distance between the current average of the estimates and the correct value y.

The question we ask is: how much time is required to reach a given precision $\delta \in (0,1]$? Namely, we will study the *communication complexity*, defined as follows:
$$\tau(\delta) := \inf\left\{t \in \mathbb{N} :\ \forall s \geq t,\ \tfrac{1}{n}\mathrm{E}[\|\boldsymbol{e}(s)\|^2] \leq \delta\right\}.$$

Here, we are considering as 'time' the number of transmitted bits. It is a fair notion of time in the case where the time needed for the transmission is considerably larger that the computation time, or in the case where we are not really interested in time but rather in the energy consumption due to the transmissions. Otherwise, it is more sensible to consider a different notion of complexity, which takes into account both transmission and computation. Such a notion of transmission/computation complexity, allowing a fair comparison among algorithm with very different computational complexity (particularly in the decoders) is defined and discussed in [4]. For example, in the IPA algorithm one might want to compare the use of different encoder/decoder pairs, exploring different correction performance vs. complexity tradeoffs: some random-tree coding (à la Forney) can ensure faster decrease of the error at the price of a higher complexity, as opposed to simple repetition-like or fountain-like codes with linear-time decoding. In particular, on the binary erasure channel (where even random-tree codes have a polynomial-time computational complexity) it is not obvious a priori which choice can ensure faster convergence. On other channels, however, it is clear that the exponential complexity of decoding such schemes is unaffordable.

In this paper, we focus on the case where the computational complexity is very low: we will consider encoders and decoders with linear complexity w.r.t. the number of transmitted bits. For this reason, we will consider as 'time' the number of transmitted bits, the actual time being the same up to a multiplicative constant.

III. Improving the algorithm by (partial) compensation of the errors

A. Partial error compensation

In [4], an idealized setting in which the agents have access to noiseless feedback about the signals received by their out-neighbors was also considered. In this case, at the end of the $(k-1)$-th transmission phase, each agent $u \in \mathcal{V}$ can use the feedback information in order to compute the corrupted estimate of its state $\hat{x}_u^{(v)}(k-1)$, which each out-neighbor $v \in \mathcal{N}_u^+$ will use in its state-update. Thus, u can compensate this error, by using the following modified state-update equation:

$$x_u(k) = x_u(k-1) - \sum_{v\in\mathcal{N}_u^+} P_{vu}\hat{x}_u^{(v)}(k-1) + \sum_{v\in\mathcal{N}_u^-} P_{uv}\hat{x}_v^{(u)}(k-1) \tag{2}$$

instead of Eq. (1). This modification ensures that the average is preserved along the iterations, i.e., $\frac{1}{n}\mathbf{1}^*\boldsymbol{x}(k) = y$ for all $k \geq 0$. In fact, this modification can be shown to significantly improve the performance. However, it is implementable only in the case where the channels provide perfect feedback, which is a strong and often unrealistic assumption, particularly in the considered broadcast setting. We shall refer to such idealized implementation of the IPA algorithm, with the state-update following Eq. (2) instead of Eq. (1), as to the Increasing Precision Algorithm with Compensation (C-IPA).

On the other hand, a simple but fundamental remark is that, even in the more realistic case where there is no feedback available from the channels, there is still one part of the error $\delta_{u\to v}(k)$ that depends on the quantization of $x_u(k-1)$ and not on the channel, and thus it is perfectly known by agent u. Indeed, one can choose an encoder $\phi^{(k)}$ that is the composition of two separate encoders: a quantizer $\mathcal{Q}^{(k)} : [0,1] \to \{0,1\}^{q(k)}$ and an encoder which adds some redundancy for error correction $\mathcal{C}^{(k)} : \{0,1\}^{q(k)} \to \{0,1\}^{\ell(k)}$. We shall also use the notation $\tilde{\mathcal{Q}}$ to denote the same quantizer as above, but where the output is seen as a rational number instead of as a string of bits, i.e., $\tilde{\mathcal{Q}}(x) := \mathcal{R} \circ \mathcal{Q}$, where $\mathcal{R}((r_1,\ldots,r_q)) := \sum_{j=1}^q r_j 2^{-j}$. With this notation, we can decompose the error that node u makes when it transmits to a neighbor v during the k-th transmission phase into two parts: $\delta_{u\to v}(k) = \nu_u(k) + \delta'_{u\to v}(k)$, where:

- $\nu_u(k) = \tilde{\mathcal{Q}}(x_u(k-1)) - x_u(k-1)$ is the quantization error;
- $\delta'_{u\to v}(k) = \hat{x}_u^{(v)}(k-1) - \tilde{\mathcal{Q}}(x_u(k-1))$ is the error due to the channel (already mitigated by error correction).

The key remark is that, even if the channel does not provide any feedback (agent u does not know $\delta_{u\to v}(k)$), at least the quantization error is known: agent u knows $\nu_u(k)$ and can apply a compensation rule for this part of the error. Clearly, the error being compensated only partially, the algorithm will not be able to preserve the average along its iterations; however, this partial compensation will be enough to improve the performance with respect to the basic IPA presented in Sect. II-B. These results are new, and will be presented here with their proofs.

