Information Theoretic Bounds for Distributed Computation

Ola Ayaso, Devavrat Shah, and Munther A. Dahleh
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Email: ayaso@alum.mit.edu, devavrat@mit.edu, dahleh@mit.edu

Abstract— **A network of nodes communicate via 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, we obtain 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' estimate is within a given interval around zero. This utilizes information theoretic inequalities reminiscent of those used in rate distortion theory along with a novel 'perturbation' techniques so as to be broadly applicable.**

To understand the tightness of the bound, we consider a specific scenario where nodes are required to learn a linear combination of the initial values in the network while communicating over erasure channels. We develop a distributed quantized algorithm whose computation time essentially scales as that 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 function in a network with minimal computation time.

I. Introduction

We consider a network of nodes communicating via 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.

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.

The formulation we consider lends itself to Information Theoretic techniques. We use Information Theoretic inequalities to derive lower bounds on information exchange necessary between nodes for the mean square error in the nodes' estimates to converge to zero. We use the Information Theoretic technique is to determine 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, as one of the important technical contribution of this paper we develop a novel 'perturbation' technique as explained in section VI-C. This allows us to apply our method to obtain non-trivial lower bound for any functional computation setup.

In section VII, we apply the lower bound developed in this paper to a specific scenario where we find our bounds to be asymptotically tight. Specifically, we consider a scenario where nodes are required to learn a linear combination of the initial values in the network while communicating over erasure channels. Our lower bound suggests that in this scenario, the computation time depends reciprocally on "conductance." 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.

To establish the tightness of our lower bound, we describe an algorithm whose 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 that a 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 an important question of recent interest of designing the fastest possible distributed algorithm for separable function computation (e.g. see works on consensus, linear estimation and distributed control [22], [23], [1]).

A. Related work

Existing results include algorithms with upper and/or lower bounds on the time for the nodes to reach agreement or compute a certain quantity with given accuracy, when communication is subject to topological constraints, but perfect when present [2], [3], [24], [23]. Another set of work investigates algorithms for computation when communication is unreliable. The channels in the network are explicitly modelled. The researchers propose an algorithm that will perform the desired computation while satisfying some performance criterion. For example, in [9], 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 [11] that this algorithm is optimal. The authors produce an algorithm-independent lower bound that is of the same order as the upper bound.

Many different formulations and corresponding bounds can be found in the literature. Two examples are [8], [13]. In [8], 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 [13], the authors present lower bounds to the number of messages that must be communicated by two sensors to a fusion center that must determine a given function of the nodes' data. However, the transmitted messages are real-valued vectors and the lower bound is on the sum of the dimensions of the message functions. Several formulations and results relevant to computation in wireless sensor networks can be found in a detailed survey by Giridhar and Kumar [10].

Our approach and, hence results, are quite different. We capitalize on Martins' successful use of Information Theoretic tools in [14], [15], [16], [17] to characterize fundamental performance limits of feedback control systems with communication constraints. 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. As noted earlier, in order to make our bounds widely applicable, we develop a novel perturbation technique.

The Information Theoretic approach captures fundamental performance limitations that arise in the network due to the communication constraints. This happens because the analysis is independent of the communication algorithm used by the nodes. The lower bound we derive in this paper enables us to characterize the effect of the network structure on algorithm running time. For nodes exchanging information over erasure channels to compute the sum of their initial conditions, 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 two main results of this paper. The first result is about lower bound on computation and the second result is about application of this lower bound along with a lower bound achieveing quantized algorithm for computation of linear combination of numbers. In section IV we illustrate how network topology, through conductance, affects the computation time using specific network structure. We compare our quantized algorithm with the popular linear iterative algorithms. The comparison suggests that for network structures with *small* conductance our algorithm convincingly outperforms the popular algorithms.

In section V we recall useful Information Theoretic definitions and properties. In section VI we prove our main theorem on the lower bound. We use information theoretic inequalities with a novel perturbation argument (introduced in section VI-C) to obtain non-trivial bound on computation for any function of interest. Next, we discuss the scenario for which our results are tight. In section VII-A we derive the lower bound that scales reciprocally with conductance. In section VII-B we describe an algorithm that can be used to compute the sum

via erasure channels. 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 Formulation and Main Results

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 $X_i(0)$. Let $X(0)$ represent the vector of all the initial condition random variables, $[X_1(0) \ \dots \ X_n(0)]'$. Each node is required to compute a given function of all the initial conditions. That is, node i is required to estimate $C_i = f_i(X(0))$. We let $C = [C_1 \ \dots \ C_n]'$. 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, $C_S = [C_1 \ \dots \ C_m]'$.

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 C_i. We denote this estimate by $X_i(k)$. Let, X_i^k denote the finite sequence of estimates at node i, $\{X_i(1), X_i(2), \dots X_i(k)\}$. The estimates of all nodes in the network at the end of time slot k are denoted by the vector $X(k) = [X_1(k) \ \dots \ X_n(k)]'$. And, the estimates of nodes in set S are denoted by $X_S(k) = [X_1(k) \ \dots \ X_m(k)]'$.

The nodes communicate via 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 $\mathbf{C}_{ij} = 0$. [1] 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 based on the k received symbols. Let this be denoted as $X_i(k)$ at node i.

To capture the limitations arising exclusively due to the communication structure, we assume no limits on the computational capabilities of the nodes, such as limited memory or power. The estimate $X_i(k)$ is generated by node i using some function of its initial condition, $X_i(0)$, and the messages it has received by the end of the k^{th} time slot. We make no assumptions on this function, except that it be measureable. 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. We make no assumptions on this function, except that it be measureable. The relationship between the node estimates and communicated symbols is made mathematically precise when need arises, in Appendix I.

We consider two mean square error criteria. The operator $\|\cdot\|$ is to be interpreted, when the argument is a vector, C, as $\|C\|^2 = \sum C_i^2$.

R1. $E(\|X(T) - C\|^2) \leq \beta 2^{-\alpha}$, and,

R2. $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$, for all $i \in \{1, \dots, n\}$,

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

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, $E(X_i(k) - C_i)^2$ 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. We seek a lower bound on the computation time, T, that holds if the desired mean square error criterion, R1 or R2, is satisfied. That is, if R1 or R2 holds, then how large must T be?

A. Features of the formulation

Our formulation (and results) are appropriate for 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.

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

[1] Note that we use bold capitalized $\mathbf{C}_{ij}$ for channel capacity, where the two letters in the subscript indicate that the channel is from node i to j. Constrast this with our notation for the function to be estimated at node i, C_i, which is not boldface and is followed by a single-letter subscript.

must elapse before it is possible to acheive the performance desired, over all communication and computation schemes that satisfy our assumptions. The necessity of having this time elapse arises due to network topology and channel capacities.

III. Main results

We state the formal statements of the main results of this section. There are two main results of this paper. The first result, stated in section III-A describes 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 summation. This involves specializing the lower bound of section III-A and developing a quantized algorithm with the computation time matching thus developed lower bound.

