HIERARCHICAL DIFFUSION ALGORITHMS FOR DISTRIBUTED ESTIMATION

Federico S. Cattivelli *Ali H. Sayed*

Department of Electrical Engineering
University of California, Los Angeles

ABSTRACT

We study the problem of distributed estimation, where a set of nodes are required to collectively estimate some parameter of interest from their measurements. Distributed implementations avoid the use of a fusion center and distribute the processing and communication across the entire network. Among distributed solutions, diffusion algorithms have been shown to achieve good performance, increased robustness and are amenable for ad-hoc implementation. In this work we focus on hierarchical diffusion algorithms, where we allow different nodes to have different responsibilities, as opposed to our previous work where every node performed exactly the same type of operations. Our results are general in the sense that they apply to any diffusion algorithm. We illustrate the concept using diffusion LMS, provide performance analysis for hierarchical collaboration and present simulation results showing improved performance over non-hierarchical methods.

1. INTRODUCTION

We study the problem of distributed estimation, where a set of nodes are required to collectively estimate some parameter of interest from their measurements. In a centralized solution to the problem, every node transmits its measurements to a fusion center for processing, requiring large amounts of energy for communication. Distributed implementations avoid the use of a fusion center and distribute the processing across the entire network.

Two types of distributed estimation methods are known as *incremental* and *diffusion*. Diffusion algorithms are more amenable for ad-hoc implementation, more robust to node and link failure, and obtain good performance in terms of estimation accuracy, though in general inferior to incremental or centralized solutions. Several distributed estimation algorithms have been proposed, including incremental LMS and RLS [1], diffusion LMS [2] , diffusion RLS [3] and diffusion Kalman filtering and smoothing [4]. Algorithms based on consensus have been proposed in [5, 6].

In this work we focus on hierarchical diffusion algorithms, where we allow different nodes to have different responsibilities in the processing and communication, as opposed to our previous work where every node performed exactly the same type of operations. Our results are general in the sense that they apply to any diffusion algorithm, though we will illustrate the concept by focusing on diffusion LMS. In [4], we proposed a hierarchical collaboration strategy for diffusion Kalman smoothing. In this work we formalize the approach to any diffusion algorithm, show how the analysis should be modified to account for hierarchical collaboration, and present simulation results showing improved performance over non-hierarchical methods. We also compare with another hierarchical approach we proposed in [7].

This work was supported in part by NSF grants ECS-0601266 and ECS-0725441. Author's emails: {fcattiv, sayed}@ee.ucla.edu.

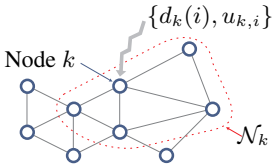

Fig. 1. Node k takes a measurement at time i.

2. PROBLEM FORMULATION

Consider a set of N nodes distributed over some region (see Fig. 1). At every time instant i, every node k takes a scalar measurement $d_k(i)$ of some random process $\boldsymbol{d}_k(i)$ and a $1 \times M$ regression vector, $u_{k,i}$, corresponding to a realization of a random process $\boldsymbol{u}_{k,i}$ which is correlated with $\boldsymbol{d}_k(i)$. The objective is for every node in the network to use the data $\{d_k(i), u_{k,i}\}$ to estimate some deterministic unknown vector w^o. In particular, we seek the linear estimator w^o that minimizes the following global cost function:

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} \mathrm{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{1}$$

where E denotes the expectation operator. An adaptive global LMS solution can be obtained [1] as follows

$$w_i = w_{i-1} + \mu \sum_{k=1}^{N} u_{k,i}^*(d_k(i) - u_{k,i}w_{i-1}) \tag{2}$$

where the operator * denotes complex conjugate transposition. Algorithm (2) is not distributed, since every node in the network needs to have access to global information (namely, the measurements and regressors of every node in the network) in order to compute the new estimate. A distributed solution based on diffusion strategies was proposed in [2] and is known as *diffusion LMS*. We will focus on the adapt-then-combine (ATC) version of diffusion LMS [8], since it exhibits better performance than the CTA version in general.

