On Energy for Progressive and Consensus Estimation in Multihop Sensor Networks

Yi Huang and Yingbo Hua, *Fellow, IEEE*

***Abstract*—This paper addresses a transmission energy problem for distributed (or decentralized) estimation in multihop wireless sensor networks (WSN). A primary advantage of distributed estimation is its energy efficiency when compared to centralized estimation. Two distributed estimation schemes are considered in this paper: progressive estimation and consensus estimation. We develop a generalized energy planning algorithm for a progressive estimation method which exploits routing tree and channel state information. We also analyze the energy cost for a consensus estimation method used in broadcast multihop WSN. We demonstrate by analysis and simulation that, subject to an equivalent performance, the total energy cost for consensus estimation is typically much higher than that for progressive estimation, but the peak energy for the former can be less than that for the latter.**

***Index Terms*—Broadcast network, consensus estimation, decentralized estimation, distributed estimation, energy and power planning, multihop sensor networks, network with routing tree, network without routing tree, peer-to-peer network, progressive estimation, wireless sensor networks (WSN).**

I. Introduction

We consider a wireless sensor network (WSN) where each sensor is a device capable of sensing, computing, and wireless communication. Although the energy cost for sensing and computing can be reduced by improving the performance of individual devices, the energy cost for communication is dominated by networking architectures, data fusion protocols and channel characteristics between devices. In this paper, we study the energy cost for communication in WSN by jointly considering networking architecture, data fusion method and channel energy model.

There appear three basic architectures for WSN: single-hop network with fusion center, multihop network with fusion center, and peer-to-peer or broadcast multihop network without fusion center. Many other architectures can be formed by combinations of the three.

Manuscript received June 15, 2010; revised December 22, 2010; accepted April 06, 2011. Date of publication April 21, 2011; date of current version July 13, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Subhrakanti Dey. This work was supported in part by the U. S. National Science Foundation by Grant No. TF-0514736 and by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program. The material in this paper was presented at ICASSP Dallas, TX, March 2010.

Y. Huang was with Department of Electrical Engineering, University of California, Riverside, CA 92521 USA. He is now with Qualcomm R&D, San Diego, CA 92121 USA (e-mail: yihuang9706@gmail.com).

Y. Hua is with Department of Electrical Engineering, University of California, Riverside, CA 92521 USA (e-mail: yhua@ee.ucr.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2144588

In a single-hop WSN with fusion center, data are directly transmitted from sensors to a fusion center before data fusion (i.e., estimation and/or detection) takes place. In recent years, there have been many research activities on distributed estimation methods aimed to reduce the communication cost for single-hop WSN, e.g., see [1]–[8]. However, single-hop WSN is often not an energy efficient architecture especially when the network is flat or more generally when the ratio of the distance between a sensor to the fusion center over the distance between adjacent sensors is large.

To reduce the transmission energy cost in a flat network, a data packet can be relayed from one sensor to another until it reaches a fusion center. This is a multihop WSN. The multihop architecture is generally more efficient in energy than the single-hop architecture for a flat network. However, if the data packets are not compressed as they hop towards the fusion center, the sensors near the fusion center can be over burdened, which is a bottleneck effect. To solve this problem, progressive estimation [10] is a useful idea where data are fused together as they hop from one sensor to another along a routing path. This idea resembles an earlier notion known as decentralized detection in [9]. To further reduce the energy cost at all sensors, the optimization of the number of bits used for quantization at each sensor has been shown in [11] and [12] to be very effective.

A peer-to-peer or broadcast multihop network is useful for situations where there is no centralized control or the network is too dynamic to be regulated centrally. A major approach for distributed fusion in peer-to-peer or broadcast multihop network is known as consensus estimation where sensors exchange information with their neighbors and iteratively update their own information. There are numerous articles in this area, e.g., see [14]–[33]. Although consensus estimation does not require centralized control, the rate of its convergence to a desired result depends on the network topology. Without the knowledge of the network topology or some global statistical information, a node in the network cannot know when an updated information is reliable enough.

In this paper, we present two contributions.

First, we develop a generalized energy planning algorithm for progressive estimation in multihop WSN with fusion center. We assume that a central planner has the knowledge of a routing tree and the channel gains of all communication links for a time window of interest. The purpose of the central planner is to determine a bit allocation for each sensor such that the energy cost of the network is minimized subject to a predetermined estimation quality at the fusion center. The bit allocation determines the number of bits required for quantization of each estimated

parameter at each sensor as well as the number of bits required for transmission through each subchannel from each sensor. The results in this paper are much stronger than that in [12] because we will use the exact channel energy model (not an upper bound) and optimize both quantization and transmission bit allocation (not just quantization bit allocation). We also consider an important special case where the transmission energy is modeled as a linear function of the number of bits transmitted.

Second, we study the energy cost for consensus estimation in a broadcast multihop WSN subject to a pre-given performance. This allows a comparison between progressive estimation and consensus estimation in terms of energy cost. It will be shown that progressive estimation consumes much less total energy than consensus estimation although the latter has an advantage in terms of the largest amount of energy cost by individual sensors. This work appears the first that reveals that there are more energy-efficient estimation schemes than consensus estimation.

In Section II, we review briefly the principle of progressive estimation for multihop WSN and then formulate an energy planning problem that is more general than that in [12]. In Section III, we develop an efficient algorithm for solving the energy planning problem. In Section IV, a simpler energy planning algorithm is presented for the linear energy model. In Section V, we introduce a consensus estimation algorithm for a broadcast multihop WSN under a similar channel condition used for progressive estimation. In Section VI, we investigate a minimized energy cost for consensus estimation. In Section VII, we present simulation examples to compare the minimized energy cost for consensus estimation with that for progressive estimation. We also illustrate the effects of several key parameters used in consensus estimation on its performance. The conclusion is given in Section VIII.

II. Progressive Estimation

A. Estimation Protocol

A multihop WSN with a routing tree is illustrated in Fig. 1. We denote by $\boldsymbol{\theta} \in R^{M\times 1}$ as the unknown deterministic parameter vector of interest during a sampling window of the network, and $\boldsymbol{\theta}_k$ as the initial estimate of $\boldsymbol{\theta}$ at sensor k in the sampling window. Here, $k = 1, 2, \cdots, K$. We assume that $E\{\boldsymbol{\theta}_k\} = \boldsymbol{\theta}$ and $E\{(\boldsymbol{\theta}_k - \boldsymbol{\theta})(\boldsymbol{\theta}_k - \boldsymbol{\theta})^T\} \le \boldsymbol{C}_{\theta,k}$ where E denotes expectation and $\boldsymbol{C}_{\theta,k}$ is known.

For progressive estimation, sensor k performs a fusion only once within one sampling window. Namely, after sensor k receives the quantized estimates of $\boldsymbol{\theta}$ from all its upstream sensors, it performs a fusion and a quantization and then forwards its quantized new estimate to its downstream sensor. More specifically, let $\mathbf{m}_i$ be the quantized new estimate at sensor i, $\mathcal{S}_k$ be the set of the indices of the upstream sensors of sensor k, and $|\mathcal{S}_k|$ be the size of $\mathcal{S}_k$. The new estimate at sensor k is

$$\hat{\boldsymbol{\theta}}_k = \frac{1}{|\mathcal{S}_k|+1}\left(\boldsymbol{\theta}_k + \sum_{i\in\mathcal{S}_k} \boldsymbol{m}_i\right). \tag{1}$$

This is an estimate by simple average. More advanced estimate such as the best linear unbiased estimate (BLUE) or minimum mean squared error (MMSE) estimate could be used here. But it would make it very hard (if not impossible) to establish an energy planning algorithm as shown later. In this paper, we only consider estimation by average.

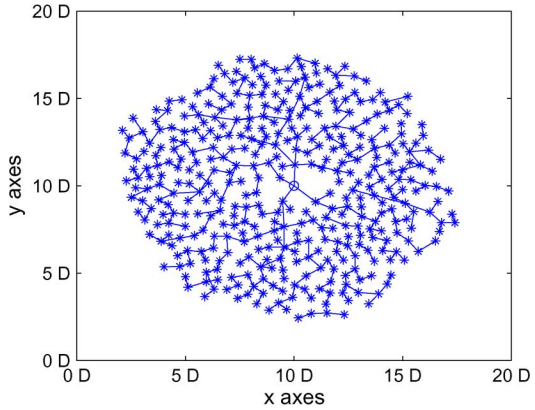


Fig. 1. A multihop sensor network of 400 nodes used for all simulation examples. The routing tree shown here is for progressive estimation, where the fusion center is at the center of the network. For consensus estimation, the routing tree is not needed, each sensor broadcasts its data to its neighbors (assuming no collision), and the neighborhood of each sensor is defined below (37).

The quantization of $\hat{\boldsymbol{\theta}}_k$ into $\mathbf{m}_k$ is done element-wise by the uniform probabilistic quantization as in [2]. In this case, we have $E_q\{\mathbf{m}_k\} = \hat{\boldsymbol{\theta}}_k$ and $E_q\{(\mathbf{m}_k - \hat{\boldsymbol{\theta}}_k)(\mathbf{m}_k - \hat{\boldsymbol{\theta}}_k)^T\} \le \mathbf{C}_{q,k} = diag(\sigma^2_{q,k,1}, \cdots, \sigma^2_{q,k,M})$ where $\sigma^2_{q,k,m} = \frac{4W_m^2}{2^{2B_{k,m}}}$, $B_{k,m}$ is the number of bits used to quantize the mth element $(\hat{\boldsymbol{\theta}}_k)_m$ of $\hat{\boldsymbol{\theta}}_k$, and we assume $-W_m < (\hat{\boldsymbol{\theta}}_k)_m < W_m$. Here, E_q is the expectation with respect to the quantization error only, or equivalently conditional upon $\hat{\boldsymbol{\theta}}_k$. Consequently, we have from (1) that $E\{\mathbf{m}_k\} = \boldsymbol{\theta}$ and

$$E\left\{(\mathbf{m}_k - \boldsymbol{\theta})(\mathbf{m}_k - \boldsymbol{\theta})^T\right\} \doteq \boldsymbol{C}_{m,k} \le \frac{1}{(|\mathcal{S}_k|+1)^2}\boldsymbol{C}_{\theta,k} + \frac{1}{(|\mathcal{S}_k|+1)^2} \times \sum_{i\in\mathcal{S}_k} \boldsymbol{C}_{m,i} + \boldsymbol{C}_{q,k}. \tag{2}$$

At the fusion center denoted by sensor K, we do not need to quantize $\hat{\boldsymbol{\theta}}_K$, and its covariance matrix is denoted by $\boldsymbol{C}_K$. By using (2), it follows (also shown in [12]) that

$$\boldsymbol{C}_K \le \sum_{k=1}^{K} \frac{\alpha_k}{(|\mathcal{S}_k|+1)^2}\boldsymbol{C}_{\theta,k} + \sum_{k=1}^{K-1} \alpha_k \boldsymbol{C}_{q,k} \tag{3}$$

where $\alpha_K = 1$, and $\alpha_l = \frac{\alpha_k}{(|\mathcal{S}_k|+1)^2}$ for $l \in \mathcal{S}_k$. Note that in order to calculate α_k for all k, one should start with the fusion center where $k = K$ and then proceed outwards recursively. The actual values of α_k depend on the routing tree. Taking the trace of (3), we have the mean squared error (MSE) equation

$$\text{MSE} \doteq Tr\{\boldsymbol{C}_K\} \le \xi + \sum_{k=1}^{K-1}\sum_{m=1}^{M} \alpha_k \frac{4W_m^2}{2^{2B_{k,m}}} \tag{4}$$

where $\xi = \sum_{k=1}^{K} \frac{\alpha_k}{(|\mathcal{S}_k|+1)^2} Tr\{\boldsymbol{C}_{\theta,k}\}$ which is invariant to $B_{k,m}$.