The modified state-update that we propose is the following:

$$x_u(k) = x_u(k-1) - (1-P_{uu})\tilde{\mathcal{Q}}(x_u(k-1)) + \sum_{v\in\mathcal{N}_u^-} P_{uv}\hat{x}_v^{(u)}(k-1). \tag{3}$$

With the notation $\boldsymbol{\nu}(k) = (\nu_u(k))_{u\in\mathcal{V}}$ for the vector of quantization errors and $\Delta'(k)$ for the matrix defined by $\Delta'(k)_{uv} = \delta'_{v\to u}(k)$ if $v \in \mathcal{N}_u^-$, and 0 otherwise, the system evolution can be described as follows:

$$\boldsymbol{x}(k) = P\boldsymbol{x}(k-1) + (P-I)\boldsymbol{\nu}(k) + (P \odot \Delta'(k))\mathbf{1}\,. \tag{4}$$

We shall refer to the IPA where the state update is done according to Eq. (3) as to the Increasing Precision Algorithm with Partial Compensation (PC-IPA). In the following subsection, we shall present our main results concerning the performance of the PC-IPA.

B. Performance of PC-IPA

Before stating our main results on performance of PC-IPA, we describe in detail a particular choice of the quantizers and of the error-correcting codes that we will use, although the choice of such specific encoders is not essential, similar results can be obtained for other coding schemes.

We will consider the probabilistic quantizer, defined as follows:

$$\tilde{\mathcal{Q}}^{(k)}(x) = \begin{cases} \frac{1}{2^{q(k)}}\left\lfloor 2^{q(k)}x\right\rfloor & \text{with prob. } \left\lceil 2^{q(k)}x\right\rceil - 2^{q(k)}x \\ \frac{1}{2^{q(k)}}\left\lceil 2^{q(k)}x\right\rceil & \text{with prob. } 2^{q(k)}x - \left\lfloor 2^{q(k)}x\right\rfloor. \end{cases}$$

We will assume that, for all $u \in \mathcal{V}$ and $k \in \mathbb{Z}^+$, given $x_u(k-1)$, $\tilde{\mathcal{Q}}^{(k)}(x_u(k-1))$ is conditionally independent from all the past and from all the neighbors' states and quantizers. A simple way to implement such quantizer is to add an independent random dither to $x_u(k-1)$ and quantize the sum by rounding [1].

As a channel encoder $\mathcal{C}^{(k)} : \{0,1\}^{q(k)} \to \{0,1\}^{\ell(k)}$, we shall use a simple repetition encoder, whereby each bit is repeated $\ell(k)/q(k)$ times. As a decoder, we shall perform a binary hypothesis test for each transmitted bit. Along this paper, we will choose $\ell(k) = Skq(k)$. Indeed, a number of repetitions of each bit (at k-th transmission round) increasing linearly with k allows to obtain transmission errors $\delta'_{u\to v}(k)$ decreasing exponentially in k.

The performance of the PC-IPA algorithm with increasing quantization lengths is estimated in the following proposition.

Proposition 1: Consider the setting described in Sect. II-A and the PC-IPA algorithm with the above-described probabilistic quantizer and repetition encoder, with increasing lengths $q(k) = k$ and $\ell(k) = Sk^2$. Then, there exists a constant

$\beta \in (0,1)$ depending only on the channels such that, defining $\alpha := \beta^S$ and ρ the second largest singular value of P, if[1] $\frac{\alpha}{\rho} \leq C < 1$ and $\frac{1}{2}\frac{1}{\rho} \leq C < 1$, then:

- for all $t \in \mathbb{Z}^+$,

$$\mathrm{E}[\zeta(t)^2] \leq \alpha^2(1-\alpha)^{-2}\,;$$

- for all $k \in \mathbb{Z}^+$, for all $t \in [h_k, h_{k+1})$,

$$\begin{aligned}\frac{1}{n}\mathrm{E}[\|\boldsymbol{z}(t)\|^2] \leq \rho^{2k}&\frac{1}{(1-\alpha/\rho)^2}\\ &+ \rho^{2k}\frac{1}{4-\rho^{-2}}\\ &+ \rho^{2k}\frac{1}{(1-\alpha/\rho)(1-1/(2\rho))}\,.\end{aligned}$$

This implies that there exists a real-valued random variable $\hat{y}$ satisfying

$$\mathrm{E}[(\hat{y}-y)^2] \leq \alpha^2(1-\alpha)^{-2}$$

and such that

$$\lim_{t\to\infty}\hat{\boldsymbol{y}}(t) = \hat{y}\mathbf{1} \text{ with prob. } 1\,.$$

Moreover, it is possible to choose the initial phase length S in such a way that

$$\tau(\delta) \leq c + c'\frac{\log^4(\delta^{-1})}{\log^3(\rho^{-1})}\,.$$

for some constants c, c' depending only on the channels and on C. □

C. Do lengths really have to be increasing?

The increasing lengths $\ell(k) = Skq(k)$ and $q(k) = k$ ensure that $\frac{1}{n}\mathrm{E}[\|\boldsymbol{z}(t)\|^2]$ decreases to zero exponentially fast with $\sqrt[3]{t}$, and thus the algorithm converges with probability 1 to a consensus. However, one might renounce asymptotic convergence and be interested in only reaching, at some time $\tau(\delta) < \infty$, a required precision δ. In the idealized case where perfect feedback is available, one would not need increasing lengths any more, as constant lengths would suffice. In contrast, in the realistic case when no feedback is available, and the partial compensation rule is adopted, one needs to keep increasing the repetition length $\ell(k)/q(k) = Sk$ in order to avoid accumulation of the errors $\delta'_{u\to v}(k)$, but can decide to use constant lengths for the quantizers $\mathcal{Q}^{(k)}$. Increasing quantizers' lengths $q(k) = k$ are needed in order to ensure convergence (Prop. 1), while constant quantizers' lengths $q(k) = q$ are enough to ensure $\tau(\delta) < \infty$, as stated in the following proposition.

