Information Theoretic Bounds for Distributed Computation Over Networks of Point-to-Point Channels

Information Theoretic Bounds for Distributed Computation Over Networks of Point-to-Point Channels

Ola Ayaso, Devavrat Shah, *Member, IEEE*, and Munther A. Dahleh, *Fellow, IEEE*

***Abstract*—A network of nodes communicate via point-to-point memoryless independent noisy channels. Each node has some real-valued initial measurement or message. The goal of each of the nodes is to acquire an estimate of a given function of all the initial measurements in the network. As the main contribution of this paper, a lower bound on computation time is derived. This bound must be satisfied by any algorithm used by the nodes to communicate and compute, so that the mean-square error in the nodes' estimate is within a given interval around zero. The derivation utilizes information theoretic inequalities reminiscent of those used in rate distortion theory along with a novel "perturbation" technique so as to be broadly applicable. To understand the tightness of the bound, a specific scenario is considered. Nodes are required to learn a linear combination of the initial values in the network while communicating over erasure channels. A distributed quantized algorithm is developed, and it is shown that the computation time essentially scales as is implied by the lower bound. In particular, the computation time depends reciprocally on "conductance", which is a property of the network that captures the information-flow bottleneck. As a by-product, this leads to a quantized algorithm, for computing separable functions in a network, with minimal computation time.**

***Index Terms*—Computation time, conductance, distributed computing, noisy networks, quantized summation.**

I. Introduction

We consider a network of nodes communicating via a network of point-to-point memory-less independent noisy channels. Each node has a single real-valued initial measurement or message. The goal of each of the nodes is to acquire an estimate of a given function of all the initial measurements in the network.

We seek to understand the limitations imposed by the communication constraints on the nodes' performance in computing the desired function. The performance is measured by the mean-square error in the nodes' estimates of the desired function. The communication constraints consist of: 1) the topology of the network, that is, the connectivity of the nodes and 2) the noisy channels between nodes that communicate. In order to capture the limitation due to the communication constraints, we assume that that the nodes have unlimited computation capability. Each node can perform any amount of computation as well as encoding and decoding for communication.

As we discuss below, the formulation of Section II is not the typical information theoretic formulation for networks. Our setup is more similar to certain distributed computation formulations. Still, we use information theoretic inequalities to derive lower bounds on information exchange between nodes necessary for the mean-square error in the nodes' estimates to converge to zero.

Both our technique and results are different from those of the distributed computation results. In Section V, we derive a lower bound on computation time that must be satisfied by *any* algorithm used by the nodes to communicate and compute, so that the mean-square error in the nodes' estimates is within a given interval around zero. The bound is in terms of the channel capacities, the size of the desired interval, and the uncertainty in the function to be computed. To obtain this bound, we develop a novel "perturbation" technique as explained in Section V-C. This allows us to apply our method to obtain non-trivial lower bound for any functional computation setup.

Our lower bound is a universal lower bound that holds for any causal distributed algorithm that can be used by the nodes to attain their goal of function computation. We make minimal assumptions on how a node encodes messages sent over the channels or decodes messages received via the channels. Furthermore, we make minimal restrictions on how the node uses the information it possesses to compute or update its estimate. Essentially, we only require that the encoders, decoders, and estimators are measurable and causal mappings, the output depends only the node's initial measurement and the messages received in the past.

As a result, our lower bound provides a means to assess the optimality of distributed causal computation algorithms. No algorithm can achieve a desired mean-square error in a computation time that is smaller than the lower bound. This limitation is due to the distributed nature of the algorithm, specifically, the need to communicate with nodes via a network of noisy point-to-point channels. Therefore, any algorithm that has a computation time that is equal to the lower bound is an optimal distributed algorithm for the given network topology. We illustrate this in the remainder of the paper for a scenario where nodes are required to learn a linear combination of the initial

Manuscript received September 03, 2008; revised October 07, 2009. Date of current version November 19, 2010. This work was supported (in part) by the National Science Foundation (NSF) HSD project 0729361, by AFOSR Grant FA9550-08-0085, and NSF EFRI-ARES Grant 0735956.

O. Ayaso is with the Georgia Institute of Technology, Atlanta, GA 30332 USA and also with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: ayaso@alum.mit.edu).

D. Shah and M. A. Dahleh are with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: devavrat@mit.edu; dahleh@mit.edu).

Communicated by M. Gastpar, Associate Editor for Shannon Theory.

Digital Object Identifier 10.1109/TIT.2010.2080850

values in the network while communicating over block erasure channels.

In Section VI, we consider a scenario where nodes are required to learn a linear combination of the initial values. Our lower bound suggests that in this scenario, the computation time depends reciprocally on a "conductance-like" term. This term is equal to "conductance" when the the channel from node i to node j has the same capacity as the channel from node j to i. Conductance essentially captures the information-flow bottleneck that arises due to topology and channel capacities. The more severe the communication limitations, the smaller the conductance. When nodes communicate over erasure channels, our conductance is identical to the graph-theoretic conductance that arises in the analysis of mixing times in Markov chains.

To establish the tightness of our lower bound, we describe an algorithm for computation linear combination of the initial values when nodes communicate over block erasure channels. For this algorithm, the computation time matches the lower bound. The algorithm that we describe here can in fact be more generally used for distributed computation of separable functions, a special case of which is the sum. The desired function, a sum, is simple, and the algorithm that we describe has computational demands that are not severe. So, the time until the performance criterion is met using this algorithm is primarily constrained by the limitations on communication.

Indeed, we show that the upper bound, on the time until this algorithm guarantees the performance criterion, depends reciprocally on conductance. Hence, we conclude that the lower bound we derive using information theoretic analysis is tight in capturing the limitations due to the network topology. Alternatively, one can interpret this tightness as the fact that the algorithm we describe here is the fastest with respect to its dependence on the network topology, as quantified by the conductance. Thus, our distributed quantized algorithm answers a question of recent interest on the design of the fastest possible distributed algorithm for separable function computation, for example, in the works on consensus, linear estimation and distributed control.

A. Related Work

Our work has similarities with a vast body of works that can be broadly categorized as distributed computation, signal processing, information theory or control. Our formulation is most similar to some formulations appearing in the distributed computation and signal processing literature. Our approach of finding a lower bound using information theoretic inequalities and demonstrating a bound-achieving algorithm is similar to work in the information theory literature. Our inspiration, for using information theoretic tools on a formulation that is not typical in information theory, comes from a similar approach that appears in the control theory literature. Below, we highlight the difference between our work and some works from the different fields mentioned above.

Our problem formulation is similar to certain formulations in the distributed computation literature, like the distributed averaging and consensus literature. Each node has an initial value, a real number or vector, and needs to exchange data with its neighbors in order to compute a function of the data in the network. Results consist of a suggested algorithm together with an analysis of the algorithm. For example, upper or lower bounds are provided on the time or number of messages exchanged until nodes compute a certain quantity with given accuracy [4], [16], [23]. Or, conditions are provided so that nodes reach agreement asymptotically in the number of algorithm iterations [3], [29], [30]. In the cited work, communication is subject to topological constraints, but perfect when present.

Recent studies explicitly assume imperfect communication of some sort, in addition to the topological constraint of having direct links with neighbors only. For example, in the average consensus problem, all nodes are required to asymptotically agree on the average of all the initial values in the network. In [2] and [31], messages are quantized but transmission is noiseless. In [14], messages are real-valued, but links between nodes may fail probabilistically. In a parameter estimation problem considered in [15], messages are quantized and links may fail probabilistically.

In all of these works, authors assume that each node updates its estimate of the quantity to be computed by linearly combining the previous estimate, received data, and other information that the node possesses. The computation algorithm is analyzed together with the encoding and decoding strategies, such as the suggested quantization scheme. Asymptotic properties, such as convergence to a consensus value, boundedness of mean-square error, or unbiasedness of the estimate are exhibited.

In contrast to the consensus literature, our goal is not for nodes to ultimately agree on the same value. Rather, the goal is for the nodes to compute a function of their measurements with desired accuracy. All nodes need not obtain the same estimate of the function. Our bounds are not asymptotic in the number of algorithm iterations; the accuracy appears explicitly in the bounds on computation time.

Furthermore, to obtain an optimal algorithm for summation over block erasure channels, we do not constrain the node update rule, for the estimate of the sum, to be linear. In fact, when exchanging information, nodes need only keep track of the minimum of received values. So, nodes need not keep track of duplicate messages or sender identity. Another consequence of not limiting our updates to be linear is that the quantization scheme that we propose is relatively simple.

Another perspective is the information theoretic one. Each node has access to a sequence of samples from its source. Alternatively, the node receives data at some bit rate. In classical network information theory, the goal is for nodes to reliably exchange these samples [5]. In [9], the authors derive information theoretic bounds on the number of bits that must be exchanged for nodes communicating via noiseless channels to acquire each other's data. In [1], the authors consider a point-to-point network of finite-rate noiseless channels. This network connects one set of nodes, the source nodes, to another, the destination nodes. Source nodes need to transmit their data sequences to the destination nodes. The admissible rate region is characterized.

Recent work investigates a variation on these information theoretic formulations. Nodes exchange information for function computation rather than transmission of data. For example, in [25] there is one encoder, a noiseless finite-rate channel and a decoder with side information. The authors determine the compression rate region so that the decoder can obtain a function

of its side information and the source. In [24], the authors investigate the "computation capacity" region so that a decoder receiving information via a multiple access channel obtains a function of the sources.

Our formulation is different in several ways. For example, for our lower bound for summation, each node has at time zero a single real-valued initial value, that is, infinite bits. Unlike our work, the results in [1], [24], and [25] hold asymptotically in the block length (number of source samples or length of messages sent over the channel, depending on the formulation).

But, like our work, results typically consist of two parts. First, there are lower bounds. These are derived using information theoretic inequalities and properties. Second, there is an algorithm, or proof of existence of an algorithm or code, achieving the lower bound.

Like our work, there is a common message that appropriate processing of data improves performance. As put in [1], "network coding has to be employed to achieve optimality". In [25], for certain functions, the rate region for computation is larger than the rate region for data exchange (equivalently, computing the identity function). That is, for computing certain functions, transmission can be made more efficient than simply transmitting the source. In [24], the authors show that for computation over multi-access channels, codes that utilize joint source-channel strategies outperform strategies which encode the source and channel messages separately. Our optimal algorithm requires that data is processed at nodes as they exchange messages; in particular, a node passes on the minimum of all the messages it receives.

In a similar flavor as our work, [10] and [12] provide an algorithm for computation together with an algorithm-independent lower bound that establishes optimality of the proposed algorithm. In [10], each node in the network has one bit. Nodes broadcast messages to each other via binary symmetric channels. The goal is for a fusion center to compute the parity of all the bits in the network. Gallager proposes an algorithm that can be used while guaranteeing a desired probability of error. He exhibits an upper bound that is a constant multiple of the bits that must be transmitted per node. Recently, it has been shown in [12] that this algorithm is optimal. The authors produce an algorithm-independent lower bound that is of the same order as the upper bound.

Several formulations and results relevant to computation in wireless sensor networks can be found in a detailed survey by Giridhar and Kumar [11].

In summary, our formulation is similar to that of distributed computation, but our approach is similar to that of information theory. We use information theoretic inequalities, reminiscent of those of rate-distortion theory, in a different setting with different objectives. In particular, we have a network of nodes whose objective is to compute a given function of the nodes' data, rather than to communicate reliably to each other their data. Hence, our results are quite different from results within either of these categories.

We capitalize on Martins' successful use of information theoretic tools in [17]–[20] to characterize fundamental performance limits of feedback control systems with communication constraints. In our setting, the information theoretic approach captures fundamental performance limitations that arise in the network due to the communication constraints. The derivation of the lower bound is independent of the communication algorithm used by the nodes. Therefore, the lower bound enables us to characterize the effect of the network structure on algorithm running time. We propose an algorithm to compute the sum of initial conditions for nodes exchanging information over block erasure channels. By showing that this algorithm's computation time achieves the lower bound, we conclude that the lower bound is indeed tight in capturing the network constraints.

B. Organization

In the next section, we describe the problem formulation and necessary formalities. In Section III we state the two main results of this paper. The first result is a general lower bound on the computation time for nodes communicating over a network of point-to-point independent memory-less channels. The second result consists of two parts. First, we specialize the general lower bound to the case of nodes computing the sum of their initial values. Second, we describe a quantized algorithm for computation of sum and show that its computation time achieves our lower bound with respect to the dependence on the network structure.

In Section IV, we illustrate how network topology, through conductance, affects the computation time. We compare our quantized algorithm with the popular linear iterative algorithms. The comparison suggests that for network structures with *small* conductance our algorithm outperforms the popular algorithms.

In Section V, we prove our main theorem on the general lower bound. Then, we illustrate the use of a novel perturbation argument, introduced in Section V-C, to obtain a non-trivial bound when nodes compute any general function. In Section VI-A, we derive the lower bound for the computation of the sum of initial values; the computation time scales reciprocally with conductance. In Section VI-B, we describe an algorithm that can be used to compute the sum via block erasure channels, where the block length depends on the number of nodes. We derive an upper bound on its computation time; we show that this upper bound also scales inversely with conductance. This establishes the optimality of our quantized algorithm for computation of summation in terms of its dependence on the graph structure.

II. Problem Description

A network consists of n nodes, each having a random initial condition or value. We represent the initial condition at node i by the random variable X_i. A realization of the random variable will be denoted by lower-case letters, x_i. Let $\underline{X}$ represent the vector of all the initial condition random variables, $[X_1 \ \dots \ X_n]'$.

Each node is required to compute a given function of all the initial conditions, with continuous support. That is, node i is required to estimate $Y_i = f_i(\underline{X})$, and Y_i is a continuous random variable. We let $\underline{Y} = [Y_1 \ \dots \ Y_n]'$. Suppose that nodes 1 to m belong to set S. Whenever we use a set as a subscript to a variable, we mean the vector whose entries are that variable subscripted by the elements of the set. For example, $Y_S = [Y_1 \ \dots \ Y_m]'$.

We assume that time is discretized into intervals, and enumerated by positive integers, $\{1, 2, \dots\}$. During each time step,

a node can communicate with its neighbors. At the end of time-slot k, node i uses the information it has received thus far to form an estimate of Y_i. We denote this estimate by $\widehat{Y}_i(k)$. The estimates of all nodes in the network at the end of time slot k are denoted by the vector $\underline{\widehat{Y}}(k) = [\widehat{Y}_1(k) \ \ldots \ \widehat{Y}_n(k)]'$. And, the estimates of nodes in set S are denoted by $\widehat{Y}_S(k) = [\widehat{Y}_1(k) \ \ldots \ \widehat{Y}_m(k)]'$.

The nodes communicate via point-to-point noisy channels. The network structure is described by a graph, $G = (V, E)$, where V is the set of nodes and E is the set of edges, (i, j). If node i communicates with node j via channel with capacity $\mathbf{C}_{ij} > 0$, then $(i, j) \in E$. If $(i, j) \notin E$, we set $\mathbf{C}_{ij} = 0$. We assume that the graph is connected.