If MSE_0 is the desired MSE value at the destination node, then the choice of $B_{k,m}$ for $1 \le k \le K-1$ and $1 \le m \le M$ must satisfy

$$\sum_{k=1}^{K-1}\sum_{m=1}^{M} \alpha_k \frac{W_m^2}{2^{2B_{k,m}}} \le \eta \tag{5}$$

where $\eta = \frac{1}{4}(\text{MSE}_0 - \xi)$.

B. Communication Protocol

As implied by the estimation protocol shown above, the communication protocol is such that within a sampling window, the leaf sensors act first by quantizing their initial estimates and transmitting the quantized estimates to their downstream nodes. Each of all other nodes acts accordingly after it receives the quantized estimates from all of its upstream sensors. Furthermore, we assume that there is no (or virtually no) collision among the transmissions, which can be achieved by various medium access control schemes, e.g., see [34], [35], and the references therein.

The total number of bits to be transmitted from sensor k to its downstream sensor is $B_k = \sum_{m=1}^{M} B_{k,m}$. If there are transmission bandwidth $\mathcal{W}$ and transmission time $\mathcal{T}$ for any sensor, we can partition it into a set of L subchannels where L is well known to be no larger than the time-bandwidth product $\mathcal{TW}$. The number of bits that can be transmitted over the lth subchannel is

$$B'_{k,l} = \varphi \frac{\mathcal{TW}}{L} \log_2\left(1 + \frac{E_{k,l}}{a_{k,l}}\right) \tag{6}$$

where $0 < \varphi < 1$ is a penalty factor due to the heading of data packet, $E_{k,l}$ is the energy used for transmission in the lth subchannel from sensor k, and $a_{k,l} = \frac{N_0}{\phi\mu|h_{k,l}|^2}$. Here, N_0 is the noise spectral density of the RF communication channel, $h_{k,l}$ is the gain of the lth subchannel from sensor k, $0 < \phi < 1$ is a penalty factor due to practical digital coding, and $\mu > 0$ is a factor due to analog waveform modulation [12]. In terms of maximizing $\sum_{l=1}^{L} B'_{k,l}$ subject to $\sum_{l=1}^{L} E_{k,l}$ being a constant, we can let $L = \mathcal{TW}$ without loss of generality. Then, assuming $L = \mathcal{TW}$ and writing $E_{k,l}$ in terms of $B'_{k,l}$, we have

$$E_{k,l} = a_{k,l}\left(2^{B'_{k,l}/\varphi} - 1\right). \tag{7}$$

C. Energy Planning Problem

In the next section, we will solve the following optimization problem to determine $B_{k,m}$ and $B'_{k,l}$ for $1 \le k \le K-1$, $1 \le m \le M$, and $1 \le l \le L$:

$$\min_{B_{k,m},B'_{k,l}} J_p = \sum_{k=1}^{K-1}\sum_{l=1}^{L} E_{k,l}^p = \sum_{k=1}^{K-1}\sum_{l=1}^{L} a_{k,l}^p\left(2^{B'_{k,l}/\varphi} - 1\right)^p \tag{8}$$

subject to

$$\sum_{k=1}^{K-1}\sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} \le \eta \tag{9}$$

$$\sum_{m=1}^{M} B_{k,m} = \sum_{l=1}^{L} B'_{k,l} \text{ for all } k \tag{10}$$

$$B_{k,m} \ge 0, B'_{k,l} \ge 0 \text{ for all } k,\ m, \text{ and } l \tag{11}$$

where $p \ge 1$ and $J_p^{\frac{1}{p}}$ is the pth norm of all components of the energy cost in the network. If we choose $p = 1$, the cost corresponds to the sum energy. If we choose a large p, the cost corresponds approximately to the largest component of the energy cost.

The problem (8) is more general than that formulated in [12] where $2^{\frac{B'_{k,l}}{\varphi}} - 1$ was replaced by its upper bound $2^{\frac{B'_{k,l}}{\varphi}}$, and $B'_{k,l}$ was chosen to be invariant to the subchannel index l assuming that $h_{k,l}$ is invariant to l. If l denotes a frequency subchannel and $\mathcal{W}$ is large, $h_{k,l}$ typically depends on l. So, it is important to treat $B'_{k,l}$ as a function of l in general. Consequently, (8) involves two types of bit allocations. One is bit allocation for quantization of estimates, which is represented by $B_{k,m}$. The other is bit allocation for transmission of estimates, which is represented by $B'_{k,l}$. The relationship between $B_{k,m}$ and $B'_{k,l}$ is not trivial except $\sum_{m=1}^{M} B_{k,m} = \sum_{l=1}^{L} B'_{k,l}$.

It is also useful to note that for the special case $p = 1$, the subproblem $\min \sum_{l=1}^{L} E_{k,l}$ subject to $\sum_{l=1}^{L} B'_{k,l}$ being a constant, or its dual $\max \sum_{l=1}^{L} B'_{k,l}$ subject to $\sum_{l=1}^{L} E_{k,l}$ being a constant, has the well known waterfilling solution. But (8) is much more complex.

One can verify that if $B_{k,m}$ and $B'_{k,l}$ are treated as real numbers, (8) is convex. The proof is given in Appendix I. Hence, the globally optimal solution to the problem, assuming real $B_{k,m}$ and $B'_{k,l}$, can be found. If a general-purpose convex optimization program such as in Matlab, the computational speed is very slow. In the next section, we present a much more efficient algorithm to solve this problem.

III. Energy Planning for Progressive Estimation

In this section, we develop an energy planning algorithm for progressive estimation by solving (8). To distinguish this algorithm from that in [12], we call this algorithm the generalized algorithm. To solve (8), we apply the KKT method [36]. The complete set of the KKT equations for (8) are given by

$$a_{k,l}^p p \frac{\ln 2}{\varphi} 2^{B'_{k,l}/\varphi}\left(2^{B'_{k,l}/\varphi} - 1\right)^{p-1} - \nu_{k,l} + \xi_k = 0, \quad \text{for all } k \text{ and } l \tag{12}$$

$$-2\mu\alpha_k W_m^2 (\ln 2) 2^{-2B_{k,m}} - \lambda_{k,m} - \xi_k = 0 \quad \text{for all } k \text{ and } m \tag{13}$$

$$\sum_{k=1}^{K-1}\sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} - \eta \le 0 \tag{14}$$

$$\sum_{m=1}^{M} B_{k,m} = \sum_{l=1}^{L} B'_{k,l} \text{ for all } k \tag{15}$$

$$\mu\left(\sum_{k=1}^{K-1}\sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} - \eta\right) = 0 \tag{16}$$

$$\lambda_{k,m} B_{k,m} = 0, \text{ for all } k \text{ and } m \tag{17}$$

$$\nu_{k,l} B'_{k,l} = 0, \text{ for all } k \text{ and } l \tag{18}$$

where $B_{k,m} \geq 0$, $B'_{k,l} \geq 0$, $\mu \geq 0$, $\lambda_{k,m} \geq 0$, and $\nu_{k,l} \geq 0$. It is easy to verify from the KKT equations that $\lambda_{k,m} = 0$ for all k and m where $B_{k,m} > 0$, and $\nu_{k,l} = 0$ for all k and l where $B'_{k,l} > 0$. It is also easy to verify that $\mu > 0$.

Finding the complete solution to the above nonlinear equations requires iterative search. To derive our search algorithm, we first define

$$\eta_k \doteq \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} \tag{19}$$

From (13), we have that for $B_{k,m} > 0$

$$\frac{\alpha_k W_m^2}{2^{2B_{k,m}}} = \frac{-\xi_k}{2\mu \ln 2} \tag{20}$$

Using (20) in (19) yields

$$\eta_k = \frac{-\xi_k \left|\mathcal{M}_k^+\right|}{2\mu \ln 2} + \sum_{m \notin \mathcal{M}_k^+} \alpha_k W_m^2 \tag{21}$$

where $\mathcal{M}_k^+ = \{m | B_{k,m} > 0\}$ and $|\mathcal{M}_k^+|$ is the size of the set $\mathcal{M}_k^+$. Substituting $\frac{\xi_k}{\mu}$ from (21) into (20), we find

$$B_{k,m} = \frac{1}{2} \log_2 \frac{\alpha_k W_m^2 \left|\mathcal{M}_k^+\right|}{\eta'_k} \tag{22}$$

where $m \in \mathcal{M}_k^+$, and $\eta'_k = \eta_k - \sum_{m \notin \mathcal{M}_k^+} \alpha_k W_m^2$. A slightly more compact form of this solution is

$$B_{k,m} = \left(\frac{1}{2} \log_2 \frac{\alpha_k W_m^2 \left|\mathcal{M}_k^+\right|}{\eta'_k} \right)^+ \tag{23}$$

where $(x)^+ = \max(x, 0)$.

To compute (23) with a given η_k, we initially assume that $\mathcal{M}_k^+$ contains all $m = 1, \cdots, M$. We then apply (23) to calculate $B_{k,m}$ for all m. For those $B_{k,m} = 0$, we exclude the corresponding indexes m from the set $\mathcal{M}_k^+$. The new set $\mathcal{M}_k^+$ is then applied to calculate new $B_{k,m}$ for all m via (23). This iterative procedure continues until $B_{k,m} > 0$ for $m \in \mathcal{M}_k^+$ and $B_{k,m} = 0$ for $m \notin \mathcal{M}_k^+$. In Appendix II, we prove that the number of iterations in computing (23) is finite.