Proposition 2: Consider the setting described in Sect. II-A and the PC-IPA algorithm with the probabilistic quantizer and repetition encoder described in Sect. III-B, with constant quantization length $q(k) = q$ and increasing phase length $\ell(k) = Skq$.

Then, there exists a constant $\beta \in (0,1)$ depending only on the channels such that, defining $\alpha := \beta^S$ and ρ the second largest singular value of P, if[1] $\frac{\alpha}{\rho^2} \leq C < 1$, then:

- for all $t \in \mathbb{Z}^+$,

$$\mathrm{E}[\zeta(t)^2] \leq \alpha^2(1-\alpha)^{-2}\,;$$

- for all $k \in \mathbb{Z}^+$, for all $k \in \mathbb{Z}^+$, for all $t \in [h_k, h_{k+1})$,

$$\begin{aligned}\frac{1}{n}\mathrm{E}[\|\boldsymbol{z}(t)\|^2] \leq \rho^{2k}&\frac{1}{(1-\alpha/\rho)^2}\\ &+ \frac{1}{4^{q+1}}\Phi(P)\\ &+ \frac{1}{2^q}\rho^{2k}(1-\alpha/\rho^2)^{-2}\,,\end{aligned}$$

where $\Phi(P) := \frac{1}{n}\sum_{r=1}^{\infty}\|P^{r+1}-P^r\|_{\mathrm{F}}^2$.

Hence, it is possible to choose the phase length S in such a way that

$$\tau(\delta) \leq c + c'\frac{\log^4(\delta^{-1})}{\log^2(\rho^{-1})}\,,$$

for some constants c, c' depending only on the channels, on C and[2] on $\Phi(P)$. □

D. Proof of Propositions 1 and 2

The proofs of Propositions 1 and 2 being very similar, we give them together. They are based on the following lemma characterizing the properties of the noises $\nu(k)$ and $\Delta'(k)$.

Lemma 1: Consider the setting described in Sect. II-A and the PC-IPA algorithm with the probabilistic quantizer and repetition encoder described in Sect. III-B, with quantization length $q(k) = q$ and phase length $\ell(k) = Skq(k)$. Then, the errors $\nu(k)$ and $\Delta'(k)$ satisfy the following properties:

1) $\mathrm{E}[\nu_u(r)] = 0$;
2) $\mathrm{E}[\nu_u(r)\nu_v(s)] = 0$ if $r \neq s$ or $u \neq v$ (or both);
3) $\mathrm{E}[\nu_u(k)^2] \leq \frac{1}{4^{q(k)+1}}$;
4) there exists $\beta \in (0,1)$ which depends only on the channel, such that $\mathrm{E}[\Delta'_{vw}(k)^2] \leq \alpha^{2k}$ with $\alpha := \beta^S$;
5) $\mathrm{E}[\Delta'_{vw}(r)\Delta'_{v'w'}(s)] \leq \alpha^{r+s}$;
6) $\mathrm{E}[\nu_u(r)\Delta_{vw}(s)] = 0$ if $s \leq r$;
7) $\mathrm{E}[\nu_u(r)\Delta_{vw}(s)] \leq \frac{1}{2^{q(r)+1}}\alpha^s$. □

Proof: Items 1), 2) and 6) are true because, for all $x(k-1)$, $\mathrm{E}[\nu_u(r)|x_{k-1}] = 0$.

[1]The assumptions $\rho > \alpha$ in Prop 1 and $\rho^2 > \alpha$ in Prop 2 are not very restrictive: one can always choose S such that they hold true. The assumption $\rho > \frac{1}{2}$ seems more restrictive, but it can be loosened by choosing $q(k) = Ak$ with A larger than 1. Moreover, all such assumptions are not really essential: they are only needed to get the exact expressions we give in the statements of the propositions, but removing them does not significantly change the behavior of the algorithms, it simply slightly modifies the expressions of the bounds. For example, if $\rho = \frac{1}{2}$ or $\rho < \frac{1}{2}$, the second term in the upper bound for $\mathrm{E}[\|\boldsymbol{z}(t\|)^2]$ in Prop. 1 becomes $\frac{1}{4}k\rho^{2k}$ or $\frac{1}{4^{k+1}}\frac{1}{1-4\rho^2}$, respectively. We have chosen to focus on the case with larger ρ because it is useful to analyze the dependence of the performance on ρ in the 'bad' case, where ρ is near 1 (see Sect. IV-C).

