Distributed Algorithm for Minimizing Delay in Multi-Hop Wireless Sensor Networks

M. Farukh Munir*, Arzad A. Kherani** and F. Filali*
*Institut Eurécom, Department of Mobile Communications, Sophia Antipolis, France
**Indian Institute of Technology Delhi, New Delhi India.
Email: {munir,filali}@eurecom.fr
alam@cse.iitd.ernet.in

Abstract

We consider a wireless sensor network with n sensor nodes. The sensed data needs to be transferred in a multi-hop fashion to a common processing center. We consider the standard data sampling/sensing scheme where the sensor nodes have a sampling process independent of the transmission scheme.

In this paper, we study the problem of optimizing the end-to-end delay in a multi-hop single-sink wireless sensor network. We prove that the delay-minimization objective function is strictly convex for the entire network. We then provide a distributed optimization framework to achieve the required objective. The approach is based on distributed convex optimization and deterministic distributed algorithm without feedback control. Only local knowledge is used to update the algorithmic steps. Specifically, we formulate the objective as a network level delay minimization function where the constraints are the reception-capacity and service-rate probabilities. Using the Lagrangian dual composition method, we derive a distributed primal-dual algorithm to minimize the delay in the network. We further develop a stochastic delay control primal-dual algorithm in the presence of noisy conditions. We also present its convergence and rate of convergence. The proposal is extensively evaluated by analysis and simulations.

I. Introduction and Related Work

Wireless Sensor networks (WSNs) is an emerging technology that has a wide range of potential applications including environment monitoring, medical systems, robotic exploration, and smart spaces. WSNs are becoming increasingly important in recent years due to their ability to detect and convey real-time, in-situ information for many civilian and military applications. Such networks consist of large number of distributed sensor nodes that organize themselves into a multihop wireless network. Each node has one or more sensors, embedded processors, and low-power radios, and is normally battery operated. Typically, these nodes coordinate to perform a common task.

Most of the research in the direction of WSNs is focused on energy-conserving routing [3], [4], where the idea of flow-splitting is also utilized to achieve the global objective of maximizing network lifetime. The idea of clustering to optimize network lifetime [6] is shown to provide better results compared to flat architecture approaches like diffusion algorithms [5], but they tend to operate on small scale sensor networks due to the limitations on transmission radius of the cluster heads. In [8], the authors outlined the state-of-the-art routing approaches used in WSNs research community. The main focus of all these research efforts was to maximize the network lifetime due to constrained energy properties of these networks. None of them focused on the delay-optimization direction. It can be a crucial element when the applications require delay-sensitive data in order to operate properly. In which case, the existing approaches might deliver data to the sinks in an energy-efficient way but this will serve no purpose if the data has spanned more-than-required time in the network. Applications where delay is critical include: emergency-response, disaster-management, patient-response, fire-prevention, and many more. In [2], we considered a layered system model for sensor networks and studied the stability properties of the network. In particular, we have shown that the stability conditions proposed in the PRN literature [1] are not correct. Therefore, we provided the correct stability conditions for this model. A cross-layered model is also proposed that is shown to outperform the layered system. This model is meant to be used in applications where a sensor network is used to observe the time variation of a random field over the space on which the network is deployed. A distributed routing algorithm [7] is proposed for both models that tends to achieve a Wardrop-equilibrium.

In this paper, we consider a sensor network with n nodes. The sensors are the sources of delay sensitive traffic that needs to be transferred in a multi-hop fashion to a common processing center. We consider a layered model, where each layer operates independently. We observe the average node delay in the network. The objective then is to minimize the total delay in the network. We provide a distributed optimization framework and deterministic distributed algorithm with no feedback control to achieve this optimization. we avoid a feedback control due to the fact that in multihop wireless networks, the feedback is obtained using error-prone measurement mechanisms. A fundamental open question is that under what conditions these algorithms would converge to the optimal solution? Therefore, our approach is based entirely on the local knowledge of the node. Specifically, we formulate the objective as a network level delay minimization function where the constraints are the reception-capacity and service-rate probabilities. Using the Lagrangian dual composition method, we derive a distributed primal-dual algorithm to minimize the delay in the network. We further develop a stochastic delay control primal-dual algorithm in the presence of noisy conditions. We also present its convergence and rate of convergence.. To the best of our knowledge, this is the first attempt to achieve delay optimizations for a general wireless sensor network.

The organization of this paper is as follows. In Section II, we detail the network model under consideration. Section III discusses the stability issues and objective functions considered in this work. We then propose a distributed optimization framework and deterministic distributed algorithm. Section IV presents the stochastic delay control algorithm under noisy conditions. Its rate of convergence is discussed in Section V. Numerical results from our simulations are presented in Section VI. In Section VII, we briefly conclude the paper and outline the future directions.