The first term in (12) is monotonically increasing function of $B'_{k,l} > 0$. So, for a given $B'_{k,l} > 0$, there is an unique ξ_k, and for a given ξ_k, there is either a unique $B'_{k,l} > 0$ or $B'_{k,l} = 0$. The computation of this one-to-one mapping can be easily implemented via the bisection method. For convenience, we can write $B'_{k,l} = f^+_{k,l}(\xi_k)$ where the function $f_{k,l}$ is the inverse function of the first term of (12). Note that we do not need any more explicit expression of $f_{k,l}$ for the reason aforementioned. Because of (15), we can find ξ_k by solving

$$\sum_{l=1}^{L} f^+_{k,l}(\xi_k) = B_k \tag{24}$$

where $B_k = \sum_{m=1}^{M} B_{k,m}$. The entire left-hand side (LHS) expression of (24) is a monotonically increasing function of ξ_k. So, with a given B_k, it is easy to find the corresponding ξ_k and hence the corresponding $B'_{k,l} = f^+_{k,l}(\xi_k)$ for all l.

Up to now, we have obtained the optimal solutions for $B_{k,m}$ and $B'_{k,l}$ for all k, m, and l provided that η_k for all k are given. To find the optimal η_k, we need a search algorithm as developed next. Since $\mu > 0$, the optimal η_k must be such that $\sum_{k=1}^{K-1} \eta_k = \eta$ due to (19) and (16). Using $\sum_{k=1}^{K-1} \eta_k = \eta$ and (21), we can eliminate μ and obtain

$$\eta_k = \eta' \frac{\xi_k \left|\mathcal{M}_k^+\right|}{\sum_{j=1}^{K-1} \xi_j \left|\mathcal{M}_j^+\right|} + \sum_{m \in \mathcal{M}_k^+} \alpha_k W_m^2 \tag{25}$$

where $\eta' = \eta - \sum_{j=1}^{K-1} \sum_{m \notin \mathcal{M}_j^+} \alpha_j W_m^2$. Here η_k is expressed as a function of $\mathcal{M}_j^+$ and ξ_j for all j. Note that $\mathcal{M}_j^+$ is a byproduct of $B_{j,m}$ for all m, and ξ_j is a byproduct of $B'_{j,l}$ for all l.

It is important to understand an important meaning of (25). This equation says that if we have an estimate of the optimal η_k, denoted by $\eta_k(s)$, and computed $B_{j,m}$ and $B'_{j,l}$ based on $\eta_k(s)$, then we can find a new $\eta_k(s+1)$ based on (25). Note that s is the iteration index. If $\eta_k(s)$ is already optimal, so is $\eta_k(s+1)$, i.e., $\eta_k(s) = \eta_k(s+1)$. But if $\eta_k(s)$ is not optimal, it is intuitive to think that $\eta_k(s+1)$ is an improved version of η_k and converges to the optimal η_k as s increases. However, this is not always true. In our simulation, we found that if $p > 1$, the iteration of $\eta_k(s)$ based on (25) alone can diverge. To solve this problem, we propose the following that dampens the change of η_k from one iteration to another

$$\eta_k(s+1) = \beta \hat{\eta}_k(s+1) + (1-\beta)\eta_k(s) \tag{26}$$

where $\hat{\eta}_k(s+1)$ in the right-hand side (RHS) is computed from (25) based on $\eta_k(s)$, and $\eta_k(s+1)$ in the LHS is the new estimate of η_k. The dampening factor $\beta > 0$ at each iteration should be chosen to be small enough so that there is no divergence. Given the convexity of (8), there should exist a small enough $\beta > 0$ such that (26) guarantees the convergence of $\eta_k(s)$. Finding an exact value of $\beta > 0$ for guaranteed convergence in all cases is difficult. According to our simulation, however, we can choose any $\beta > 0$ that meets $\beta \hat{\eta}_k(s+1) \leq (1-\beta)\eta_k(s)$ at each iteration. This simple rule of choosing $\beta > 0$ at each iteration worked well for all cases in simulation.

With the derivations and discussions shown above, our algorithm for computing the solution to the problem (8) can be summarized here.

1) Initialization: Choose $\mathcal{M}_k^+ = \{1, 2, \cdots, M\}$ for all k. Also set $s = 0$ and $\eta_k(0) = \frac{\eta}{K-1}$ for all k.
2) Step 1: Compute $B_{k,m}$ and $\mathcal{M}_k^+$ by (23).
3) Step 2: Compute $B'_{k,l}$ and ξ_k by (24).
4) Step 3: Compute $\eta_k(s+1)$ by (26).
5) Step 4: Set $s := s+1$, and go to Step 1 until convergence.

Upon convergence of the above algorithm, we have real valued $B_{k,m}$ and $B'_{k,l}$. In practice, we must have $B_{k,m}$ in integers. But $B'_{k,l}$ can be nonintegers such as fractional numbers

since the coding over different subchannels can be done jointly. In our simulation examples shown later, upon convergence of the above algorithm, we will round up each of $B_{k,m}$ to its nearest integer, i.e., $\bar{B}_{k,m} = \lceil B_{k,m} \rceil$ and then apply (24) one more time to calculate $B'_{k,l}$ with $B_k = \sum_{m=1}^{M} \lceil B_{k,m} \rceil$.

The complexity of the above algorithm can be determined as follows (assuming large K, M and L). It should be noted that $B_{k,m}$ and $B'_{k,l}$ constitute $KM+KL$ variables. Step 1 computes $B_{k,m}$ with no more than M number of iterations to solve (23) for each k and m. Its complexity is $\mathcal{O}(KM^2)$. Step 2 computes $B'_{k,l}$ with two layers of 1-D bisection search which converges exponentially. Its complexity is $\mathcal{O}(KL)$. Step 3 has the complexity $\mathcal{O}(K)$. So, each iteration of the above algorithm has the complexity $\mathcal{O}(KM^2 + KL)$. As a contrast, a general-purpose convex optimization method (using the interior-point method for example) requires an inverse of Hessian matrix of the dimension $(KM + KL) \times (KM + KL)$ at each iteration. The complexity of that is typically $\mathcal{O}((KM + KL)^3)$, assuming that no structure of the problem is exploited. This is why using Matlab to directly solve (8) is much slower than using the algorithm developed in this section.

IV. Linear Energy Model for Progressive Estimation

Suppose that there is a large number of subchannels for each transmission and the gains of the subchannels are the same. Then, the optimal $B'_{k,l}$ should be very small for each k and l, and hence (7) becomes $E_{k,l} = a_{k,l} \frac{\ln 2}{\varphi} B'_{k,l}$ which is a linear energy model.

Under the linear model, if $a_{k,l}$ is not invariant to l, then the optimal $B'_{k,l}$ would be such that only the largest $a_{k,l}$ over all l is allocated with nonzero $B'_{k,l}$ which could violate the assumption that all $B'_{k,l}$ are small subject to $\sum_{l=1}^{L} B'_{k,l} = \sum_{m=1}^{M} B_{k,m} = B_k$. However, if we simply force $B'_{k,l} = \frac{B_k}{L}$ which is invariant to l, then the linear model implies $\sum_{l=1}^{L} E_{k,l} = \frac{1}{L} \left(\sum_{l=1}^{L} a_{k,l} \frac{\ln 2}{\varphi} \right) B_k$.

The linear energy model also applies to many existing communication devices where the energy cost is simply proportional to the number of packets transmitted. Therefore, there is a need to consider this special case. We will write

$$E_k = b_k B_k \tag{27}$$

where E_k is the energy spent by sensor k to transmit B_k bits, and b_k is a constant associated with sensor k. For comparison with the algorithm in Section III, we will choose $b_k = \frac{1}{L} \sum_{l=1}^{L} a_{k,l} \frac{\ln 2}{\varphi}$ in the simulation section. To minimize the sum energy subject to $B'_{k,l} = \frac{B_k}{L}$ and the linear energy model, we need to solve

$$\min_{B_{k,m}} \sum_{k=1}^{K-1} \sum_{m=1}^{M} b_k B_{k,m} \tag{28}$$

subject to

$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} \leq \eta \tag{29}$$

$$B_{k,m} \geq 0, \text{ for all} P \; k \text{ and } m. \tag{30}$$

This problem could be reduced from (8) if $p = 1$, $a_{k,l}$ is invariant to l, and $B'_{k,l} \ll 1$. However, this problem formulation stands on its own without the above conditions on $a_{k,l}$ and $B'_{k,l}$. We also like to mention that for the linear energy model, the basic energy component is E_k. It is hence tempting to minimize the more general cost $\sum_{k=1}^{K-1} E_k^p$ with any $p \geq 1$. For large p, the cost automatically weighs more for sensors that consume more energy. But unfortunately, the resulting optimization problem with respect to the variables $B_{k,m}$ is much more difficult, which will not be addressed in this paper. Next, we present a simple algorithm to solve (28).

The KKT conditions of this problem are given by

$$b_k - 2(\ln 2)\mu\alpha_k W_m^2 2^{-2B_{k,m}} - \lambda_{k,m} = 0 \quad \text{for all } k \text{ and } m \tag{31}$$

$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} \leq \eta \tag{32}$$

$$\mu \left(\sum_{k=1}^{K-1} \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} - \eta \right) = 0 \tag{33}$$

$$\lambda_{k,m} B_{k,m} = 0, \text{ for all } k \text{ and } m \tag{34}$$

where $\mu \geq 0$, $\lambda_{k,m} \geq 0$ and $B_{k,m} \geq 0$. We define the set $\mathcal{KM}^+$ containing the indices of (k,m) where $B_{k,m} > 0$. For $(k,m) \notin \mathcal{KM}^+$, $B_{k,m} = 0$. From (34), we can see $\lambda_{k,m} = 0$ where $(k,m) \in \mathcal{KM}^+$. Plugging this result into (31), we can see $\mu > 0$. We then see from (33) that

$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} = \eta. \tag{35}$$

For $(k,m) \in \mathcal{KM}^+$, we know that $\lambda_{k,m} = 0$ and hence from (31) that

$$B_{k,m} = \left(\frac{1}{2} \log \frac{2(\ln 2)\mu\alpha_k W_m^2}{b_k} \right)^+ \tag{36}$$

where the constant μ is chosen such that (35) holds.

The solution (36) is simpler than the general algorithm described in Section III because in this case there are no iterations between the number of quantization bits $B_{k,m}$ and the number of transmission bits $B'_{k,l}$. Namely, $B_{k,m}$ is determined by (36) while $B'_{k,l}$ is determined by $B'_{k,l} = \sum_{m=1}^{M} \frac{B_{k,m}}{L}$.

V. Consensus Estimation

Consensus estimation has been extensively studied as in [14]–[33]. The purpose of this section and the next is to formulate and analyze an energy efficient version of consensus estimation. This analysis will allow us to compare a minimized energy cost for consensus estimation with that for progressive estimation.