A. Result I: 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.

Theorem III.1. *For the communication network described above, if at time, T, the mean square error is in an interval prescribed by α, $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$, 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,

$$\bar{L}(S) = h(C_S | X_S(0)) - \frac{|S|}{2} \log 2\pi e \beta + |S| \frac{\alpha}{2}.$$

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.

B. Result II: An optimal summation algorithm

Setup. 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 $X_i(0)$. Here we assume that these initial values are distributed independently and uniform at random in the interval $[1, B+1]$. Here $B > 0$ is a constant and should be treated as problem parameter. Each node wishes to compute the same quantity C, where $C = \sum_{j=1}^{n} \beta_j X_j(0)$. Finally, let $X_i(k)$ denote the estimate of node i at the end of time k under any computation algorithm.

As before, nodes will communicate over noisy channels that are independent and discrete-time memoryless. In addition, we assume that these are 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 $logM$ bits, where we shall decide value M later. Thus effective capacity of 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.

Definition III.2 (Conductance)**.** The conductance of a capacitated graph G with edge capacities $\mathbf{C}_{ij}, (i,j) \in E$ is defined as

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

The conductance, $\Phi(G)$ captures the *information bottle-neck* in the capacitated graph G. It depends on the connectivity or toplogy of the graph along with the channel magnitude. We use the word 'conductance' as it co-incides 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.

Specifically, consider a Markov chain with irreducible and aperiodic probability transition matrix P on 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). For such a P, its conductance which is denoted by $\Phi(P)$ is defined as

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

where $\pi(S) = \sum_{i \in S} \pi_i$. 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 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 we have P as symmetric and doubly stochastic. In such a case the stationary distribution π is uniform. That is, $\pi_i = 1/n$ for all i. Therefore, the conductance can be simplified to

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

Thus, $\Phi(G) = \Phi(P)\log M$. In this sense, the $\Phi(G)$ is related to the standard definition of conductance utilized in the context of Markov chain theory.

Let A represent a realization of the initial conditions, $A = \{X_1(0) = x_1, \ldots, X_n(0) = x_n\}$. The performance of an algorithm, $\mathcal{H}$, used by the nodes to compute an estimate of $f(x,V) = \sum_{j=1}^n \beta_j x_j$ at each node, is measured by the algorithm's (ε,δ)-computation time, $T^{\text{cmp}}_{\mathcal{H}}(\varepsilon,\delta)$. It is the time until the estimates at all nodes are within a factor of $1\pm\varepsilon$ of $f(x,V)$, with probability larger than $1-\delta$. The definition follows, where $\hat{y}_i(k)$ denotes the estimate of $f(x,V)$ at node i at time k.

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

$$T^{\text{cmp}}_{\mathcal{H}}(\varepsilon,\delta) = \sup_{x\in\mathbb{R}^n} \inf\{k : \mathbf{P}(\cup_{i=1}^n \{\hat{y}_i(k) \notin [(1-\varepsilon)f(x,V),(1+\varepsilon)f(x,V)]\}) \leq \delta\}$$

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

Lower bound. Consider any algorithm, $\mathcal{H}$, 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 C, at time T. The Information Theoretic lower bound maintains that such algorithm must have a computation time, $T = T^{\text{cmp}}_{\mathcal{H}}(\varepsilon,\delta)$, that is inversely proportional to conductance.

Theorem III.4. *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 $i \in \{1,\ldots,n\}$,*

$$\mathbf{P}\left(|X_i(T) - C| \leq \varepsilon C \Big| A\right) \geq 1-\delta,$$

must have

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

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

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

Upper bound: Through an algorithm. 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. We provide an upper bound on this algorithm's computation time. The computation time is inversely proportional to conductance.

Theorem III.5. *Suppose that node i has an initial condition, x_i. There exists a distributed algorithm $\mathcal{AP}^{\mathcal{Q}}$ by which nodes compute a linear sum, $f(x,V) = \sum_{j=1}^n \beta_j x_j$, 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 $\varepsilon \in (\gamma f(x,V), \gamma f(x,V) + \frac{1}{2})$ and $\delta \in (0,1)$, the computation time of the algorithm will be*

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

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

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

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. Motivation: 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 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.

A. Conductance: Two examples

Consider two networks, each has n nodes. We calculate conductance for two extreme cases of connectivity shown in Figure 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.

For the purpose of illustrating the computation of conductance 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 $\mathbf{C}_{ij} = \frac{1}{n}$. So, in both cases, we have that the sum of the capacities of channels leaving a node is 1, $\sum_j \mathbf{C}_{ij} = 1$.

Now, we compute the conductance of the ring graph. Recall that conductance is

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

Consider any cut that divides the ring graph into two sets, S and S^c. For any such cut, there will be exactly two links crossing the cut, going from S to S^c. So, $\sum_{i \in S, j \notin S} \mathbf{C}_{ij} = \frac{1}{2}$, and

$$\Phi(G) = \min_{\substack{S \subset V \\ 0<|S|\leq n/2}} \frac{\frac{1}{2}}{|S|}.$$

Since we minimize over all cuts such that $|S| \leq n/2$, the ratio is minimized when the cut divides the ring into two sets of equal size, and $|S| = n/2$. So, $\Phi(G) = \frac{1}{n}$.

Next, we compute the conductance of the fully connected graph. Consider any cut that divides the graph into two sets, S and S^c. For any such cut, there will be $|S||S^c|$ links crossing the cut, going from S to S^c. So,

$$\begin{aligned} \frac{\sum_{i \in S, j \notin S} \mathbf{C}_{ij}}{|S|} &= \frac{|S||S^c|\frac{1}{n}}{|S|} \\ &= \frac{|S^c|}{n} \\ &= \frac{n - |S|}{n} \end{aligned}$$

The last equality is minimized where $|S| = n/2$, so, $\Phi(G) = \frac{1}{2}$.

So, 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 $\Phi(G) = O(1)$.

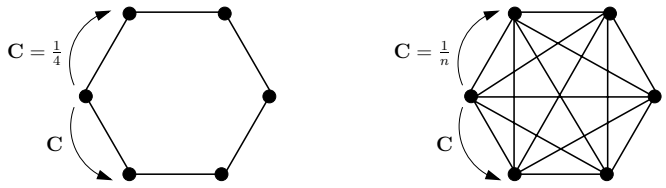

Fig. 1. Two ways to connect six nodes: a ring graph and a fully connected graph.

B. Comparison with iterative algorithms

A popular approach for computing a linear function of the initial conditions are based on linear iterations. For example, to compute linear estimator or reaching consensus, such linear iterative algorithms are popular. These algorithms assume that nodes can communicate real numbers between them in each time instance.