II. Network Model

Consider a static wireless sensor network with n sensor nodes. Given is an $n \times n$ neighborhood relation matrix N that indicates the node pairs for which direct communication is possible. We will assume that N is a symmetric matrix, i.e., if node i can transmit to node j, then j can also transmit to node i. For such node pairs, the $(i,j)^{th}$ entry of the matrix N is unity, i.e., $N_{i,j} = 1$ if node i and j can communicate with each other; we will set $N_{i,j} = 0$ if nodes i and j can not communicate. For any node i, we define $N_i = \{j : N_{i,j} = 1\}$, which is the set of neighboring nodes of node i.

Each sensor node is assumed to be *sampling* (or, sensing) its environment at a predefined rate; we let τ_i denote this sampling rate for node i. The units of τ_i will be packets per second, assuming same packet size for all the nodes in the network. In this work we will assume that the readings of each of these sensor nodes are statistically independent of each other so that distributed compression techniques are not employed.

Each sensor node wants to use the sensor network to forward its sampled data to a *common* fusion center (assumed to be a part of the network). Thus, each sensor node acts as a forwarder of data from other sensor nodes in the network. We will assume that the buffering capacity of each node is infinite, so that there is no data loss in the network. We will also assume that a sensor node does not discriminate among its own packets and the packets to be forwarded.

We let R denote the $n \times n$ routing matrix. The $(i,j)^{th}$ element of this matrix is unity if node j is the next-hop node on the route from node i to the fusion center; $R_{i,j} = 0$ otherwise[1]. Clearly, $R_{i,j} = 1$ is possible only if $N_{i,j} = 1$. Similarly, for any node i, we define $F_i = \{j : N_{i,j} = 1 \cap R_{j,i} > 0\}$, which is the set of neighboring nodes of node i that are transmitting data to node i to be forwarded to the fusion center.

We assume that the system operates in discrete time, so that the time is divided into (conceptually) fixed length slots. Since the system operates on CSMA/CA MAC, we will assume that there is no exponential backoff and that the channel access rate of node i (if it has a packet to be transmitted) is $0 \leq \alpha_i \leq 1$. Thus, α_i is the probability that node i, if it has packet to be transmitted, attempts a transmission in any slot.

III. Deterministic Distributed Optimization Framework

We first provide the correct stability condition for the layered system.

A. Rate Balance Equations

Lemma 1: The minimum rate at which a node can serve its transmit queue is

$$\mu_i \triangleq \alpha_i \sum_{j \in N_i} R_{i,j} \left(1 - \alpha_j\right) \Pi_{k \in N_j \setminus \{i\}} \left(1 - \alpha_k\right).$$

Lemma 2: The minimum reception rate of node i is

$$\gamma_i \triangleq \left(1 - \alpha_i\right) \sum_{j \in N_i} R_{j,i} \alpha_j \Pi_{k \in N_i \setminus j} \left(1 - \alpha_k\right).$$

Let the total arrival rate into the transmit buffer of node i be denoted by a_i. If all the transmit queues in the network are *stable*, then the following relation is obtained for a_is

Lemma 3:

$$a_i = \tau_i + \sum_j R_{j,i} \left(a_j \wedge (\tau_j + \gamma_j) \wedge \mu_j\right).$$

Lemma 4: The transmit queue at node i is stable if

$$\sum_{j \in N_i} R_{i,j} \left((\tau_i + \gamma_i) \wedge \mu_i\right) > a_i.$$

Lemma 5: If all the nodes in the network are stable, then

$$a_i = \tau_i + \sum_j R_{j,i} a_j.$$

[1] The formulation of this paper and all the equations are written in a manner that allows for values of $R_{i,j}$ in the interval $(0, 1)$. This would mean a probabilistic flow splitting as in the model of [1]. For simplicity of discussion, however, we will restrict our attention to the $0-1$ value scheme for $R_{i,j}$s.

Lemma 6: The probability that a transmission from node i is successful is

$$s_i = \sum_{j \in N_i} R_{i,j} \left(1 - \pi_j \alpha_j\right) \Pi_{k \in N_j \setminus \{i\}} \left(1 - \pi_k \alpha_k\right).$$

The proofs for the given lemmas can be found in [2].

B. Optimization Problem

We will call a routing matrix feasible if the following constraint is met $\sum_{1<j\leq n} \tau_j = a_1$, where, without loss of generality, we have given an index 1 to the fusion center. This requirement says that all the data generated in the network must end-up at the fusion center. We have the following consideration now: Minimize the total delay in the network

$$\sum_i w_i \frac{1}{1 - \frac{a_i}{\mu_i}} \left(1 - \frac{a_i}{2\mu_i}\right) \tag{1}$$

where we have used the average delay formula for the M/D/1 queue with mean service requirement of unity. Here $w_i > 0$ is a weight given to the node i, for example, the node close to fusion center may be heavily loaded, hence we may want to give more attention to this node. Here $\frac{a_j}{\mu_j}$ is the load on node j. s.t. $\sum_i w_i \left(\mu_i - a_i\right)$. which says, maximize the difference between the service rate and the arrival rate into any node, while in the stable region. It is important to be noted that we first fix the routing in the network, and thus, fixing the arrival rate at each node. We then look at the optimization criteria assuming the network is operating in the stable region. We thus want to maximize the system performance while in the stable region.