Like the progressive estimation problem shown earlier, the consensus estimation problem we consider is about the estimation of the unknown vector $\boldsymbol{\theta}$ by averaging $\boldsymbol{\theta}_k$. But unlike progressive estimation, consensus estimation requires network-wise iterations within each sampling window. For each iteration, each sensor performs a localized averaging. After iteration time t, the estimate of $\boldsymbol{\theta}$ at the kth sensor is denoted by $\hat{\boldsymbol{\theta}}_k(t)$, and its

quantized version by $\mathbf{m}_k(t)$. Note that $\hat{\boldsymbol{\theta}}_k(0) \equiv \boldsymbol{\theta}_k$. The quantization error is modeled the same way as before. For example, $E_q\{\mathbf{m}_k(t)\} = \hat{\boldsymbol{\theta}}_k(t)$ and $E_q\{(\mathbf{m}_k - \hat{\boldsymbol{\theta}}_k(t))(\mathbf{m}_k - \hat{\boldsymbol{\theta}}_k(t))^T\} \leq \mathbf{C}_{q,k} = diag(\sigma^2_{q,k,1}, \cdots, \sigma^2_{q,k,M})$.

For the communication protocol, we assume that after an iteration of fusion has been done at all sensors, each sensor broadcasts its latest quantized estimate of $\boldsymbol{\theta}$. We assume that sensor j receives the packet successfully from sensor k if and only if the following holds:

$$\sum_{l=1}^{L} \varphi \log_2 \left(1 + \phi\mu \frac{E_k |h_{j,k,l}|^2}{LN_0}\right) \geq \sum_{m=1}^{M} B_m \qquad (37)$$

where $h_{j,k,l}$ is the fading factor of subchannel l from sensor k to sensor j, E_k is the transmission energy used by sensor k. We will also assume that all links are symmetric, i.e., $|h_{j,k,l}| = |h_{k,j,l}|$. The other parameters are the same as those described in the previous sections. In practice, distributed transmission scheduling scheme [34], [35] is required to avoid packet collision. But we neglect the energy cost required for coordinating the broadcast from all sensors. The above principle is similar to the broadcast gossip algorithm as in [27]. We do not consider the strict peer-to-peer multihop network where for each transmitter there is only one receiver. This is because for a given transmission range, consensus estimation in a peer-to-peer network consumes more transmission energy than that in a broadcast network.

Once sensor k has received the packets from all of its neighbors, sensor k updates its estimate of $\boldsymbol{\theta}$ as follows:

$$\hat{\boldsymbol{\theta}}_k(t+1) = W_{k,k}\hat{\boldsymbol{\theta}}_k(t) + \sum_{j \in \mathcal{N}_k} W_{k,j} \boldsymbol{m}_j(t) \qquad (38)$$

where $\mathcal{N}_k$ is the neighbor of sensor k, i.e., the set of the indices of the sensors from which sensor k successfully receives $\boldsymbol{m}_j(t)$, and $W_{k,j}$ is a scalar weight or equivalently the (k,j)th element of a $K \times K$ matrix $\mathbf{W}$. After this computation, sensor k quantizes $\hat{\boldsymbol{\theta}}_k(t+1)$ into $\mathbf{m}_k(t+1)$ and then broadcasts $\mathbf{m}_k(t+1)$ to its neighbors.

In the absence of the quantization noise, we want $\lim_{t\to\infty} \hat{\boldsymbol{\theta}}_i(t) = \frac{1}{K}\sum_{k=1}^{K} \hat{\boldsymbol{\theta}}_k(0)$ for all k, which is called average consensus. As shown in [16], this is equivalent to $\lim_{t\to\infty} \boldsymbol{W}^t = \frac{\mathbf{1}\mathbf{1}^T}{K}$ where $\mathbf{1}$ is a $K \times 1$ vector consisting of all 1 s. This is also equivalent to the following conditions: $\mathbf{1}^T\boldsymbol{W} = \mathbf{1}^T$, $\boldsymbol{W}\mathbf{1} = \mathbf{1}$, and $\rho\left(\boldsymbol{W} - \frac{\mathbf{1}\mathbf{1}^T}{K}\right) < 1$ where $\rho(\boldsymbol{A})$ denotes the spectral radius of the matrix $\boldsymbol{A}$. We will assume that $\boldsymbol{W}$ always meets the above conditions. For more general convergence conditions under random perturbations, see [28].

Let $\boldsymbol{m}_k(t) = \hat{\boldsymbol{\theta}}_k(t) + \boldsymbol{n}_k(t)$, $\hat{\boldsymbol{\theta}}(t) = (\hat{\boldsymbol{\theta}}_1(t)^T, \cdots, \hat{\boldsymbol{\theta}}_k(t)^T)^T$ and $\boldsymbol{n}(t) = (\boldsymbol{n}_1(t)^T, \cdots, \boldsymbol{n}_k(t)^T)^T$. We can then write (38) for all k in a vector form as

$$\hat{\boldsymbol{\theta}}(t+1) = (\boldsymbol{W} \otimes \boldsymbol{I})\hat{\boldsymbol{\theta}}(t) + (\boldsymbol{V} \otimes \boldsymbol{I})\boldsymbol{n}(t) \qquad (39)$$

where $\boldsymbol{I}$ is the $M \times M$ identity matrix, $\otimes$ is the Kronecker product, $\boldsymbol{V} = \boldsymbol{W} - \boldsymbol{W}_D$, $\boldsymbol{W}_D$ is a diagonal matrix with the same diagonal elements as $\boldsymbol{W}$.

With the above conditions, one can verify that the mean of $\hat{\boldsymbol{\theta}}(t)$ is given by $E\{\hat{\boldsymbol{\theta}}(t)\} = \mathbf{1} \otimes \boldsymbol{\theta}$, and the MSE of $\hat{\boldsymbol{\theta}}(t)$ is given by

$$\begin{aligned} \text{MSE}(t) &\doteq Tr\left(E\left\{\left(\hat{\boldsymbol{\theta}}(t) - \mathbf{1}\otimes\boldsymbol{\theta}\right)\left(\hat{\boldsymbol{\theta}}(t) - \mathbf{1}\otimes\boldsymbol{\theta}\right)^T\right\}\right) \\ &\leq Tr\left\{\boldsymbol{D}_c(\boldsymbol{W}^t)^T\boldsymbol{W}^t\right\} \\ &\quad + \left(\sum_{i=0}^{t-1} Tr\left\{(\boldsymbol{W}^i\boldsymbol{V})(\boldsymbol{W}^i\boldsymbol{V})^T\right\}\right) \\ &\quad \times \left(\sum_{m=1}^{M} \frac{4W_m^2}{2^{2B_m}}\right) \doteq \text{MSE}_U(t) \end{aligned} \qquad (40)$$

where $\boldsymbol{D}_c$ is a diagonal matrix whose (k,k)th entry is $Tr\{\boldsymbol{C}_{\theta,k}\}$. Here, we have used $E\{(\hat{\boldsymbol{\theta}}_k(0) - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}}_k(0) - \boldsymbol{\theta})^T\} \leq \boldsymbol{C}_{\theta,k}$, $(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{C}\otimes\boldsymbol{D}) = \boldsymbol{A}\boldsymbol{C}\otimes\boldsymbol{B}\boldsymbol{D}$, $(\boldsymbol{A}\otimes\boldsymbol{B})^T = \boldsymbol{A}^T\otimes\boldsymbol{B}^T$, and $Tr\{\boldsymbol{A}\otimes\boldsymbol{B}\} = Tr\{\boldsymbol{A}\}Tr\{\boldsymbol{B}\}$. $\text{MSE}_U(t)$ is an upper bound on $\text{MSE}(t)$.

It is useful to note a couple of other fusion algorithms. In [29], the following algorithm is considered:

$$\hat{\boldsymbol{\theta}}_k(t+1) = W_{k,k}\boldsymbol{m}_k(t) + \sum_{j\in\mathcal{N}_k} W_{k,j}\boldsymbol{m}_j(t). \qquad (41)$$

In [31], the authors proposed

$$\hat{\boldsymbol{\theta}}_k(t+1) = \hat{\boldsymbol{\theta}}_k(t) + \sum_{j\in\mathcal{N}_k} W_{k,j}\left(\boldsymbol{m}_j(t) - \boldsymbol{m}_k(t)\right). \qquad (42)$$

Both (41) and (42) are small variations of (38). In fact, if $\boldsymbol{m}_k(t)$ in (41) and (42) is replaced by its unquantized version $\hat{\boldsymbol{\theta}}_k(t)$, both (41) and (42) are identical to (38). Furthermore, due to the "earlier" quantization used in (41) and (42), $\text{MSE}(t)$ of (41) and (42) is larger than that of (38). Since (41) and (42) are convergent as shown in [29] and [31], so is (38).

VI. Energy Planning for Consensus Estimation

In this section, we further minimize the energy cost of consensus estimation by optimizing a number of parameters as follows:

$$\min_{t, E_0, B_m, \boldsymbol{W}} : tKE_0 \qquad (43)$$

subject to :

$$\text{MSE}_U(t, \boldsymbol{W}, E_0, B_m)/K \leq \text{MSE}_0 \qquad (44)$$

$$\mathbf{1}^T\boldsymbol{W} = \mathbf{1}^T \qquad (45)$$

$$\boldsymbol{W}\mathbf{1} = \mathbf{1} \qquad (46)$$

$$\rho\left(\boldsymbol{W} - \frac{\mathbf{1}\mathbf{1}^T}{K}\right) < 1 \qquad (47)$$

$$t \geq 0 \qquad (48)$$

where MSE_0 is the target per-sensor average MSE.

This problem in general is a very difficult one. To simplify the problem, we choose $B_m = B$ for all m and select $\boldsymbol{W}$ as shown in [16], i.e., $\boldsymbol{W} = \boldsymbol{I} - \gamma\boldsymbol{L}$ where $\boldsymbol{L}$ is the Laplacian matrix of the network graph

$$(\boldsymbol{L})_{j,k} = \begin{cases} -1, & j \in \mathcal{N}_k \\ |\mathcal{N}_k|, & k = j \\ 0, & otherwise. \end{cases} \qquad (49)$$

This $\boldsymbol{W}$ satisfies (45) and (46). Furthermore, the condition (47) is equivalent to $0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})}$ as shown in [16]. Here $\lambda_1(\boldsymbol{L})$ is the largest eigenvalue of matrix $\boldsymbol{L}$.

With the above simplifications, the previous problem becomes

$$\min_{t,E_0,B,\gamma} : tKE_0 \tag{50}$$
$$\text{s.t.} \quad \text{MSE}_U(t,\gamma,E_0,B)/K \le \text{MSE}_0 \tag{51}$$
$$0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})} \tag{52}$$
$$t \ge 0 \tag{53}$$

Note that both $\text{MSE}_U(t,E_0,B,\gamma)$ and $\boldsymbol{L}$ depend on E_0 and B.