[2]In most cases, even for families of graphs for which $\rho \to 1$ for $n \to \infty$, $\Phi(P)$ remains bounded. In particular, [9, Coroll. 9] ensures that, if P is normal (i.e., $P^*P = PP^*$), then $\Phi(P) \leq \frac{1-p}{p}$ where $p := \min_u P_{uu}$. A loose bound for $\Phi(P)$ wich does not require any assumption is the following: $\Phi(P) \leq (1-\rho^2)^{-1}$.

For item 3), notice that, with the short-hand notation $\tilde{x} := 2^{q(k)}x$, we have

$$\begin{aligned}&\mathrm{E}[(x-\tilde{\mathcal{Q}}(x))^2]\\&=\tfrac{1}{4^{q(k)}}\left[(\tilde{x}-\lfloor\tilde{x}\rfloor)^2(\lceil\tilde{x}\rceil-\tilde{x})+(\tilde{x}-\lceil\tilde{x}\rceil)^2(\tilde{x}-\lfloor\tilde{x}\rfloor)\right]\\&=\tfrac{1}{4^{q(k)}}(\tilde{x}-\lfloor\tilde{x}\rfloor)(\lceil\tilde{x}\rceil-\tilde{x})\\&\le\tfrac{1}{4^{q(k)+1}}\,.\end{aligned}$$

Item 4). Using [6, Th. 11.9.1] on the Bayesian error probability of binary hypothesis tests of length Sk, one has that the probability of mis-decoding any of the $q(k)$ bits can be upper-bounded by ε^{Sk}, for some constant $\varepsilon\in(0,1)$ depending on the channel only. Moreover, if the most significant mis-decoded bit is the j-th, then $\Delta'_{vw}(k)^2\le 2^{-2(j-1)}$. It follows that

$$\mathrm{E}[\Delta'_{vw}(k)^2]\le\sum_{j=1}^{q(k)}\left(\frac{1}{2}\right)^{2(j-1)}\varepsilon^{Sk}\le\frac{4}{3}\varepsilon^{Sk}\,.$$

Finally, items 5) and 7) are obtained from 3) and 4) with Cauchy-Schwarz inequality:

$$\mathrm{E}[\Delta'_{vw}(r)\Delta'_{v'w'}(s)]\le\sqrt{\mathrm{E}[\Delta'_{vw}(r)^2]\mathrm{E}[\Delta'_{v'w'}(s)^2]}\le\alpha^{r+s}$$

and similarly for 7). ■

The bound $\mathrm{E}[\zeta(t)^2]\le\alpha^2(1-\alpha)^{-2}$ comes immediately from [4, Prop. 3]. For the bound on $\frac{1}{n}\mathrm{E}[\|\boldsymbol{z}(t)\|^2]$, we need to go through the proof of [4, Prop. 3], so as to take care of the additional terms involving quantization noise. First of all, from Eq. (4), we get:

$$\boldsymbol{z}(t)=P^k\boldsymbol{z}(0)+\sum_{r=1}^{k}P^{k-r}(P-I)\boldsymbol{\nu}(r)+\sum_{r=1}^{k}P^{k-r}\boldsymbol{u}(r)$$

for all $t\in[h_k,h_{k+1})$, where $\boldsymbol{u}(k):=(P\odot\Delta'(k))\mathbf{1}-\xi(k)\mathbf{1}$ and $\xi(k):=\frac{1}{n}\mathbf{1}^*(P\odot\Delta'(k))\mathbf{1}$. Thus, $\forall t\in[h_k,h_{k+1})$

$$\begin{aligned}\tfrac{1}{n}\mathrm{E}[\|\boldsymbol{z}(t)\|^2]=&\,\mathrm{E}[\|P^k\boldsymbol{z}(0)+\sum_{r=1}^{k}P^{k-r}\boldsymbol{u}(r)\|^2]\\&+\tfrac{1}{n}\mathrm{E}[\|\sum_{r=1}^{k}P^{k-r}(P-I)\boldsymbol{\nu}(r)\|^2]\\&+\tfrac{2}{n}(P^k\boldsymbol{z}(0))^*\sum_{r=1}^{k}P^{k-r}(P-I)\mathrm{E}[\boldsymbol{\nu}(r)]\\&+\tfrac{2}{n}\sum_{r=1}^{k}\sum_{s=1}^{k}\mathrm{E}[(P^{k-r}(P-I)\boldsymbol{\nu}(r))^*P^{k-s}\boldsymbol{u}(s)]\end{aligned}\tag{5}$$

Now, we will give bounds for the four terms in Eq. (5). Along the proof, we will use the fact that both $(P-I)\boldsymbol{\nu}(k)$ and $\boldsymbol{u}(k)$ are perpendicular to $\mathbf{1}$, and that $\|P\boldsymbol{x}\|\le\rho\|\boldsymbol{x}\|$ for all $\boldsymbol{x}\perp\mathbf{1}$.

i. From [4, Prop. 3],

$$\mathrm{E}[\|P^k\boldsymbol{z}(0)+\sum_{r=1}^{k}P^{k-r}\boldsymbol{u}(r)\|^2]\le\rho^{2k}(1-\alpha/\rho)^{-2}\,.$$

ii. Thanks to item 2) in Lemma 1,

$$\begin{aligned}&\tfrac{1}{n}\mathrm{E}[\|\sum_{r=1}^{k}P^{k-r}(P-I)\boldsymbol{\nu}(r)\|^2]\\&=\tfrac{1}{n}\sum_{r=1}^{k}\mathrm{E}[\|P^{k-r}(P-I)\boldsymbol{\nu}(r)\|^2]\\&\le\tfrac{1}{n}\sum_{r=1}^{k}\rho^{2(k-r)}\frac{n}{4^{q(r)+1}}.\end{aligned}$$