For an iterative algorithm based on a doubly stochastic matrix P, the computation time is proportional to its mixing time, say $\mathcal{H}(P)$ (e.g. see results by Boyd, Ghosh, Prabhakar and Shah [3]). As noted earlier, the mixing time $\mathcal{H}(P)$ (and 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 quick iterative algorithm, it is important to obtain P with small mixing time $\mathcal{H}(P)$. The standard approach of finding such a P is based on the method of Metropolis [18] and Hastings [12]. This results into a symmetric and doubly stochastic P on G.

Now, for *expander* graphs the resulting P induced by Metropolis-Hasting method is likely to have $\Phi(P) = \Theta(1)$ and hence mixing time is $O(\log n)$ which is *essentially* fastest possible. For example, for complete graph the resulting will be $P = [1/n]$. Therefore, as discussed earlier it has $\Phi(P) = \Theta(1)$. Thus, algorithm based on linear iterations or our algorithm have essentially optimal computation time. It should be noted that our algorithm (described later) is quantized. However, quantized version of the linear iterative algorithm is far from obvious and an important open question to the best of our knowledge.

Now the graph topologies such as those arising in wireless sensor network deployed in some geographic area [3], [7] or a nearest neighbor network of unmanned vehicle [21], do possess *geometry* and are far from being expanders. One of the simplest example of 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. As discussed, it has $\Phi(P) = \Theta(1/n)$. But the mixing time scales at least as $\Omega(n^2)$. 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 [3]). Thus, the linear iterative algorithms based on symmetric P have computation time scaling as n^2. In contrast, our quantized algorithm will have computation time scaling as n (which is $1/\Phi(P)$) for the ring. Now the diameter of the ring graph is n and obviously no algorithm takes less than n or no P can have mixing time smaller than this diameter n.

In the general, it can be checked that diameter of a graph G is at most $1/\Phi(P)$ for any irreducible probability matrix P. Also, it can be checked that for graphs with bounded degree and geometry, the P induced by the Metropolis-Hasting method have diameter scaling like $1/\Phi(P)$. By graph with geometry, precisely we mean a graph with polynomial growth – number of nodes within distance r for any given node scales as $O(r^d)$ for some fixed constant d. Now Diaconis and Saloff-Coste [6] 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 algorithm will take computation time scaling like D^2. In contrast, our algorithm will take computation time $1/\Phi(P)$ which will be equal to diameter D for 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. Information Theoretic Preliminaries

The differential entropy of C is denoted by $h(C)$. The mutual information between X and C is denoted by $I(X;C)$.

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. We repeat this definition from [20, p.9]. The conditional mutual information is similarly defined; see [20, Ch. 3].

Suppose X and C are random variables that take values in the measurable spaces (Ω_X, S_X) and (Ω_C, S_C), respectively. S_X denotes the sigma algebra of subsets of Ω_X. Let the probability distributions of X and C be $\mathbf{P}_X$ and $\mathbf{P}_C$. Let $\mathbf{P}_{XC}$ be the joint distribution of X and C.

The mutual information between X and C is

$$I(X;C) = \sup \sum_{i,j} \mathbf{P}_{XC}(E_i \times F_j) \log \frac{\mathbf{P}_{XC}(E_i \times F_j)}{\mathbf{P}_X(E_i)\mathbf{P}_C(F_j)},$$

where the supremum is taken over all partitions $\{E_i\}$ of Ω_X and partitions $\{F_i\}$ of Ω_C.

In the remainder of this section, let C and X be continuous random variables. The definitions for mutual information and differential entropy for this case are repeated from [4, Ch.9].

Let X and C have the probability densities $p(x)$ and $p(c)$. Let their joint density be $p(x,c)$. Then their mutual information is defined as

$$I(X;C) = \int p(x,c) \log \frac{p(x,c)}{p(x)p(c)} \, dx \, dc.$$

The differential entropy of C is defined as

$$h(C) = -\int p(c) \log p(c) \, dc.$$

The conditional differential entropy $h(C|X)$ is

$$h(C|X) = -\int p(x,c) \log p(c|x) \, dx \, dc.$$

The following properties of differential entropy will be used.

(1) Conditioning reduces entropy, $h(C|X) \leq h(C)$. Equality holds if C and X are independent.

(2) Differential entropy, $h(X)$, is maximized, over all distributions with the variance $Var(X) = \sigma^2$, by the normal distribution. If X had a Normal distribution, it would have entropy $\frac{1}{2}\log 2\pi e\sigma^2$. Hence, for any distribution of X with $Var(X) = \sigma^2$,

$$h(X) \leq \frac{1}{2}\log 2\pi e\sigma^2.$$

Further, if X is a vector of random variables, $X = [X_1 \ \ldots \ X_n]'$, then

$$h(X_1, \ldots, X_n) \leq \frac{1}{2}\log(2\pi e)^n |Z|,$$

where Z is the covariance matrix of X and $|Z|$ is the determinant of Z.

Next, the following properties of mutual information will be needed.

(1) Mutual information can be written in terms of differential entropies as

$$I(X;C) = h(C) - h(C|X).$$

(2) By the chain rule for mutual information,

$$I(X_1, X_2, \ldots X_n; C) = \sum_{i=1}^{n} I(X_i; C|X_1, \ldots X_{i-1}).$$

(3) By the data processing inequality, if $Y = f(C)$ for any (measurable) function f, then $I(C;X) \geq I(Y;X)$.

Finally, when the argument in $h(\cdot)$ is a vector of length n, for example, $C = [C_1, \ldots, C_n]'$, it is interpreted as the joint differential entropy $h(C_1, \ldots, C_n)$. Similarly, when the arguments in $I(\cdot;\cdot)$ are vectors of length n, for example C and X, it is to be interpreted as $I(C_1, \ldots, C_n; X_1, \ldots, X_n)$.

VI. 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 $S^c = \{1, \dots n\} \backslash S$. 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 $X_S(T) = [X_1(T) \ \dots \ X_m(T)]'$. The initial conditions of the nodes in sets S and S^c are denoted by $X_S(0) = [X_1(0) \ \dots \ X_m(0)]'$ and $X_{S^c}(0) = [X_{m+1}(0) \ \dots \ X_n(0)]'$.

The quantity that will play a central role in the proof of Theorem III.1 is the mutual information term, $I(X_S(T); X_{S^c}(0)|X_S(0))$. 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 $I(X_S(T); X_{S^c}(0)|X_S(0))$.

In the first of our series of lemmas, we bound from above $I(X_S(T); X_{S^c}(0)|X_S(0))$, by the mutual information between the inputs and the outputs of the channels that traverse the cut.