We assume that all channels in the network are independent, memory-less and are operating in discrete-time. For each channel, one channel symbol is sent per unit time. Each node generates an input for its encoder every τ time units. For simplicity, we assume that $\tau = 1$. Thus, by the end of time k, each node has generated its kth estimate, $\widehat{Y}_i(k)$, based on the k received symbols and its initial value.

A. Features of the Formulation

Our formulation (and results) are appropriate when high accuracy computation must take place over networks with severe communication constraints. These include cases where

1) channel capacities are diminished, due to loss of transmission power, for example, or;
2) network topology creates information-flow bottlenecks.

B. Notation

The differential entropy of Y is denoted by $h(Y)$. The mutual information between X and Y is denoted by $I(X;Y)$. Most of the definitions and properties we will need can be found in texts like [5]. When indicated, we will need to use the most general definition of mutual information. It can be used when the random variables are arbitrary ensembles, not necessarily both continuous or both discrete [26, p. 9]. The conditional mutual information is similarly defined; see [26, Ch. 3].

Finally, when the argument in $h(\cdot)$ is a vector of length n, for example, $\underline{Y} = [Y_1, \ldots, Y_n]'$, it is interpreted as the joint differential entropy $h(Y_1, \ldots, Y_n)$. Similarly, when the arguments in $I(\cdot;\cdot)$ are vectors of length n, for example $\underline{Y}$ and $\underline{X}$, it is to be interpreted as $I(Y_1, \ldots, Y_n; X_1, \ldots, X_n)$.

III. Main Results

This section contains the formal statements of our main results. The first result, stated in Section III-A is a general lower bound on computation time. The second result, stated in Section III-B establishes the tightness of this lower bound in the specific scenario of the distributed computation of a sum. This involves, first, specializing the lower bound of Section III-A to the case where nodes compute a linear combination of the initial values in the network. Second, it involves developing a quantized algorithm for nodes computing the sum while communicating over erasure channels and showing that the computation time of the algorithm matches the lower bound for summation.

A. Result I: A General Lower Bound

The first main theorem of this paper provides a lower bound to computation time as a function of the accuracy desired, as specified by the mean-square error, and the uncertainty in the function that nodes must learn, as captured by the differential entropy.

We place few assumptions on how the nodes communicate and compute their estimates. Namely, each node can use only its own initial measurement and past received messages. But, we do not specify how the node makes its computation or exchanges messages. Hence, our lower bound reveals the smallest time that must elapse before it is possible to achieve the performance desired, over all communication and computation schemes that satisfy our assumptions. The necessity of this time elapsing is due to the fact that initial measurements are distributed and communication must occur over a network with a given topology and channel capacities.

Let $V_i^T = \{V_i(1), \ldots, V_i(T)\}$ be the symbols received by the decoder of node i up to time T. Then, $\widehat{Y}_i(T) = g_i(X_i, V_i^T)$. To capture the limitations arising exclusively due to the communication structure, in deriving our lower bound, we assume no limits on the computational capabilities of the nodes, such as limited memory or power. So, we make no assumptions on g_i, except that it is a measurable function.

Similarly, the messages that the node communicates with other nodes are a function of the node's initial condition and messages it has received in the past. Let U_i be transmitted by the node i encoder. The message transmitted by i in the l^{th} channel use, $U_i(l)$, is a function of the received messages at that node, V_i^{l-1} and its own data, X_i, $U_i(l) = \psi_i(V_i^{l-1}, X_i)$. We make no assumptions on ψ_i, except that it is a measurable function. The notation of this paragraph will not be needed until Appendix A.

We consider two mean-square error criteria. The operator $\|\cdot\|$ is to be interpreted, when the argument is a vector, $\underline{Y}$, as $\|\underline{Y}\|^2 = \sum Y_i^2$.

R1. $E(\|\underline{\widehat{Y}}(T) - \underline{Y}\|^2) \leq \beta 2^{-\alpha}$;

R2. $E(\widehat{Y}_i(T) - Y_i)^2 \leq 2^{-\alpha}$, for all $i \in \{1, \ldots, n\}$;

where $\alpha \in \mathbb{R}^+ \backslash \{0\}$.

The first criterion requires that as the number of nodes increases, the per node error is also smaller. It suggests that as the number of nodes, n, increases, we require the mean-square errors at each of the nodes, $E(\widehat{Y}_i(k) - Y_i)^2$ to decrease like $1/n$. This criterion is appropriate if, for example, the initial values at the nodes are independent and each node is to estimate the average of the initial values in the network. As the number of nodes increases, the variance of the average decreases. In circumstances where this does not happen, the second criterion may be more appropriate.

The "computation time" is the first time at which the desired performance criterion holds. In the first of our main results, we seek a lower bound on the computation time, T, that holds if the desired mean-square error criterion, R1 or R2, is satisfied.

Theorem III.1: For the communication network described above, if at time, T, the mean-square error is in an interval prescribed by α, $E(\widehat{Y}_i(T)-Y_i)^2 \leq 2^{-\alpha}$, for every node, then T is lower bounded by

$$T \geq \max_{S\subset V} \frac{\bar{L}(S)}{\sum_{i\in S^c}\sum_{j\in S}\mathbf{C}_{ij}}$$

where $S^c = V\backslash S$ and

$$\bar{L}(S) = h(Y_S|X_S) - \frac{|S|}{2}\log 2\pi e + |S|\frac{\alpha}{2}.$$

This theorem captures the fact that the larger the uncertainty in the function to be estimated, or the larger the desired accuracy, the longer it must take for any algorithm to converge. Specifically, when the mean-square error decreases exponentially in the accuracy, α, the computation time increases linearly in α, at best.

B. Result II: An Optimal Summation Algorithm

Here, we consider a specific scenario of the general formulation described in Section II. As before, we have a network of n nodes each having a random initial condition denoted by X_i. Each node needs to compute the same separable function of the initial values.

Definition III.2: $f:\mathbb{R}^n \to \mathbb{R}$ is separable if there exist functions $f_1,\ldots,f_n$ such that

$$f(\underline{x}) = \sum_{i=1}^{n} f_i(x_i).$$

Furthermore, we assume $f \in \mathcal{F}$ where $\mathcal{F}$ is the class of all separable functions with $f_i(x_i) \geq 1$ for all $x_i \in \mathbb{R}$ and $i = 1,\ldots,n$.

Remark: In the algorithm we describe, node i generates samples from an exponential distribution with mean $1/f_i(x_i)$. For the algorithm to work, we must have $f_i(x_i) > 0$ for all i. That is there is a constant $c > 0$ such that for all i, $f_i(x_i) \geq c$. Let $c = 1$. There is no loss of generality. And, this simplifies our expressions where constants do not matter, as our results are $O(.)$.

We assume that node i can compute $f_i(x_i)$ without communication. Further, we assume that there exists a constant B such that for all i, $f_i(x_i) \in [1, B+1]$. $B > 0$ is a constant and should be treated as a problem parameter.

In what follows, we will assume that $f_i(x_i) = \beta_i x_i$. This causes no loss of generality as we have assumed that each node can compute $f_i(x_i)$. So, essentially, we have relabeled $f_i(x_i)/\beta_i$ with x_i.

In terms of our formulation, we have that each node needs to compute the same quantity Y, where $Y = \sum_{j=1}^{n}\beta_j X_j$. Here, we assume that these initial values are distributed independently and uniformly in the interval $[1, B+1]$. The assumption that the distributions are uniform and independent simplifies computations in the derivation of our lower bound for summation.

Let A represent a realization of the initial conditions, $A = \{X_1 = x_1,\ldots,X_n = x_n\}$. The performance of an algorithm, $\mathcal{AG}$, used by the nodes to compute an estimate of $f(\underline{x}) = \sum_{j=1}^{n}\beta_j x_j$ at each node, is measured by the algorithm's (ε,δ)-computation time, $T^{\mathrm{cmp}}_{\mathcal{AG}}(\varepsilon,\delta)$. It is the time until the estimates at all nodes are within a factor of $1\pm\varepsilon$ of $f(\underline{x})$, with probability larger than $1-\delta$. Recall that $\widehat{Y}_i(k)$ denotes the estimate of node i at the end of time k.

Definition III.3: For $\varepsilon > 0$ and $\delta \in (0,1)$, the (ε,δ)-computing time of an algorithm, $\mathcal{AG}$, denoted as $T^{\mathrm{cmp}}_{\mathcal{AG}}(\varepsilon,\delta)$ is defined as

$$T^{\mathrm{cmp}}_{\mathcal{AG}}(\varepsilon,\delta) = \sup_{x\in\mathbb{R}^n}\inf\{k : \mathbf{P}(\cup_{i=1}^{n}\{\widehat{Y}_i(k) \notin [(1-\varepsilon)f(\underline{x}),(1+\varepsilon)f(\underline{x})]\}) \leq \delta\Big\}.$$

Here, the probability is taken with respect to $\widehat{Y}_i(k)$. This is random because nodes communicate over noisy channels.

As before, nodes communicate over noisy channels that are independent and discrete-time memory-less. Besides the assumptions of Section II, we make no additional assumptions about the channels in deriving our lower bound for summation. Additional assumptions will be stated where they are necessary.

The conductance $\Phi(G)$ captures the *information bottle-neck* in the capacitated graph G. It depends on the connectivity or topology of the graph along with the channel magnitude.

Definition III.4 (Conductance): The conductance of a capacitated graph G with edge capacities $\mathbf{C}_{ij}, (i,j) \in E$ is defined as

$$\Phi(G) = \min_{\substack{S\subset V \\ 0<|S|\leq n/2}} \frac{\sum_{i\in S, j\notin S}\mathbf{C}_{ij}}{|S|}.$$

We use the word "conductance" as it coincides with the notion of conductance or "Cheeger" constant for a Markov chain based on a symmetric and doubly stochastic matrix P on the network graph G. We will have more to say about conductance in Section IV.

A Lower Bound for Summation: Consider any algorithm, $\mathcal{AG}$, that guarantees that for any realization of the initial values, with high probability each node has an estimate within $1\pm\varepsilon$ of the true value of Y, at time T. The information theoretic lower bound maintains that such algorithm must have a computation time, $T = T^{\mathrm{cmp}}_{\mathcal{AG}}(\varepsilon,\delta)$, that is inversely proportional to conductance.

Theorem III.5: Nodes communicate in order for each node to compute a linear combination of all initial values in the network. Any algorithm that guarantees that for all $i \in \{1,\ldots,n\}$,

$$\mathbf{P}\left(|\widehat{Y}_i(T) - Y| \leq \varepsilon Y \Big| A\right) \geq 1-\delta$$

must have

$$T \geq \frac{1}{\widetilde{\Phi}(G)}\log\frac{1}{B\varepsilon^2 + \frac{1}{B}^{\frac{2}{n}} + \kappa\delta}$$

where, $B\varepsilon^2 \in \left[0, 1-\frac{1}{B}^{\frac{2}{n}} - \kappa\delta\right]$, κ is a constant, and

$$\widetilde{\Phi}(G) = \min_{\substack{S\subset V \\ 0<|S|\leq n/2}} \frac{\sum_{i\in S^c}\sum_{j\in S}\mathbf{C}_{ij}}{|S|}.$$

If $\mathbf{C}_{ij} = \mathbf{C}_{ji}$ then $\widetilde{\Phi}(G) = \Phi(G)$.

Again, the probability in this theorem is taken with respect to the measure on $\widehat{Y}_i(T)$, conditional on A, and induced by the randomness due to communication over channels.

An Upper Bound for an Algorithm for Summation Over Block Erasure Channels: Next, we provide an algorithm that guarantees, with high probability, the nodes' estimates are within the desired ε-error interval around the true value of the sum.

Here, we assume that nodes communicate via block-erasure channels. Specifically, if a node i sends a channel symbol to node j then it is successful with probability p_{ij} independently of everything else. The channel symbol is of length $\log M$ bits, where we shall decide value M later. Thus, the effective capacity of the channel between nodes i and j is $\mathbf{C}_{ij} = p_{ij} \log M$. We assume that $p_{ij} = p_{ji}$. Further, we assume that the matrix $P = [p_{ij}]$ is a doubly stochastic matrix.

We provide an upper bound on our algorithm's computation time. The computation time is inversely proportional to conductance.

Theorem III.6: Suppose that node i has an initial condition, x_i. There exists a distributed algorithm $\mathcal{AP}^{\mathcal{Q}}$ by which nodes compute a sum, $f(\underline{x}) = \sum_{j=1}^{n} \beta_j x_j$, via communication of quantized messages. If each quantized message is $\log M$ bits and $\log M = O(\log n)$, the quantization error will be no more than a given $\gamma = \Theta(\frac{1}{n})$, and for any $\varepsilon \in (\gamma f(\underline{x}), \gamma f(\underline{x}) + \frac{1}{2})$ and $\delta \in (0,1)$, the computation time of the algorithm will be

$$T^{\mathrm{cmp}}_{\mathcal{AP}^{\mathcal{Q}}}(\varepsilon, \delta) = O\left(\varepsilon^{-2} \log e \delta^{-1} \frac{\log n \delta^{-1} \log n}{\Phi(G)}\right).$$

So, setting $\delta = \frac{1}{n^2}$ in the above bound, we have

$$T^{\mathrm{cmp}}_{\mathcal{AP}^{\mathcal{Q}}}\left(\varepsilon, \frac{1}{n^2}\right) = O\left(\varepsilon^{-2} \frac{\log^3 n}{\Phi(G)}\right).$$

The computation time of this algorithm depends on the network topology, via the conductance of the graph, in the same reciprocal manner manifested by the lower bound. Thus, we conclude that the lower bound is tight in capturing the effect of the network topology on computation time. Conversely, the algorithm's running time is optimal with respect to its dependence on the network topology, as captured by the conductance.

IV. Conductance: Capturing the Effect of Topology

The conductance of a graph, $\Phi(G)$, is a property that captures the bottle-neck of information flow. It depends on the connectivity, or topology, of the graph, and the magnitudes of the channel capacities. The more severe the network constraints, the smaller the conductance. It is also related to time it takes for information to spread in a network; the smaller the conductance, the longer it takes.

$\Phi(G)$ is related to the standard definition of conductance utilized in Markov chain theory. Specifically, consider a Markov chain with irreducible and aperiodic probability transition matrix P on the n nodes of graph G. The P may not be necessarily symmetric or doubly stochastic. It is, however always stochastic since it is a probability matrix. It is well known that such a Markov chain has a unique stationary distribution $\pi = [\pi_i]$ (cf. Perron–Frobenius Theorem).