We first consider the delay minimization objective function

$$\min \sum_i w_i \frac{1}{1 - \frac{a_i}{\mu_i}} \left(1 - \frac{a_i}{2\mu_i}\right) = \sum_i w_i \left(1 + \frac{1}{2} a_i x_i\right) \tag{2}$$

where $x_i = \frac{1}{\mu_i - a_i}$. Suppose that $\mu_i - a_i = A$. First of all, we will prove that x_i is a convex function. Let $f\left(\mu_i\right)$ be a function of a single variable defined on the interval I, then $f\left(\mu_i\right)$ is convex, if for all $a \in I$, all $b \in I$, and all $t \in [0, 1]$, we have

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \bar{t} f\left(\underline{b}\right) \leq 0$$

where $\bar{t} = 1 - t$. Also, we assume the feasible region of x_i is $[a, b]$. We have

$$\underline{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_i \end{bmatrix}, \underline{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \end{bmatrix} \Rightarrow \underline{c} = t\underline{a} + \bar{t}\underline{b} = \begin{bmatrix} ta_1 + \bar{t}b_1 \\ \vdots \\ ta_i + \bar{t}b_i \end{bmatrix}$$

Then, we need to prove that

$$f\left(t\underline{a} + \bar{t}\underline{b}\right) - tf\left(\underline{a}\right) - \bar{t} f\left(\underline{b}\right) \leq 0$$

or

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \bar{t} f\left(\underline{b}\right) \leq 0$$

n-Node Example: Let us consider an example of n-nodes and use this method to prove convexity. According to the objective function (2), we can rewrite it as

$$f\left(\underline{\mu}\right) = \underline{w}^{\mathrm{T}} \left(\mathrm{I} + \frac{1}{2} \underline{ax}\right)$$

where $\underline{\mu} = [\mu_1 \cdots \mu_i]^{\mathrm{T}}$, $\underline{x} = \left[\frac{1}{\mu_1 - a_1} \cdots \frac{1}{\mu_i - a_i}\right]^{\mathrm{T}}$, $\underline{w} = [w_1 \cdots w_i]^{\mathrm{T}}$ and $\underline{a} = [a_1 \cdots a_i]^{\mathrm{T}}$. For the ease of understanding in the proof, here we use q_i replacing a_i in (1). So

$$f\left(\underline{\mu}\right) = \underline{w}^{\mathrm{T}} \left(\mathrm{I} + \frac{1}{2} \underline{qx}\right) = \underline{w}^{\mathrm{T}} \left[\mathrm{I} + \frac{1}{2} \underline{q} \left(\underline{\mu} - \underline{q}\right)^{-1}\right]$$

where $\underline{x} = \left[\frac{1}{\mu_1 - q_1} \cdots \frac{1}{\mu_i - q_i}\right]^{\mathrm{T}}$. Therefore, we can write $f\left(\underline{a}\right)$ and $f\left(\underline{b}\right)$ as follows

$$f\left(\underline{a}\right) = \underline{w}^{\mathrm{T}} \left[\mathrm{I} + \tfrac{1}{2} \underline{q} \left(\underline{a} - \underline{q}\right)^{-1}\right], \; f\left(\underline{b}\right) = \underline{w}^{\mathrm{T}} \left[\mathrm{I} + \tfrac{1}{2} \underline{q} \left(\underline{b} - \underline{q}\right)^{-1}\right] \tag{3}$$

Similarly, we can write

$$f\left(\underline{c}\right) = \underline{w}^{\mathrm{T}} \left[\mathrm{I} + \tfrac{1}{2} \underline{q} \left(t\underline{a} + \bar{t}\underline{b} - \underline{q}\right)^{-1}\right] = \underline{w}^{\mathrm{T}} \left[\mathrm{I} + \tfrac{1}{2} \underline{q} \left(\underline{b} - \underline{q} + t\left(\underline{a} - \underline{b}\right)\right)^{-1}\right] \tag{4}$$

where $\mathrm{I} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$. And for $\underline{a}$ and $\underline{b}$, we have

$$tf\left(\underline{a}\right) + \overline{t}f\left(\underline{b}\right) = \underline{w}^{\mathrm{T}}\mathrm{I} + \frac{1}{2}\underline{w}^{\mathrm{T}}\underline{q}\left[t\left(\underline{a}-\underline{q}\right)^{-1} + \overline{t}\left(\underline{b}-\underline{q}\right)^{-1}\right] \tag{5}$$

Therefore, we need to prove that

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \overline{t}f\left(\underline{b}\right) \leq 0 \tag{6}$$

By plugging (4) and (5) into (6) and solving, we get

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \overline{t}f\left(\underline{b}\right) = \tfrac{1}{2}\underline{w}^{\mathrm{T}}\underline{q}\left[\left(\underline{b}-\underline{q}+t\left(\underline{a}-\underline{b}\right)\right)^{-1} - \ t\left(\underline{a}-\underline{q}\right)^{-1} - \overline{t}\left(\underline{b}-\underline{q}\right)^{-1}\right] \tag{7}$$