We say that two nodes are connected if they can communicate directly with each other. Let $\mathcal{N}_k$ denote the neighborhood of node k, i.e., the set of nodes that are connected to node k, including k itself. Consider $N \times N$ matrices A and C with individual non-negative real entries $\{a_{l,k}\}$ and $\{c_{l,k}\}$, respectively, such that

$$c_{l,k} = a_{l,k} = 0 \ \text{ if } l \notin \mathcal{N}_k, \quad C\mathbb{1} = \mathbb{1}, \quad \mathbb{1}^T C = \mathbb{1}^T, \quad \mathbb{1}^T A = \mathbb{1}^T \tag{3}$$

where $\mathbb{1}$ denotes the $N \times 1$ vector with unit entries.

978-1-4244-2710-9/09/$25.00

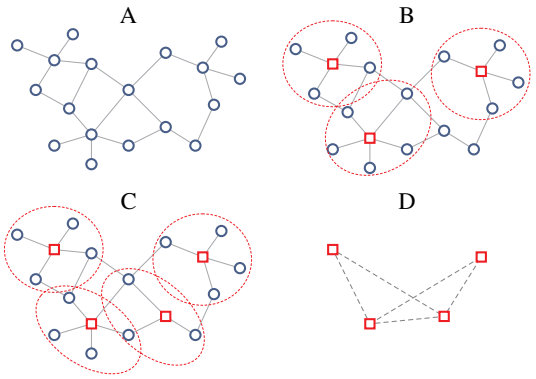

Fig. 2. Network clustering algorithm.

The ATC diffusion LMS algorithm is given by the following iterations. Start with $\{w_{k,-1} = 0\}$ for all k; at every time instant $i \geq 0$ and for each node k, do:

$$\begin{cases} \psi_{k,i} = w_{k,i-1} + \mu_k \sum_{l=1}^{N} c_{l,k} u_{l,i}^{*} (d_l(i) - u_{l,i} w_{k,i-1}) \\ w_{k,i} = \sum_{l=1}^{N} a_{l,k} \psi_{l,i} \end{cases} \quad (4)$$

The first and second equations of (4) are known as the adaptation (or incremental) and combination (or diffusion) updates, respectively. The algorithm requires that at every time instant, every node perform four operations. First, every node communicates its data $\{d_k(i), u_{k,i}\}$ to its neighbors. Second, every node adapts its current estimate using the measurements from its neighborhood to obtain $\psi_{k,i}$. Third, all nodes exchange their pre-estimates $\psi_{k,i}$ with their neighbors, and finally, every node combines (or averages) the pre-estimates to obtain the new estimate $w_{k,i}$. Notice that the first step can be avoided for the common choice $C = I$.

3. HIERARCHICAL DIFFUSION ALGORITHMS

In Algorithm (4), nodes communicate with their neighbors in an isotropic manner, and every node does the same type of processing. This setup is very robust to node and link failure, since the network keeps working whenever a node fails. In this work we show that we may obtain performance improvement if we exploit hierarchy in the network, that is, if we assign different responsibilities to different nodes. These benefits come at the expense of reduced robustness to node failure, increased complexity required to assign and maintain the hierarchies, and the need for multi-hop communications.

3.1. Network clustering

We now group sets of nodes into clusters, and for every cluster we designate one of the nodes as the cluster leader, or *clusterhead*. The degree of a node is defined as the number of neighbors connected to it, including itself. We say that a node is *covered*, if it is already connected to a clusterhead, otherwise it is *uncovered*. We consider the following maximum-degree rules to designate a clusterhead:

- Every node is a neighbor to at least one clusterhead.
- Every clusterhead has a degree larger than or equal to the degree of each of its uncovered neighbors.
- Two clusterheads cannot be neighbors.

An algorithm to cluster the network in such a way can be found in [9], and is summarized as follows:

1. Every node broadcasts its degree to its neighbors
2. Nodes with higher degrees than all of their neighbors automatically become clusterheads. Ties are resolved arbitrarily.
3. A node connected to a clusterhead becomes a covered node.
4. Uncovered nodes with higher degrees than all of their uncovered neighbors become clusterheads. Ties are resolved arbitrarily.