We propose a two-loop search algorithm to this problem. In the inner loop, we fix E_0 and B and search for the optimal pair of γ and t, which optimizes the local fusion algorithm. In the outer loop, we search for the optimal pair of E_0 and B, which optimizes the network connectivity. For the outer loop, we use the brute force search. For the inner loop, the following algorithm will be used.

A. Optimal Selection of γ and t

With a given pair of E_0 and B, $\mathcal{N}_k$ for all k are determined, and hence so is $\boldsymbol{L}$. The problem of the inner loop search can be formulated as

$$\min_{\gamma,t} : t \tag{54}$$
$$\text{s.t.} \quad \text{MSE}_U(t,\gamma)/K \le \text{MSE}_0 \tag{55}$$
$$0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})} \tag{56}$$
$$t \ge 0. \tag{57}$$

We can rewrite (40) as

$$\text{MSE}_U(t,\gamma) = Tr\left\{\boldsymbol{D}_c\boldsymbol{U}\boldsymbol{\Lambda}^{2t}\boldsymbol{U}^T\right\} + \gamma^2 Tr\left\{\boldsymbol{U}\left(\sum_{i=0}^{t-1}\boldsymbol{\Lambda}^{2i}\right)\boldsymbol{U}^T(\boldsymbol{L}-\boldsymbol{L}_D)^2\right\}\left(\sum_{m=1}^{M}\frac{4W_m^2}{2^{2B_m}}\right) \tag{58}$$

where we have used the eigenvalue decomposition $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, $(\boldsymbol{\Lambda})_{l,l} = \lambda_l(\boldsymbol{W})$, and $\boldsymbol{L}_D$ is the diagonal part of $\boldsymbol{L}$. Also recall that $\lambda_1(\boldsymbol{W}) = 1$ and $|\lambda_l(\boldsymbol{W})| < 1$ for $l \ge 2$. We can then write

$$\text{MSE}_U(t,\gamma) = Tr\left\{\boldsymbol{D}_c\boldsymbol{U}\boldsymbol{\Lambda}^{2t}\boldsymbol{U}^T\right\} + \gamma^2 Tr\left\{\boldsymbol{U}\bar{\boldsymbol{D}}\boldsymbol{U}^T(\boldsymbol{L}-\boldsymbol{L}_D)^2\right\}\left(\sum_{m=1}^{M}\frac{4W_m^2}{2^{2B_m}}\right) \tag{59}$$

where $\bar{\boldsymbol{D}}$ is diagonal, $(\bar{\boldsymbol{D}})_{1,1} = t$, and $(\bar{\boldsymbol{D}})_{l,l} = \frac{1-\lambda_l^{2t}(\boldsymbol{W})}{1-\lambda_l^2(\boldsymbol{W})}$ for $l \ge 2$.

With the above expressions, $\text{MSE}_U(t,\gamma)$ can be treated as a continuous function of t to simplify the problem. It is easy to check that $\text{MSE}_U(t)$ has two components. The first is $\mathcal{O}(t) = Tr\{\boldsymbol{D}_c\boldsymbol{U}\boldsymbol{\Lambda}^{2t}\boldsymbol{U}^T\}$ which is a decreasing function of t, and the second is $\mathcal{Q}(t) = \gamma^2 Tr\{\boldsymbol{U}\bar{\boldsymbol{D}}\boldsymbol{U}^T(\boldsymbol{L}-\boldsymbol{L}_D)^2\}(\sum_{m=1}^{M}\frac{4W_m^2}{2^{2B_m}})$ which is an increasing function of t. The behavior of $\text{MSE}_U(t,\gamma)$ with respect to γ is not as clear. But knowing $\boldsymbol{W} = \boldsymbol{I} - \gamma\boldsymbol{L}$, we know that $\boldsymbol{\Lambda}$ is a linear function of γ and hence according to (59), $\text{MSE}_U(t,\gamma)$ is a polynomial function of γ of degree $2t$.

The Lagrangian function of this problem is

$$\mathcal{L} = t + \mu(\text{MSE}_U(t,\gamma) - \text{MSE}_0K) + \nu_1(-\gamma) + \nu_2\left(\gamma - \frac{2}{\lambda_1(\boldsymbol{L})}\right) + \nu_3(-t) \tag{60}$$

and the KKT conditions are

$$1 + \mu\frac{\partial \text{MSE}_U(t,\gamma)}{\partial t} - \nu_3 = 0 \tag{61}$$
$$\mu\frac{\partial \text{MSE}_U(t,\gamma)}{\partial \gamma} - \nu_1 + \nu_2 = 0, \tag{62}$$
$$\text{MSE}_U(t,\gamma) - \text{MSE}_0K \le 0 \tag{63}$$
$$0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})} \tag{64}$$
$$\mu \ge 0, \nu_1 \ge 0, \nu_2 \ge 0, \nu_3 \ge 0 \tag{65}$$
$$\mu(\text{MSE}_U(t,\gamma) - \text{MSE}_0K) = 0 \tag{66}$$
$$\nu_1(-\gamma) = 0,\ \nu_2\left(\gamma - \frac{2}{\lambda_1(\boldsymbol{L})}\right) = 0,\ \nu_3(-t) = 0. \tag{67}$$

This system seems complex. But it can be simplified as follows. If $\text{MSE}(0) = Tr\{\boldsymbol{D}_c\} \le \text{MSE}_0K$, then $t = 0$ is the solution to the original optimization, which is trivial. Otherwise for $t > 0$, according to (67), we have $\nu_3 = 0$. From (61), we know $\mu \ne 0$, which leads to $\text{MSE}_U(t,\gamma) - \text{MSE}_0K = 0$ according to (63). From the constraint $0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})}$, we know $\nu_1 = 0$ and $\nu_2 = 0$ as well. Finally, the KKT system is simplified to

$$\frac{\partial \text{MSE}_U(t,\gamma)}{\partial t} + \frac{1}{\mu} = 0 \tag{68}$$
$$\frac{\partial \text{MSE}_U(t,\gamma)}{\partial \gamma} = 0 \tag{69}$$
$$\text{MSE}_U(t,\gamma) - \text{MSE}_0K = 0 \tag{70}$$
$$0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})}, \mu > 0, t > 0. \tag{71}$$

Although there are three unknowns t, γ and μ in (68)–(71), μ can be obtained readily by plugging t and γ into (68). Now we only need to solve (69) and (70) for t and γ with constraint in (71).

We can solve the nonlinear system (69) and (70) by minimizing the following cost function with logarithmic barriers [36]:

$$J(t,\gamma) = \left(\frac{\partial \text{MSE}_U(t,\gamma)}{\partial \gamma}\right)^2 + (\text{MSE}_U(t,\gamma) - \text{MSE}_0K)^2 - \frac{1}{\tau}\ln t - \frac{1}{\tau}\ln\gamma - \frac{1}{\tau}\ln\left(\frac{2}{\lambda_1(\boldsymbol{L})} - \gamma\right) \tag{72}$$

Using gradient descent and Armijo backtracking linear search, we can minimize $J(t,\gamma)$ for each choice of τ. Until convergence, the constant τ is increased after each gradient search for t and γ.

After convergence, if $\text{MSE}(\lfloor t \rfloor) \le \text{MSE}_0$, the solution is $t^\star = \lfloor t \rfloor$; else if $\text{MSE}(\lceil t \rceil) \le \text{MSE}_0$, the solution is $t^\star = \lceil t \rceil$; otherwise, the solution $t^\star$ does not exist.

B. Other Selections of γ and t

The previous subsection has presented an optimization of γ with consideration of quantization errors. We will refer to that γ as γ_1. Prior literature has established other choices of γ without any consideration of quantization errors. Two common choices are

$$\gamma_2 = \frac{2}{\lambda_1(\boldsymbol{L}) + \lambda_{K-1}(\boldsymbol{L})} \tag{73}$$

$$\gamma_3 = \frac{1}{d_{max}} \tag{74}$$

where $d_{max} = \max_{k=1}^{K} |\mathcal{N}_k|$. The choice γ_2 was established in [16] as optimal in the absence of quantization errors. The choice γ_3 was also used in [16] as a simpler option which does not need any further knowledge of $\boldsymbol{L}$.

With γ fixed to γ_2 or γ_3, the corresponding optimal t can be found by choosing the smallest t to satisfy $\frac{\text{MSE}_U(t,\gamma)}{K} \leq \text{MSE}_0$.

Note that the outer loop algorithm for E_0 and B is not affected by the inner loop algorithm for γ and t.

VII. Simulation

A. Progressive Estimation

The network we consider is shown in Fig. 1 where there are $K = 400$ nodes. The routing tree is generated by following the minimal distance principle [37]. The distance between a sensor and its upper-stream sensor is $D\delta$ where δ is uniformly distributed within the range [0.5, 1.5] and D is a normalizing factor.

For the simulation, we assume that sensor k observes the data vector $\boldsymbol{x}_k = \boldsymbol{G}_k\boldsymbol{\theta} + \boldsymbol{\omega}_k$, where $\boldsymbol{G}_k$ is the observation matrix associated with sensor k, which is assumed to be known to sensor k, and $\boldsymbol{\omega}_k$ is white noise with the identity covariance matrix $\boldsymbol{C}_\omega = \boldsymbol{I}$. With this observation model, the initial estimate of $\boldsymbol{\theta}$ at sensor k is obtained by the best linear unbiased estimate: $\boldsymbol{\theta}_k = (\boldsymbol{G}_k^T\boldsymbol{G}_k)^{-1}\boldsymbol{G}_k^T\boldsymbol{x}_k$ for which we have $\boldsymbol{C}_{\theta,k} = (\boldsymbol{G}_k^T\boldsymbol{G}_k)^{-1}$. For each k, $\boldsymbol{G}_k$ is chosen independently to be a $N \times M$ real matrix with i.i.d. elements from a Gaussian distribution with zero mean and standard deviation equals to 10.

Each entry of $\boldsymbol{\theta}$ is chosen randomly from $[-1, 1]$. The squared channel gain of channel l from sensor k to its downstream sensor is $|h_{k,l}|^2 = d_k^{-\alpha}\rho_{k,l}$, where d_k is the distance from sensor k to its downstream sensor, $\alpha = 4$ and $\rho_{k,l}$ is randomly chosen from an exponential distribution with mean equal to one. The same model is applied to $|h_{k,j,l}|^2 = |h_{j,k,l}|^2$ for the link between sensor k and sensor j. We also choose $N = 20$, $M = 10$, $\mu = 1$, $\phi = 1$, $N_0 = 1$, $W_m = 1$, for $m = 1, 2, \cdots, M$.