Assume: $A = \underline{a} - \underline{q}$, $B = \underline{b} - \underline{q}$. Then

$$\begin{aligned} &\left(\underline{b}-\underline{q}+t\left(\underline{a}-\underline{b}\right)\right)^{-1} - t\left(\underline{a}-\underline{q}\right)^{-1} - \overline{t}\left(\underline{b}-\underline{q}\right)^{-1} = \tfrac{1}{B+t(A-B)} - \tfrac{t}{A} - \tfrac{\overline{t}}{B} \\ &= \tfrac{1}{tA+\overline{t}B} - \tfrac{\overline{t}A+tB}{AB} = \tfrac{AB-\left(\overline{t}\cdot A+B\cdot t\right)\left(A\cdot t+\overline{t}\cdot B\right)}{AB\left(t\cdot A+\overline{t}\cdot B\right)} = \tfrac{Q_1}{Q_2} \end{aligned} \tag{8}$$

$$\begin{aligned} Q_1 &= AB - \left(\overline{t}\cdot t\left(A^2+B^2\right) + AB\left(\overline{t}^2+t^2\right)\right) = AB - \left[\left(t-t^2\right)\left(A^2-B^2\right) + \left(1-2t+2t^2\right)AB\right] \\ &= -\left(t-t^2\right)\left[A^2+B^2+2AB\right] = -\left(t-t^2\right)\left(A+B\right)^2 \end{aligned}$$

therefore $Q_1 \leq 0$. Also, $Q_2 = AB\left(t\cdot A + \overline{t}\cdot B\right) > 0$. Therefore we have proved that

$$f\left(\underline{c}\right) - tf\left(\underline{a}\right) - \overline{t}f\left(\underline{b}\right) \leq 0$$

The proof is complete and is for any number of nodes $n \in [1, N]$. We have shown that the function $\min \sum_i w_i \frac{1}{1-\frac{a_i}{\mu_i}}(1 - \frac{a_i}{2\mu_i})$ is a strictly convex function in μ_i. Therefore, we now use the Lagrange method to find the optimal value of μ_i. We want to $\min \sum_i w_i\left(1+\frac{1}{2}a_i x_i\right)$, where $x_i = \frac{1}{\mu_i - a_i}$ s.t. $\mu_i > a_i$. Let

$$f\left(\cdot\right) = \min \sum_i w_i\left(1 + \frac{1}{2}a_i x_i\right)$$

Then the Lagrange of $f\left(\cdot\right)$ is

$$L = \min \sum_i w_i\left[1 + \frac{1}{2}a_i\left(\frac{1}{\mu_i - a_i}\right)\right] + \sum_i \lambda_i\left(\mu_i - a_i\right)$$

$$\frac{\delta L}{\delta \mu_i} = \begin{bmatrix} \frac{\delta L}{\delta \mu_1} \\ \vdots \\ \frac{\delta L}{\delta \mu_n} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}w_1 a_1\left(\frac{1}{\mu_1 - a_1}\right)^2 + \lambda_1 \\ \vdots \\ -\frac{1}{2}w_n a_n\left(\frac{1}{\mu_n - a_n}\right)^2 + \lambda_n \end{bmatrix} = 0$$

$$\frac{\delta L}{\delta \mu_i} = -\frac{1}{2}w_i a_i\left(\frac{1}{\mu_i - a_i}\right)^2 + \lambda_i = 0,\ \frac{1}{2}w_i a_i = \lambda_i\left(\mu_i - a_i\right)^2$$

$$\lambda_i\mu_i^2 - 2\lambda_i\mu_i a_i - \frac{1}{2}w_i a_i + \lambda_i a_i^2 = 0,\ \mu_i = \frac{2\lambda_i a_i \pm \sqrt{\left(2\lambda_i a_i\right)^2 - 4\lambda_i\left(\frac{1}{2}w_i a_i + \lambda_i a_i^2\right)}}{2\lambda}$$

$$= a_i \pm \frac{1}{2\lambda_i}\sqrt{4\lambda_i^2 a_i^2 + 2\lambda_i w_i a_i - 4\lambda_i^2 a_i^2},\ = a_i \pm \frac{1}{2\lambda_i}\sqrt{2\lambda_i w_i a_i}$$

We suppose that $\lambda_i > 0$, because then the condition $\mu_i > a_i$ is satisfied.

$$\mu_{i-opt} = a_i + \underbrace{\sqrt{\frac{w_i a_i}{2\lambda_i}}}_{>0}$$

As discussed before, w_i is the weight given to the node i. We want to give a higher weight to those nodes that are heavily loaded so that they can have a higher priority for transmissions over nodes that are not. Here is how we calculate the weight w_i for each node i

$$w_i = \frac{F_i}{N_i} a_i$$

Here, F_i is the set of neighboring nodes that are transmitting data to the node i, N_i is the entire set of one-hop neighbors of node i, and a_i is the total arrival rate into node i. Here we take into consideration the total load on the node i in terms of arrival rate a_i along with the neighborhood of the node. Because, we do not want to destabilize the neighborhood a node by assigning it a high priority over transmissions. Therefore, we consider both the load a_i and the neighborhood $\frac{F_i}{N_i}$ of node i while assigning weight in order to be fair.