Now, for Prop. 1 we have $q(r)=r$ and thus the latter line is equal to

$$\tfrac{1}{4}\rho^{2k}\sum_{r=1}^{k}(2\rho)^{-2r}\le\tfrac{1}{4}\rho^{2k}\frac{1}{1-1/(4\rho^2)}\,.$$

For Prop. 2, instead, we have $q(r)=q$ and thus we have $\frac{1}{4^{q+1}}\sum_{r=1}^{k}\rho^{2(k-r)}\le\frac{1}{4^{q+1}}\frac{1}{1-\rho^2}$. However, following [9, Thm. 7] we can get the following tighter bound, which exploits linearity of expectation and of trace, the simple remark that for a scalar value a, $a=\operatorname{tr}a$, and the property $\operatorname{tr}(ABC)=\operatorname{tr}(CBA)$ whenever the size of the matrices A,B,C allows to write such products.

$$\begin{aligned}&\tfrac{1}{n}\sum_{r=1}^{k}\mathrm{E}[\|P^{k-r}(P-I)\boldsymbol{\nu}(r)\|^2]\\&=\tfrac{1}{n}\sum_{r=1}^{k}\mathrm{E}[\operatorname{tr}\{\boldsymbol{\nu}(r)^*(P-I)^*(P^{k-r})^*P^{k-r}(P-I)\boldsymbol{\nu}(r)\}]\\&=\tfrac{1}{n}\sum_{r=1}^{k}\operatorname{tr}\{\mathrm{E}[\nu(r)\nu(r)^*](P-I)^*(P^{k-r})^*P^{k-r}(P-I)\}\\&=\tfrac{1}{n}\frac{1}{4^{q+1}}\sum_{s=1}^{k}\operatorname{tr}\{(P-I)^*(P^s)^*P^s(P-I)\}\\&=\frac{1}{4^{q+1}}\Phi(P).\end{aligned}$$

iii. Thanks to Lemma 1, the third term is zero.

iv. Using Lemma 1 and Cauchy-Schwarz inequality, we get:

$$\begin{aligned}&\tfrac{2}{n}\sum_{r=1}^{k}\sum_{s=1}^{k}\mathrm{E}[(P^{k-r}(P-I)\boldsymbol{\nu}(r))^*P^{k-s}\boldsymbol{u}(s)]\\&\le\tfrac{2}{n}\sum_{r=1}^{k}\sum_{s=1}^{k}\sqrt{\mathrm{E}[\|P^{k-r}(P-I)\boldsymbol{\nu}(r)\|^2\mathrm{E}[\|P^{k-s}\boldsymbol{u}(s)\|^2}\\&\le 2\sum_{r=1}^{k}\sum_{s=r+1}^{k}\rho^{k-r}\frac{1}{2^{q(r)+1}}\rho^{k-s}\alpha^s\end{aligned}$$

For Prop. 1 we have $q(r)=r$ and thus the latter line is equal to

$$\rho^{2k}\sum_{r=1}^{k}(2\rho)^{-r}\sum_{s=r+1}^{k}\left(\frac{\alpha}{\rho}\right)^s\le\rho^{2k}\frac{1}{1-(2\rho)^{-1}}\,\frac{1}{1-(\alpha/\rho)^{-1}}\,.$$

For Prop. 2, instead, we have $q(r) = q$ and thus we have (exchanging the order of summation):

$$\begin{aligned}\frac{1}{2^q}\rho^{2k}\sum_{s=1}^{k}\left(\frac{\alpha}{\rho}\right)^s\sum_{r=1}^{s-1}\rho^{-r} &\le \frac{1}{2^q}\rho^{2k}\sum_{s=1}^{k}\left(\frac{\alpha}{\rho}\right)^s(s-1)\rho^{-(s-1)}\\ &\le \frac{1}{2^q}\rho^{2k}\sum_{s=1}^{k}\left(\frac{\alpha}{\rho}\right)^{s-1}s\rho^{-(s-1)}\\ &\le \frac{1}{2^q}\rho^{2k}\sum_{s'=0}^{\infty}(s'+1)(\alpha/\rho^2)^{s'}\\ &= \frac{1}{2^q}\rho^{2k}[1-(\alpha/\rho^2)]^{-2}.\end{aligned}$$

This concludes the first part of the proof. Then, convergence with probability 1 follows from Markov inequality and Borel-Cantelli Lemma, as in [4, Thm. 4].

Finally, the bounds on $\tau(\delta)$ are obtained as follows. For the case with $q(k) = k$, we know from the first part of Prop. 1 that the following conditions are sufficient to ensure that $\frac{1}{n}\mathrm{E}[\|\boldsymbol{e}(t)\|^2] \le \delta$ for all $t \ge h(k) = \sum_{r\le k}\ell(r)$:

1) $\alpha^2(1-\alpha)^{-2} \le \delta/4$;
2) $\rho^{2k}\frac{1}{(1-\alpha/\rho)^2} \le \delta/4$;
3) $\rho^{2k}\frac{1}{4-\rho^{-2}} \le \delta/4$;
4) $\rho^{2k}\frac{1}{(1-\alpha/\rho)(1-1/(2\rho))} \le \delta/4$.