Fig. 2 shows schematically how the clustering algorithm works. Fig. 2A shows the initial network before clustering, where all the nodes are represented as circles. Subsequently, all nodes broadcast their degrees, and the nodes with higher degree than all of their neighbors become clusterheads, and are denoted by squares in Fig. 2B. The nodes connected to these clusterheads become covered, and are not eligible to become clusterheads. Of the remaining uncovered nodes, the ones with highest degree among their uncovered neighborhoods become clusterheads. The final result is shown in Fig. 2C, where a total of four clusters were formed. Note that some nodes may belong to more than one cluster.

Consider now the network formed only by clusterhead nodes. We denote this network by Level-1 network, whereas the network formed by all the nodes is denoted Level-0 network. The above clustering algorithm guarantees that two clusterheads will never be neighbors, that is, clusterheads are never connected in the Level-0 network. Moreover, any clusterhead is at most 3 hops away from the closest clusterhead. Thus, we will say that two clusterheads are connected in the Level-1 network, whenever they are at most 3 hops away from each other. This guarantees that the Level-1 network will be connected. The resulting Level-1 network is shown in Fig. 2D, where the dashed links represent Level-1 connections. Note that communication between clusterheads will require multi-hop transmissions, and therefore some routing protocol will be necessary to transmit the information from one clusterhead to the other.

3.2. Hierarchical diffusion

We now present a general form of hierarchical diffusion algorithms and then specialize the algorithm to diffusion LMS. The pseudo-code for the algorithm is as follows (see Fig. 3):

Hierarchical diffusion algorithm

- Nodes are clustered and clusterheads are designated according to the rules of the Section 3.1.
- Given some diffusion algorithm (e.g. diffusion LMS), for every measurement at time i, do:

1. Every node k that is not a clusterhead transmits its measurements (e.g. $\{d_k(i), u_{k,i}\}$) to its neighboring clusterheads.
2. Every clusterhead k updates its pre-estimate $\psi_{k,i}$ using the incremental update of the diffusion algorithm.
3. Neighboring clusterheads in the Level-1 network exchange their pre-estimates $\psi_{k,i}$ and average them as in the diffusion update of the diffusion algorithm to obtain $w_{k,i}$.
4. Clusterheads transmit the estimates $w_{k,i}$ to their Level-0 neighbors. Nodes connected to more than one clusterhead will keep the estimate coming from the one with higher degree.

We now specialize the hierarchical diffusion algorithm to the case of diffusion LMS. In the first step, all nodes send their measurements $\{d_l(i), u_{l,i}\}$ to their corresponding clusterheads. Then every clusterhead k calculates the pre-estimate through:

$$\psi_{k,i} = w_{k,i-1} + \mu_k \sum_{l \in \mathcal{N}_k} c_{l,k} u_{l,i}^{*} (d_l(i) - u_{l,i} w_{k,i-1}) \quad (5)$$

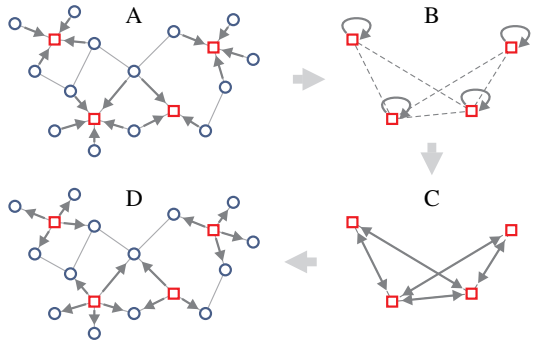

Fig. 3. Hierarchical diffusion algorithm. A) Nodes send their measurements to their clusterheads. B) Clusterheads adapt their pre-estimates using the received measurements. C) Clusterheads exchange their pre-estimates and aggregate the results to obtain the estimates. D) Clusterheads send their estimates to their neighbors.