The results of the energy planning algorithm depend on the choice of p in (8). However, for any p used for computing $B'_{k,l}$, the sum energy is determined by

$$J_1 = \sum_{k=1}^{K-1}\sum_{l=1}^{L} a_{k,l}\left(2^{B'_{k,l}} - 1\right). \tag{75}$$

The sum of transmission energy to be illustrated is given by $\frac{J_1}{D^\alpha}$.

For convenience of reference, we will refer to the algorithm developed in Section III as the "generalized" algorithm, the algorithm under linear energy model in Section IV as the "linear" algorithm, and the algorithm shown in [12] as the "previous" algorithm. We also have a "uniform" algorithm for which the same number of quantization bits is assigned to each element of the estimate at each sensor (i.e., $B_{k,m}$ is independent of k and m), the same number of transmission bits is assigned to each subchannel (i.e., $B'_{k,l}$ is independent of k and l), and however $\sum_{m=1}^{M} B_{k,m} = \sum_{l=1}^{L} B'_{k,l}$ and the MSE constraint (4) hold.

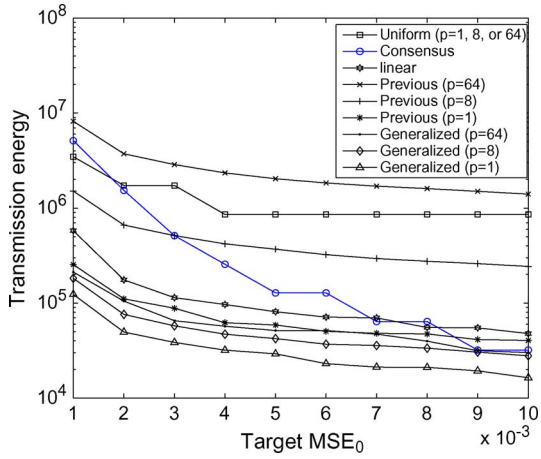


Fig. 2. The sum energy required by different energy planning algorithms for progressive estimation. Also shown is the minimized sum energy required by consensus estimation under (38).

We like to note here that for $K = 40$, $N = 4$, $M = 2$, $L = 2$, $p = 1$, it took the generalized algorithm 2.3125 s (on a computer with Pentium (R) 4 CPU 3.00 GHz, and 1G memory) to find the solution of (8). In contrast, it took the Matlab nonlinear constrained optimization routine *fmincon* 219.56 s to find the same solution.

1) Comparison of Sum Energy: Fig. 2 compares several curves of the sum energy versus the target MSE. Each of these curves (except one) is determined by one of the following energy planning algorithms for progressive estimation: the generalized algorithm with $p = 1$, $p = 8$, and $p = 64$, the previous algorithm with $p = 1$, $p = 8$, and $p = 64$, the linear algorithm, and the uniform algorithm.

Also shown in this figure is a curve of the sum energy cost for consensus estimation. The network we consider for consensus estimation is the same as in Fig. 1 except that there is no routing tree. All other simulation parameters remain the same as those used previously. Also note that the target MSE for progressive estimation is achieved at the fusion center, but the target MSE for consensus estimation is achieved "on average" at each sensor.

We see that with $p = 1$, progressive estimation with the generalized algorithm consumes much less sum energy than consensus estimation especially when the target MSE is small. With increased p, the sum energy determined by the generalized algorithm increases as expected. The same is true for the previous algorithm. However, the sum energy by the generalized algorithm is always smaller than that by the previous algorithm for each given p. This is because the generalized algorithm uses the exact energy model (as opposed to its upper bound) and also exploits the variation of the subchannel gains.

We also see that the linear algorithm requires more sum energy than the previous algorithm with $p = 1$, which is expected.

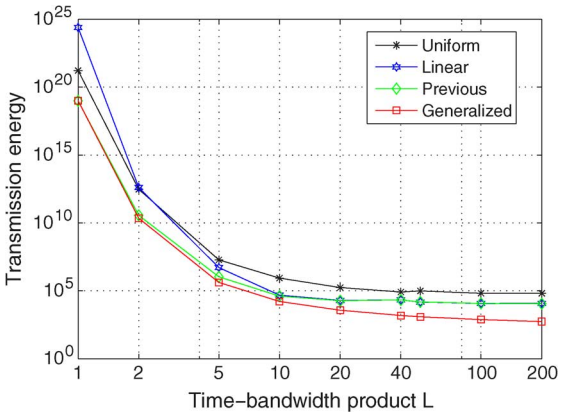


Fig. 3. The sum energy required by four different energy planning algorithms for progressive estimation versus the time-bandwidth product L, where $\text{MSE}_0 = 0.01$. For both the previous and generalized algorithms, $p = 1$ was used.

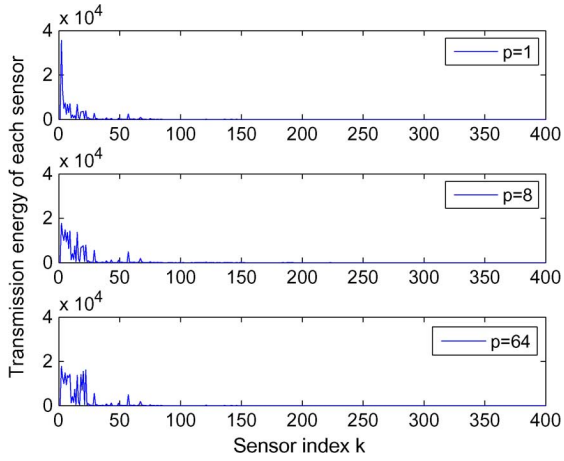


Fig. 4. Amount of transmission energy consumed by an individual sensor versus the sensor index for the generalized algorithm for progressive estimation, with $p = 1$, 8, and 64, where $\text{MSE}_0 = 0.001$. The sensor index is sorted increasingly as the distance between the sensor and the fusion center increases. For progressive estimation, sensor zero in this and the following figures is the fusion center.

Note that although the energy model used for developing the linear algorithm is linear, i.e., (27), the actual amount of energy required by all algorithms (as shown in all figures) is computed by using the original energy model (7) where for the linear algorithm we use $B'_{k,l} = \frac{B_k}{L}$. But when the number of subchannels L becomes large, the linear algorithm and the previous algorithm with $p = 1$ should require the same sum energy, which will be illustrated in the next figure.

The uniform algorithm is clearly a bad choice for progressive estimation in terms of the sum energy cost.

This figure also illustrates that progressive estimation consumes less sum energy than consensus estimation only when a proper energy planning algorithm is applied for progressive estimation. With the previous energy planning algorithm, progressive estimation can even consume more energy than consensus estimation when the target MSE is large.

2) Effect of the Number of Subchannels: Fig. 3 illustrates the sum energy versus the number L of subchannels, where the target MSE is 0.01. For all algorithms, the sum energy cost decreases as L increases, and becomes less sensitive to L when L is large. An explanation of this is available in [12]. As expected, the sum energy required by the linear algorithm becomes the same as that by the previous algorithm when L is large. In this figure, the curves for the previous algorithm and the generalized algorithm appear overlapping in the region of small L because of the large scale. In fact, the generalized algorithm is always better than the previous algorithm.

3) Effect of p on Energy Distribution: Fig. 4 shows the effect of p used in the generalized algorithm for progressive estimation on the peak energy consumed by individual sensors. We see that the peak energy is reduced when p is increased. However, as shown in Fig. 5, the peak energy required by progressive estimation (which always occurs near the fusion center) is generally larger than that by consensus estimation. Fig. 6 shows the average number of quantization bits $\frac{B_k}{M}$ allocated for each element at individual sensors. We see that the distribution of $\frac{B_k}{M}$ is almost invariant to the choice of p, and the sensors near the fusion center always uses larger $\frac{B_k}{M}$. Different from this property, the distribution of the quantization bits generated by the previous algorithm shown in [12] depends significantly on p, and becomes more uniform when p becomes large.

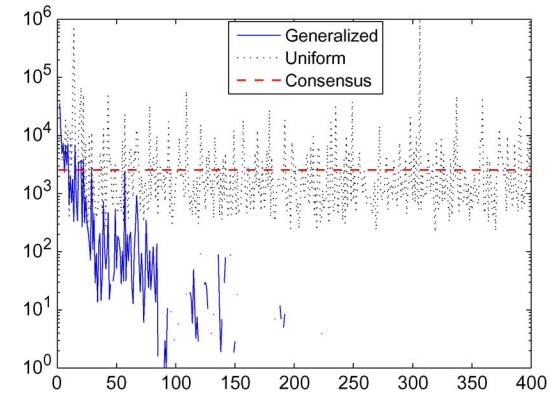


Fig. 5. Amount of transmission energy (in log scale) consumed by an individual sensor versus the sensor index, where $\text{MSE}_0 = 0.001$. For the generalized algorithm, the curve for $p = 1$ is shown here, and the curves for other values of p would differ slightly due to the log scale.

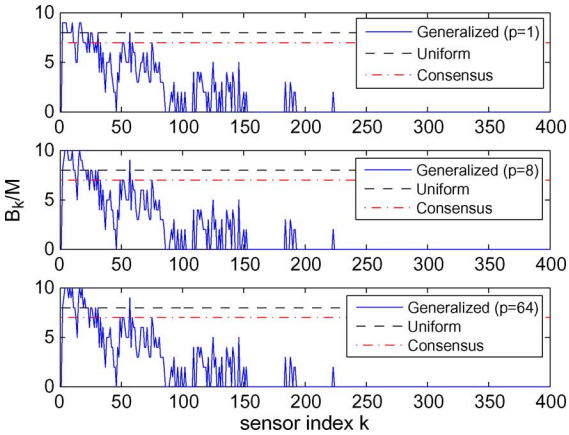


Fig. 6. Averaged number of quantization bits per element allocated by different energy planning algorithms versus the sensor index, where $\text{MSE}_0 = 0.001$.