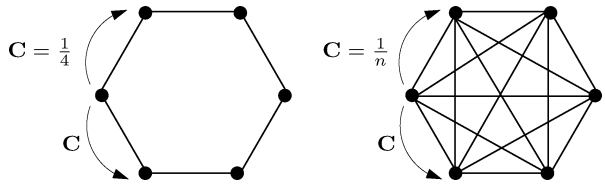

Fig. 1. Two ways to connect six nodes: a ring graph and a fully connected graph.

In the context of mixing times of Markov chains, conductance for the above P, $\Phi(P)$, is defined as

$$\Phi(P) = \min_{\substack{S \subset V \\ \pi(S) \le 1/2}} \frac{\sum_{i \in S, j \notin S} \pi_i p_{ij}}{\pi(S)}$$

where $\pi(S) = \sum_{i \in S} \pi_i$.

For a reversible Markov chain, the conductance is related to the spectral gap λ, where $\lambda = 1 - \lambda_2$ and λ_2 is the second largest eigenvalue of the transition matrix P. By the Cheeger bound [28]

$$\frac{1}{2}\Phi^2(P) \le \lambda \le 2\Phi(P).$$

In general, the $\Phi(P)$ is used to bound the mixing time of the Markov chain with transition matrix P. Let $\mathcal{H}(P)$ be the mixing time of the Markov chain, based on the notion of stopping time, then the following is a well-known bound

$$\Omega\left(\frac{1}{\Phi(P)}\right) = \mathcal{H}(P) = O\left(\frac{\log n}{\Phi^2(P)}\right).$$

Now, in our setup P is symmetric and doubly stochastic. In this case the stationary distribution π is uniform. That is, $\pi_i = 1/n$ for all i. Therefore, the conductance can be simplified to

$$\Phi(P) = \min_{\substack{S \subset V \\ 0 < |S| \le n/2}} \frac{\sum_{i \in S, j \notin S} p_{ij}}{|S|}.$$

In the case of our $\log M$-bit erasure channels, $\Phi(G) = \Phi(P) \log M$. In this sense, the $\Phi(G)$ is related to the standard definition of conductance utilized in the context of Markov chain theory. For more details on mixing times of Markov chains see [22] and [28].

A. Conductance: Two Examples

Consider two networks, each has n nodes. We calculate conductance for two extreme cases of connectivity shown in Fig. 1. On the one hand, we have severe topological constraints: a ring graph. Each node may contact only the node on its left or the node on its right. On the other hand, we have a case of virtually no topological constraints: a fully connected graph. Each node may contact every other node in the network.

To compare the conductances for the two topologies, suppose that in both cases, the links from a given node to different nodes are equally weighted. So, for the ring graph, let $\mathbf{C}_{ij} = \mathbf{C} = \frac{1}{4}$, for all $i \neq j$; for the fully connected graph, let $\mathbf{C}_{ij} = \mathbf{C} = \frac{1}{n}$, for all $i \neq j$. Assume that for the ring graph, $\mathbf{C}_{ii} = \frac{1}{2}$. If the channels were erasure channels, this would be the probability

that node i makes contact with no other nodes. For the fully connected graph, let $\mathbf{C}_{ii} = \frac{1}{n}$. So, in both cases, we have that the sum of the capacities of channels leaving a node is 1, $\sum_j \mathbf{C}_{ij} = 1$.

Using definition III.4 and some straightforward simplifications we have that for the ring, $\Phi(G) = \frac{1}{n}$. For the fully connected graph we have $\Phi(G) = \frac{1}{2}$. For two networks with the same number of nodes, the network with the more severe topological constraints has smaller conductance. In general, for a ring graph, we have $\Phi(G) = O(\frac{1}{n})$, while for a fully connected graph we have $\Phi(G) = O(1)$.

Remark: In both of these examples, conductance scales like the reciprocal of diameter. These examples were chosen to illustrate that conductance does capture the topological properties of the network. In general, however, conductance and diameter are not the same.

Conductance is the natural generalization that captures the infomation bottleneck. Conductance depends on both channel capacities and topology of the network while diameter is purely topological property. Generally, the channel capacities will cause conductance and diameter to be different.

To illustrate this, consider a complete graph with $\mathbf{C}_{ij} = \mathbf{C}_{ji} = 1/n^2$ instead of $\mathbf{C}_{ij} = \mathbf{C}_{ji} = 1/n$ in the above example. In this case, the diameter is still 1 but the reciprocal of conductance will be n. Here, our bound is a much better lower bound than a diameter based lower bound.

More generally, different edges may have different channel capacities, in which case conductance and diameter will again be very different. For example, if only one of the nodes of a complete graph had incident edges with capacities $1/n^2$ while all the rest had capacity $1/n$, conductance again evaluates to n. This node creates the information bottleneck in the network, and this is captured by the conductance.

B. Comparison With Iterative Algorithms

A popular approach for computing a linear function of the initial values is based on linear iterations. If nodes can communicate real numbers between them in each time instance, the computation time for a linear iterative algorithm based on a doubly stochastic matrix P is proportional to the mixing time of the matrix, $\mathcal{H}(P)$[4]. As noted earlier, the mixing time $\mathcal{H}(P)$, hence computation time of iterative algorithm, is bounded as

$$\frac{1}{\Phi(P)} \leq \mathcal{H}(P) \leq O\left(\frac{\log n}{\Phi^2(P)}\right).$$

Therefore, in order to obtain a fast iterative algorithm, P must have a small mixing time $\mathcal{H}(P)$. The standard approach of finding such a P is based on the method of Metropolis [21] and Hastings [13]. This method does indeed yield a symmetric and doubly stochastic P on G.

For *expander* graphs, the resulting P induced by the Metropolis-Hastings method is likely to have $\Phi(P) = \Theta(1)$. Hence, the mixing time is $O(\log n)$, and this is *essentially* the fastest possible mixing time. For example, the P for a complete graph will be $P = [1/n]$, and it has $\Phi(P) = \Theta(1)$. In this case, both our algorithm and the linear-iteration algorithms, based the Metropolis-Hastings induced P, will have essentially optimal computation time. It should be noted that our algorithm, described later, is quantized. On the other hand, a quantized version of the linear iterative algorithm is far from obvious and subject of recent interest and on-going research. To the best of our knowledge, how to optimally deal with finite-rate constraints in conjunction with the linear iterative updates is an open question.

Certain graph topologies of interest do possess *geometry* and are far from being expanders. Examples of these graphs include those arising in wireless sensor network deployed in some geographic area [4], [8] or a nearest neighbor network of unmanned vehicles [27]. The simplest example of a graph with geometry is the *ring* graph that we considered above. The Metropolis-Hastings method will lead to a P as discussed in Section IV-A. It has $\Phi(P) = \Theta(1/n)$. But it is known that for this topology, mixing time scales like $\Omega(n^2)$, at the least. That is, mixing time scales like $1/\Phi^2(P)$ and not $1/\Phi(P)$. More generally, for any symmetric P, the mixing time is known to be at least n^2 (e.g., see [4]). Thus, the linear iterative algorithms based on a symmetric P have computation time that scales like n^2. In contrast, our quantized algorithm will have computation time that scales with n (which is $1/\Phi(P)$) for the ring. Now the diameter of the ring graph is n and no algorithm takes less than n or no P can have mixing time smaller than this diameter n.

In general, it can be checked that the diameter of a graph G is at most $1/\Phi(P)$ for any irreducible probability matrix P. For graphs with bounded degree and with geometry, the P induced by the Metropolis-Hastings method has a diameter that scales like $1/\Phi(P)$. By a graph with geometry, we mean a graph with polynomial growth: for any given node, the number of nodes within distance r from that node scales as $O(r^d)$ for some fixed constant d. Diaconis and Saloff-Coste [7] have established that for graphs with geometry the mixing time of any symmetric doubly stochastic P scales like at least D^2, where D is the diameter of the graph G. Therefore, linear iterative algorithms will have computation time that scales like D^2. In contrast, our algorithm will have computation time $1/\Phi(P)$ which will be equal to diameter D for a P given by the Metropolis-Hastings method.

In summary, our algorithm will provide the best possible computation time scaling with respect to graph structure for both expander graphs and graphs with geometry.

V. Proof of Theorem III.1

In this section, we present the proof of Theorem III.1. The core idea is to characterize the information flow between arbitrary "cut-sets" of the network. A cut divides the network into two sets, S and $S^c = \{1, \ldots n\} \backslash S$. Suppose that nodes 1 to m belong to set S and nodes $m+1$ to n belong to set S^c. So, the estimates of the nodes in set S at time T are $\widehat{Y}_S(T) = [\widehat{Y}_1(T) \ \ldots \ \widehat{Y}_m(T)]'$. The initial conditions of the nodes in sets S and S^c are denoted by $X_S = [X_1 \ \ldots \ X_m]'$ and $X_{S^c} = [X_{m+1} \ \ldots \ X_n]'$.

The quantity that will play a central role in the proof of Theorem III.1 is the mutual information term, $I(\widehat{Y}_S(T); X_{S^c}|X_S)$. This is mutual information between the estimates of the nodes in set S and the initial conditions of the nodes in set S^c, assuming that all nodes in S have each other's initial conditions. Leading

up to the proof of Theorem III.1, we prove 3 lemmas related to $I(\widehat{Y}_S(T); X_{S^c}|X_S)$.

In the first of our series of lemmas, we bound $I(\widehat{Y}_S(T); X_{S^c}|X_S)$ from above by the mutual information between the inputs and the outputs of the channels that traverse the cut.

Lemma V.1: For a given cut in the network, and corresponding cut-sets S^c and S

$$I(\widehat{Y}_S(T); X_{S^c}|X_S) \leq \sum_{l=1}^{T} I(V_S(l); U_{S^c}(l)|U_S(l))$$

where U_{S^c} is a vector of the variables transmitted by the encoders of the nodes in S^c and V_S is a vector of the variables received via channels by the decoders of the nodes in S. The (l) refers to the l^{th} channel use.

In the second lemma, we bound from above $I(V_S(l); U_{S^c}(l)|U_S(l))$ by the sum of the capacities of the channels traversing the cut.

Lemma V.2: Suppose a network is represented by the graph $G = (V, E)$. The edges of the graph represent channels with positive capacity. If the channels connecting the nodes are memory-less and independent, then

$$I(V_S(l); U_{S^c}(l)|U_S(l)) \leq \sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij}.$$

The proof of this lemma makes apparent the value of the conditioning in the mutual information terms. This conditioning is equivalent to assuming that all nodes in S have access to all information that is available at the nodes of the set S, including information about X_S. In this way, we capture the information that is traversing the cut, without including the effect of information exchanged between nodes in the same set.

Finally, in the third lemma, we bound from below the term $I(\widehat{Y}_S(T); X_{S^c}|X_S)$. We show that this term is bounded from below by the information that must be communicated from the nodes of S^c to the nodes of S in order for the nodes of S to compute their estimates, $I(\widehat{Y}_S(T); Y_S|X_S)$. We then bound this from below by an expression that involves the desired performance criterion and the desired function.

For the mean-square error criterion R1, we have the following lemma.

Lemma V.3: If $E(||\underline{\widehat{Y}}(T) - \underline{Y}||^2) \leq \beta 2^{-\alpha}$ then

$$I(\widehat{Y}_S(T); X_{S^c}|X_S) \geq L(S)$$

where

$$L(S) = h(Y_S|X_S) - \frac{|S|}{2}\log 2\pi e\beta + \frac{|S|}{2}\log|S| + |S|\frac{\alpha}{2}$$

and, $|S|$ is the size of the set S, specifically, $|S| = m$.

The lower bound involves two terms. These are (1) the desired accuracy in the nodes' estimates, specified by the mean-square error criterion, and (2) the uncertainty in the function to be estimated, Y_S, quantified by its differential entropy. The larger the desired accuracy, the larger the α in the mean-square error criterion. This implies a larger lower bound on the information that must be conveyed. Also, the larger the uncertainty in the function to be learned by the nodes in set S, the larger the differential entropy term. Hence, the lower bound is larger.

For the mean-square error criterion R2, we have the following corollary.

Corollary V.4: If, for all $i \in \{1, \ldots, n\}$, $E(\widehat{Y}_i(T) - Y_i)^2 \leq 2^{-\alpha}$, then

$$I(\widehat{Y}_S(T); X_{S^c}|X_S) \geq \bar{L}(S)$$

where $\bar{L}(S) = h(Y_S|X_S) - \frac{|S|}{2}\log 2\pi e + |S|\frac{\alpha}{2}$.

When, for all i, $E(\widehat{Y}_i(T) - Y_i)^2 \leq 2^{-\alpha}$, we again have a lower bound that depends on the desired accuracy and the uncertainty in the function to be estimated. However, $\bar{L}(S)$ is smaller than $L(S)$ due to the weaker error requirement of R2.

The proofs of Lemma V.1 and V.2 are in Appendix A. In the next sections, we prove Lemma V.3 and Corollary V.4. Then, we prove Theorem III.1.

A. Proof of Lemma V.3 and Corollary V.4

Recall that the lemma stated that if $E(||\underline{\widehat{Y}}(T) - \underline{Y}||^2) \leq \beta 2^{-\alpha}$ then

$$I(\widehat{Y}_S(T); X_{S^c}|X_S) \geq L(S)$$

where

$$L(S) = h(Y_S|X_S) - \frac{|S|}{2}\log 2\pi e\beta + \frac{|S|}{2}\log|S| + |S|\frac{\alpha}{2}$$

and $|S|$ is the size of the set S, specifically, $|S| = m$.

We start the proof by observing the following:

$$\begin{aligned} &I(\widehat{Y}_S(T); X_{S^c}|X_S) \\ &\quad \overset{(a)}{=} I(\widehat{Y}_S(T); \underline{X}|X_S) \\ &\quad \overset{(b)}{\geq} I(\widehat{Y}_S(T); Y_S|X_S) \end{aligned}$$

where

a) that is, $I(W; Y, U|U) = I(W; Y|U)$, can be verified by the chain rule for mutual information

$$\begin{aligned} I(W; Y, U|U) &= I(W; Y|U) + I(W; U|U, Y) \\ &= I(W; Y|U) \end{aligned}$$

because $I(W; U|U, Y) = 0$.

b) follows by the data processing inequality, because $Y_i = f_i(\underline{X})$.

Second, we obtain a lower bound on $I(\widehat{Y}_S(T); Y_S|X_S)$ in terms of the desired mean-square criterion. We have the following series of inequalities:

$$\begin{aligned} &I(\widehat{Y}_S(T); Y_S|X_S) \\ &\quad = h(Y_S|X_S) - h(Y_S|\widehat{Y}_S(T), X_S) \\ &\quad = h(Y_S|X_S) - h(Y_S - \widehat{Y}_S(T)|\widehat{Y}_S(T), X_S) \\ &\quad \overset{(c)}{\geq} h(Y_S|X_S) - h(Y_S - \widehat{Y}_S(T)) \end{aligned} \tag{1}$$

where (c) follows because conditioning reduces entropy.