where $\{c_{l,k}\}$ satisfy (3). Next, clusterheads exchange their pre-estimates $\psi_{k,i}$ with their Level-1 neighbors. Let $\mathcal{N}_k^{(1)}$ denote the Level-1 neighborhood of node k. Then every clusterhead aggregates the pre-estimates from its Level-1 neighbors as follows

$$w_{k,i} = \sum_{l \in \mathcal{N}_k^{(1)}} a_{l,k}^{(1)} \psi_{l,i} \tag{6}$$

where it is assumed that $\{a_{l,k}^{(1)}\}$ are non-negative and satisfy $\sum_{l \in \mathcal{N}_k^{(1)}} a_{l,k}^{(1)} = 1$ for all k. In the final step, every clusterhead k sends its estimate $w_{k,i}$ to its Level-0 neighbors. If a node receives an estimate from two clusterheads, it will prioritize the highest-degree clusterhead as follows. Let l_k denote the highest-degree clusterhead of node k. Then, the estimate of a non-clusterhead node k, at time i is formed by just copying the estimate from its clusterhead l_k, i.e.,

$$w_{k,i} = w_{l_k,i} \text{ if } k \text{ is not a clusterhead} \tag{7}$$

Equations (5)-(7) represent the hierarchical diffusion LMS algorithm. Next, we analyze this algorithm and show that by appropriately selecting the matrices A and C, we can re-utilize the theoretical expressions derived in our previous work [8].

4. PERFORMANCE ANALYSIS

4.1. Mean-square performance of diffusion LMS

In what follows we will consider the estimates $\boldsymbol{w}_{k,i}$ to be random processes, and will analyze their performance in terms of their mean-square behavior, in terms of the excess mean-square error (EMSE) and mean-square deviation (MSD). We start by introducing the following assumptions:

- The measurements are related to the unknown vector as follows:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \tag{8}$$

  where $\boldsymbol{v}_k(i)$ is a zero-mean random variable with variance $\sigma_{v,k}^2$, independent of $\boldsymbol{u}_{k,i}$ for all k and i, and independent of $\boldsymbol{v}_l(j)$ for $l \neq k$ or $i \neq j$.
- All regressors $\boldsymbol{u}_{k,i}$ are zero-mean with covariance $R_{u,k}$, and are both spatially and temporally independent.
- The step-sizes $\{\mu_k\}_{k=1}^N$ are small (see [8] for details).

In [8] we showed that under these assumptions, and for diffusion LMS algorithms of the form (4), the steady-state EMSE and MSD at node k are given by

$$\text{MSD}_k = \text{E}\,||\tilde{\boldsymbol{w}}_\infty||_{q_k}^2 = [\text{vec}(\mathcal{A}^T \mathcal{M} \mathcal{G}^T \mathcal{M} \mathcal{A})]^T (I - F)^{-1} q_k \tag{9}$$

$$\text{EMSE}_k = \text{E}\,||\tilde{\boldsymbol{w}}_\infty||_{r_k}^2 = [\text{vec}(\mathcal{A}^T \mathcal{M} \mathcal{G}^T \mathcal{M} \mathcal{A})]^T (I - F)^{-1} r_k \tag{10}$$

where $\tilde{\boldsymbol{w}}_i = \text{col}\{\tilde{\boldsymbol{w}}_{1,i}, \ldots, \tilde{\boldsymbol{w}}_{N,i}\}$, $\tilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$, and

$$q_k = \text{vec}(\text{diag}(e_k) \otimes I_M) \qquad r_k = \text{vec}(\text{diag}(e_k) \otimes R_{u_k})$$

$$\mathcal{A} = A \otimes I_M \quad \mathcal{C} = C \otimes I_M \quad \mathcal{M} = \text{diag}\{\mu_1 I_M, \ldots, \mu_N I_M\} \tag{11}$$

$$F = \left\{ \mathcal{A} \otimes \mathcal{A} - \mathcal{A} \otimes (\mathcal{D}\mathcal{M}\mathcal{A}) - (\mathcal{D}^T \mathcal{M}\mathcal{A}) \otimes \mathcal{A} \right\}$$

$$\mathcal{D} = \text{diag}\left\{ \sum_{l=1}^N c_{l,1} R_{u,l} \;,\; \ldots \;,\; \sum_{l=1}^N c_{l,N} R_{u,l} \right\}$$

$$\mathcal{G} = \mathcal{C}^T \times \text{diag}\left\{ \sigma_{v,1}^2 R_{u,1} \;,\; \ldots \;,\; \sigma_{v,N}^2 R_{u,N} \right\} \times \mathcal{C}$$

The *network* MSD and EMSE are defined as the average MSD and EMSE, respectively, across all nodes in the network.