C. Lagrange Dual Approach

In what follows, we use the Lagrange dual decomposition method to solve the minimization problem. The Lagrangian function with the Lagrange multipliers $(\lambda_i)$is given as follows:

$$L(\mu, \lambda) = \sum_i w_i \left(1 + \frac{1}{2} a_i x_i\right) + \sum_i \lambda_i (\mu_i - a_i)$$

where $\mu = \{\mu_i,\ i = 1, \ldots, n\}$ and $\lambda = \{\lambda_i,\ i = 1, \ldots, n\}$.
Then, the Lagrange dual function is: $Q(\lambda) = \min_\mu L(\mu, \lambda)$
Thus, the dual problem is given by: $D : \max_{\lambda > 0} Q(\lambda)$

D. Deterministic Primal-Dual Algorithm

The delay minimization problem can be solved via the following deterministic distributed algorithm

- The $\mu_i's$ are updated by $\mu_i(n+1) = \mu_i(n) - \epsilon_n \nabla_{\mu_i} L(\mu(n), \lambda(n))$
- The Lagrange multipliers are update by $\lambda_i(n+1) = \lambda_i(n) + \epsilon_n \nabla_{\lambda_i} L(\mu(n), \lambda(n))$, where

$$\nabla_{\mu_i} L = \frac{\delta}{\delta \mu_i} \left[\sum_j w_j \left(1 + \frac{1}{2} a_j x_j\right) + \sum_j \lambda_j (\mu_j - a_j)\right] = w_i \frac{1}{2} a_i \frac{-1}{(\mu_i - a_i)} + \lambda_i = \frac{-w_i a_i}{2(\mu_i - a_i)^2} + \lambda_i$$

$$\nabla_{\lambda_i} L = \frac{\delta}{\delta \lambda_i} \left[\sum_j w_j \left(1 + \frac{1}{2} a_j x_j\right) + \sum_j \lambda_j (\mu_j - a_j)\right] = \mu_i - a_i$$

We note that in the above algorithm, we have used the same step size ϵ_n for both the primal and the dual algorithms. We can finally write the Primal-Dual algorithm as follows

$$\mu_i(n+1) = \mu_i(n) + \epsilon_n \left(\frac{w_i(n)\, a_i(n)}{2(\mu_i(n) - a_i(n))} - \lambda_i(n)\right),\ \lambda_i(n+1) = \lambda_i(n) + \epsilon_n (\mu_i(n) - a_i(n))$$

IV. Stochastic Delay Control And Stability Under Noisy Conditions

In this section, we examine the convergence performance of the above distributed algorithms under stochastic perturbations, due to noisy feedback information.

A. Stochastic Primal-Dual Algorithm For Delay Control

In the presence of noisy feedback information, the gradients are estimators. More specifically, the stochastic version of the primal-dual algorithm is given as follows

$$\mu_i(n+1) = \mu_i(n) - \epsilon_n \cdot \hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)),\ \varphi_i(n+1) = \varphi_i(n) + \epsilon_n \cdot \hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) \tag{9}$$

where $\hat{L}_{\mu_i}$ is an estimator of $\nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))$ and $\hat{L}_{\varphi_i}$ is an estimator of $\nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))$.

B. Probability One Convergence Of Stochastic Delay Control Algorithm

Next, we examine in detail the models for stochastic perturbations. Let $\{\mathcal{F}_n\}$ be a sequence of σ-algebras generated by $\{(\mu_i(m), \varphi_i(m)), \forall m \leq n\}$. For convenience, we use $\mathrm{E}_n[\cdot] = \mathrm{E}[\cdot|\mathcal{F}_n]$ to denote the conditional expectation.

1) Stochastic gradient $\hat{L}_{\mu_i}$: Observe that $\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) = \nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) + \alpha_i(n) + \zeta_i(n)$, where

$$\alpha_i(n) \triangleq \mathrm{E}_n\left[\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] - \nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)),\ \zeta_i(n) \triangleq \hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) - \mathrm{E}_n\left[\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] \tag{10}$$

i.e. $\alpha_i(n)$ is the biased random error of $\nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))$ and $\zeta_i(n)$ is a martingale difference noise since $\mathrm{E}_n[\zeta_i(n)] = 0$.

2) Stochastic gradient $\hat{L}_{\varphi_i}$: Observe that $\hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) = \nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) + \beta_i(n) + \xi_i(n)$, where

$$\beta_i(n) \triangleq \mathrm{E}_n\left[\hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] - \nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)),\ \xi_i(n) \triangleq \hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) - \mathrm{E}_n\left[\hat{L}_{\varphi_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right]$$

i.e. $\beta_i(n)$ is the biased random error of $\nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))$ and $\xi_i(n)$ is a martingale difference noise.