Recalling that $\alpha = \beta^S$, clearly there exists constants c_1 and c_2 depending only on β and on C such that the first inequality is true for all $S \ge c_1\log(\delta^{-1})$ and the three other inequalities are true for all $k \ge c_2\log(\delta^{-1})/\log(\rho^{-1})$. The claim on $\tau(\delta)$ then follows by recalling that $h(k) = \sum_{r\le k}\ell(r) = \sum_{r\le k}Sr^2 \le Sk^3$. The proof of the bound on $\tau(\delta)$ in Prop. 2 is obtained with the same technique. ■

IV. Simulations and Discussion

This section is devoted to some examples illustrating the averaging algorithms proposed in this paper. Specifically, in Sect. IV-A we provide a practical implementation of the PC-IPA. In Sect. IV-B we provide a comparison between the IPA, PC-IPA, and C-IPA algorithms. Finally in Sect. IV-C we comment on how the complexity of the presented algorithms depends on the topology of the network.

In all our simulations, we use as a simple example of channel the binary erasure channel (BEC), i.e., the output alphabet is $\mathscr{A} = \{0,1,?\}$ and each sent bit is erased with probability ε, while it is correctly received with probability $1-\varepsilon$. In our simulations, we use $\varepsilon = 1/2$.

We describe here the communication graph and the consensus matrix P adopted in our simulations. We consider $n = 30$ agents, and a communication graph which is a strongly connected realization of a two-dimensional random geometric graph, where vertices are 30 points uniformly distributed in the unit square, and there is a pair of edges (u,v) and (v,u) whenever points u,v have a distance smaller than 0.4. The communication graph is bidirectional, in the sense that $\mathcal{N}_v^- = \mathcal{N}_v^+$ for all $v \in \mathcal{V}$. The consensus matrix P is built according to the Metropolis weights:

$$P_{uv} = \begin{cases}\frac{1}{1+\max\{\deg(u),\deg(v)\}} & \text{if } (u,v)\in\mathcal{E}\\ 1-\sum_{w\in\mathcal{N}_u^-}P_{uw} & \text{if } u=v\\ 0 & \text{otherwise}\end{cases}$$

where $\deg(v)$ is the number of neighbors of node v. The construction is distributed, as it uses only information on neighbors.

The initial condition $\boldsymbol{\theta}$ of each experiment is randomly sampled from a uniform distribution on $[0,1]^n$. All plots show curves averaged over 1000 simulations: a different random geometric graph and a different initial condition, as well as a different channel noise sequence, are independently generated for each simulation.

In all our simulations we use the quantization and channel coding described in Sect. III-B, which a choice of the lengths $q(k)$ and $\ell(k)$ which is specified for each simulation. Notice that, with this simple repetition code and for the BEC, the decoding is trivial: a bit is correctly decoded whenever at least one of its repeated copies is received unerased.

A. Performance of PC-IPA

We start by providing a practical implementation of the PC-IPA and commenting on its performance.

As indicated in Proposition 1, we set $q(k) = k$ and $\ell(k) = Sk^2$, $S > 0$.

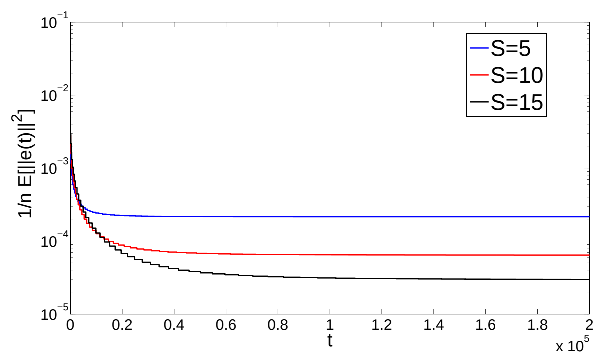

Fig. 1. Behavior of $n^{-1}\mathrm{E}[\|\boldsymbol{e}(t)\|^2]$ for different values of S when the PC-IPA algorithm is adopted.

The simulation results obtained are plotted in Figures 1 and 2, which show $n^{-1}\mathrm{E}\left[\|\boldsymbol{e}(t)\|^2\right]$ and $n^{-1}\mathrm{E}\left[\|\boldsymbol{z}(t)\|^2\right]$ respectively, for different values of S. Observe that the larger is the value of S, the better is the attainable performance in terms of $\boldsymbol{e}(t)$. On the other hand, larger values of S also imply a slower convergence to 0 of $\boldsymbol{z}(t)$.