TABLE I
TOTAL TRANSMISSION ENERGY CONSUMED BY THE NETWORK UNDER DIFFERENT CHOICES OF E_0 AND B. THE MSE TARGET IS SET $\text{MSE}_0 = 0.001$

E_{total}	$B=6$	$B=7$	$B=8$	$B=9$	$B=10$	$B=11$	$B=12$
$E_0 = 640$	∞	∞	∞	∞	∞	∞	∞
$E_0 = 1.28\times10^3$	∞	7.68×10^6	2.25×10^7	∞	∞	∞	∞
$E_0 = 2.56\times10^3$	∞	5.12×10^6	6.14×10^6	1.54×10^7	∞	∞	∞
$E_0 = 5.12\times10^3$	∞	6.14×10^6	1.02×10^7	2.66×10^7	2.87×10^7	∞	∞
$E_0 = 1.024\times10^4$	∞	8.19×10^6	2.05×10^7	4.51×10^7	5.32×10^7	5.32×10^7	∞
$E_0 = 2.048\times10^4$	∞	1.02×10^7	4.1×10^7	8.19×10^7	9.83×10^7	1.06×10^8	1.06×10^8

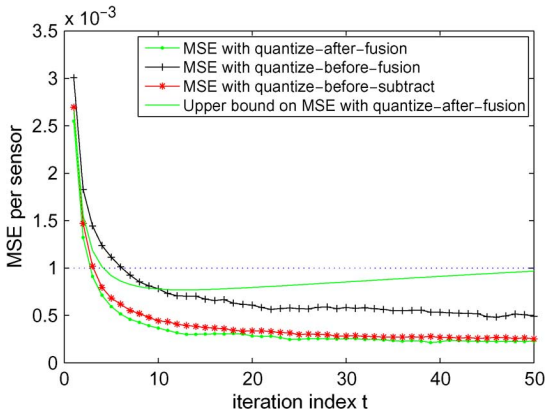


Fig. 7. Gap between the MSE upper bound $\text{MSE}_U(t)$ and the actual MSE $\text{MSE}(t)$ over iteration index t. $E_0 = 2560$, $B = 7$, $\gamma = 0.0909$, and $\text{MSE}_0 = 0.001$. Here, "quantize-after-fusion" follows (38), "quantization-before-fusion" follows (41), and "quantization-before-subtract" follows (42).

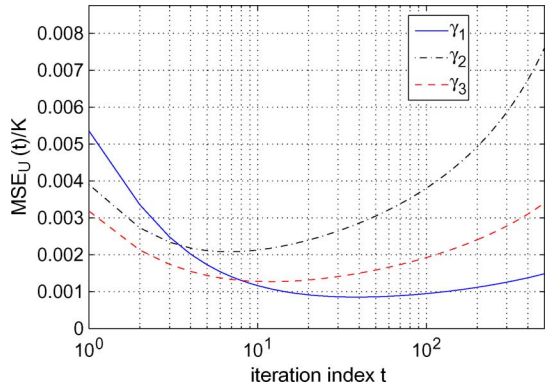


Fig. 8. $\frac{\text{MSE}_{U(t,\gamma)}}{K}$ of consensus estimation under (38) versus the iteration index t, with different γ values: $\gamma_1 = 0.0771$ (optimized with quantization errors), $\gamma_2 = 0.2163$ (optimized without quantization errors) and $\gamma_3 = 0.1429$ (using maximum degree of connectivity), where $E_0 = 1280$, $B = 7$, $\text{MSE}_0 = 0.001$.

B. Consensus Estimation

Here we use the same channel model and the same network as previously except that there is no routing tree.

1) Gap between MSE and Its Upper Bound: As discussed in Section VI, the MSE upper bound $\text{MSE}_U(t)$ (59) is used for develop the energy planning algorithm of the consensus estimation. This upper bound becomes loose when t is large due to the quantization error introduced at each iteration. Therefore, it is important to exam the tightness of the bound in the region of interest, i.e., the region where t is small before the upper bound actually diverges. With $E_0 = 2560$ and $B = 7$, Fig. 7 shows the actual averaged MSE per sensor $\frac{\text{MSE}(t)}{K}$ for the three different fusion algorithms (38), (41), and (42), and the MSE upper bound $\frac{\text{MSE}_U(t)}{K}$ for (38). We see that the MSE of all three algorithms converge as t increases. This is expected given the study of [29, eq. (41)]. One should notice that we only use the decreasing segment of the curve $\text{MSE}_U(t)$ in the energy planning. If a selection of E_0 and B cannot allow $\frac{\text{MSE}_U(t)}{K} \le \text{MSE}_0$ to be achieved before $\text{MSE}_U(t)$ starts to increase, it will cause $t = \infty$ and $E_{\text{total}} = \infty$. We see that the gap between $\text{MSE}(t)$ of (38) and its $\text{MSE}_U(t)$ is not very large in this region.

2) Effect of γ Selection: Fig. 8 illustrates the effects of γ on $\text{MSE}_U(t,\gamma)$ versus t, where $\gamma_1 = 0.0771$, $\gamma_2 = 0.2163$, and $\gamma_3 = 0.1429$. Here, we assumed $E_0 = 1280$ and $B = 7$ to compute $\boldsymbol{L}$. Recall that $\gamma_3 = \frac{1}{d_{max}}$ where d_{max} represents a small amount of information from $\boldsymbol{L}$, $\gamma_2 = \frac{2}{\lambda_1(\boldsymbol{L})+\lambda_{K-1}(\boldsymbol{L})}$ which depends much more strongly on $\boldsymbol{L}$, and γ_1 is optimized to minimize t to meet a target MSE where we used $\text{MSE}_0 = 0.001$. We have found that $t = 15$ with γ_1. However, with γ_2 and γ_3, the target MSE $\text{MSE}_0 = 0.001$ is not achievable with any t.

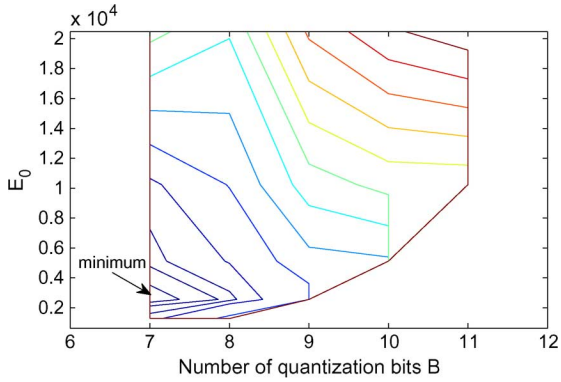


Fig. 9. The contour of the sum energy required by consensus estimation under (38) in terms of E_0 and B. The target MSE is set at $\text{MSE}_0 = 0.001$.

3) Effect of E_0 and B: The optimization of E_0 and B corresponds to the outer-loop of the energy planning algorithm for consensus estimation. For each given pair of E_0 and B, the total energy is minimized by the inner-loop of the algorithm. Table I and Fig. 9 illustrate the total energy cost as function of E_0 and B. If E_0 becomes too small, the network loses connectivity between nodes. If B becomes too small, the quantization errors dominate. In either case, the target MSE may become unachievable even after infinite number of iterations, which corresponds to the case $E_{\text{total}} = \infty$.

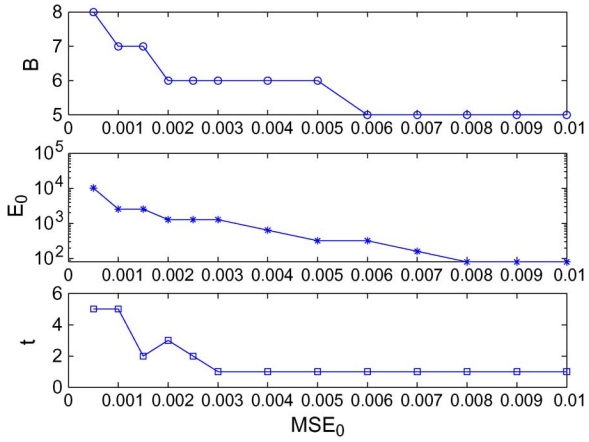


Fig. 10. The optimal B, E_0 and t required by consensus estimation versus MSE_0.

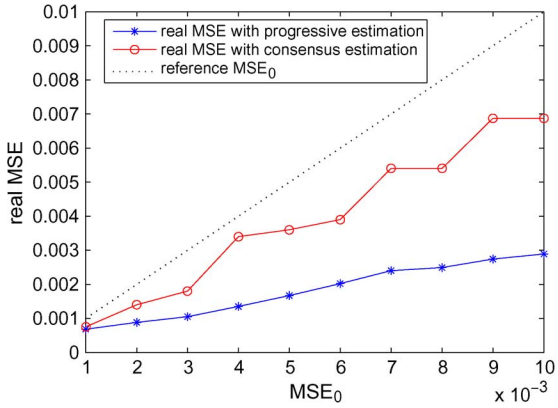


Fig. 11. Comparison of the real MSEs for progressive estimation (with $p = 1$ for energy planning) and consensus estimation.

Corresponding to the curve for consensus estimation in Fig. 2, Fig. 10 shows the optimized B, E_0 and t versus the target MSE.

C. Comparison of Actual MSE

As shown in Fig. 2, under the same target MSE, progressive estimation (with $p = 1$ for energy planning) consumes much less energy than consensus estimation. Since the target MSE used for energy planning is only an upper bound on the actual MSE, it is useful to compare the actual MSE between the two estimation schemes. Fig. 11 shows the actual or real MSE of the two schemes versus a common target MSE. The values of the actual MSE were computed based on 50 independent realizations. For each realization, each entry of $\boldsymbol{\theta}$ was chosen randomly between -1 and 1, and each quantized estimate of it (at various stages) was obtained via the probabilistic quantization with the number of bits determined by the corresponding energy planning algorithm. The quantization errors over 50 realizations were used to compute the actual MSE shown in Fig. 11. We see that the actual MSE of progressive estimation is smaller than that of consensus estimation. This is because the upper bound used for progressive estimation is conservative. This conservativeness only helps to support the conclusion that consensus estimation consumes more energy than progressive estimation subject to the same MSE (for both the target MSE and actual MSE).

VIII. Conclusion

In this paper, we have made contributions in two aspects. One is a generalized energy planning algorithm for progressive estimation in multihop sensor network with routing tree. Using the exact energy model and taking advantage of diverse channel gains of multiple subchannels, this algorithm is more general and yields more energy saving than the previous algorithm developed in [12]. The other aspect is an in-depth study of the energy cost for consensus estimation in broadcast multihop sensor networks. Insights have been developed into the energy cost in terms of transmission energy, quantization bits allocation and fusion rule. Furthermore, this study is the first that has provided a quantitative comparison of the energy cost between progressive estimation and consensus estimation. In applications, the flexibility of consensus estimation on one hand and the total energy efficiency of progressive estimation on the other hand may both affect the decision by the network designers.