Now, because the multivariate Normal maximizes entropy over all distributions with the same covariance

$$h(\widehat{Y}_S(T) - Y_S) \leq \frac{1}{2}\log(2\pi e)^m |Z| \tag{2}$$

where, Z is a covariance matrix whose diagonal elements are $Z_{ii} = Var(\widehat{Y}_i(T) - Y_i)$, and $|Z|$ denotes the determinant. Recall that S is the set containing nodes 1 to m, so it has size m. Also, $\widehat{Y}_S(T) - Y_S$ is a vector of length m. So, Z is an m by m matrix. Now

$$\begin{aligned} |Z| &\overset{(d)}{\leq} \prod_{i=1}^{m} Var(\widehat{Y}_i(T) - Y_i) \\ &\leq \prod_{i=1}^{m} E(\widehat{Y}_i(T) - Y_i)^2 \\ &\overset{(e)}{\leq} \left(\frac{\beta 2^{-\alpha}}{m}\right)^m. \end{aligned} \tag{3}$$

Here, (d) is due to Hadamard's inequality [5, Ch. 9]. To see (e), we have the following proposition.

Proposition V.5: For $\gamma > 0$, subject to $\sum_{i=1}^m y_i \leq \gamma$ and $y_i \geq 0$, $\prod_{i=1}^m y_i$ is maximized when $y_i = \frac{\gamma}{m}$.

Now, (e) follows by setting $y_i = E(\widehat{Y}_i(T) - Y_i)^2$ and observing that

$$\begin{aligned} \sum_{i=1}^{m} y_i &= E(\|\widehat{Y}_S(T) - Y_S\|^2) \\ &\leq E(\|\widehat{\underline{Y}}(T) - \underline{Y}\|^2) \\ &\leq \beta 2^{-\alpha} \end{aligned}$$

where the last inequality follows by the assumption of our lemma.

Finally, using (3) and (2), we bound (1) from below and obtain $L(S)$. ∎

Proof of Corollary V.4: Recall that in this corollary, we had the weaker condition that for all $i \in \{1, \ldots, n\}$, $E(\widehat{Y}_i(T) - Y_i)^2 \leq 2^{-\alpha}$. In this case, we show that we have the smaller lower bound,

$$\bar{L}(S) = h(Y_S|X_S) - \frac{|S|}{2}\log 2\pi e + |S|\frac{\alpha}{2}..$$

To see this, observe that $E(\widehat{Y}_i(T) - Y_i)^2 \leq 2^{-\alpha}$ implies $E(\|\widehat{Y}_S(T) - Y_S\|^2) \leq |S|2^{-\alpha}$. So, replacing β in $L(S)$ of the previous lemma by $|S|$ yields the desired result. ∎

B. Proof of Theorem III.1

The proof proceeds in several steps. First, as shown in Lemma V.1, for a given cut in the network and corresponding cut-sets S^c and S

$$I(\widehat{Y}_S(T); X_{S^c}|X_S) \leq \sum_{l=1}^{T} I(V_S(l); U_{S^c}(l)|U_S(l)) \tag{4}$$

where U_{S^c} is a vector of the variables transmitted by the encoders of the nodes in S^c and V_S is a vector of the variables received via channel by the decoders of the nodes in S.

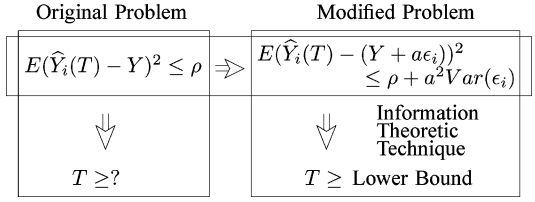

Fig. 2. Diagram illustrating the use of the information theoretic technique to obtain a lower bound in a situation where all nodes learn the same function.

Second, by Lemma V.2, because we have assumed that the channels connecting the nodes are memory-less and independent

$$I(V_S(l); U_{S^c}(l)|U_S(l)) \leq \sum_{i \in S^c}\sum_{j \in S} \mathbf{C}_{ij}. \tag{5}$$

Third, we combine (4) and (5) with Corollary V.4 to obtain

$$T \geq \frac{\bar{L}(S)}{\sum_{i \in S^c}\sum_{j \in S} \mathbf{C}_{ij}}. \tag{6}$$

Finally, we have that

$$T \geq \max_{S \subset V} \frac{\bar{L}(S)}{\sum_{i \in S^c}\sum_{j \in S} \mathbf{C}_{ij}}$$

because (6) holds for any cut. ∎

C. A Technical Difficulty and Its Resolution

Making use of the lower bounds derived above involves computing the differential entropy of the random variables to be learned in the network, specifically, $h(Y_S|X_S)$, where $Y_S = [Y_1 \ \ldots \ Y_m]'$. If the Y_i's are different random variables, then the differential entropy term is well-defined. However, if two entries of Y_S are the same random variable, for example if both are $f(\underline{X})$, then $h(Y_S|X_S)$ will be $-\infty$.

But, our technique and lower bound can still be used in situations where all nodes need to learn the same function of the initial conditions. In order to have a nontrivial lower bound, we modify the problem slightly. We introduce auxiliary random variables associated with the nodes of set S^c, to be learned by nodes in S. This enables us to obtain a nontrivial lower bound for the modified problem. This is also a lower bound for the original problem. By proper choice of the auxiliary random variables, the lower bound of the modified problem can be made as large as possible, and hence the best possible approximation for the lower bound of the original problem. This procedure is illustrated in Fig. 2.

The aforementioned technique will be used in the next section. In the examples below, we demonstrate the computation of $h(Y_S|X_S)$ when we introduce the auxiliary random variables.

Example V.6 (The Solution): Let nodes $\{1, \ldots, m\}$, $m \leq n/2$, belong to set S, so that $Y_S = [Y_1 \ \ldots \ Y_m]$.′ Let $Y_1 = f(\underline{X})$ and $Y_i = f(\underline{X}) + a_i\epsilon_{j_i}$ for $i \in \{2, \ldots, m\}$. One can think of ϵ_{j_i} being associated with a node in set S^c, that is, $j_i \in \{m+1, \ldots n\}$. So, node j_i's initial condition would be $(X_{j_i}, \epsilon_{j_i})$.

Furthermore, we assume that f is separable, meaning $f(\underline{X}) = f_S(X_S) + f_{S^c}(X_{S^c})$. Finally, we assume that the X_i's and ϵ_i's are mutually independent. Then

$$\begin{aligned} &h(Y_S|X_S) \\ &\quad = h\left(f_{S^c}(X_{S^c}), f_{S^c}(X_{S^c}) + a_2\epsilon_{j_2}, \right. \\ &\qquad \left. \ldots, f_{S^c}(X_{S^c}) + a_m\epsilon_{j_m}|X_S\right) \\ &\quad \overset{(a)}{=} h(f_{S^c}(X_{S^c}), f_{S^c}(X_{S^c}) + a_2\epsilon_{j_2}, \\ &\qquad \ldots, f_{S^c}(X_{S^c}) + a_m\epsilon_{j_m}) \\ &\quad \overset{(b)}{=} h(f_{S^c}(X_{S^c})) + \sum_{i=2}^{m} h(a_i\epsilon_{j_i}) \\ &\quad \overset{(c)}{=} h(f_{S^c}(X_{S^c})) + \sum_{i=2}^{m} h(\epsilon_{j_i}) + \log\prod_{i=2}^{m}|a_i| \end{aligned}$$

where

a) follows because we have assumed that the X_i's and ϵ_i's are mutually independent;
b) follows by the chain rule for differential entropy, and again using the fact that the X_i's and ϵ_i's are mutually independent;
c) follows using the fact that $h(a_i\epsilon_{j_i}) = h(\epsilon_{j_i}) + \log|a_i|$, as shown in shown in [5, Ch. 9].

In the next example, we assume that the function f is a linear function and that the auxiliary random variables are independent Gaussian random variables. For this scenario, we then obtain the expression for the lower bound of Corollary V.4.

Example V.7 (Using the Solution for a Linear Function): In addition to the assumptions in Example V.6, let $f(\underline{X}) = \sum_{j=1}^{n}\beta_j X_j$. We assume that $\epsilon_{j_2}, \ldots, \epsilon_{j_m}$ are independent and identically distributed Gaussian random variables, with mean zero and variance η. Then, the differential entropy of ϵ_{j_i} is $h(\epsilon_{j_i}) = \frac{1}{2}\log 2\pi e\eta$.

So, substituting in the expression from Example V.6, we have that

$$h(Y_S|X_S) = h\left(\sum_{j\in S^c}\beta_j X_j\right) + \frac{m-1}{2}\log 2\pi e\eta + \log\prod_{i=2}^{m}|a_i|. \quad (7)$$

To evaluate $h\left(\sum_{j\in S^c}\beta_j X_j\right)$, we use the Entropy Power Inequality, namely, for independent X_i's

$$2^{2h\left(\sum_{j\in S^c}\beta_j X_j\right)} \geq \sum_{j\in S^c} 2^{2h(\beta_j X_j)}$$

which implies that

$$h\left(\sum_{j\in S^c}\beta_j X_j\right) \geq \frac{1}{2}\log\left(\sum_{j\in S^c} 2^{2h(\beta_j X_j)}\right).$$

Now, if we assume that each X_i is uniformly distributed in the interval between 1 and $B+1$, $X_i \sim U[1, B+1]$, then

$$h(\beta_j X_j) = \log|\beta_j| B.$$

So

$$\begin{aligned} h\left(\sum_{j\in S^c}\beta_j X_j\right) &\geq \frac{1}{2}\log\left(B^2\sum_{j\in S^c}\beta_j^2\right) \\ &= \log B + \frac{1}{2}\log\sum_{j\in S^c}\beta_j^2. \end{aligned} \quad (8)$$

Finally, we evaluate the lower bound of Corollary V.4 for this scenario. Recall that we had

$$\bar{L}(S) = h(Y_S|X_S) - \frac{|S|}{2}\log 2\pi e + |S|\frac{\alpha}{2}$$

and $|S| = m$. Using (7) together with the inequality of (8), we have that

$$\bar{L}(S) \geq \log\frac{B\left(\sum_{j\in S^c}\beta_j^2\right)^{\frac{1}{2}}\prod_{i=2}^{m}|a_i|}{\sqrt{2\pi e\eta}} + \frac{m}{2}(\alpha + \log\eta). \quad (9)$$

In summary, our use of basic information theoretic definitions and inequalities has led to a lower bound that we have applied to a formulation for distributed function computation. The lower bound on information consists of a term that arises due to the mean-square error criterion and a term due to the function that is to be estimated. Using techniques of network information theory, we have shown how the bound on information can be used to obtain a lower bound on computation time time.

VI. A Tight Bound: Computation of the Sum via Erasure Channels

In this section, we use the techniques of the previous section to find a lower bound on computation time when nodes compute a sum. We present a distributed algorithm for computation of the sum over block erasure channels and provide an upper bound for the run-time of the algorithm. Both bounds depend inversely on conductance, which captures the limitations due to the network topology. Therefore, we conclude that our lower bound is tight in capturing the effect of the network topology via the conductance.

A. Information-Theoretic Lower Bound for Summation

In this section, we provide the proof of Theorem III.5. We will use the techniques that we have developed in Section V. In particular, we will use the results of Examples V.6 and V.7, namely (9).

Proof of Theorem III.5: Recall that $Y = \sum_{j=1}^{n}\beta_j X_j$. Suppose that we have any realization of the initial conditions, $A = \{X_1 = x_1, \ldots, X_n = x_n\}$. We are given an algorithm that guarantees, for every such realization, that at time T each

node, i, has an estimate, $\widehat{Y}_i(T)$, of Y: $\sum_{j=1}^{n}\beta_j x_j$. Furthermore, for this algorithm, the estimate $\widehat{Y}_i(T)$ is within an ε-interval of the true value of Y, with desired probability. That is

$$\mathbf{P}\left(|\widehat{Y}_i(T)-Y|\leq \varepsilon Y\Big|A\right)\geq 1-\delta. \tag{10}$$

The proof proceeds in several steps. The proofs for steps 1 and 2 follow this proof.

1) Any algorithm that satisfies the probability condition of (10) must satisfy, for small enough δ, a mean-square error criterion:

$$E(\widehat{Y}_i(T)-Y)^2\leq \varepsilon^2 E(Y^2)+\kappa\delta.$$

2) Let $Y_1=Y$ and $Y_i=Y+a\epsilon_{j_i}$ for $i\in\{2,\ldots,m\}$, where $\epsilon_{j_2},\ldots,\epsilon_{j_m}$ are independent and identically distributed Gaussian random variables, with mean zero and variance η. Let the ϵ_{j_i}'s be independent of the initial conditions, X_i. Then,

$$E(\widehat{Y}_i(T)-Y_i)^2\leq \varepsilon^2 E(Y^2)+a^2\eta+\kappa\delta.$$

3) Next, let S^* and $(S^*)^c$ be the sets for which

$$\frac{\sum_{i\notin S,j\in S}\mathbf{C}_{ij}}{|S|}$$

is minimized, and assume S^* is the set with smaller size, $|S^*|\leq\frac{n}{2}$. For purposes of this proof, we enumerate the nodes in set S^* from 1 to m. Then, let $Y_{S^*}=[Y_1\ \ldots\ Y_m]'$, where the Y_i's are those of Step 2.

4) Now, we can apply our information theoretic inequalities to this set-up. We think of ϵ_{j_i} being associated with a node in set $(S^*)^c$, that is, $j_i\in\{m+1,\ldots n\}$. So, node j_i's initial condition would be (X_{j_i},ϵ_{j_i}). Denote $[\epsilon_{j_1}\ \ldots\ \epsilon_{j_m}]$ by ϵ. Using the derivations of Section V, we have that

$$\begin{aligned}T\sum_{i\in(S^*)^c}\sum_{j\in S^*}\mathbf{C}_{ij}&\geq I(\widehat{Y}_{S^*}(T);X_{(S^*)^c}|X_{S^*})\\&\overset{(a)}{=} I(\widehat{Y}_{S^*}(T);X_{(S^*)^c},\epsilon|X_{S^*})\\&\geq I(\widehat{Y}_{S^*}(T);Y_{S^*}|X_{S^*})\\&\geq \bar{L}(S^*),\end{aligned}$$

where, (a) follows because $\widehat{Y}_{S^*}(T)$ is the vector of estimates produced by the algorithm, and depends on the initial conditions, X_i's, while the ϵ_{j_i}'s are independent of X_i's. Recall that

$$\bar{L}(S^*)=h(Y_{S^*}|X_{S^*})-\frac{|S^*|}{2}\log 2\pi e+|S^*|\frac{\alpha}{2}.$$

Note that from Step 2, we have that $2^{-\alpha}=\varepsilon^2E(Y^2)+a^2\eta+\kappa\delta$. So, we have $\alpha=-\log(\varepsilon^2E(Y^2)+a^2\eta+\kappa\delta)$.