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

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \leq \sum_{l=1}^{N} I(V_S(l); U_{S^c}(l)|U_S(l)),$$

where N is the channel code block length, 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 VI.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 memoryless 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 $X_S(0)$. 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 $I(X_S(T); X_{S^c}(0)|X_S(0))$. 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, $I(X_S(T); C_S|X_S(0))$. 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 VI.3. If $E(\|X(T) - C\|^2) \leq \beta 2^{-\alpha}$ then

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \geq L(S)$$

where,

$$L(S) = h(C_S|X_S(0)) - \frac{|S|}{2} \log 2\pi e\beta + \frac{|S|}{2} \log |S| + |S| \frac{\alpha}{2},$$

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, C_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 VI.4. If, for all $i \in \{1, \dots, n\}$, $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$, then,

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \geq \bar{L}(S),$$

where $\bar{L}(S) = h(C_S|X_S(0)) - \frac{|S|}{2} \log 2\pi e\beta + |S| \frac{\alpha}{2}$.

When, for all i, $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$, 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 VI.1 and VI.2 are in Appendix I. In the next sections, we prove Lemma VI.3 and Corollary VI.4. Then, we prove Theorem III.1.

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

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

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \geq L(S)$$

where,

$$L(S) = h(C_S|X_S(0)) - \frac{|S|}{2}\log 2\pi e\beta + \frac{|S|}{2}\log|S| + |S|\frac{\alpha}{2},$$

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

We start the proof by observing the following.

$$\begin{aligned} & I(X_S(T); X_{S^c}(0)|X_S(0)) \\ \overset{(a)}{=} & I(X_S(T); X(0)|X_S(0)) \\ \overset{(b)}{\geq} & I(X_S(T); C_S|X_S(0)) \end{aligned}$$

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 $C_i = f_i(X(0))$.

Second, we obtain a lower bound on $I(X_S(T); C_S|X_S(0))$ in terms of the desired mean square criterion. We have the following series of inequalities.

$$\begin{aligned} & I(X_S(T); C_S|X_S(0)) \\ =& h(C_S|X_S(0)) - h(C_S|X_S(T), X_S(0)) \\ =& h(C_S|X_S(0)) - h(C_S - X_S(T)|X_S(T), X_S(0)) \\ \overset{(c)}{\geq} & h(C_S|X_S(0)) - h(C_S - X_S(T)) \end{aligned} \tag{1}$$

where, (c) follows because conditioning reduces entropy.

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

$$h(X_S(T) - C_S) \leq \frac{1}{2}\log(2\pi e)^m |Z|, \tag{2}$$

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

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

Here, (d) follows due to Hadamard's inequality. To see (e), we have the following proposition.

Proposition VI.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 $y_i = E(X_i(T) - C_i)^2$ and observing that

$$\begin{aligned} \sum_{i=1}^{m} y_i &= E(\|X_S(T) - C_S\|^2) \\ &\leq E(\|X(T) - C\|^2) \\ &\leq \beta 2^{-\alpha}, \end{aligned}$$

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 VI.4. Recall that in this corollary, we had the weaker condition that for all $i \in \{1, \ldots, n\}$, $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$. In this case, we show that we have the smaller lower bound,

$$\bar{L}(S) = h(C_S|X_S(0)) - \frac{|S|}{2}\log 2\pi e\beta + |S|\frac{\alpha}{2}.$$

To see this, observe that $E(X_i(T) - C_i)^2 \leq \beta 2^{-\alpha}$ implies $E(\|X_S(T) - C_S\|^2) \leq |S|\beta 2^{-\alpha}$. So, replacing β in $L(S)$ of the previous lemma by $|S|\beta$ yields the desired result. □ □

B. Proof of Theorem III.1

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

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \leq \sum_{l=1}^{N} I(V_S(l); U_{S^c}(l)|U_S(l)), \tag{4}$$

where N is the channel code block length, 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.

Second, by Lemma VI.2, because we have assumed that the channels connecting the nodes are memoryless 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 equations (4) and (5) with Corollary VI.4 to obtain

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

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 (i) (6) holds for any cut, and, (ii) by assumption, each of the channels transmits one symbol per second, so $T = N$. □

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, $h(C_S|X_S(0))$, where $C_S = [C_1 \ \ldots \ C_m].'$ If the C_i's are different random variables, then the differential entropy term is well-defined. However, if two entries of C_S are the same random variable, for example if both are $f(X(0))$, then $h(C_S|X_S(0))$ 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 non-trivial 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 non-trivial lower bound for the modified problem, that also holds for our 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 Figure 2.

The aforementioned technique will be used in the next section. In the examples below, we demonstrate the computation of $h(C_S|X_S(0))$ when we introduce the auxiliary random variables.

Example VI.6 (The Solution)**.** Let nodes $\{1, \ldots, m\}$, $m \leq n/2$, belong to set S, so that $C_S = [C_1 \ \ldots \ C_m].'$ Let $C_1 = f(X(0))$ and $C_i = f(X(0)) + a_i \epsilon_{j_i}$ 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 $(X_{j_i}(0), \epsilon_{j_i})$.

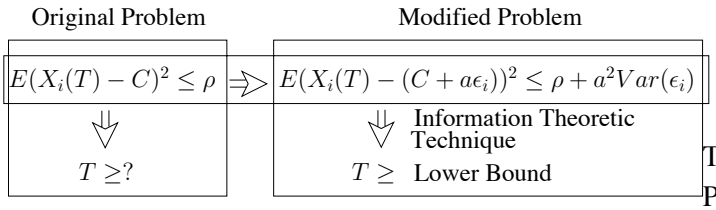

Fig. 2. Diagram illustrating the use of the Information Theoretic technique to obtain a lower bound in a situation where all nodes learn C.

Furthermore, we assume that f is separable, meaning $f(X(0)) = f_S(X_S(0)) + f_{S^c}(X_{S^c}(0))$. Finally, we assume that the $X_i(0)$'s and ϵ_i's are mutually independent. Then,

$$\begin{aligned}
&h(C_S|X_S(0)) \\
=&h\left(f_{S^c}(X_{S^c}(0)), f_{S^c}(X_{S^c}(0)) + a_2\epsilon_{j_2}, \right. \\
&\qquad \left. \ldots, f_{S^c}(X_{S^c}(0)) + a_m\epsilon_{j_m} | X_S(0)\right) \\
\overset{(a)}{=}&h(f_{S^c}(X_{S^c}(0)), f_{S^c}(X_{S^c}(0)) + a_2\epsilon_{j_2}, \\
&\qquad \ldots, f_{S^c}(X_{S^c}(0)) + a_m\epsilon_{j_m}) \\
\overset{(b)}{=}&h(f_{S^c}(X_{S^c}(0))) + \sum_{i=2}^{m} h(a_i\epsilon_{j_i}) \\
\overset{(c)}{=}&h(f_{S^c}(X_{S^c}(0))) + \sum_{i=2}^{m} h(\epsilon_{j_i}) + \log \prod_{i=2}^{m} |a_i|,
\end{aligned}$$

where,

(a) follows because we have assumed that the $X_i(0)$'s and ϵ_i's are mutually independent,

(b) follows by the chain rule for differential entropy, and again using the fact that the $X_i(0)$'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 [4, 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 VI.4.