Appendix I
Proof of Convexity of (8)

Here, we assume that $B_{k,m}$ and $B'_{k,l}$ are nonnegative real numbers, and $p \geq 1$. The objective function J_p in (8) is a sum of $E^p_{k,l}$. We know that $E^p_{k,l}$ does not depend on $B'_{k',l'}$ unless $k' = k$ and $l' = l$. We also know

$$\frac{\partial^2 E^p_{k,l}}{\partial {B'_{k,l}}^2} = \begin{cases} a_{k,l} 2^{B'_{k,l}/\varphi} \left(\frac{\ln 2}{\varphi}\right)^2 \geq 0, & p = 1 \\ a^p_{k,l} p \left(2^{B'_{k,l}/\varphi} - 1\right)^{p-2} 2^{B'_{k,l}/\varphi} \\ \quad \times \left(\frac{\ln 2}{\varphi}\right)^2 \left(p 2^{B'_{k,l}/\varphi} - 1\right) \geq 0, & p > 1. \end{cases} \tag{76}$$

Hence, $E^p_{k,l}$ is a convex function of $B'_{k,l}$, and hence J_p is a convex function of all $B'_{k,l}$. Since J_p does not depend on $B_{k,m}$, J_p is also a convex function of all $B'_{k,l}$ and $B_{k,m}$. The above is equivalent to the fact that the Hessian matrix of J_p with respect to all $B_{k,m}$ and $B'_{k,l}$ is diagonal and each of its diagonal elements is nonnegative.

In a similar way, it is easy to verify that the LHS function, denoted by f, of (9) is a sum of convex functions, and hence f is also convex. Then, the set defined by (9) is convex. It is obvious that the set defined by each of (10) and (11) is convex. So, we see that the set defined by all of (9)–(11) is convex. Hence, (8) with the associated constraints is convex.

Appendix II
Proof of Convergence of (23)

We now show that the number of iterations in computing (23) until convergence is always finite. Without loss of generality, we assume $W_1 \leq W_2 \leq \cdots \leq W_M$. Therefore, according to (23), the set $\mathcal{M}^+_k$ contains a set of contiguous integers from a number no larger than M to the number M, i.e., if $m \in \mathcal{M}^+_k$ then $m+1, m+2, \cdots, M$ are all in the set $\mathcal{M}^+_k$. The next proposition shows that once an index is excluded from $\mathcal{M}^+_k$, it will never come back into $\mathcal{M}^+_k$ in later iterations. Since M is finite, we have proved that the number of iterations is finite, i.e., no larger than M.

$$B_{k,m_j-1}^{(j+1)} = \left(\frac{1}{2}\log_2 \frac{\alpha_k W_{m_j-1}^2 (M-m_j+1)}{\eta_k - \sum_{j=1}^{m_j-1} \alpha_k W_j^2}\right)^+$$
$$= \left(\frac{1}{2}\log_2 \frac{\alpha_k W_{m_j-1}^2 (M-m_{j-1}+1) - \alpha_k W_{m_j-1}^2 (m_j-m_{j-1})}{\eta_k - \sum_{j=1}^{m_{j-1}-1} \alpha_k W_j^2 - \sum_{j=m_{j-1}}^{m_j-1} \alpha_k W_j^2}\right)^+$$
$$\triangleq \left(\frac{1}{2}\log_2 \frac{A-C}{B-D}\right)^+ \tag{78}$$

Proposition 1: Suppose $W_1 \le W_2 \le \cdots \le W_M$. Denote $\mathcal{M}_{k,j}^+ = \{m_j, \cdots, M\}$ as the set containing the indices m of all positive $B_{k,m}$ after the jth iteration. Then, m_j is nondecreasing with j, and hence $\mathcal{M}_{k,j}^+$ is a nongrowing set with j.

Proof: We prove this lemma using induction. Initially, we know $\mathcal{M}_{k,0}^+ = \{1, 2, \cdots, M\}$, i.e., $m_0 = 1$. It is obvious to see $m_1 \ge 1$. We can now assume $m_j \ge m_{j-1}$ in order to prove $m_{j+1} \ge m_j$.

We know that using $\mathcal{M}_{k,j-1}^+ = \{m_{j-1}, \cdots, M\}$, the iteration j yields the solution of B_{k,m_j-1} as given by

$$B_{k,m_j-1}^{(j)} = \left(\frac{1}{2}\log_2 \frac{\alpha_k W_{m_j-1}^2 (M-m_{j-1}+1)}{\eta_k - \sum_{j=1}^{m_{j-1}-1} \alpha_k W_j^2}\right)^+$$
$$\triangleq \left(\frac{1}{2}\log_2 \frac{A}{B}\right)^+ \tag{77}$$

where $A = \alpha_k W_{m_j-1}^2 (M - m_{j-1} + 1)$ and $B = \eta_k - \sum_{j=1}^{m_{j-1}-1} \alpha_k W_j^2$. Since $B_{k,m_j-1} = 0$, it implies that $A \le B$.

Then, using $\mathcal{M}_{k,j}^+ = \{m_j, \cdots, M\}$, the iteration $j+1$ yields the solution of B_{k,m_j-1} as given by (78), shown at the top of the page, where $C = \alpha_k W_{m_j-1}^2 (m_j - m_{j-1})$ and $D = \sum_{j=m_{j-1}}^{m_j-1} \alpha_k W_j^2$. As $m_j \ge m_{j-1}$ and $W_1 \le W_2 \le \cdots \le W_M$, we know that $C \ge D$. Combining this with $A \le B$, we obtain $B_{k,m_j-1}^{(j+1)} = 0$. Recall that because of $W_1 \le W_2 \le \cdots \le W_M$, (23) implies that $B_{k,m}^{(j+1)} = 0$ for $m \le m_j - 1$. Therefore, $m_{j+1} \ge m_j$. ∎

REFERENCES

[1] Z.-Q. Luo, "Universal decentralized estimation in bandwidth constrained sensor network," *IEEE Trans. Inf. Theory*, vol. 51, pp. 2210–2219, Jun. 2005.

[2] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Power scheduling decentralized estimation in sensor networks," *IEEE Trans. Signal Process.*, vol. 54, pp. 413–422, Feb. 2006.

[3] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, Part I: Gaussian PDF," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[4] I. D. Schizas, G. B. Giannakis, and Z.-Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4284–4299, Aug. 2007.

[5] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. Goldsmith, "Linear coherent decentralized estimation," *IEEE Trans. Signal Process.*, vol. 56, no. 2, Feb. 2008.

[6] J. Fang and H. Li, "Power constrained distributed estimation with cluster-based sensor collaboration," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3822–3832, Jul. 2009.

[7] J. Li and G. Alregib, "Distributed estimation in energy-constrained wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, Oct. 2009.

[8] L. Zhang, X. Zhang, T. Ho, and T. Dikaliotis, "Progressive distributed estimation over noisy channels in wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2009.

[9] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part I—Fundamentals," *IEEE Proc.*, vol. 85, no. 1, pp. 54–63, Jan. 1997.

[10] Y. Hua and Y. Huang, "Progressive estimation and detection," in *Proc. Workshop on Sens. Signal Inf. Process. (SenSIP)*, Sedona, AZ, May 2008.

[11] Y. Huang and Y. Hua, "Multihop progressive decentralized estimation in wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 14, no. 12, pp. 1004–1007, Dec. 2007.

[12] Y. Huang and Y. Hua, "Energy planning for progressive estimation in multihop sensor networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4052–4065, Oct. 2009.

[13] Y. Huang and Y. Hua, "Energy cost for estimation in multihop wireless sensor networks," in *Proc. IEEE ICASSP*, Dallas, TX, Mar. 2010.

[14] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.

[15] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, Jan. 2007.

[16] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Contr. Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[17] D. Scherber and H. C. Papadopoulos, "Distributed computation of averages over *ad hoc* networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 776–787, Apr. 2005.

[18] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, Apr. 2005.

[19] S. Barbarossa and G. Scutari, "Decentralized maximum-likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems," *IEEE Trans. Signal Process.*, vol. 55, no. 7, Jul. 2007.

[20] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[21] I. D. Schizas, G. B. Giannakis, S. D. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy links—Part II: Distributed estimation and smoothing of random signals," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1650–1666, Apr. 2008.

[22] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[23] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

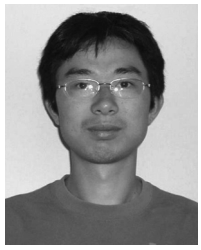
[24] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[25] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[26] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, 2005.

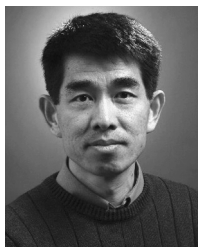
[27] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, Jul. 2009.

[28] T. C. Aysal and K. E. Barner, "Convergence of consensus models with stochatic disturbances," *IEEE Trans. Inf. Theory*, vol. 56, no. 8, pp. 4101–4113, Aug. 2010.
[29] T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Trans. Signal Process.*, vol. 56, no. 10, Oct. 2008.
[30] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, pp. 33–46, 2007.
[31] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus on networks with quantized communication," *Int. J. Robust Nonlin. Contr.*, pp. 1787–1816, 2009.
[32] M. G. Rabbat, R. D. Nowak, and J. A. Bucklew, "Generalized consensus computation in networked system with erasure links," in *Proc. IEEE Workshop on Signal Process. Adv. Wireless Commun.*, 2005.
[33] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.
[34] Y. Yu, Y. Huang, B. Zhao, and Y. Hua, "Further development of synchronous array method for ad hoc wireless networks," *EURASIP J. Adv. Signal Process.—Special Issue on Cross-Layer Design for the Phys., MAC, and Link Layer in Wireless Syst.* vol. 2009, Sep. 2008, 873202, 14 pp..
[35] B. Zhao and Y. Hua, "A distributed medium access control scheme for a large network of wireless routers," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1614–1622, May 2008.
[36] S. Boyd and L. Vandenberghe*, Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[37] F. C. Harris, Jr., "Steiner minimal trees: An introduction, parallel computation, and future work," in *Handbook of Combinatorial Optimization*. Boston, MA: Kluwer Academic, 1998.



Yi Huang received B.E. degree from the University of Science and Technology of China in 2002, the M.Phil. degree from the Chinese University of Hong Kong in 2004, and the Ph.D. degree from University of California, Riverside (UCR), in 2010.

He was the recipient of UCR Dean's Dissertation Fellowship. He is currently with Qualcomm Research Center, San Diego, CA. His research interests include signal processing and protocol design for next generation wireless communication systems.



Yingbo Hua (S'86–M'88–SM'92–F'02) received the B.S. degree in 1982 from Nanjing Institute of Technology (Southeast University), Nanjing, China, and the M.S. and Ph.D. degrees from Syracuse University, Syracuse, NY, in 1983 and 1988, respectively.

He is a Senior Full Professor with the University of California, Riverside, which he joined in 2001. Since 1990, he was a Professor with the University of Melbourne, Australia. He was a visiting professor with Hong Kong University of Science and Technology during 1999–2000. He consulted with Microsoft Research, WA, in 2000. He has published more than 280 articles, with thousands of citations, in the fields of signal processing, sensing and wireless communications. He has edited two books.

Dr. Hua served as steering and/or editorial member for five IEEE and one EURASIP journals, as member on several IEEE SPS technical, organizing, and/or advisory Committees, and other technical, organizing and/or advisory committees for numerous international conferences.