5) Next, we compute $h(Y_{S^*}|X_{S^*})$ given the assumptions of our formulation. Recall that we have performed these computations in Example V.7. We obtained the following:

$$\begin{aligned}\bar{L}(S^*)\geq\log\frac{B\left(\sum_{j\in S^c}\beta_j^2\right)^{\frac{1}{2}}|a|^{m-1}}{\sqrt{2\pi e\eta}}\\+\frac{|S^*|}{2}\left(\log\frac{\eta}{\varepsilon^2E(Y^2)+a^2\eta+\kappa\delta}\right),\end{aligned}$$

where we have substituted in $\alpha=-\log(\varepsilon^2E(Y^2)+a^2\eta+\kappa\delta)$.

6) Finally, we make the appropriate choice of our parameters, a and η. Assume, without loss of generality, that

$$\left(\frac{\sum_{j\in S^c}\beta_j^2}{2\pi e}\right)^{\frac{1}{2}}\geq 1$$

otherwise, we can just scale our choices for a and η. Let $a=\left(\frac{\eta^{\frac{1}{2}}}{B}\right)^{\frac{1}{m-1}}$, then

$$\bar{L}(S^*)\geq\frac{|S^*|}{2}\left(\log\frac{1}{\frac{\varepsilon^2E(Y^2)}{\eta}+a^2+\frac{\kappa}{\eta}\delta}\right).$$

Next, let $\eta=B$. Then, because $m-1<\frac{n}{2}$,

$$a^2<\left(\frac{1}{B}\right)^{\frac{2}{n}}.$$

Observe that $E(Y^2)\leq MB^2$, where M is some integer. So,

$$\frac{\varepsilon^2E(Y^2)}{\eta}+a^2\leq\varepsilon^2MB+\left(\frac{1}{B}\right)^{\frac{2}{n}}.$$

Combining with Step 4, we have that

$$T\sum_{i\in(S^*)^c}\sum_{j\in S^*}\mathbf{C}_{ij}\geq\frac{|S^*|}{2}\log\frac{1}{\varepsilon^2MB+\left(\frac{1}{B}\right)^{\frac{2}{n}}+\frac{\kappa}{B}\delta}.$$

Rearranging, we have that

$$T\geq\frac{1}{2}\frac{1}{\frac{\sum_{i\in(S^*)^c}\sum_{j\in S^*}\mathbf{C}_{ij}}{|S^*|}}\log\frac{1}{\varepsilon^2MB+\left(\frac{1}{B}\right)^{\frac{2}{n}}+\frac{\kappa}{B}\delta}.$$

Here, we must have $\varepsilon^2M\in\left[0,\frac{1}{B}\left(1-\left(\frac{1}{B}\right)^{\frac{2}{n}}-\kappa\delta\right)\right)$, in order for the lower bound to be positive.

Finally, because we had chose our S^* such that $\frac{\sum_{i\in(S)^c}\sum_{j\in S}\mathbf{C}_{ij}}{|S|}$ is minimized, we have that

$$\widetilde{\Phi}(G)=\frac{\sum_{i\in(S^*)^c}\sum_{j\in S^*}\mathbf{C}_{ij}}{|S^*|}.$$

■

Remark: We show in the next section that our lower bound is tight in its reciprocal dependence on the conductance term.

So, for fixed n, we have a scaling law that is tight in the case of severe communication constraints, such as very small channel capacities due to low transmission power.

In the case of increasing number of nodes, however, B must increase exponentially with n for our lower bound to remain valid. The requirement is a by-product of using a formulation based on random variables together with Information Theoretic variables. This requirement ensures that as n increases, our bound properly captures the number of bits that are transferred.

When we consider sums of independent identically distributed random variables, Central Limit Theorem type arguments imply that as the number of the random variables increases, there is some randomness lost, because we know that the distribution of the sum must converge to the Normal distribution. However, in a setting where the initial conditions are fixed values, as in the case of the algorithm we describe below, the addition of a node clearly will not reduce the information that needs to be communicated in the network. To counterbalance the probabilistic effects, we need to have B increase as the number of nodes increases.

Next, we complete the proof of Theorem III.5 by proving the statements of Step 1 and Step 2.

Proof of Step 1: We show that for small enough δ, $\mathbf{P}\left(|\widehat{Y}_i(T)-Y|\leq \varepsilon Y \middle| A\right) \geq 1-\delta$ implies $E(\widehat{Y}_i(T)-Y)^2 \leq \varepsilon^2 E(Y^2)+\kappa\delta$.

First, observe that,

$$\mathbf{P}\left(|\widehat{Y}_i(T)-Y|\geq \varepsilon Y \middle| A\right)\leq \delta,$$

is equivalent to

$$\mathbf{P}\left((\widehat{Y}_i(T)-Y)^2\geq \varepsilon^2 Y^2 \middle| A\right)\leq \delta,$$

Next, when we condition on A, Y is a fixed number. So, we have we have that

$$\begin{aligned}
&E\left((\widehat{Y}_i(T)-Y)^2\middle|A\right)\\
&=\int_0^\infty \mathbf{P}\left((\widehat{Y}_i(T)-Y)^2\geq x\middle|A\right)dx\\
&=\int_0^{\varepsilon^2Y^2} \mathbf{P}\left((\widehat{Y}_i(T)-Y)^2\geq x\middle|A\right)dx\\
&\quad+\int_{\varepsilon^2Y^2}^\infty \mathbf{P}\left((\widehat{Y}_i(T)-Y)^2\geq x\middle|A\right)dx\\
&\leq \varepsilon^2Y^2+\delta\kappa,
\end{aligned}$$

where the last inequality follows

- for the first term, because $\mathbf{P}\left((\widehat{Y}_i(T)-Y)^2\geq x\middle|A\right)\leq 1$, and,
- for the second term, because $\mathbf{P}\left((\widehat{Y}_i(T)-Y)^2\geq x\middle|A\right)\leq \delta$ for all $x\in[\varepsilon^2Y^2,\infty)$. We have also assumed that for every A, $(\widehat{Y}_i(T)-Y)^2$ is bounded from above.

Finally, we have that

$$E(\widehat{Y}_i(T)-Y)^2=E\left(E\left((\widehat{Y}_i(T)-Y)^2\middle|A\right)\right),$$

where the outermost expectation is with respect to the joint distribution of the initial conditions. ■

Proof of Step 2: We show that if $E(\widehat{Y}_i(T)-Y)^2\leq \varepsilon^2E(Y^2)+\kappa\delta$, then $E(\widehat{Y}_i(T)-Y_i)^2\leq \varepsilon^2E(Y^2)+a^2\eta+\kappa\delta$, where $Y_i=Y+a\epsilon_{j_i}$, and ϵ_{j_i} has mean zero and variance η and is independent of all the X_i's.

$$\begin{aligned}
&E(\widehat{Y}_i(T)-Y_i)^2\\
&=E(\widehat{Y}_i(T)-Y-a\epsilon_{j_i})^2\\
&=E(\widehat{Y}_i(T)-Y)^2+E(a\epsilon_{j_i})^2-2E(\widehat{Y}_i(T)-Y)(a\epsilon_{j_i})\\
&\overset{(a)}{=}E(\widehat{Y}_i(T)-Y)^2+E(a\epsilon_{j_i})^2-2E(\widehat{Y}_i(T)-Y)E(a\epsilon_{j_i})\\
&\overset{(b)}{=}E(\widehat{Y}_i(T)-Y)^2+E(a\epsilon_{j_i})^2
\end{aligned}$$

where

(a) follows because $\widehat{Y}_i(T)$ is the estimate produced by the algorithm, and depends on the initial conditions, X_i's, while ϵ_{j_i} is independent of X_i's, and,

(b) follows because ϵ_{j_i} has mean zero.

■

B. An Algorithm for Summation via Block Erasure Channels

Next, we describe the algorithm that achieves the lower bound. That is, we exhibit the reciprocal dependence of the algorithm's computation time on the conductance of the graph. Because the function that is to be computed, the sum, is relatively simple, and the algorithm requires little computation overhead, the limitations that arise are due primarily to the communication constraints. In fact, the dependence on the algorithm's run-time on conductance arises due to the fact that the algorithm uses an information spreading algorithm as a subroutine. Information spreading depends reciprocally on conductance: the more severe the connectivity constraints, the smaller the conductance and the longer it takes for information to spread in the network.

The algorithm that we describe is based on an algorithm by Mosk-Aoyama and Shah [23]. In Section VI-B1 we discuss this algorithm and its applicability to our formulation. In Section VI-B2, we describe the contributions of [23] in the design of an algorithm for distributed computation of a separable function, in a network of nodes using repeated communication of real-valued messages. In Section VI-B3, we describe the algorithm when the communicated messages are quantized, and analyze how the performance of the algorithm changes relative to the performance of the unquantized algorithm of [23].

1) Background: The algorithm that we describe is based on an algorithm by Mosk-Aoyama and Shah [23]. In that formulation, each node has a fixed real-valued initial condition, that is bounded away from zero. Nodes compute a separable function[1] of the initial values in the network. The algorithm guarantees that with some specified probability, all nodes have an estimate of the function value within a desired ε-interval of accuracy around the true value. In [23], each node may contact one of its neighbors once in each time slot. If the edge (i,j) belongs to E, node i sends its real-valued message to node j with probability p_{ij} and with probability p_{ii} sends its message to no other nodes; if $(i,j)\notin E$, $p_{ij}=0$.

[1]A linear function of the initial conditions is a separable function.

The algorithm of [23] is a simple randomized algorithm that is based on each node generating an exponentially distributed random variable with mean equal to the reciprocal of the node's initial value. The nodes sample from their respective distributions and make use of an information spreading algorithm to make computations and ultimately obtain an estimate of the desired function.

The advantage of this algorithm is that it is completely distributed. Nodes need not keep track of the identity of the nodes from which received information originates. Furthermore, the algorithm is not sensitive to the order in which information is received. In terms of its performance, the algorithm's computation time is almost optimal in its dependence on the network topology, as the computation time scales inversely with conductance of the matrix representing the communication topology.

The drawback of the algorithm in [23], however, is that it requires nodes to exchange real numbers. As such, the algorithm is not practically implementable.

Below, we quantize this algorithm, so that instead of sending real-valued messages, nodes communicate an appropriate number of bits. In the process of quantization, we determine the needed number of bits; for now, we call it $\log M$. Now, node i can send to j a $\log M$-bit message each time it makes contact. Again, the contact between the nodes is random: node i contacts node j with probability p_{ij}. This is equivalent[2] to node i communicating to j via a $\log M$-bit erasure channel, where $\log M$ bits are sent noiselessly with probability p_{ij}, and there is an erasure otherwise. In this case, capacity of the channel is $\mathbf{C}_{ij} = p_{ij} \log M$, so, $\Phi(G) = \Phi(P) \log M$. We will show that the effect of communicating bits instead of real-valued messages is to slow down the original algorithm by $\log n$; however, the dependence of computation time on conductance is unchanged.

Another difference between our formulation and the one in [23], is that we assume that the initial conditions lie in a bounded interval, $[1, B+1]$, whereas in [23] there is no upper bound. We need this assumption to show that our algorithm will also guarantee that with some specified probability, all nodes have an estimate of the function value within a desired ε-interval of accuracy around the true value. However, due to communicating a finite number of bits, ε cannot be arbitrarily close to zero.

Finally, we recall that in deriving the lower bound of the previous section, we had assumed a joint probability distribution on the initial conditions. However, we will describe the algorithm for fixed initial-values at the nodes. If the initial conditions were in fact distributed according to some joint probability density function, the algorithm that we describe below can be used for any realization of the initial values to guarantee, with the desired probability, the ε-accuracy criterion. So, the algorithm satisfies the "if" condition in the statement of Theorem III.5. As such, the computation time of the algorithm we describe below must be bounded from below by the expression in Theorem III.5 which includes the reciprocal of conductance.

[2]In [23], it is assumed that each node can contact at most one other node; but it can be contacted by more than one nodes. Under our independent erasure channel model, each node can contact more than one node. However, for our purposes, this is only beneficial as it results in faster information dissemination.

We provide an upper bound on the run-time and show that, indeed, it does scale inversely with conductance. Thus, the contribution of this work includes the non-trivial quantized implementation of the algorithm of [23] and its analysis. As a consequence, we obtain the fastest, in terms of dependence on network topology, quantized distributed algorithm for separable function computation.

2) Unquantized Function Computation: In [23], a randomized algorithm is proposed for distributed computation of a separable function of the data in the network, so that with some specified probability, all nodes have an estimate of the function value within the desired interval of accuracy. The computation algorithm assumes that the nodes exchange real-valued messages whenever a communication takes place. The algorithm depends on:

- the properties of exponentially distributed random variables;
- an information spreading algorithm used as a subroutine for the nodes to communicate their messages and determine the minimum of the messages.

a) The Algorithm: The following property of exponential random variables plays a central role in the design of this algorithm. Let $W^1, \ldots, W^n$ be independent exponentially distributed random variables, where W^i has mean $1/\theta_i$. Then, the minimum, $W^* = \min_{i=1,\ldots,n} W^i$, will also be exponentially distributed, and its mean is $1/\sum_{i=1}^n \theta_i$.

Suppose that node i has an initial value θ_i. Each node needs to compute $\sum_{i=1}^n \theta_i$. Node i generates an exponential distribution with mean $1/\theta_i$. It then draws a sample, $W^i = w^i$, from that distribution. All nodes do this. They exchange their samples so that each node knows every sample. Then, each node may compute the minimum of the samples, $w^* = \min_{i=1,\ldots,n} w^i$. w^* is a realization of W^*, which is exponentially distributed, with mean $1/\sum_{i=1}^n \theta_i$.

For the algorithm proposed in [23], the nodes perform the above procedure on r samples from each node rather than one. That is, node i draws independently r samples from its exponential distribution, $W_1^i, \ldots, W_r^i$. The nodes exchange information using the information spreading algorithm described below. Ultimately, each node acquires $W_1^*, \ldots, W_r^*$, where W_l^* is the sample-wise minimum, $W_l^* = \min_{i=1,\ldots,n} W_l^i$. Then, for its estimate of $\sum_{i=1}^n \theta_i$, each of the nodes computes

$$\frac{r}{\sum_{l=1}^r W_l^*}.$$

Recall that as r increases, $\frac{1}{r}\sum_{l=1}^r W_l^*$ approaches the mean of W_1^*, namely $1/\sum_{i=1}^n \theta_i$. It is shown that, for large enough r, the nodes' estimates of $\sum_{i=1}^n \theta_i$ will satisfy the desired accuracy criterion with the desired probability.

b) Computation of Minima Using Information Spreading: The computation of the minimum using the information spreading algorithm occurs as follows. Suppose that each node i has an initial vector $W^i = (W_1^i, \ldots, W_r^i)$ and needs to obtain $\bar{W} = (\bar{W}_1, \ldots, \bar{W}_r)$, where $\bar{W}_l = \min_{i=1,\ldots,n} W_l^i$. To compute $\bar{W}$, each node maintains an r-dimensional vector, $\hat{\bar{w}}^i = (\hat{\bar{w}}_1^i, \ldots, \hat{\bar{w}}_r^i)$, which is initially $\hat{\bar{w}}^i(0) = W^i$, and evolves such that $\hat{\bar{w}}^i(k)$ contains node i's estimate of $\bar{W}$ at time k. Node i communicates this vector to its neighbors;

and when it receives a message from a neighbor j at time k containing $\hat{\hat{w}}^j(k^-)$, node i will update its vector by setting $\hat{\hat{w}}_l^i(k^+) = \min(\hat{\hat{w}}_l^i(k^-), \hat{\hat{w}}_l^j(k^-))$, for $l = 1, \ldots, r$.