Example VI.7 (Using the Solution for a Linear Function)**.** In addition to the assumptions in Example VI.6, let $f(X(0)) = \sum_{j=1}^{n} \beta_j X_j(0)$. 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 VI.6, we have that

$$\begin{aligned}
h(C_S|X_S(0)) =& h\left(\sum_{j \in S^c} \beta_j X_j(0)\right) + \frac{m-1}{2}\log 2\pi e\eta \\
&+ \log \prod_{i=2}^{m} |a_i|. \qquad (7)
\end{aligned}$$

To evaluate $h\left(\sum_{j \in S^c} \beta_j X_j(0)\right)$, we use the Entropy Power Inequality, namely, for independent $X_i(0)$'s,

$$2^{2h\left(\sum_{j \in S^c} \beta_j X_j(0)\right)} \geq \sum_{j \in S^c} 2^{2h(\beta_j X_j(0))},$$

which implies that

$$h\left(\sum_{j\in S^c}\beta_j X_j(0)\right) \geq \frac{1}{2}\log\left(\sum_{j\in S^c} 2^{2h(\beta_j X_j(0))}\right).$$

Now, if we assume that each $X_i(0)$ is uniformly distributed in the interval between 1 and $B+1$, $X_i(0) \sim U[1, B+1]$, then,

$$h\left(\beta_j X_j(0)\right) = \log|\beta_j| B.$$

So,

$$h\left(\sum_{j\in S^c}\beta_j X_j(0)\right) \geq \frac{1}{2}\log\left(B^2\sum_{j\in S^c}\beta_j^2\right) = \log B + \frac{1}{2}\log\sum_{j\in S^c}\beta_j^2. \quad (8)$$

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

$$\bar{L}(S) = h(C_S|X_S(0)) - \frac{|S|}{2}\log 2\pi e\beta + |S|\frac{\alpha}{2},$$

and $|S| = m$. Assuming that $\beta = 1$ and using equation (7) together with the inequality of equation (8), we have that

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

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.

VII. 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 via erasure channels. We present a distributed algorithm for computation in this scenario 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. The Information Theoretic Lower Bound

In this section, we provide the proof of Theorem III.4. We will use the techniques that we have developed in section VI. In particular, we will use the results of Examples VI.6 and VI.7, namely equation (9).

Proof of Theorem III.4. Recall that $C = \sum_{j=1}^n \beta_j X_j(0)$. Suppose that we have any realization of the initial conditions, $A = \{X_1(0) = x_1, \ldots, X_n(0) = x_n\}$. We are given an algorithm that guarantees, for every such realization, that at time T each node, i, has an estimate, $X_i(T)$, of C: $\sum_{j=1}^n \beta_j x_j$. Furthermore, for this algorithm, the estimate $X_i(T)$ is within an ε-interval of the true value of C, with desired probability. That is,

$$\mathbf{P}\left(|X_i(T) - C| \leq \varepsilon C \Big| A\right) \geq 1 - \delta. \quad (10)$$

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 equation (10) must satisfy, for small enough δ, a mean square error criterion:
$$E(X_i(T) - C)^2 \leq \varepsilon^2 E(C^2) + \kappa\delta.$$
2) Let $C_1 = C$ and $C_i = C + a\epsilon_{j_i}$ 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, $X_i(0)$. Then,
$$E(X_i(T) - C_i)^2 \leq \varepsilon^2 E(C^2) + a^2\eta + \kappa\delta.$$
3) Next, let S^* and $(S^*)^c$ be the sets for which
$$\frac{\sum_{i\in S, j\notin S}\mathbf{C}_{ij}}{|S|}$$
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 $C_{S^*} = [C_1 \ \ldots \ C_m]'$, where the C_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 $(X_{j_i}(0), \epsilon_{j_i})$. Denote $[\epsilon_{j_1} \ \ldots \ \epsilon_{j_m}]$ by ϵ. Using the derivations of section VI, we have that
$$\begin{aligned} T\sum_{i\in(S^*)^c}\sum_{j\in S^*}\mathbf{C}_{ij} &\geq I(X_{S^*}(T); X_{(S^*)^c}(0)|X_{S^*}(0)) \\ &\overset{(a)}{=} I(X_{S^*}(T); X_{(S^*)^c}(0), \epsilon|X_{S^*}(0)) \\ &\geq I(X_{S^*}(T); C_{S^*}|X_{S^*}(0)) \\ &\geq \bar{L}(S^*), \end{aligned}$$

where, (a) follows because $X_{S^*}(T)$ is the vector of estimates produced by the algorithm, and depends on the initial conditions, $X_i(0)$'s, while the ϵ_{j_i}'s are independent of $X_i(0)$'s.
Recall that

$$\bar{L}(S^*) = h(C_{S^*}|X_{S^*}(0)) - \frac{|S^*|}{2}\log 2\pi e\beta + |S^*|\frac{\alpha}{2}.$$

Note that from Step 2, we have that $\beta 2^{-\alpha} = \varepsilon^2 E(C^2) + a^2\eta + \kappa\delta$. So, we let $\beta = 1$ and we have $\alpha = -\log(\varepsilon^2 E(C^2) + a^2\eta + \kappa\delta)$.

5) Next, we compute $h(C_{S^*}|X_{S^*}(0))$ given the assumptions of our formulation. Recall that we have performed these computations in Example VI.7. We obtained the following:

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

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

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,

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

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

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

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

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

Combining with Step 4, we have that

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

Rearranging, we have that

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

Here, we must have $\varepsilon^2 M \in \left[0, \frac{1}{B}\left(1 - \left(\frac{1}{B}\right)^{\frac{2}{n}} - \kappa\delta\right)\right)$, 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

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

□

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.4 by proving the statements of Step 1 and Step 2.

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

First, observe that,

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

is equivalent to

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

Next, when we condition on A, C is a fixed number. So,

we have we have that

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

where the last inequality follows

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

Finally, we have that

$$E(X_i(T)-C)^2=E\left(E\left((X_i(T)-C)^2\Big|A\right)\right),$$

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

Proof of Step 2. We show that if $E(X_i(T)-C)^2\leq\varepsilon^2E(C^2)+\kappa\delta$, then $E(X_i(T)-C_i)^2\leq\varepsilon^2E(C^2)+a^2\eta+\kappa\delta$, where $C_i=C+a\epsilon_{j_i}$, and ϵ_{j_i} has mean zero and variance η and is independent of all the $X_i(0)$'s.

$$\begin{aligned}
&E(X_i(T)-C_i)^2\\
=&E(X_i(T)-C-a\epsilon_{j_i})^2\\
=&E(X_i(T)-C)^2+E(a\epsilon_{j_i})^2-2E(X_i(T)-C)(a\epsilon_{j_i})\\
\overset{(a)}{=}&E(X_i(T)-C)^2+E(a\epsilon_{j_i})^2-2E(X_i(T)-C)E(a\epsilon_{j_i})\\
\overset{(b)}{=}&E(X_i(T)-C)^2+E(a\epsilon_{j_i})^2,
\end{aligned}$$

where,

(a) follows because $X_i(T)$ is the estimate produced by the algorithm, and depends on the initial conditions, $X_i(0)$'s, while ϵ_{j_i} is independent of $X_i(0)$'s, and,
(b) follows because ϵ_{j_i} has mean zero.

