Distributed Power Minimization for Data Aggregation in Wireless Sensor Networks

Chun-Chia Chen∗, Ness B. Shroff† and Duan-Shin Lee∗,
∗ Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C.
† Departments of ECE and CSE, The Ohio State University, Columbus, OH 43210, USA
Email: ∗{AlanChen, lds}@cs.nthu.edu.tw, †shroff@ece.osu.edu

Abstract—Wireless sensor networks attract more and more attention since they are capable of monitoring the environment. Since wireless sensor nodes typically have limited energy and power, power efficiency is a main concern in designing protocols for wireless sensor networks. Data aggregation is one of the strategies that can reduce the power consumption in wireless sensor networks. In this paper, we propose a cross layer algorithm with data aggregation to minimize the power consumption. Most importantly, our proposed algorithm is distributed and therefore, it is suitable for wireless sensor networks. From numerical results, we conclude that not all data packets should be aggregated before they arrive the destination nodes.

I. INTRODUCTION

Wireless sensor networks (WSNs) play an increasingly important role in monitoring the environment. By deploying sensor nodes in an observation area of interest, sensor nodes can take readings (e.g. temperature, humidity, or pictures). Then, by wireless communications, the sensor nodes forward the data through the neighboring sensor nodes to a base station for collection and analysis. Sensor nodes typically have a limited battery life. Therefore, power efficiency is a main concern in designing protocols for wireless sensor networks.

Data aggregation [1], [2], [3] is a strategy that saves battery life in wireless sensor networks. For the data collected by the sensor nodes, we are mostly interested in data aggregated in some form rather than the individual raw data. For example, we may be interested in the minimum temperature, the average humidity, or the maximum rainfall in a particular area. One method to measure the desired quantity is to let base stations collect all raw data from all sensor nodes. Then let base stations calculate the desired quantities from the raw data. An alternative method is to aggregate data in the relay sensor nodes before the data are transmitted to the base stations. Clearly in the second method, the number of data packets in wireless sensor networks can be reduced and therefore, the overall battery life of sensors could be extended.

Routing based on data aggregation trees was proposed in [4], [5], [6] to save energy in wireless sensor networks. An aggregation tree [4], [5] or a spanning tree [6], which is rooted at the base station, can provide the routing for the packets of all sensor nodes which have packets destined to the same base station. However, it may quickly exhaust the batteries of the internal nodes in the aggregation tree. One solution to this problem is to construct multiple aggregation trees and try to find the best policy to allocate data rates to each aggregation tree to maximize the network lifetime [7]. Another solution focuses on the allocation of data rates on each link to maximize the network lifetime [8]. These studies tried to find a routing tree to minimize the power consumption or maximize the network lifetime. These studies have not considered the fact that in most wireless networks wireless links operate in a node-exclusive mode. That is, only a subset of wireless links in a wireless network can be active, and at any time no two active wireless links can share a common wireless node. In addition, a distributed method would be more preferable.

In this paper, we propose a cross layer algorithm which can deal with the routing, scheduling, and power control with data aggregation in wireless sensor networks. It is a distributed algorithm. Our model assumes that wireless links operate in a node-exclusive mode. Our main contribution is to provide the power efficient routing for wireless sensor networks with data aggregation.

The rest of the paper is organized as follows. In Section II, we present the system model and problem formulation. This is followed by a detailed description of the distributed algorithm in Section III. The numerical result is in Section IV. Finally, the conclusion is presented in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider the minimization of power consumption with data aggregation in wireless sensor networks. In this paper, we only consider the time slotted wireless sensor networks. There are \( n \) nodes in the system and the set of nodes is denoted by \( N \). Among these nodes, let \( S \) denote the set of source nodes. Let \( D \) denote the set of destination nodes or called base stations. Clearly \( S \subseteq N \) and \( D \subseteq N \). Note that a source node could also be a relay node. We regard the wireless sensor network as a directed graph \( G = \{V, E\} \) in which \( V \) denotes the set of vertices representing nodes and \( E \) denotes the set of direct edges representing wireless links. \( N_o(v) \) and \( N_i(v) \) denote the sets of outgoing and incoming links for node \( v \), respectively. We assume wireless links operate under the node-exclusive mode. We also assume that the packets which are from different source nodes and destined to the same destination node could be aggregated in relay nodes. Finally, we assume that the power consumption for each link only depends on the data rate of that link.