Denote with $\mathcal{D}$ the information spreading algorithm, used as a subroutine to disseminate messages and compute the minimum. The performance of this algorithm is captured by the δ-information-spreading time, $T_{\mathcal{D}}^{\text{spr}}(\delta)$, at which with probability larger than $1-\delta$ all nodes have all messages. More formally, let $S_i(k)$ is the set of nodes that have node i's message at time k, and V is the set of nodes, the definition of $T_{\mathcal{D}}^{\text{spr}}(\delta)$ is the following.

Definition VI.1: For a given $\delta \in (0,1)$, the δ-information-spreading time, of the algorithm $\mathcal{D}$, $T_{\mathcal{D}}^{\text{spr}}(\delta)$, is

$$T_{\mathcal{D}}^{\text{spr}}(\delta) = \inf\{k : \mathbf{P}(\cup_{i=1}^n \{S_i(k) \neq V\}) \leq \delta\}.$$

As argued in [23], when an information spreading algorithm $\mathcal{D}$ is used where one real-number is transferred between two nodes every time there is a communication, then with probability larger than $1-\delta$, for all i, $\hat{\hat{w}}^i(k) = \bar{W}$ when $k = rT_{\mathcal{D}}^{\text{spr}}(\delta)$, because the nodes propagate in the network an evolving estimate of the minimum, an r-vector, as opposed to the n r-vectors $W^1, \ldots, W^n$.

c) The Performance: The first of the two main theorems of [23] provides an upper bound on the computing time of the proposed computation algorithm and the second provides an upper bound on the information spreading time of a randomized gossip algorithm. These theorems are repeated below for convenience as our results build on those of [23].

Theorem VI.2: Given an information spreading algorithm $\mathcal{D}$ with δ-spreading time $T_{\mathcal{D}}^{\text{spr}}(\delta)$ for $\delta \in (0,1)$, there exists an algorithm $\mathcal{A}$ for computing separable functions $f \in \mathcal{F}$ such that for any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$,

$$T_{\mathcal{A}}^{\text{cmp}}(\varepsilon, \delta) = O\left(\varepsilon^{-2} \log e\delta^{-1} T_{\mathcal{D}}^{\text{spr}}\left(\frac{\delta}{2}\right)\right).$$

In the next section, we state a theorem analogous to this one, but for the case where the nodes are required to communicate a finite number of bits.

Next, the upper bound on the information spreading time is derived for the communication scheme, or equivalently, the randomized gossip algorithm. We refer the reader to [23] for further details on the information spreading algorithm, including an analysis of the case of asynchronous communication. The theorem relevant to this section follows.

Theorem VI.3: Consider any stochastic and symmetric matrix P such that if $(i,j) \notin E$, $p_{ij} = 0$. There exists an information spreading algorithm, $\mathcal{P}$, such that for any $\delta \in (0,1)$,

$$T_{\mathcal{P}}^{\text{spr}}(\delta) = O\left(\frac{\log n + \log \delta^{-1}}{\Phi(P)}\right).$$

3) Quantized Function Computation: The nodes need to each acquire an estimate of $f(\underline{x}) = \sum_{i=1}^n f_i(x_i)$. For convenience, we denote $f_i(x_i)$ by θ_i. Recall that we have assumed that node i can compute θ_i without any communication. Further, we've assumed that there exists a B for which: for all i, $\theta_i \in [1, B+1]$.

Let $y = f(\underline{x}) = \sum_{i=1}^n \theta_i$ be the quantity to be estimated by the nodes. We denote the estimate of y at node i by $\widehat{Y}_i^Q$. The Q is added to emphasize that this estimate was obtained using an algorithm for nodes that can only communicate quantized values using messages consisting a finite number of bits. The randomness in $\widehat{Y}_i^Q$ is due to the fact that the links between the nodes may fail probabilistically, as captured by P.

Recall that the goal is to design an algorithm such that, for large enough k,

$$\mathbf{P}\left\{\cap_{i=1}^n \{|\widehat{Y}_i^Q(k) - y| \leq \varepsilon y\}\right\} \geq 1 - \delta$$

while communicating only a finite number of bits between the nodes. Again, we take advantage of the properties of exponentially distributed random variables, and an information spreading algorithm used as a subroutine for the nodes to determine the minimum of their values.

a) Computation of Minima Using Information Spreading: We use the same scheme that was described in Section VI-B2 for computation of minima using information spreading. Now, node i quantizes a value $\hat{\hat{w}}_l^i$ that it needs to communicate to its neighbor, j, where node i maps the value $\hat{\hat{w}}_l^i$ to a finite set $\{1, \ldots M\}$ according to some quantization scheme. Then, $\log M$ bits have to be communicated between the nodes before j can decode the message and update its $\hat{\hat{w}}_l^j$. But, when each communication between nodes is $\log M$-bits, the time until all nodes' estimates are equal to $\bar{W}$ with probability larger than $1-\delta$ will still be $k = rT_{\mathcal{D}}^{\text{spr}}(\delta)$. However, there will be quantization error. Our choice of M will determine this error.

b) Summary of Algorithm & Main Theorem: The proposed algorithm, $\mathcal{A}^{\mathcal{Q}}$ is summarized below.

1) Independently from all other nodes, node i generates r independent samples from an exponential distribution, with parameter θ_i. If a sample is larger than an m (which we will specify later), the node discards the sample and regenerates it.
2) The node quantizes each of the samples according to a scheme we describe below. The quantizer maps points in the interval $[0, m]$ to the set $\{1, 2, \ldots, M\}$.
3) Each of the nodes performs steps 1 and 2 and communicates its messages via the information spreading algorithm, $\mathcal{D}$, to the nodes with which it is connected. The nodes use the information spreading algorithm to determine the minimum of each of the r sets of messages. After $rT_{\mathcal{D}}^{\text{spr}}(\delta)$ time has elapsed, each node has obtained the r minima with probability larger than $1-\delta$.
4) Node i sets its estimate of y, $\widehat{Y}_i^Q$, to be the reciprocal of the average of the r minima that it has computed.

Here, r is a parameter that will be designed so that $\mathbf{P}\left\{\cap_{i=1}^n \{|\widehat{Y}_i^Q - y| \leq \varepsilon y\}\right\} \geq 1-\delta$ is achieved. Determining how large r and M must be leads to the main theorem of this section.

Theorem VI.4: Given an information spreading algorithm $\mathcal{D}$ with δ-spreading time $T_{\mathcal{D}}^{\text{spr}}(\delta)$ for $\delta \in (0,1)$, there exists an algorithm $\mathcal{A}^{\mathcal{Q}}$ for computing separable functions $f \in \mathcal{F}$ via communication of quantized messages. If each quantized message is $\log M$ bits and $\log M = O(\log n)$, the quantization error

will be no more than a given $\gamma = \Theta(\frac{1}{n})$. Furthermore, for any $\varepsilon \in (\gamma f(\underline{x}), \gamma f(\underline{x}) + \frac{1}{2})$ and $\delta \in (0,1)$

$$T^{\text{cmp}}_{\mathcal{A}^{\mathcal{Q}}}(\varepsilon,\delta) = O\left(\varepsilon^{-2}\log e\delta^{-1} T^{\text{spr}}_{\mathcal{D}}\left(\frac{\delta}{2}\right)\right).$$

Remark: Here, we point out that the condition in the theorem that $\varepsilon \in (y\gamma, y\gamma + 1/2)$ reflects the fact that due to quantization, $\widehat{Y}_i^Q$ can never get arbitrarily close to y, no matter how large r is chosen.

Before proving this theorem, it is convenient to consider the algorithm described above, excluding step 2; that is, with no sample quantization. The derivation of the computation time of this modified algorithm will lead to determining the appropriate truncation parameter, m. Next, we introduce a quantization scheme and determine the number of bits to use in order to guarantee that the node estimates of y converge with desired probability; we find that this number of bits, $\log M$, is of the order of $\log n$. The details can be found in Appendix B.

Thus, we have shown how a distributed algorithm for computing separable functions may be quantized so that the effect of the quantization scheme will be to slow down the information spreading by $\log n$, while the remaining performance characteristics of the original algorithm will be virtually unchanged, especially with respect to its dependence on conductance. This result is stated in Theorem VI.4.

Combining the result of Theorem VI.4 with that of Theorem VI.3 yields Theorem III.6. Comparison with a lower bound obtained via information theoretic inequalities in Section VI-A reveals that the reciprocal dependence between computation time and graph conductance in the upper bound of Theorem III.6 matches the lower bound. Hence the upper bound is tight in capturing the effect of the graph conductance $\Phi(G)$.

VII. Discussion and Conclusions

We've studied a network of nodes communicating over point-to-point memoryless independent noisy channels. Each node has an initial value. The objective of each of the nodes is to compute a given function of the initial values in the network. We have derived a lower bound to the time at which the mean-square error in the nodes' estimates is within a prescribed accuracy interval.

The lower bound is a function of the channel capacities, the accuracy specified by the mean-square error criterion, and the uncertainty in the function that is to be estimated. The bound reveals that, first, the more randomness in the function to be estimated, the larger the lower bound on the computation time. Second, the smaller the mean-square error that is tolerated, the larger the lower bound on the computation time. Hence, there is a tradeoff captured between computation accuracy and computation time. In addition, the lower bound can be used to capture the dependence of the convergence time on the structure of the underlying communication network.

We've considered a network of nodes communicating to compute a sum of the initial values in the network. Each of the nodes is required to acquire an estimate that is, with a specified probability, within a desired interval of the true value of the sum. We've applied our information theoretic technique to derive a lower bound on the computation time for this scenario. We've shown that when $\mathbf{C}_{ij} = \mathbf{C}_{ji}$, the computation time is inversely related to a property of the network called "conductance." It captures the effect of both the topology and channel capacities by quantifying the bottle-neck of information flow.

Next, we have described an algorithm that can be used in this setting of nodes computing a sum via block erasure channels, and guarantees that with the specified probability, each of the nodes' estimate is within the desired interval. We've determined an upper bound on the algorithm's computation time. We've shown that it too is inversely related to conductance.

Hence, we conclude that our lower bound is tight in capturing the effect of the communication network, via conductance. Equivalently, our algorithm's run-time is optimal in its dependence on conductance. That is, we have obtained a scaling law for convergence time as a function of a network property, conductance. When the number of nodes is fixed, this scaling law becomes tighter as the communication constraints are more severe, like diminished channel capacities.

A critical assumption in our work is that the network is a point-to-point network of independent memoryless channels. In this context, there is no interference or collisions from other users. The limitations imposed by the communication network are its pattern of connectivity and the noisy channels. And in this case, the capacity of each of the channels quantifies the bit constraints.

Furthermore, when the function to be computed is simple, like a sum, limitations arise primarily due to communication constraints, not the computational abilities of the nodes. In such a scenario, and when in addition we assume initial measurements are independent and channels are block-erasure channels, our results capture the effect of topology and imperfect transmission on the performance of nodes.

Our general lower bound depends on the assumption that communication occurs over a network of point-to point independent memoryless channels. Our lower bound for summation depends on the additional assumption that the initial measurements at the nodes are independent. This assumption primarily simplifies the computation of the entropy term in the general lower bound.

Our achievability result depends on the further assumption that the channels are block-erasure channels, whose block length depends on the number of nodes. In general, the algorithm we describe for summation will work for any network of point-to-point channels. When the channels are block erasure channels, the computation time of the algorithm depends reciprocally on conductance and hence achieves the lower bound. For these channels, and for the summation task, our code is relatively simple. We believe that an area for future work is to design optimal codes for more general point-to-point channels. These codes will necessarily be more sophisticated than the one we have here. Some insights to the issues that arise in coding for computation over multi-access channels are highlighted in the work of Nazer and Gastpar [24].

APPENDIX A
PROOFS OF LEMMAS V.1 AND V.2

In this appendix, we present the proofs of Lemmas V.1 and V.2, that we used in Section V to derive the lower bound of Theorem III.1.

1) Proof of Lemma V.1: We prove the following inequality:

$$I(\widehat{Y}_S(T); X_{S^c}|X_S) \leq \sum_{l=1}^{T} I(V_S(l); U_{S^c}(l)|U_S(l)) \quad \text{(A.11)}$$

where U_{S^c} is a vector of the variables transmitted by the encoders of the nodes in S^c and V_S is a vector of the variables received via channels by the decoders of the nodes in S.

For this proof, we use the general formulation for multi-terminal networks of [5, Sect. 14.10]. let U_i be transmitted by the node i encoder and V_i be received by the node i decoder. We denote a sequence of length N transmitted by i as $U_i^N = (U_i(1), U_i(2), \ldots U_i(N))$. The indices in brackets represent channel use. As before, if nodes 1 to m belong to S, we have that $V_S = (V_1, \ldots, V_m)$. Similarly, we have that $V_S(l) = (V_1(l), \ldots, V_m(l))$, representing the variables received after the l-th use of the channel.

We assume that the estimate at node i, $\widehat{Y}_i(T)$, is a function of the received messages at that node, V_i^T and its own data, X_i, $\widehat{Y}_i(T) = g_i(V_i^T, X_i)$. The message transmitted by i in the l^{th} channel use, $U_i(l)$, is also a function of the received messages at that node, V_i^{l-1} and its own data, X_i, $U_i(l) = \psi_i(V_i^{l-1}, X_i)$.

As in [5], the channel is a memoryless discrete-time channel. In our case, for convenience, we assume the channel to be continuous, represented by the conditional probability distribution function $p(v_1, \ldots, v_n|u_1, \ldots, u_n)$. However, we note that the inequalities below hold even in the case that the channel is discrete. In this case, the random variable arguments of $I(\cdot;\cdot|\cdot)$ would be arbitrary ensembles, and so we use the general definition for $I(\cdot;\cdot|\cdot)$ as the "average conditional information" in [26, Ch. 3], and for the conditional entropy, $h(X|Y)$, we use $h(X|Y) = I(X; X|Y)$. All the equalities and inequalities below will continue to hold. We refer the reader to [26, Ch. 3] for technical details.