□

B. A Tight Upper Bound: An Algorithm

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.

We describe in detail the problem formulation in the next section. The algorithm that we describe is based on an algorithm by Mosk-Aoyama and Shah [19]. In section VII-B.2 we discuss this algorithm and its applicability to our formulation. In section VII-B.3 we summarize our main results. In section VII-B.4, we describe the contributions of [19] 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 VII-B.5, 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 [19].

1) Problem Formulation: Let an arbitrary connected network of n nodes be represented by the undirected graph $G=(V,E)$. The nodes are arbitrarily enumerated and are the vertices of the graph, $V=\{1,\ldots,n\}$; the enumeration is for the purpose of analysis only as the computation algorithm does not depend on the identities of the nodes. If nodes i and j communicate with each other, then the edge (i,j) belongs to the set E.

Each node i has a measurement or initial value $x_i(0)\in\mathbb{R}$. We let the vector x represent all the initial values in the network, $x=(x_1(0)\ \ldots\ x_n(0))$. The goal of the nodes is to each acquire an estimate of a given function,f, of all the initial values. In this section, the function f is separable, defined as follows. Here, 2^V denotes the power set of V.

Definition VII.1. $f:\mathbb{R}^n\times 2^V\to\mathbb{R}$ is separable if there exist functions $f_1,\ldots,f_n$ such that for all $S\subseteq V$,

$$f(x,S)=\sum_{i\in S}f_i(x_i(0)).$$

Furthermore, we assume $f\in\mathcal{F}$ where $\mathcal{F}$ is the class of all separable functions with $f_i(x_i(0))\geq 1$ for all $x_i(0)\in\mathbb{R}$ and $i=1,\ldots,n$.

The performance of an algorithm, $\mathcal{C}$, used by the nodes to compute an estimate of $f(x,V)$ at each node, is measured by the algorithm's (ε,δ)-computation time, $T_{\mathcal{C}}^{\text{cmp}}(\varepsilon,\delta)$. It is the time until the estimates at all nodes are within a factor of $1\pm\varepsilon$ of $f(x,V)$, with probability larger than $1-\delta$. The definition follows, where $\hat{y}_i(k)$ denotes the estimate of $f(x,V)$ at node i at time k.

Definition VII.2. For $\varepsilon > 0$ and $\delta \in (0,1)$, the (ε,δ)-computing time of an algorithm, $\mathcal{C}$, denoted as $T_{\mathcal{C}}^{\text{cmp}}(\varepsilon,\delta)$ is defined as

$$T_{\mathcal{C}}^{\text{cmp}}(\varepsilon,\delta) = \sup_{f\in\mathcal{F}} \sup_{x\in\mathbb{R}^n} \inf\{k : \mathbf{P}(\cup_{i=1}^n \{\hat{y}_i(k) \notin [(1-\varepsilon)f(x,V),(1+\varepsilon)f(x,V)]\}) \leq \delta\}.$$

The algorithm described here depends on the nodes' use of an information spreading algorithm, $\mathcal{D}$, as a subroutine to communicate to each other their messages. 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 VII.3. For a given $\delta \in (0,1)$, the δ-information-spreading time, of the algorithm $\mathcal{D}$, $T_{\mathcal{D}}^{\text{spr}}(\delta)$, is

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

Consider a model where 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 messages to node j with probability p_{ij} and with probability p_{ii} sends its messages to no other nodes; if $(i,j) \notin E$, $p_{ij} = 0$. So, the matrix $P = [p_{ij}]$ is a stochastic matrix that describes the information spreading algorithm. The information spreading time if this algorithm is derived in terms of the "conductance" of P.

Definition VII.4. For a stochastic matrix P, the conductance of P, denoted $\Phi(P)$, is

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

2) Background: The algorithm that we describe is based on an algorithm by Mosk-Aoyama and Shah [19]. In that formulation, each node has a fixed real-valued initial condition, that is bounded away from zero. Nodes compute a separable function [2] 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 [19], 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 $(i,j) \notin E$, $p_{ij} = 0$.

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

The algorithm of [19] 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 graph representing the communication topology. For a large class of graphs, conductance grows like $O(1/\text{diameter})$.

The drawback of the algorithm in [19], 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 quantizing, 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[3] 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 $\mathbf{C}_{ij} = p_{ij}$, 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 [19], is that we assume that the initial conditions lie in a bounded interval, $[1,B]$, whereas in [19] 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

[3]In [19], it is assumed that each node can "contact" at most one other node; but it can be contacted by more than one nodes. Under independent "erasure" channel model, each node can "contact" more than one node. However, for our purpose this is only beneficial as interest here is in "quicker" information dissemination. We will refrain from discussing this difference in further detail.

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.4.

As such, the computation time of the algorithm we describe below must scale reciprocally with conductance. 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 [19] and its analysis. As a consequence, we obtain the fastest, in terms of dependence on network topology, quantized distributed algorithm for separable function computation.

3) Main Result: The main result of this section is stated in the following theorem.

Theorem VII.5. *Let P be a stochastic and symmetric matrix for which if $(i,j) \notin E$, $p_{ij} = 0$. There exists an algorithm $\mathcal{AP}^{\mathcal{Q}}$ 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 $\varepsilon \in (\gamma f(x,V), \gamma f(x,V) + \frac{1}{2})$ and $\delta \in (0,1)$,*

$$T^{cmp}_{\mathcal{AP}^{\mathcal{Q}}}(\varepsilon,\delta) = O\left(\varepsilon^{-2}\log e\delta^{-1}\frac{\log n\delta^{-1}}{\Phi(P)}\right). \qquad (11)$$

For example, the bound implied by the above theorem when $\delta = \frac{1}{n^2}$ is

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

Recall that by the Information Theoretic lower bound derived in section VII-A, we have that the computation time is lower bounded as

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

where B is a constant such that for all i, $f_i(x_i) \leq B$.

Because the computation time and graph conductance are reciprocally related in both this lower bound and the upper bound in (11), we conclude that our results are tight in capturing the scaling of the computation time with respect to the graph conductance. So, our algorithm is optimal in its dependence on the network topology.