4.2. Mean-square performance of hierarchical diffusion LMS

The hierarchical diffusion LMS algorithm is given by Equations (5)-(7). In this section, we will show that these equations can be put into the form of (4) by appropriately choosing the matrices A and C.

We start by noting that (5) has the same form as the adaptive step of (4). The difference is that in the former, the update is done only for nodes k that are clusterheads. Thus given a matrix C satisfying (3), we can form a new matrix $C^{(1)}$ that has the same columns as C for those columns corresponding to clusterheads, and zeros elsewhere. Let γ denote a column vector of size N, with individual entries $\gamma_l = 1$ if l is a clusterhead, and zero otherwise. Then $C^{(1)}$ is given by

$$C^{(1)} = C \cdot \text{diag}(\gamma)$$

Next, we can also see that (6) has the same form as the combination update of (4). However, the sums need to be performed over Level-1 neighbors only. Thus, consider a new matrix $A^{(1)}$ with individual non-negative entries $a_{l,k}^{(1)}$ satisfying

$$a_{l,k}^{(1)} = \begin{cases} 0 & \text{if } \gamma_k = 1 \text{ and } l \notin \mathcal{N}_k^{(1)} \\ a_{l,l_k}^{(1)} & \text{if } \gamma_k = 0 \end{cases} \tag{12}$$

$$\mathbb{1}^T A^{(1)} = \mathbb{1}^T$$

where l_k is the highest-degree clusterhead of node k. The first condition in (12) specifies that every clusterhead k will not communicate with other clusterheads that are not its Level-1 neighbors. Note that we have not defined the values of $a_{l,k}^{(1)}$ for the case when $\gamma_k = 1$ and $l \in \mathcal{N}_k^{(1)}$. These values must be non-negative and add-up to unity for each k. The second condition of (12) corresponds to Equation (7), where nodes that are not clusterheads obtain a copy of the estimate of their highest-degree clusterhead.

The performance analysis of the hierarchical diffusion LMS in terms of EMSE and MSD can be obtained from (9) and (10) by replacing A and C in (11) with $A^{(1)}$ and $C^{(1)}$ respectively.

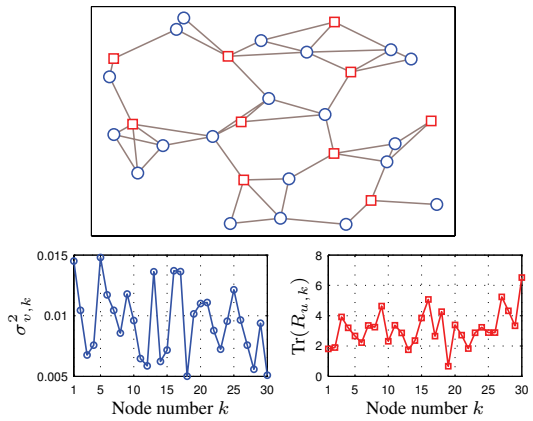

Fig. 4. Network topology (top), noise variances $\sigma^2_{v,k}$ (bottom, left) and trace of regressor covariances $\mathrm{Tr}(R_{u,k})$ (bottom, right).

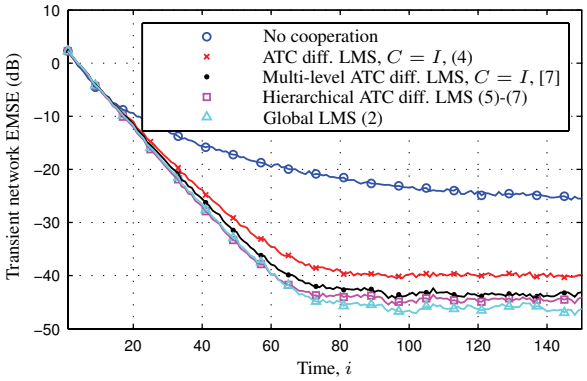

Fig. 5. Transient average EMSE for different algorithms.