The following inequalities proceed in the same manner as Theorem 14.10.1 in [5]. For convenience, we repeat the steps here using our notation.

$$\begin{aligned}
&I(\widehat{Y}_S(T); X_{S^c}|X_S) \\
&\quad = I(\widehat{Y}_S(T), X_S; X_{S^c}|X_S) \\
&\quad \overset{(a)}{\leq} I(V_1^T, \ldots, V_m^T, X_S; X_{S^c}|X_S) \\
&\quad = I(V_S(1), \ldots, V_S(T); X_{S^c}|X_S) \\
&\quad \overset{(b)}{=} \sum_{l=1}^{T} I(V_S(l); X_{S^c}|X_S, V_S(l-1), \ldots, V_S(1)) \\
&\quad \overset{(c)}{=} \sum_{l=1}^{T} h(V_S(l)|X_S, V_S(l-1), \ldots, V_S(1)) \\
&\qquad - h(V_S(l)|X_{S^c}, X_S, V_S(l-1), \ldots, V_S(1)) \\
&\quad \overset{(d)}{\leq} \sum_{l=1}^{T} h(V_S(l)|X_S, V_S(l-1), \ldots V_S(1), U_S(l)) \\
&\qquad - h(V_S(l)|X_{S^c}, X_S, V_S(l-1), \\
&\qquad \ldots, V_S(1), U_S(l), U_{S^c}(l)) \\
&\quad \overset{(e)}{\leq} \sum_{l=1}^{T} h(V_S(l)|U_S(l)) - h(V_S(l)|U_S(l), U_{S^c}(l)) \\
&\quad \overset{(f)}{=} \sum_{l=1}^{T} I(V_S(l); U_{S^c}(l)|U_S(l)).
\end{aligned}$$

Above:

a) holds by the data processing inequality, because $\widehat{Y}_i(T) = g_i(V_i^T, X_i)$;
b) follows by the chain rule for mutual information;
c) follows by the definition of mutual information, (or, in the discrete channel case, it follows by Kolmogorov's formula [26, Ch. 3] and by noting that the entropy term is well-defined since V_i would take values in a discrete set);
d) follows, for the first term, because $U_i(l) = \psi_i(V_i^{l-1}, X_i)$, so it does not change the conditioning; and the second part follows because conditioning reduces entropy;
e) holds, for the first term, because conditioning reduces entropy, and for the second term, because the channel output depends only on the current input symbols;
f) from the definition of mutual information. ■

2) Proof of Lemma V.2: In this lemma, we consider a network that is represented by the graph $G = (V, E)$. The edges of the graph represent channels with positive capacity. If the channels connecting the nodes are memoryless and independent, we show that

$$I(V_S(l); U_{S^c}(l)|U_S(l)) \leq \sum_{i \in S^c} \sum_{j \in S} \mathbf{C}_{ij}.$$

For simplicity of notation in the rest of the proof, we omit the braces after the random variables, (l). For example, instead of $V_S(l)$ we write V_S.

As we had in the previous lemma, U_i is transmitted by the node i encoder. Previously, we had not specified which nodes will receive this code letter. In our set up, however, there is a dedicated channel between every two nodes that have an edge between them. So, the transmitter at node i will send out codewords to each of the neighbors of i, that is all j, such that $(i, j) \in E$. We denote the encoder's code letter from i to j as U_{ij}. U_i represents all messages transmitted by the encoder of node i. So, $U_i = \{U_{ij}\}$, for all j, such that $(i, j) \in E$.

Similarly, V_i is received by the node i decoder. It consists of all the digits received by i from its neighbors, all j such that $(j, i) \in E$. If there is a link from node j to i, the code letter from node j arrives at the decoder of i through a channel. We denote the digit received at i from j as V_{ji}. V_i represents all the received messages; so, $V_i = \{V_{ji}\}$, for all j, such that $(j, i) \in E$.

In order to make our notation in the proof simpler, we introduce dummy random variables. In particular, we will use U_{ij} and V_{ij} even if $(i, j) \notin E$. Effectively, we are introducing a link between nodes i and j. But, in this case, we set $\mathbf{C}_{ij} = 0$. So now, we let $U_i = \{U_{i1}, \ldots, U_{in}\}$ and $V_i = \{V_{1i}, \ldots, V_{ni}\}$.

The key to the proof is the memorylessness and independence of the channels. That is, the output of a channel at any instant, $V_{ij}(l)$, depends only on the channel input at that instant, $U_{ij}(l)$. Because of this, we have that

$$I(V_S;U_{S^c}|U_S) \leq \sum_{i\in S^c}\sum_{j\in S} I(V_{ij};U_{ij}).$$

To obtain this expression, we express the mutual information in terms of the entropy

$$I(V_S;U_{S^c}|U_S) = h(V_S|U_S) - h(V_S|U_{S^c},U_S).$$

Next, we express the entropy terms using the chain rule. We assume that nodes 1 to m belong to set S and nodes $m+1$ to n belong to S^c. Then,

$$h(V_S|U_S) = \sum_{j=1}^{m} h(V_j|V_{j-1},\ldots,V_1,U_S)$$

and

$$h(V_S|U_{S^c},U_S) = \sum_{j=1}^{m} h(V_j|V_{j-1},\ldots,V_1,U_{S^c},U_S).$$

Because conditioning reduces entropy, we have that

$$h(V_S|U_S) \leq \sum_{j=1}^{m} h(V_j|U_S).$$

For every channel, given its input, the channel output is independent of all other channel outputs. So

$$h(V_S|U_{S^c},U_S) = \sum_{j=1}^{m} h(V_j|U_{S^c},U_S).$$

Combining the two inequalities, we have

$$I(V_S;U_{S^c}|U_S) \leq \sum_{j=1}^{m} h(V_j|U_S) - h(V_j|U_{S^c},U_S).$$

Now, let $j=1$ and consider the expression $h(V_1|U_S) - h(V_1|U_{S^c},U_S)$. Recall that we have assumed that $V_1 = \{V_{11},\ldots,V_{n1}\}$. Also, we have that $U_i = \{U_{i1},\ldots,U_{in}\}$. So, U_S includes $\{U_{11},\ldots,U_{m1}\}$.

For the first differential entropy term we have the following sequence of inequalities:

$$\begin{aligned} h(V_1|U_S) &\overset{(a)}{=} \sum_{i=1}^{n} h(V_{i1}|V_{(i-1)1},\ldots,V_{11},U_S) \\ &\overset{(b)}{=} \sum_{i=1}^{m} h(V_{i1}|U_{i1}) \\ &\quad + \sum_{i=m+1}^{n} h(V_{i1}|V_{(i-1)1},\ldots,V_{11},U_S) \\ &\overset{(c)}{\leq} \sum_{i=1}^{m} h(V_{i1}|U_{i1}) + \sum_{i=m+1}^{n} h(V_{i1}) \end{aligned}$$

where

a) follows by the chain rule;
b) follows because the channels are independent; so, given U_{i1}, V_{i1} is independent of all of the other random variables;
c) holds because conditioning reduces entropy.

Next, observe that

$$\begin{aligned} h(V_1|U_{S^c},U_S) &\overset{(d)}{=} \sum_{i=1}^{n} h(V_{i1}|V_{(i-1)1},\ldots,V_{11},U_{S^c},U_S) \\ &\overset{(e)}{=} \sum_{i=1}^{n} h(V_{i1}|U_{i1}) \end{aligned}$$

where

d) follows by the chain rule;
e) follows because the channels are independent; so, given U_{i1}, V_{i1} is independent of all of the other random variables.

Finally, combining these inequalities:

$$\begin{aligned} h(V_1|U_S) - h(V_1|U_{S^c},U_S) &\leq \sum_{i=m+1}^{n} h(V_{i1}) - h(V_{i1}|U_{i1}) \\ &= \sum_{i=m+1}^{n} I(V_{i1};U_{i1}). \end{aligned}$$

Hence, we have the desired expression

$$I(V_S;U_{S^c}|U_S) \leq \sum_{i\in S^c}\sum_{j\in S} I(V_{ij};U_{ij}).$$

Finally, to complete the proof, we note that

$$I(V_{ij};U_{ij}) \leq \mathbf{C}_{ij}.$$

This is because, by definition

$$\mathbf{C}_{ij} = \max I(V_{ij};U_{ij}),$$

where the maximum is taken over all distributions of the channel input, U_{ij}. ■

APPENDIX B
PROOF OF THEOREM VI.4

1) Determining m: Before we state the lemma of this section, we describe the modified computation algorithm, $\mathcal{A}^{\mathcal{Q}}_{\mathcal{M}}$, which consists of steps 1 to 4 above excluding 2, and we introduce the necessary variables.

First, node i, independently from all other nodes, generates r samples drawn independently from an exponential distribution, with parameter θ_i. If a sample is larger than m, the node discards the sample and regenerates it. This is equivalent to drawing the samples from an exponential distribution truncated at m.

Let $(W_l^i)_T$ be the random variable representing the l^{th} sample at node i, where the subscript "T" emphasizes that the distribution is truncated. Then, the probability density function of $(W_l^i)_T$ is that of an exponentially distributed random variable, W_l^i, with probability density function $f_{W_l^i}(w) = \theta_i e^{-\theta_i w}$

for $w \geq 0$, conditioned on the the event $A_l^i = \{W_l^i \leq m\}$. For $w \in [0, m]$

$$f_{(W_l^i)_T}(w) = \frac{\theta_i e^{-\theta_i w}}{1 - e^{\theta_i m}}$$

and $f_{(W_l^i)_T}(w) = 0$ elsewhere.

Second, the nodes use a spreading algorithm, $\mathcal{D}$, so that each determines the minimum over all n for each set of samples, $l = 1, \ldots, r$. Recall that we consider the random variables at this stage as if there was no quantization. In this case, the nodes compute an estimate of $\bar{W}_l = \min_{i=1\ldots n}(W_l^i)_T$; we denote the estimate of $\bar{W}_l$ at node i by $\widehat{\bar{W}_l}^i$. Furthermore, we denote the estimates at node i of the minimum of each of each of the r set of samples by $\widehat{\bar{W}}^i = (\widehat{\bar{W}_1}^i, \ldots, \widehat{\bar{W}_r}^i)$, and the actual minima of the r set of samples by $\bar{W} = (\bar{W}_1, \ldots, \bar{W}_r)$.

It it is shown in [23] that by the aforementioned spreading algorithm, with probability at least $1 - \delta/2$, the estimates of the r minima, $\widehat{\bar{W}}^i$, will be be equal to the actual minima, $\bar{W}$, for all nodes, $i = 1, \ldots, n$, in $rT_{\mathcal{D}}^{\text{spr}}(\delta/2)$ time slots.

Last, each of the nodes computes its estimate, $\widehat{Y}_i$, of y by summing the r minimum values it has computed, inverting the sum, and multiplying by r:

$$\widehat{Y}_i = \frac{r}{\sum_{l=1}^r \widehat{\bar{W}_l}^i}.$$

The following lemma will be needed in the proof of Theorem VI.4.

Lemma B.1: Let $\theta_1, \ldots, \theta_n$ be real numbers such that for all i, $\theta_i \geq 1$, $y = \sum_{i=1}^n \theta_i$ and $\bar{W} = (\bar{W}_1, \ldots, \bar{W}_r)$. Furthermore, let $\widehat{\bar{W}}^i = (\widehat{\bar{W}_1}^i, \ldots, \widehat{\bar{W}_r}^i)$ and let $\widehat{Y}_i$ denote node i's estimate of y using the modified algorithm of this section, $\mathcal{A}_{\mathcal{M}}^{\mathcal{Q}}$.

For any $\mu \in (0, 1/2)$, and for $I = ((1-\mu)\frac{1}{y}, (1+\mu)\frac{1}{y})$, if $m \geq \ln n - \ln(1 - e^{-\frac{\mu^2}{6}})$

$$\mathbf{P}\left(\cup_{i=1}^n \{\widehat{Y}_i^{-1} \notin I\} | \forall i \in V, \widehat{\bar{W}}^i = \bar{W}\right) \leq e^{-r\frac{\mu^2}{6}}$$

where, $\widehat{Y}_i^{-1} = \frac{1}{r}\sum_{l=1}^r \widehat{\bar{W}_l}^i$.

Proof: First, note that when $\{\forall i \in V, \widehat{\bar{W}}^i = \bar{W}\}$, we have that for all i, $\widehat{Y}_i^{-1} = \frac{1}{r}\sum_{l=1}^r \bar{W}_l$. So, it is sufficient to show that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r \bar{W}_l \notin I\right) \leq e^{-r\frac{\mu^2}{6}}.$$

Let $W_l^* = \min_{i=1,\ldots,n} W_l^i$, the minimum of independent exponentially distributed random variables, W_l^i, with parameters $\theta_1, \ldots, \theta_n$ respectively, then W_l^* will itself be exponentially distributed with parameter $y = \sum_i \theta_i$. Observe that the cumulative distribution function of $\bar{W}_l$, $\mathbf{P}(\bar{W}_l \leq w)$, is identical to that of W_l^*, conditioned on the event $A_l = \{\cap_{i=1}^n A_l^i\}$, where $A_l^i = \{W_l^i \leq m\}$, $\mathbf{P}(W_l^* \leq w | A_l)$, (see Appendix for proof). Hence, we have that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r \bar{W}_l \notin I\right) = \mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r W_l^* \notin I | \cap_{l=1}^r A_l\right).$$

Now, because $\mathbf{P}(A \cap B) \leq \mathbf{P}(A)$, it follows that:

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r W_l^* \notin I | \cap_{l=1}^r A_l\right)\mathbf{P}\left(\cap_{l=1}^r A_l\right) \leq \mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r W_l^* \notin I\right).$$

From Cramer's Theorem, see [6], and the properties of exponential distributions, we have that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r W_l^* \notin I\right) \leq e^{-r(\mu - \ln(1+\mu))}$$

and for $\mu \in (0, 1/2)$, $e^{-r(\mu-\ln(1+\mu))} \leq e^{-r\frac{\mu^2}{3}}$.

Next, we have that $\mathbf{P}\left(\cap_{l=1}^r A_l\right) = \left(\mathbf{P}\left(A_l\right)\right)^r$, because the $A_1, \ldots, A_r$ are mutually independent. Furthermore, $\mathbf{P}\left(A_l\right) \geq 1 - ne^{-m}$. To see this, note that the complement of A_l is $A_l^c = \{\cup_{i=1}^n \{W_l^i > m\}\}$, and $\mathbf{P}\left(W_l^i > m\right) = e^{-\theta_i m}$. So, by the union bound, we have

$$\mathbf{P}\left(A_l^c\right) \leq \sum_{i=1}^n e^{-\theta_i m} \leq ne^{-m},$$

where the last inequality follows because $\forall i, \theta_i \geq 1$.