4) Unquantized Function Computation: In [19], 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, and,
- 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 [19], 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, $W^i_1, \ldots, W^i_r$. 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, $W^*_l = \min_{i=1,\ldots,n} W^i_l$. 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 $W^i = (W^i_1, \ldots, W^i_r)$ and needs to obtain $\bar{W} = (\bar{W}_1, \ldots, \bar{W}_r)$, where $\bar{W}_l = \min_{i=1,\ldots,n} W^i_l$. To compute $\bar{W}$, each node

maintains an r-dimensional vector, $\hat{\bar{w}}^i = (\hat{\bar{w}}^i_1, \ldots, \hat{\bar{w}}^i_r)$, 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 $\hat{\bar{w}}^j(k^-)$, node i will update its vector by setting $\hat{\bar{w}}^i_l(k^+) = \min(\hat{\bar{w}}^i_l(k^-), \hat{\bar{w}}^j_l(k^-))$, for $l = 1, \ldots, r$.

As argued in [19], 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, $\hat{\bar{w}}^i(k) = \bar{W}$ when $k = rT^{\text{spr}}_{\mathcal{D}}(\delta)$, 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 [19] 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 [19].

Theorem VII.6. *Given an information spreading algorithm $\mathcal{D}$ with δ-spreading time $T^{spr}_{\mathcal{D}}(\delta)$ 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 $\delta \in (0,1)$,*

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

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, described in section VII-B.3. We refer the reader to [19] 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 VII.7. *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 $\delta \in (0,1)$,*

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

5) Quantized Function Computation: The nodes need to each acquire an estimate of $f(x,V) = \sum_{i=1}^n f_i(x_i(0))$. For convenience, we denote $f_i(x_i(0))$ by θ_i, and $y = f(x,V) = \sum_{i=1}^n \theta_i$ is the quantity to be estimated by the nodes. We denote the estimate of y at node i by $\hat{y}^Q_i$. 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.

We assume that node i can compute θ_i without any communication. Further, we assume that there exists a B for which: for all i, $\theta_i \in [1, B]$.

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

$$\mathbf{P}\left\{\cap_{i=1}^n \{|\hat{y}^Q_i(k) - y| \leq \varepsilon y\}\right\} \geq 1 - \delta,$$

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 VII-B.4 for computation of minima using information spreading. Now, node i quantizes a value $\hat{\bar{w}}^i_l$ that it needs to communicate to its neighbor, j, where node i maps the value $\hat{\bar{w}}^i_l$ 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 $\hat{\bar{w}}^j_l$. 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 $k = rT^{\text{spr}}_{\mathcal{D}}(\delta)$. 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 $rT^{\text{spr}}_{\mathcal{D}}(\delta)$ time has elapsed, each node has obtained the r minima with probability larger than $1-\delta$.
4) Node i sets its estimate of y, $\hat{y}^Q_i$, 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 $\mathbf{P}\left\{\cap_{i=1}^{n}\{|\hat{y}_i^Q - y| \leq \varepsilon y\}\right\} \geq 1-\delta$ is achieved. Determining how large r and M must be leads to the main theorem of this section.

Theorem VII.8. *Given an information spreading algorithm $\mathcal{D}$ with δ-spreading time $T_{\mathcal{D}}^{spr}(\delta)$ 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 $\varepsilon \in (\gamma f(x,V), \gamma f(x,V) + \frac{1}{2})$ and $\delta \in (0,1)$,*

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

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, $\hat{y}_i^Q$ 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 II.

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 VII.8.

Combining the result of Theorem VII.8 with that of Theorem VII.7 yields Theorem VII.5. Comparison with a lower bound obtained via Information Theoretic inequalities in section VII-A reveals that the reciprocal dependence between computation time and graph conductance in the upper bound of Theorem VII.5 matches the lower bound. Hence the upper bound is tight in capturing the effect of the graph conductance $\Phi(G)$.

VIII. Discussion and Conclusions

In this paper, we've studied a network of n nodes communicating over 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 trade-off 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 via erasure channels 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 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've described an algorithm that can be used in this setting of nodes computing a sum via 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 and show 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.

Appendix I
Proofs of Lemmas VI.1 and VI.2

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

A. Proof of Lemma VI.1

We prove the following inequality:

$$I(X_S(T); X_{S^c}(0)|X_S(0)) \leq \sum_{l=1}^{N} I(V_S(l); U_{S^c}(l)|U_S(l)), \tag{A.12}$$

where N is the channel code block length, 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.

Recall that by the assumptions we made in our problem formulation, the end of the k^{th} time slot corresponds to time T. So, $X_S(T) = X_S(k)$. In this proof, it is convenient to use $X_S(k)$. We need to refer to the sequence of estimates at node i up until time T. The most natural way is to enumerate using integers: $X_i^k = \{X_i(1), X_i(2), \dots X_i(k)\}$.

For this proof, we use the general formulation for multi-terminal networks of [4, section 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 $U_i^N = (U_i(1), U_i(2), \dots U_i(N))$. The indices in brackets represent channel use. As before, if nodes 1 to m belong to S, we have that $V_S = (V_1, \dots, V_m)$. Similarly, we have that $V_S(l) = (V_1(l), \dots, V_m(l))$, representing the variables received after the l-th use of the channel.

We assume that the estimates at node i, X_i^k, are a function of the received messages at that node, V_i^N and its own data, $X_i(0)$, $X_i^k = \phi_i(V_i^N, X_i(0))$. 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, $X_i(0)$, $U_i(l) = \psi_i(V_i^{l-1}, X_i(0))$.

As in [4], 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 $p(v_1, \dots, v_n|u_1, \dots, u_n)$. 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 [20, 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 [20, Ch.3] for technical details.

The following inequalities proceed in the same manner as Theorem 14.10.1 in [4]. For convenience, we repeat the steps here using our notation.

$$\begin{aligned}
&I(X_S(k); X_{S^c}(0)|X_S(0)) \\
&\overset{(a)}{\leq} I(X_S(1), \dots, X_S(k); X_{S^c}(0)|X_S(0)) \\
&= I(X_S(1), \dots, X_S(k), X_S(0); X_{S^c}(0)|X_S(0)) \\
&\overset{(b)}{\leq} I(V_1^N, \dots, V_m^N, X_S(0); X_{S^c}(0)|X_S(0)) \\
&= I(V_S(1), \dots, V_S(N); X_{S^c}(0)|X_S(0)) \\
&\overset{(c)}{=} \sum_{l=1}^{N} I(V_S(l); X_{S^c}(0)|X_S(0), V_S(l-1), \dots, V_S(1)) \\
&\overset{(d)}{=} \sum_{l=1}^{N} h(V_S(l)|X_S(0), V_S(l-1), \dots, V_S(1)) \\
&\quad - h(V_S(l)|X_{S^c}(0), X_S(0), V_S(l-1), \dots, V_S(1)) \\
&\overset{(e)}{\leq} \sum_{l=1}^{N} h(V_S(l)|X_S(0), V_S(l-1), \dots V_S(1), U_S(l)) \\
&\quad - h(V_S(l)|X_{S^c}(0), X_S(0), V_S(l-1), \\
&\qquad\qquad \dots, V_S(1), U_S(l), U_{S^c}(l)) \\
&\overset{(f)}{\leq} \sum_{l=1}^{N} h(V_S(l)|U_S(l)) - h(V_S(l)|U_S(l), U_{S^c}(l)) \\
&\overset{(g)}{=} \sum_{l=1}^{N} I(V_S(l); U_{S^c}(l)|U_S(l)).
\end{aligned}$$