We impose the following standard assumptions in order to examine the convergence of the stochastic primal-dual algorithm:

- A1. *We assume that the estimator of the gradients are based on the measurements in each iteration only.*
- A2. *Condition on the step size:* $\epsilon_n > 0$, $\epsilon_n \to 0$, $\sum_n \epsilon_n \to \infty$ *and* $\sum_n \epsilon_n^2 < \infty$.
- A3. *Condition on the biased error:* $\sum_n \epsilon_n |\alpha_i(n)| < \infty$ *and* $\sum_n \epsilon_n |\beta_i(n)| < \infty$, $\forall i$.
- A4. *Condition on the martingale difference noise:* $\sum_n \epsilon_n \left[\zeta_i(n)^2\right] < \infty$ *and* $\sum_n \epsilon_n \left[\xi_i(n)^2\right] < \infty$, $\forall i$.

Proposition 1: We have the following proposition. Under Conditions **A1** - **A4,** the iterates, generated by stochastic approximation algorithm (9), converge with probability one to the optimal solutions of Problem.

Sketch of the proof: The proof consists two steps. First, using the stochastic Lyapunov Stability Theorem, we establish that the iterates generated by (9) return to a neighborhood of the optimal points infinitely often. Then we show that the recurrent iterates eventually reside in an arbitrary small neighborhood of the optimal points, and this is proved by using *local analysis* .

We use the following example to illustrate how to characterize sufficient conditions for the almost sure convergence of stochastic gradient algorithms.

We assume that the exponential marking technique is used to feedback the price information, φ_i, to the source nodes. Therefore the overall non-marking probability is that $p_i = \exp(\varphi_i)$.

To estimate the overall price, source i sends N_i packets during round n and counts the non-marked packets. For example, if K non-marked packets have been counted, then the estimation of the overall price $\hat{p}_i$can be K/N_i. Therefore

$$\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) = -\frac{w_i a_i}{2(\mu_i - a_i)^2} + \log(\hat{p}_i) \tag{11}$$

By the definition of (10), we have

$$\alpha_i(n) = \mathrm{E}_n[\log(\hat{p}_i)] - \log(p_i)$$

Note that K is a Binomial random variable with distribution $B(N_i, q)$. When N_s is sufficiently large, it follows that $\hat{p}_i \sim \aleph(p_i, p_i(1-p_i)/N_i)$ and $\hat{p}_i \in \left[P_i - c/\sqrt{N_i}, P_i + c/\sqrt{N_i}\right]$ with high probability, where c is a positive constant. Then the estimation bias of the price information can be upper-bounded as $|\alpha_i(n)| \leq \frac{c'}{\sqrt{N_i}}$ for large N_i, where c' is some positive constant.

To ensure the convergence of primal-dual algorithm, from condition **A3,** it suffices to have that

$$\sum_n \frac{\epsilon_n}{\sqrt{N_i}} < \infty$$

Next, we discuss that the variance condition **A4** is satisfied for $\zeta_i(n)$. By (10) and (11),

$$\begin{aligned} \mathrm{E}_n\left[\zeta_i(n)^2\right] = \mathrm{E}_n\left[\hat{L}^2_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] - \mathrm{E}^2_n\left[\hat{L}_{\mu_i}(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n))\right] = \mathrm{E}_n\left[\log^2(\hat{p}_i)\right] - \mathrm{E}^2_n[\log(\hat{p}_i)] \\ \leq \mathrm{E}_n\left[\log^2(\hat{p}_i)\right] \leq \mathrm{E}_n\left[\log^2(p_i + c)\right] \quad \forall N_i \gg 0 \end{aligned}$$

Similar studies can be done for $\beta_i(n)$ and $\xi_i(n)$.

V. Rate Of Convergence of Stochastic Delay Control Algorithm

The rate of convergence is concerned with the asymptotic behavior of normalized errors about the optimal points. Our primal-dual algorithm can be rewritten as a general constrained form as follows:

$$\begin{bmatrix} \mu_i(n+1) \\ \varphi_i(n+1) \end{bmatrix} = \begin{bmatrix} \mu_i(n) \\ \varphi_i(n) \end{bmatrix} + \epsilon_n \begin{bmatrix} -\nabla_{\mu_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) \\ \nabla_{\varphi_i} L(\boldsymbol{\mu}(n), \boldsymbol{\varphi}(n)) \end{bmatrix} + \epsilon_n \begin{bmatrix} \alpha_i(n) + \zeta_i(n) \\ \beta_i(n) + \xi_i(n) \end{bmatrix} + \epsilon_n \begin{bmatrix} Z_n^{\mu_i} \\ Z_n^{\varphi_i} \end{bmatrix}$$

where $\epsilon_n Z_n^{\mu_i}$ and $\epsilon_n Z_n^{\varphi_i}$ are the correction term which force μ_i and φ_i to reside inside the constraint set. As is standard in the study on the rate of convergence, we assume that the iterates generated by the stochastic primal-dual algorithm have entered in a small neighborhood of an optimal solution (μ_i^*, φ_i^*).