The distributed power minimization scheduling without data aggregation in wireless networks has been proposed [10]. Our scheme is a generalization of the above work. We first briefly describe the joint power minimization problem proposed in [10]. Let \( \mathbf{f} = [f_e^d] \) be a \( |D| \times |E| \) matrix, where \( f_e^d \) denotes the achieved data rate destined to destination node
d on link e. The e\textsuperscript{th} column of f is denoted by f\textsubscript{e}, i.e.,
\begin{equation}
f_e = [f_e^1, f_e^2, f_e^3, \ldots, f_e^{|D|}]^T.
\end{equation}
Let t = [t_e] be an |E| × 1 vector, where t\textsubscript{e} denotes the fraction of time slots that link e is activated. T\textsubscript{d} denotes the average amount of data rate generated from node v to destination d. The joint power minimization problem in [10] is shown below:
\begin{align}
\text{(A)} \min_{f, t} \quad & \sum_{e \in E} \Theta(f_e, t_e) \tag{1} \\
\text{subject to} \quad & \sum_{e \in N(v)} t_e \leq \beta, \forall v \in V, \tag{2} \\
& \sum_{e \in N_e(v)} f_e^d - \sum_{e \in N_i(v)} f_e^d - T_v^d \geq 0, \forall v \neq d \text{ and } \forall d \in D, \tag{3}
\end{align}
where Y = \{(f, t) : f_e^d \geq 0, \forall e, d; 0 \leq t_e \leq \beta, \sum_{d} f_e^d \leq a_t, \forall e, \text{ for some } a \geq 0\}, N(v) = N_i(v) \cup N_o(v) \text{ and } \beta > 0 \text{ is the scheduling efficient ratio.} \text{ Basically, (3) corresponds to the flow constraint at node v. With } \beta = 1/2 - \eta, \text{ where } \eta \text{ is a small arbitrary number, (2) corresponds to the node-exclusive constraint. Note that } \Theta \text{ is defined as}
\begin{equation}
\Theta(f_e, t_e) = \begin{cases} 
0, & t_e = 0, \ 
t_e h(\frac{\sum_{d \in B} f_e^d}{t_e}), & t_e > 0,
\end{cases}
\end{equation}
where h is the power consumption function and we assume that h is a non-decreasing and convex function with h(0) = 0. We refer the reader to [10] for more information on the objective function and these constraint inequalities.

In this paper we generalize problem (A) to model the effect of data aggregation. To illustrate our model on data aggregation, consider an example shown in Fig. 1. In this example, we only consider the packets destined to the same destination node, say d\textsubscript{1}, since only the packets destined to the same destination could be aggregated. Without the data aggregation operation, the total incoming flow must equal to the total outgoing flow. That is,
\begin{equation}
f_e^{d_1} + f_e^{d_2} = f_e^{d_1} + f_e^{d_1} + T_e^{d_1},
\end{equation}
where T_e^{d_1} is the average data workload generated by Node 4 to destination d\textsubscript{1}. With data aggregation, let f_e^{d_1}(\tau) be the number of packets transmitted on link e, in time slot \tau, i.e., 1, 2, \ldots, 5. The data aggregation operation at Node 4 combines and aggregates data such that the total number of outgoing packets in time slot \tau is
\begin{equation}
f_e^{d_1}(\tau) + f_e^{d_2}(\tau) = \max(f_e^{d_1}(\tau), f_e^{d_1}(\tau), f_e^{d_1}(\tau), T_e^{d_1}).
\end{equation}
From the above identity, the time-averaged flow rates for Node 4 must satisfy
\begin{align}
\lim_{T \to \infty} \frac{1}{T} \sum_{\tau=0}^{T} (f_e^{d_1}(\tau) + f_e^{d_2}(\tau)) &= \lim_{T \to \infty} \frac{1}{T} \sum_{\tau=0}^{T} \max(f_e^{d_1}(\tau), f_e^{d_1}(\tau), f_e^{d_1}(\tau), T_e^{d_1}). \tag{4}
\end{align}
One can interchange the order of the limit operation and the summation by replacing the equality in (4) by an inequality. We obtain the following inequality
\begin{align}
f_e^{d_1} + f_e^{d_2} &\geq \max \left( \lim_{T \to \infty} \frac{1}{T} \sum_{\tau=0}^{T} f_e^{d_1}(\tau), \right. \\
&\left. \lim_{T \to \infty} \frac{1}{T} \sum_{\tau=0}^{T} f_e^{d_1}(\tau) \right) \\
&= \max(f_e^{d_1}, f_e^{d_1}, f_e^{d_1}, T_e^{d_1}). \tag{5}
\end{align}
From the above example, we can deal with the flow constraint of a network with data aggregation. We formulate the power minimization scheduling with data aggregation in wireless networks as follows:
\begin{align}
\text{(B)} \min_{f, t} \quad & \sum_{e \in E} \Theta(f_e, t_e) \tag{6} \\
\text{subject to} \quad & \sum_{e \in N(v)} t_e \leq \beta, \forall v \in V, \tag{7} \\
& \sum_{e \in N_e(v)} f_e^d - \sum_{e \in N_i(v)} f_e^d \geq 0, \forall v \neq d \text{ and } \forall d \in D, \tag{8}
\end{align}
The goal of designing the network with data aggregation is to reduce the power consumption. We summarize this property in the following proposition.