Finally, putting all this together, we have that

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r \bar{W}_l \notin I\right) \leq (1 - ne^{-m})^{-r} e^{-r\frac{\mu^2}{3}}.$$

Letting $1 - ne^{-m} \geq e^{-\frac{\mu^2}{6}}$ completes the proof. ■

2) Proof of Theorem VI.4: Before we proceed with the proof of the Theorem, we describe the quantization scheme. In step 2 of the algorithm $\mathcal{A}^{\mathcal{Q}}$, node i quantizes the sample it draws, a realization of $(W_l^i)_T$ denoted by w_l^i. The quantizer Q maps points in the interval $[0, m]$ to the set $\{1, 2, \ldots, M\}$. Each node also has a "codebook," Q^{-1}, a bijection that maps $\{1, 2, \ldots, M\}$ to $\{w_{q_1}, w_{q_2}, \ldots, w_{q_M}\}$, chosen such that for a given γ, $|w_l^i - Q^{-1}Q(w_l^i)| \leq \gamma$. We will denote $Q^{-1}Q(w_l^i)$ by $(w_l^i)_Q$.

While we do not further specify the choice of the quantization points, w_{q_k}, we will use the fact that the quantization error criterion can be achieved by a quantizer that divides the interval $[0, m]$ to no more than M intervals of length γ each. Then, the number of messages will be $M = m/\gamma$, and the number of bits that the nodes communicate is $\log M$.

Proof: We seek an upper bound on the (ε, δ)-computation time of the algorithm $A^{\mathcal{Q}}$, the time until, with probability at least

$1-\delta$, all nodes $i=1,\ldots,n$ have estimates $\widehat{Y}_i^Q$ that are within a factor of $1\pm\varepsilon$ of y. That is

$$\mathbf{P}(\cup_{i=1}^n\{\widehat{Y}_i^Q\notin[(1-\varepsilon)y,(1+\varepsilon)y]\})\leq\delta.$$

First, suppose that we may communicate real-valued messages between the nodes. We analyze the effect of quantization on the convergence of the node estimates to the desired $1\pm\varepsilon$ factor of y. For this, we compare the quantized algorithm, $\mathcal{A}^{\mathcal{Q}}$, with the modified algorithm $\mathcal{A}^{\mathcal{Q}}_{\mathcal{M}}$.

Note that for the above quantization scheme, for all i,l and any realization of $(W_l^i)_T$ denoted by w_l^i

$$(w_l^i)_Q\in\left[w_l^i-\gamma,w_l^i+\gamma\right]$$

hence

$$\min_{i=1,\ldots,n}(w_l^i)_Q\in\left[\min_{i=1,\ldots,n}w_l^i-\gamma,\min_{i=1,\ldots,n}w_l^i+\gamma\right]$$

and

$$\frac{1}{r}\sum_{l=1}^r\min_{i=1,\ldots,n}(w_l^i)_Q\in\left[\frac{1}{r}\sum_{l=1}^r\min_{i=1,\ldots,n}w_l^i-\gamma,\frac{1}{r}\sum_{l=1}^r\min_{i=1,\ldots,n}w_l^i+\gamma\right]. \quad (A.12)$$

Note that $\frac{1}{r}\sum_{l=1}^r\min(w_l^i)_Q$ is a realization of $(\widehat{Y}_i^Q)^{-1}$.

Now, suppose that the information spreading algorithm, $\mathcal{D}$, is used so that in $O(rT_{\mathcal{D}}^{\mathrm{spr}}(\delta/2))$ time

$$\mathbf{P}\left(\cup_{i=1}^n\{\widehat{\bar{W}}^i\neq\bar{W}\}\right)\leq\frac{\delta}{2}. \quad (A.13)$$

Consider the case where $\{\cap_{i=1}^n\{\widehat{\bar{W}}^i=\bar{W}\}\}$, we have from Lemma B.1 that, for any $\mu\in(0,1/2)$, if $m=\ln n-\ln(1-e^{-\frac{\mu^2}{6}})$

$$\mathbf{P}\left(\frac{1}{r}\sum_{l=1}^r\bar{W}_l\notin\left((1-\mu)\frac{1}{y},(1+\mu)\frac{1}{y}\right)\right)\leq e^{-r\frac{\mu^2}{6}}.$$

Combining with (A.12), we have that

$$\mathbf{P}\left(\cup_{i=1}^n\left\{(\widehat{Y}_i^Q)^{-1}\notin\left((1-\mu)\frac{1}{y}-\gamma,(1+\mu)\frac{1}{y}+\gamma\right)\right\}\,\middle|\,\cap_{i=1}^n\{\widehat{\bar{W}}^i=\bar{W}\}\right)\leq e^{-r\frac{\mu^2}{6}}.$$

But the event

$$\left\{(\widehat{Y}_i^Q)^{-1}\notin\left((1-\mu)\frac{1}{y}-\gamma,(1+\mu)\frac{1}{y}+\gamma\right)\right\}$$

is equivalent to

$$\left\{(\widehat{Y}_i^Q)\notin\left((1+(\mu+y\gamma))^{-1}y,(1-(\mu+y\gamma))^{-1}y\right)\right\}.$$

And, letting $\varepsilon=\mu+y\gamma$,

$$\left((1+\varepsilon)^{-1},(1-\varepsilon)^{-1}\right)\subset(1-2\varepsilon,1+2\varepsilon).$$

So

$$\mathbf{P}\left(\cup_{i=1}^n\left\{|\widehat{Y}_i^Q-y|>2\varepsilon y\right\}|\cap_{i=1}^n\{\widehat{\bar{W}}^i=\bar{W}\}\right)\leq e^{-r\frac{\mu^2}{6}}.$$

Letting $r\geq6\mu^{-2}\ln2\delta^{-1}$, we have that

$$e^{-r\frac{\mu^2}{6}}\leq\frac{\delta}{2}.$$

Combining this with (A.13) in the Total Probability Theorem, we have the desired result

$$\mathbf{P}(\cup_{i=1}^n\{\widehat{Y}_i^Q\notin[(1-2\varepsilon)y,(1+2\varepsilon)y]\})\leq\delta.$$

Finally, recall that when the nodes communicate their real-valued messages, with high probability all nodes have estimates of the minima that they need in the computation of the estimate of y in $O(rT_{\mathcal{D}}^{\mathrm{spr}}(\delta/2))$ time. So, the computation time is of that order.

Now, for the quantization algorithm described in this section the nodes need to communicate $\log M$ bit messages before the appropriate minima are computed. Because we assume that this is the case, that the nodes exchange $\log M$ bits at a time, $T_{\mathcal{D}}^{\mathrm{spr}}(\delta)$ time slots are needed until the quantized messages are disseminated and the minima computed. Consequently, the computation time of the quantized algorithm will be $O(rT_{\mathcal{D}}^{\mathrm{spr}}(\delta/2))$.

But, $M=m/\gamma$, and by design, for a given μ we choose $m=\ln n-\ln(1-e^{-\frac{\mu^2}{6}})$; so $m=O(\log(n))$. Furthermore, we choose γ, such that $\gamma=\Theta(\frac{1}{n})$. Then,

$$\log M\leq\log\log n+\log n$$

so, $\log M=O(\log n)$ bits are needed.

As we have previously seen, for $\mu\in(0,1/2)$, $r\geq6\mu^{-2}\ln2\delta^{-1}$. But, $\mu=\varepsilon-y\gamma$ and $\gamma=\Theta(1/n)$, so $y\gamma=O(1)$. We therefore have, for $\varepsilon\in(y\gamma,y\gamma+1/2)$

$$T_{\mathcal{A}^{\mathcal{Q}}}^{\mathrm{cmp}}(\varepsilon,\delta)=O\left(\varepsilon^{-2}(1+\log\delta^{-1})T_{\mathcal{D}}^{\mathrm{spr}}(\delta/2)\right).$$

■

ACKNOWLEDGMENT

O. Ayaso would like to thank Prof. N. C. Martins (University of Maryland, College Park) for suggesting the utility of information theoretic techniques in the context of distributed computation.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 1204–1216, Jul. 2000.

[2] 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, Oct. 2008.

[3] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conf. Decision and Control and European Control Conf. (CDC-ECC'05)*, Seville, Spain, 2005.
[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley-Interscience, 1991.
[6] A. Dembo and O. Zeitouni, *Large Deviation Techniques and Applications*. New York: Springer, 1998.
[7] P. Diaconis and L. Saloff-Coste, 2006, private communication.
[8] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip : Efficient aggregation for sensor networks," in *Proc. 5th Int. ACM/IEEE Symp. Information Processing in Sensor Networks (IPSN '06)*, Apr. 2006.
[9] A. E. Gamal and A. Orlitsky, "Interactive data compression," in *Proc. 25th IEEE Symp. Foundations of Computer Science*, Oct. 1984, pp. 100–108.
[10] R. Gallager, "Finding parity in a simple broadcast network," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 176–180, Feb. 1988.
[11] A. Giridhar and P. R. Kumar, "Towards a theory of in-network computation in wireless sensor networks," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 98–107, Apr. 2006.
[12] N. Goyal, G. Kindler, and M. Saks, "Lower bounds for the noisy broadcast problem," *SIAM J. Comput.*, vol. 37, no. 6, pp. 1806–1841, Mar. 2008.
[13] W. K. Hastings, "Monte carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.
[14] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3315–3326, Jul. 2008.
[15] S. Kar, J. M. F. Moura, and K. Ramanan, Distributed Parameter Estimation in Sensor Networks: Nonlinear Observation Models and Imperfect Communication 2008 [Online]. Available: http://arxive.org/abs/0809.0009
[16] Z.-Q. Luo and J. N. Tsitsiklis, "Data fusion with minimal communication," *IEEE Trans. Inf. Theory*, vol. 40, no. 9, pp. 1551–1563, Sep. 1994.
[17] N. C. Martins, "Information Theoretic Aspects of the Control and Mode Estimation of Stochastic Systems," Ph.D. dissertation, Dept. Elect. Eng. and Comput. Sci., Massachusetts Inst. Technol., Cambridge, 2004.
[18] N. C. Martins, "Finite gain LP stabilization requires analog control," *Syst. Control Lett.*, vol. 55/1, pp. 949–954, Nov. 2006.
[19] N. C. Martins and M. A. Dahleh, "Feedback control in the presence of noisy channels: Bode-like fundamental limits of performance," *IEEE Trans. Autom. Control*, vol. 53, no. 7, pp. 1604–1615, May 2008.
[20] N. C. Martins, M. A. Dahleh, and J. C. Doyle, "Fundamental limitations of disturbance attenuation in the presence of side information," *IEEE Trans. Autom. Control*, vol. 52, no. 1, pp. 56–66, Jan. 2007.
[21] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chemical Phys.*, vol. 21, pp. 1087–1092, 1953.
[22] R. Montenegro and P. Tetali, "Mathematical aspects of mixing times in Markov chains," in *Found. and Trends in Theoret. Comput. Sci.*, 2006, vol. 1, pp. 237–354.
[23] D. Mosk-Aoyama and D. Shah, "Computing separable functions via gossip," in *ACM Principles of Distrib. Comput.* , Denver, CO, Jul. 2006.
[24] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.
[25] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 903–917, Mar. 2001.
[26] M. S. Pinsker, *Information and Information Stability of Random Variables and Processes*. San Francisco, CA: Holden-Day, 1964.
[27] K. Savla, F. Bullo, and E. Frazzoli, "On traveling salesperson problems for Dubins' vehicle: Stochastic and dynamic environments," in *Proc. CDC-ECC*, Seville, Spain, Dec. 2005, pp. 4530–4535.
[28] A. Sinclair, *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Boston, MA: Birkhauser, 1993.
[29] J. N. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Trans. Autom. Control*, vol. AC-29, no. 1, pp. 42–50, Jan., 1984.
[30] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. AC-31, no. 9, pp. 803–812, Sep. 1986.
[31] M. E. Yildiz and A. Scaglione, "Coding with side information for rate-constrained consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3753–3764, Aug. 2008.

Ola Ayaso received the B.S. degree (with high distinction) from the American University of Beirut, the M.S. degree from the Massachusetts Institute of Technology (MIT), Cambridge, and the Ph.D. degree from MIT in 2008, and all in electrical engineering.

She is a Postdoctoral Fellow in the Electrical and Computer Engineering Department, Georgia Institute of Technology, Atlanta. Her research interests include algorithms and limits of computation and coordination in distributed systems and dynamic networks.

Devavrat Shah (M'05) received the B.Tech. degree in computer science and engineering from IIT-Bombay, Bombay, India, in 1999 (with the honor of the President of India Gold Medal) and the Ph.D. degree from the Computer Science Department, Stanford University, Stanford, CA, in October 2004.

He was a Postdoctoral Associate in the Statistics Department, Stanford University, during 2004–2005. He is currently a Jamieson Career Development Associate Professor with the Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA. He is a member of the Laboratory of Information and Decision Systems (LIDS) and affiliated with the Operations Research Center (ORC). His research focus is on theory of large complex networks which includes network algorithms, stochastic networks, network information theory and large scale statistical inference.

Dr. Shah was co-awarded the Best Paper awards at the IEEE INFOCOM '04, ACM SIGMETRICS/Performance '06, and Best Student Paper awards at Neural Information Processing Systems '08 and ACM SIGMETRICS/Performance '09. He received the 2005 George B. Dantzig Best Disseration Award from the INFORMS and the first ACM SIGMETRICS Rising Star Award 2008 for his work on network scheduling algorithms.

Munther A. Dahleh (F'00) was born in 1962. He received the B.S. degree from Texas A&M University, College Station, in 1983 and the Ph.D. degree from Rice University, Houston, TX, in 1987, all in electrical engineering.

Since then, he has been with the Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, where he is now a Full Professor and the Associate Director of the Laboratory for Information and Decision Systems. He was a Visiting Professor at the Department of Electrical Engineering, California Institute of Technology, Pasadena, in Spring 1993. He has held consulting positions with several companies in the U.S. and abroad. He is the co-author (with I. Diaz-Bobillo) of the book *Control of Uncertain Systems: A Linear Programming Approach* (Upper Saddle River, NJ: Prentice-Hall, 1995) and the co-author (with N. Elia) of the book *Computational Methods for Controller Design* (New York: Springer, 1998).

Dr. Dahleh was an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL and for *Systems and Control Letters*. He was the recipient of the Ralph Budd Award in 1987 for the best thesis at Rice University, the George Axelby Outstanding Paper Award (coauthored with J.B. Pearson) in 1987, an NSF Presidential Young Investigator Award in 1991, the Finmeccanica Career Development Chair in 1992, the Donald P. Eckman Award from the American Control Council in 1993, the Graduate Students Council Teaching Award in 1995, the George Axelby Outstanding Paper Award (coauthored with Bamieh and Paganini) in 2004, and the Hugo Schuck Award for Theory (for the paper coauthored with Martins). He was a Plenary Speaker at the 1994 American Control Conference, at the Mediterranean Conference on Control and Automation in 2003, at the MTNS in 2006, at SYSID in 2009, at the Asian Control Conference in 2009, and at SING6 in 2010.