B. Effect of (partial) compensation on performance

It is natural to expect that the partial compensating rule introduced in PC-IPA permits to improve performance with respect to the simple IPA, and that the total compensation of C-IPA improves performance even more, provided channel feedback is available. Indeed, according to Prop. 1 and to [4, Theorems 4 and 5], the three algorithms ensure almost sure

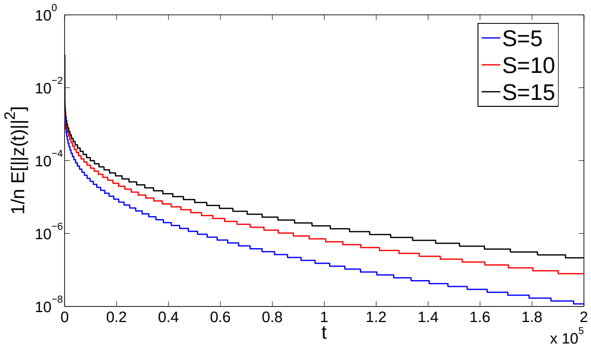

Fig. 2. Behavior of $n^{-1}\mathrm{E}[\|\boldsymbol{z}(t)\|^2]$ for different values of S when the PC-IPA algorithm is adopted.

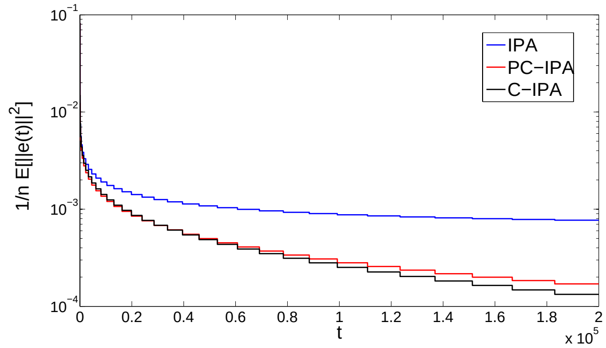

Fig. 3. Behavior of $n^{-1}\mathrm{E}[\|\boldsymbol{e}(t)\|^2]$ for the IPA, PC-IPA and C-IPA algorithms.

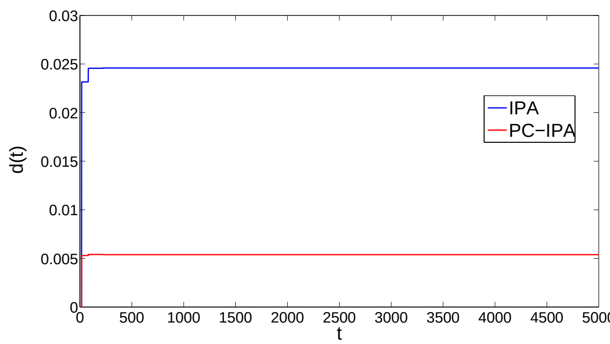

Fig. 4. Behavior of $d(t)$ for the IPA and PC-IPA algorithms.

convergence to consensus and their communication complexities $\tau(\delta)$ satisfy

- for IPA: $\tau(\delta) \leq c + c' \dfrac{\log^5(\delta^{-1})}{\log^3(\rho^{-1})}$;
- for PC-IPA: $\tau(\delta) \leq c + c' \dfrac{\log^4(\delta^{-1})}{\log^3(\rho^{-1})}$;
- for C-IPA: $\tau(\delta) \leq c + c' \dfrac{\log^3(\delta^{-1})}{\log^3(\rho^{-1})}$;

using communication phases of length $\ell(k) = Sk^2$.

In order to illustrate why these bounds depend on $\log(\delta^{-1})$ in a different way, we compare the three algorithms assuming that coding is done by quantization followed by a simple repetition as described in Sect. III-B, with lengths $q(k) = Ak$ and $\ell(k)/q(k) = Bk$, for some positive A, B and so $\ell(k) = Sk^2$ with $S = AB$. In the case of IPA, at each iteration k both a quantization error $\sim 2^{-Ak}$ and a channel error $\sim \varepsilon^{-Bk}$ accumulate on the marginally stable subspace of the dynamical system, so that choosing $A = B = \sqrt{S}$ ensures that $\delta_{u \to v}(k)$ decreases exponentially in $\sqrt{S}k$, and that $|\hat{y} - y| \leq \beta^{\sqrt{S}}$, for some β depending only on the channel. Thus, to get within a distance δ from y, we need to choose $\sqrt{S}$ that grows (at least) linearly in $\log(\delta^{-1})$. Instead, for PC-IPA, only the channel error accumulates, so that one can choose $q(k) = k$ and have the channel error decrease exponentially in Sk, which implies that S (as opposed to $\sqrt{S}$ for IPA) needs to grow linearly in $\log(\delta^{-1})$. Finally, for the C-IPA, $\hat{y} = y$ and no error accumulates, so that one can choose $q(k) = k$ and S independent of δ.

In Figures 3 and 4 we reported the results obtained simulating the IPA, PC-IPA and C-IPA strategies, adopting the same coding/decoding scheme for all three algorithms. Precisely we set $q(k) = \sqrt{S}k$ and $\ell(k) = Sk^2$ with $S = 16$. In Figure 3 we depict the behavior of $n^{-1}\mathrm{E}\left[\|\boldsymbol{e}(t)\|^2\right]$ while in Figure 4 we plot the quantity

$$d(t) := |\zeta(t)| = \left|\tfrac{1}{n}\mathbf{1}^* \hat{\boldsymbol{y}}(t) - y\right|.$$

Since $d(t)$ is equal to 0 in the C-IPA algorithm, in Figure 4 we depict only the curves for IPA and PC-IPA. As expected, the C-IPA algorithm outperforms the other two strategies. However, it is clear that perfect channel feedback is a strong and often unrealistic assumption. Remarkably, we can see from Figure 4 that also the partial compensation on the quantization error introduced in PC-IPA allows for a significant improvement over the IPA.