\textbf{Proposition 1} The power consumption by the solution of Problem (B) is less than or equal to that by the solution of Problem (A).

\textbf{Proof:} Let \Omega_A and \Omega_B be the feasible sets to Problem (A) and Problem (B), respectively. Let (f, t) be a feasible solution of Problem (A), i.e., (f, t) \in \Omega_A. Then (f, t) also satisfies the constraints in (7) and (8) in Problem (B). Therefore, the (f, t) is a feasible solution of Problem (B), i.e., (f, t) \in \Omega_B. It follows that \Omega_A \subseteq \Omega_B. Let \phi_A and \phi_B be the optimal values of Problem (A) and Problem (B), respectively. Since \Omega_A \subseteq \Omega_B, we can obtain \phi_B \leq \phi_A.

Since the maximum function in (8) is not differentiable, we decompose (8) into two inequalities. In other words, we
rewrite the above problem formulation as follows:

\begin{align}
\text{(P1)} \min_{f, t} & \quad \sum_{e \in E} \Theta(f_e, t_e) \\
\text{subject to} & \quad \sum_{e \in \mathcal{N}(v)} t_e \leq \beta, \forall v \in V, \\
& \quad \sum_{e \in \mathcal{N}(v)} f_e \geq f_e', \forall v \in \mathcal{N}(v), \\
& \quad \forall v \in \mathcal{N}(v) \text{ and } \forall e \in D, \sum_{e \in \mathcal{N}(v)} f_e \geq T_v', \\
& \quad \forall v \neq d \text{ and } \forall e \in D, (f, t) \in Y.
\end{align}

Since the objective function and the constraint functions of Problem (P1) are convex, Problem (P1) is a convex optimization problem which has no duality gap. As a result, we can use the duality method to solve it.

### III. DISTRIBUTED ALGORITHM BASED ON LAGRANGE DUALITY METHOD

We briefly show the steps to apply the duality method. First, we define a Lagrangian function:

\[
L(f, t, \mu, q^d, \ldots, q^{d|D|}, r) = \sum_{e \in E} \Theta(f_e, t_e) + \sum_{v \in V} \mu_v \left( \sum_{e \in \mathcal{N}(v)} t_e - \beta \right) - \sum_{d, e' \in \mathcal{N}(v)} q_{d,e'} \left( f_e - f_{e'} \right) - \sum_{d, v} r_{d,v} \left( f_e - T_v' \right),
\]

where vector \( \mu = [\mu_v]_{v \in V} \geq 0 \), matrix \( q^d = [q_{d,e'}]_{e', v \in \mathcal{N}(v)} \geq 0 \), \( d \in D \) and vector \( r = [r_{d,v}]_{v \in V, d \in D} \geq 0 \) contain Lagrange multipliers. In addition, we define \( q_{d,e'}^d = 0 \), for all \( d \) and \( e' \in \mathcal{N}(d) \). We rearrange the order of the summation and rewrite the above equation as follows:

\[
L(f, t, \mu, q^d, \ldots, q^{d|D|}, r) = \sum_{e \in E} \Theta(f_e, t_e) + \sum_{e \in E} \left( \mu_{x(e)} + \mu_{r(e)} \right) t_e - \sum_{d, e' \in \mathcal{N}(x(e))} f_{d,e'} \sum_{e \in \mathcal{N}(e)} q_{d,e'} + \sum_{d, e' \in \mathcal{N}(r(e))} f_{d,e'} - \sum_{d, v} r_{d,v} \sum_{e \in \mathcal{N}(e)} f_e - T_v',
\]