To characterize the asymptotic properties, we define $U_{\mu_i}(n) \triangleq \left(\mu_i\left(n\right) - \mu_i^*\right)/\sqrt{\epsilon_n}$ and $U_{\varphi_i}(n) \triangleq \left(\varphi_i\left(n\right) - \varphi_i^*\right)/\sqrt{\epsilon_n}$, and we construct $U^n(t)$ to be the piecewise constant interpolation of $U(n) = \{U_{\mu_i}\left(n\right), U_{\varphi_i}\left(n\right)\}$, i.e., $U^n(t) = U_{n+1}$, for $t \in [t_{n+i} - t_n, t_{n+i+1} - t_n]$, where $t_n \triangleq \sum_{i=0}^{n-1} \epsilon_n$.

A5. Let $\theta(n) \triangleq (\mu_i(n), \varphi_i(n))$ and $\phi(n) \triangleq (\zeta(n), \xi(n))$. Suppose for any given small $\rho > 0$, there exists a positive definite symmetric matrix $\Sigma = \sigma\sigma'$ such that

$$\mathrm{E_n}\left[\phi_\mathrm{n}\phi_\mathrm{n}^\mathrm{T} - \Sigma\right] I\left\{|\theta(n) - \theta^*| \le \rho\right\} \to 0, \;\; as \;\; n \to \infty.$$

Define

$$A \triangleq \begin{bmatrix} L_{\mu_i\mu_i}\left(\boldsymbol{\mu}^*, \boldsymbol{\varphi}^*\right) & L_{\varphi_i\mu_i}\left(\boldsymbol{\mu}^*, \boldsymbol{\varphi}^*\right) \\ -L_{\varphi_i\mu_i}\left(\boldsymbol{\mu}^*, \boldsymbol{\varphi}^*\right) & 0 \end{bmatrix}$$

A6. Let $\epsilon_n = 1/n$, and assume $A + I/2$ is a Hurwitz matrix. Note that it can be easily shown that the real parts of the eigenvalues of A are all non-positive (cf. page 449 in [9]).

Proposition 2: We have the following proposition.

a) Under Conditions A1 and A3-A6. $U^n\left(\cdot\right)$ converges weakly to the solution (denoted as U) to the Skorohod problem

$$\begin{pmatrix} dU_{\mu_i} \\ dU_{\varphi_i} \end{pmatrix} = \left(A + \frac{I}{2}\right)\begin{pmatrix} U_{\mu_i} \\ U_{\varphi_i} \end{pmatrix} dt + \sigma dw(t) + \begin{pmatrix} dZ_{\mu_i} \\ dZ_{\varphi_i} \end{pmatrix}$$

b) If $(\mu_i^*, \varphi_i^*)$is an interior point in the constraint set, the limiting process U is a stationary Gaussian diffusion process, and $U(n)$ converges in distribution to a normally distributed random variable with mean zero and covariance Σ.

c) If (μ_i^*, φ_i^*) is on the boundary of the constraint set, then the limiting process U is a stationary reflected linear diffusion process.

Proposition 2 can be proved by appealing to a combination of tools used in the proofs of Theorem 5.1 in [10] and Theorem 2.1 in Chapter 6 in [11]. Roughly, we can expand, via a truncated Taylor series, the interpolated process $U^n(t)$ around the chosen saddle point (μ_i^*, φ_i^*). Then, the main new step is to show that the tightness of $U^n(t)$. To this end, we can follow part 3 in the proof of Theorem 2.1 in Chapter 6 in [11] to establish that the biased term in the interpolated process diminishes asymptotically. Then, the rest follows from the proof of Theorem 5.1 in [10].

The rate of convergence depends heavily on the smallest eigenvalue of $\left(A + \frac{I}{2}\right)$. The more negative the smallest eigenvalue is, the fast the rate of convergence would be. The reflection terms would help increase the speed of convergence, which unfortunately cannot be characterized exactly.

VI. Simulation Results

In this section, we implement the proposed deterministic distributed primal-dual algorithm. Specifically, we consider a simple 8-node wireless sensor network as shown in Fig.1. All the sensors sample data with $\tau_i = 0.1$. We use a random access CSMA/CA like MAC without backoff. We first fix the routing in the network, and thus, fixing the arrival rate at each node. We then look at the convergence of primal-dual algorithm. The results obtained by the proposed primal-dual algorithm, together with the theoretical optimal solution, are presented in Table I. It can be easily seen that the results obtained from the primal-dual algorithm is very close to the optimal solution.

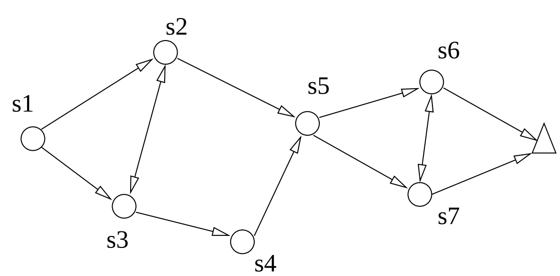

Fig. 1. A Simple Network Topology

We now look at the convergence of the distributed primal-dual algorithm for some nodes in the network w.r.t time. Fig. 2 shows the convergence of distributed primal-dual algorithm for node 3, 4, 5, and 6 in the network. It can be seen that the optimal values of μ_3, μ_4, μ_5, and μ_6 are obtained by the distributed primal-dual algorithm in less than 100 iterations of the algorithm. This shows a very fast convergence of the distributed primal-dual algorithm.