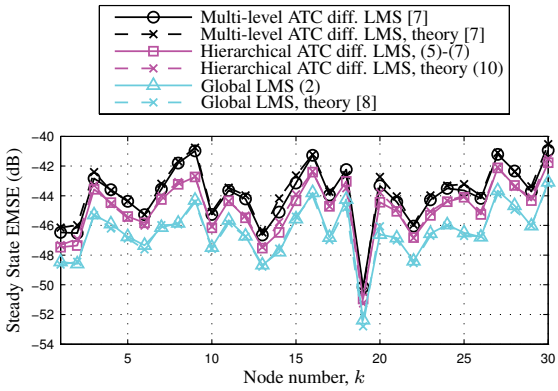

Fig. 6. Steady-state EMSE for different algorithms.

5. SIMULATIONS

We present a simulation example to illustrate the performance of the hierarchical diffusion LMS algorithm. We use a network with $N = 40$ nodes and a topology as shown in Fig. 4, where the cluster-heads are denoted by red squares. The regressors have size $M = 2$, are zero-mean Gaussian, and independent in time and space. Fig. 4 shows $\sigma^2_{v,k}$ and $\mathrm{Tr}(R_{u,k})$ for every node k. The results were averaged over a 100 experiments of duration 150 iterations each.

We compare five algorithms. "No cooperation" corresponds to the case where every node runs an LMS algorithm on its data and does not communicate with its neighbors. Also shown are the ATC diffusion LMS algorithm (4), multi-level diffusion LMS from [7], the proposed hierarchical diffusion LMS given by (5)-(7), and the global solution (2). The step-size $\mu_k = 0.05$ is constant for all nodes and algorithms, except for the second level of multi-level diffusion, where we use $\mu_k = 0.0083$. Relative-degree weights [3] are used for all the combination updates of the diffusion algorithms. Metropolis weights [5] were employed for $C^{(1)}$ in hierarchical diffusion, while we used $C = I$ in (4) and [7]. The reason for this choice is that multi-level diffusion with $C = I$ will have about the same communication requirements as hierarchical diffusion.

Fig. 5 shows the learning curves for the different LMS algorithms in terms of the instantaneous EMSE, $\mathrm{E}\,|\boldsymbol{u}_{k,i}\tilde{\boldsymbol{w}}_{k,i-1}|^2$, averaged across all nodes. We can observe that the hierarchical diffusion LMS algorithm outperforms the non-hierarchical version, and achieves a performance closer to the global solution. Moreover, its performance is similar to that of multi-level diffusion, although it is both faster and has a lower steady-state error in this example. Fig. 6 shows the steady-state EMSE for the same LMS algorithms, showing agreement with the theoretical results (9) and (10).

6. CONCLUSIONS

We proposed a cooperation scheme denoted *hierarchical diffusion* where node hierarchy is introduced. We specialized the method to the case of diffusion LMS, and showed through theory and simulation that the method outperforms non-hierarchical solutions, at the expense of requiring multi-hop communications.

7. REFERENCES

[1] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. on Fund. of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 8, pp. 1504–1510, August 2007.

[2] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proc. IEEE ICASSP*, Honolulu, Hawaii, April 2007, pp. 917–920.

[3] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[4] F. S. Cattivelli and A. H. Sayed, "Diffusion mechanisms for fixed-point distributed Kalman smoothing," in *Proc. EUSIPCO*, Lausanne, Switzerland, August 2008.

[5] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc IPSN*, Nashville, TN, April 2006, pp. 168–176.

[6] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Stability analysis of the consensus-based distributed LMS algorithm," in *Proc. ICASSP*, Las Vegas, NV, March 2008, pp. 3289–3292.

[7] F. S. Cattivelli and A. H. Sayed, "Multilevel diffusion adaptive networks," in *Proc. ICASSP*, Taipei, Taiwan, April 2009.

[8] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," in *Proc. Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, October 2008.

[9] M. Gerla and J. T. C. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, September 1995.