Above,

(a) holds by the data processing inequality,
(b) holds again by the data processing inequality, because $X_i^k = \phi_i(V_i^N, X_i(0))$,
(c) follows by the chain rule for mutual information,
(d) follows by the definition of mutual information, (or, in the discrete channel case, it follows by Kolmogorov's formula [20, Ch.3] and by noting that the entropy term is well-defined since V_i would take values in a discrete set),
(e) follows, for the first term, because $U_i(l) = \psi_i(V_i^{l-1}, X_i(0))$, so it does not change the conditioning; and the second part follows because conditioning reduces entropy,
(f) 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,
(g) from the definition of mutual information. □

B. Proof of Lemma VI.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,

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

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 II
PROOF OF THEOREM VII.8

A. 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 [19] 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 $rT_{\mathcal{D}}^{spr}(\delta/2)$ time slots.

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

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

The following lemma will be needed in the proof of Theorem VII.8.

Lemma A II.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 $\hat{y}_i$ denote node i's estimate of y using the modified algorithm of this section, $\mathcal{A}^{\mathcal{Q}}_{\mathcal{M}}$.

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

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

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

Proof. First, note that when $\{\forall i \in V, \widehat{\bar{W}}^i = \bar{W}\}$, we have that for all i, $\hat{y}_i^{-1} = \frac{1}{r}\sum_{l=1}^r \bar{W}_l$. 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 [5], 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. □ □

B. Proof of Theorem VII.8

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^Q, the time until, with probability at least $1 - \delta$, all nodes $i = 1, \ldots, n$ have estimates $\hat{y}_i^Q$ that are within a factor of $1 \pm \varepsilon$ of y. That is,

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

First, suppose that we may communicate real-valued messages between the nodes. We analyse 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,

$$\frac{1}{r}\sum_{l=1}^{r} \min_{i=1,\ldots,n}(w_l^i)_Q \in \left[\frac{1}{r}\sum_{l=1}^{r} \min_{i=1,\ldots,n} w_l^i - \gamma, \frac{1}{r}\sum_{l=1}^{r} \min_{i=1,\ldots,n} w_l^i + \gamma\right]. \tag{A.13}$$

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

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

$$\mathbf{P}\left(\cup_{i=1}^{n}\{\widehat{\bar{W}}^i \neq \bar{W}\}\right) \leq \frac{\delta}{2}. \tag{A.14}$$

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

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

Combining with (A.13), we have that

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

But the event

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

is equivalent to

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

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

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

So,

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

Letting $r \geq 6\mu^{-2}\ln 2\delta^{-1}$, we have that

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

Combining this with (A.14) in the Total Probability Theorem, we have the desired result,

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

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 $O(rT_{\mathcal{D}}^{\text{spr}}(\delta/2))$ 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, $T_{\mathcal{D}}^{\text{spr}}(\delta)$ time slots are needed until the quantized messages are disseminated and the minima computed. Consequently, the computation time of the quantized algorithm will be $O(rT_{\mathcal{D}}^{\text{spr}}(\delta/2))$.

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 $\mu \in (0, 1/2)$, $r \geq 6\mu^{-2}\ln 2\delta^{-1}$. But, $\mu = \varepsilon - y\gamma$; and, $\gamma = \Theta(1/n)$ so, $y\gamma = O(1)$. We therefore have, for $\varepsilon \in (y\gamma, y\gamma + 1/2)$,

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

□

ACKNOWLEDGMENT

O. Ayaso would like to thank Professor Nuno C. Martins (University of Maryland, College Park) for suggesting the utility of Information Theoretic techniques in the context of distributed computation.

REFERENCES

[1] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, December 2005.

[2] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, Seville, Spain, 2005. IEEE.

[3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, June 2006.

[4] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[5] A. Dembo and O. Zeitouni. *Large Deviation Techniques and Applications*. Springer, 1998.

[6] Diaconis and Saloff-Coste. *personal communication*, 2006.

[7] A. G. Dimakis, A.D. Sarwate, and M.J. Wainwright. Geographic gossip : Efficient aggregation for sensor networks. In *5th International ACM/IEEE Symposium on Information Processing in Sensor Networks (IPSN '06)*, April 2006.

[8] A. El Gamal and A. Orlitsky. Interactive data compression. In *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science*, pages 100–108. IEEE, October 1984.

[9] R. Gallager. Finding parity in a simple broadcast network. *IEEE Transactions on Information Theory*, 34:176–180, 1988.

[10] A. Giridhar and P. R. Kumar. Towards a theory of in-network computation in wireless sensor networks. *IEEE Communications Magazine*, 44(4):98–107, April 2006.

[11] N. Goyal, G. Kindler, and M. Saks. Lower bounds for the noisy broadcast problem. *SIAM Journal on Computing*, 37(6):1806–1841, March 2008.

[12] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[13] Z.-Q. Luo and J. N. Tsitsiklis. Data fusion with minimal communication. *IEEE Transactions on Information Theory*, 40(5):1551–1563, September 1994.

[14] N. C. Martins. *Information Theoretic aspects of the control and mode estimation of stochastic systems*. PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Laboratory of Information and Decision Systems, August 2004.

[15] N. C. Martins. Finite gain lp stabilization requires analog control. *Systems and Control Letters*, 55/1:949–954, Novemeber 2006.

[16] N. C. Martins and M. A. Dahleh. Feedback control in the presence of noisy channels: Bode-like fundamental limits of performance. *IEEE Transaction on Automatic Control*, May 2008.

[17] N. C. Martins, M. A. Dahleh, and J. C. Doyle. Fundamental limitations of disturbance attenuation in the presence of side information. *IEEE Transaction on Automatic Control*, 52:56–66, January 2007.

[18] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[19] D. Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *ACM Principles of Distributed Computation*, 2006.

[20] M. S. Pinsker. *Information and Information Stability of Random Variables and Processes*. Holden-Day, Inc., San Francisco, 1964.

[21] K. Savla, F. Bullo, and E. Frazzoli. On traveling salesperson problems for Dubins' vehicle: stochastic and dynamic environments. In *CDC-ECC*, pages 4530–4535, Seville, Spain, December 2005.

[22] J. N. Tsitsiklis and M. Athans. Convergence and asymptotic agreement in distributed decision problems. *IEEE Transactions on Automatic Control*, 29(1):42–50, 1984.

[23] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[24] J.N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD dissertation, Massachusetts Institute of Technology, Laboratory of Information and Decision Systems, Department of Electrical Engineering and Computer Science, 1984.