TABLE I

COMPARISON BETWEEN THE RESULTS OF THE PROPOSED PRIMAL-DUAL ALGORITHM AND THE THEORETICAL OPTIMAL SOLUTION

Node	a_i	μ_{i-opt}	$\mu_{i-primal-dual}$
1	0.1	0.102	0.121
2	0.2	0.208	0.225
3	0.1	0.1220	0.125
4	0.2	0.241	0.256
5	0.35	0.383	0.412
6	0.7	0.719	0.743
7	1.05	1.058	1.072

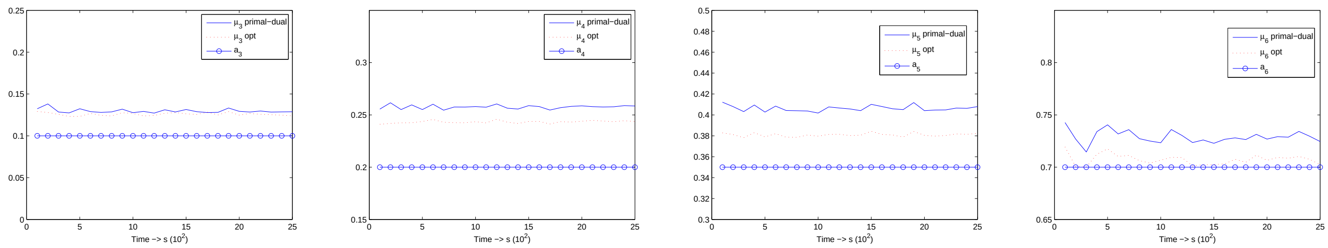

Fig. 2. Convergence of μ_3, μ_4, μ_5 and μ_6 using distributed primal-dual algorithm

VII. CONCLUSIONS AND FUTURE WORK

We consider a general purpose wireless sensor network with n sensor nodes. The objective for the open system was to minimize the total delay in the network where the constraints are the arrival-rate and service-rate of a node. Particularly, we have shown that the objective function is strictly convex for the entire network. We then use the Lagrangian dual decomposition method to devise a distributed primal-dual algorithm to minimize the delay in the network. The deterministic distributed primal-dual algorithm requires no feedback control and therefore converges almost surely to the optimal solution. The results show that the required optimal value of *service rate* is achieved for every node in the network by the distributed primal-dual algorithm. It is important to pay equal attention to both the observed delay in the network and energy consumption for data transmissions. A fast convergence means that only a little extra energy is consumed to perform local calculations to achieve the desired optimizations. Only energy-efficient routing might not serve any purpose for some sensor network applications. Similarly for the stochastic delay control algorithm, we have shown a probability one convergence and its rate of convergence which is entirely distributed in nature.

In the future, we will consider a dynamic routing approach and go into a stochastic distributed primal-dual algorithm approach to observe the effect of routing on system performance. We also consider the problem of minimizing the probability of collision for random medium access wireless sensor networks in order to maximize the network lifetime of battery operated sensor nodes.

REFERENCES

[1] Jr. R. L. Hamilton and H.-C. Yu. Optimal routing in multihop packet radio networks. IEEE Infocom Conference Proceedings, pp. 389-396, San Francisco, CA, USA, 1990.

[2] M. Farukh Munir, Arzad A. Kherani, F. Filali. On stability and sampling schemes for wireless sensor networks. IEEE/IFIP WiOpt Conference Proceedings, Limassol, Cyprus, 2007.

[3] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad hoc networks. In IEEE Infocom Conference Proceedings, Tel Aviv, Israel, Mar. 26-30, 2000, pp. 22-31.

[4] M. Bhardwaj and A. P. Chandrakasan. Bounding the lifetime of sensor networks via optimal role assignments. In IEEE Infocom Conference Proceedings, New York, Jun. 23-27, 2002, pp. 1587-1596.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In ACM Mobicom Conference Proceedings, pp. 56-67, Boston, MA, USA, 2000.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application specific protocol architecture for wireless microsensor networks. IEEE Trans. Wireless Comm., 1(4), 660-670, Oct. 2002.

[7] M. Farukh Munir, Arzad A. Kherani, F. Filali. A Distributed Algorithm to achieve Cesaro-Wardrop equlibrium in wireless sensor networks, pp. 1278-1283, IEEE CCNC Conference Proceedings, Las Vegas, NV, USA, 2008.

[8] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks, IEEE Wireless Communications, Vol. 11, No. 6, PP. 6-28, 2004.

[9] D. P. Bertsekas. *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[10] R. Buche and H. J. Kushner. Rate of convergence for constrained stochastic approximation algorithms. *SIAM Journal on Control and Optimization*, vol. 40, pp. 1011-1041, 2001.

[11] H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer 2003.