C. Performance scaling with network size

We conclude with some considerations on how the complexity of the presented algorithms depends on the topology of the network. The matrix P is adapted to the communication graph $\mathcal{G}$, and ρ, the second largest singular value of P, depends on the network topology. A precise characterization of this dependence can be obtained for some families of graphs. For instance, consider a regular grid with $n = m^d$ nodes on a d-dimensional torus: such graph can be seen as a Cayley graph on the Abelian group $\mathbb{Z}_m^d$. When there is a family of Abelian Cayley graphs of increasing size with bounded neighborhoods and the coefficients in the adapted matrices P do not vary with n either, then the asymptotic behavior of the second largest singular value is the following (see, e.g., [16]):

$$\rho = 1 - \tfrac{C}{n^{2/d}} + O\left(\tfrac{1}{n^{4/d}}\right) \qquad n \to \infty,$$

where C is a positive constant, depending only on d and on the coefficients assigned to neighbors.

It is worth to notice that this qualitative behavior for $n \to \infty$ does not strictly rely on the group structure of $\mathbb{Z}_m^d$, but is more general and can be observed in other families of dimension-dependent graphs, like geometric graphs, that is graphs whose nodes belong to an hypercube $[0,1]^d \subset \mathbb{R}^d$ and have 'local' neighborhoods in a Euclidean sense. Under mild non-degeneracy assumptions, matrices P adapted to such graphs are shown in [16] to satisfy $\rho = 1 - \Theta(n^{-d/2})$.

These facts and Prop. 1 imply that on these families of d-dimensional graphs, the communication complexity of the PC-IPA with $\ell(k) = Sk^2$ is $O(n^{6/d})$ for $n \to \infty$. Similar bounds can be obtained for all the algorithms discussed above. Although such results suggest that the performance of our algorithms may not scale nicely with the size of the network, simulations do not seem to show such a drawback. Likely, ρ does not provide the most significant information about the role of topology in the achievable performance. A similar remark has been made in [9] and in [19], in the case of quantization or additive noise, respectively. In those cases, it was possible to obtain different and tighter bounds, involving all the eigenvalues of P instead of only the dominant one, and having a nicer scaling with respect to n for many classes of graphs (see also the survey [16]). It is an open problem to look for similar bounds in our setting, where we have the additional difficulty that the errors do not have zero mean and are not independent.

Acknowledgment

The research leading to this paper has been partly funded by the European Community Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack, and by the French government's CONNECT grant (ANR PSIROB06-174215).

References

[1] T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4905–4918, 2008.

[2] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[3] R. Carli, F. Bullo, and S. Zampieri, "Quantized average consensus via dynamic coding/decoding schemes," *Intern. J. Robust and Nonlin. Control*, vol. 20, no. 2, pp. 156–175, 2010.

[4] R. Carli, G. Como, P. Frasca, and F. Garin, "Distributed averaging on digital erasure networks," *Automatica*, vol. 47, no. 1, pp. 115–121, 2011.

[5] G. Como, F. Fagnani, and S. Zampieri, "Anytime reliable transmission of real-valued information through digital noisy channels," *SIAM J. Control and Optim.*, vol. 48, no. 6, pp. 3903–3924, 2010.

[6] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley, New York, 1991.

[7] A. K. Das, Y. Hatano, and M. Mesbahi, "Agreement over noisy networks," *IET Control Theory and Applications*, vol. 4, no. 11, pp. 2416–2426, 2010.

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

[9] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus on networks with quantized communication," *Intern. J. Robust and Nonlin. Control*, vol. 19, no. 16, pp. 1787–1816, 2009.

[10] M. Huang and J. H. Manton, "Coordination and consensus of networked agents with noisy measurements: Stochastic algorithms and asymptotic behavior," *SIAM J. Control and Optim.*, vol. 48, no. 1, pp. 134–161, 2009.

[11] ——, "Stochastic consensus seeking with noisy and directed inter-agent communication: fixed and randomly varying topologies." *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 235–241, 2010.

[12] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, 2009.

[13] ——, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1383–1400, 2010.

[14] T. Li, M. Fu, L. Xie, and J.-F. Zhang, "Distributed consensus with limited communication data rate," *IEEE Trans. Autom. Control*, 2010, in press.

[15] T. Li and J.-F. Zhang, "Mean square average-consensus under measurement noises and fixed topologies: Necessary and sufficient conditions," *Automatica*, vol. 45, no. 8, pp. 1929 – 1936, 2009.

[16] E. Lovisari and S. Zampieri, "Performance metrics in the consensus problem: a survey," in *IFAC Symposium on System, Structure and Control*, 2010.

[17] R. Rajagopal and M. J. Wainwright, "Network-based consensus averaging with general noisy channels," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 373–385, 2011.

[18] S. S. Stanković, M. S. Stanković, and D. S. Stipanović, "Decentralized parameter estimation by consensus based stochastic approximation," *IEEE Trans. Autom. Control*, in press.

[19] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.