The dual objective function is

\[
D(u, q^d, \ldots, q^{d|D|}, r) = \min_{(f, t) \in Y} L(f, t, \mu, q^d, \ldots, q^{d|D|}, r),
\]

\[
= \sum_{e \in E} \left[ \min_{(f_e, t_e) \in Y_e} c_e(f_e, t_e) - \beta \sum_{v} \mu_v + \sum_{d} r_{d,v} T_v' \right],
\]

where \( Y_e \) denotes the constraint set on link \( e \), i.e.,

\[
Y_e = \{(f_e, t_e) : 0 \leq f_e' \leq H, \forall v; 0 \leq t_e \leq \beta; \sum_{d} f_{d,e} \leq at_e, \text{ for some } a \geq 0 \}.
\]

Note that the condition \( 0 \leq f_e' \leq H \), where \( H \) is a large enough constant, ensures the convergence of the duality method [10]. Moreover, from (13), the minimization of the Lagrangian function can be decomposed into the minimization on each link. That is, the information for the minimization of \( c_e(f_e, t_e) \) is local on link \( e \).

The dual optimization problem is

\[
\text{(D1)} \quad \max_{u \geq 0, q^d \geq 0, \ldots, q^{d|D|} \geq 0, r \geq 0} \quad D(u, q^d, \ldots, q^{d|D|}, r)
\]

We solve the above dual problem in a distributed manner. Since the primal problem (P1) is convex, the dual problem (D1) is also convex and there is no duality gap between primal and dual problems. Hence, the sub-gradient of \( D(u, q^d, \ldots, q^{d|D|}, r) \) is given by

\[
\frac{\partial D}{\partial \mu_v} = \sum_{e \in \mathcal{N}(v)} t_e - \beta,
\]

\[
\frac{\partial D}{\partial q_{d,e'}} = - \sum_{e' \in \mathcal{N}(e)} f_{d,e'} - f_{d,e'}', \quad \text{where } e' \in \mathcal{N}(v),
\]

\[
\frac{\partial D}{\partial r_{d,v}} = - \sum_{e \in \mathcal{N}(e)} f_e - T_v'.
\]

We then use the sub-gradient method to solve the dual problem (D1). The distributed algorithm is shown below:

**Algorithm 2 : Distributed Algorithm**

At each iteration \( m \),

- At link \( e \), the data rate \( f_e \) and the link assignment \( t_e \) are determined by

\[
(f_e(m), t_e(m)) = \arg \min_{(f_e, t_e) \in Y_e} c_e(f_e, t_e, m)
\]

\[
= \arg \min_{(f_e, t_e) \in Y_e} \left[ \Theta(f_e, t_e) + \left( \mu_{x(e)}(m) + \mu_{r(e)}(m) \right) t_e - \sum_{d} f_{d,e} \sum_{e' \in \mathcal{N}(x(e))} q_{d,e'} \right. \\
+ \sum_{d} q_{d,r(e),e} f_e - \sum_{d} q_{d,x(e),e} f_e - \left. \sum_{d} r_{d,e} f_e(m) \right]
\]

and \( x(e) \) and \( r(e) \) denote the transmission and reception nodes of link \( e \), respectively.
• At node $v$, the dual variables are updated according to

$$
\mu_v(m+1) = \left\{ \mu_v(m) + \alpha^\mu_m \left[ \sum_{e \in N(v)} t_e(m) - \beta \right] \right\}^+;
$$

$$
q_{v,e'}^d(m+1) = \left\{ q_{v,e'}^d(m) - \alpha^q_m \left[ \sum_{e \in N_e(v)} f_e^d(m) \right] - f_{e'}^d(m) \right\}^+ , \text{ where } e' \in N_t(v); \displaybreak[0]
$$

$$
r_{v}^d(m+1) = \left\{ r_{v}^d(m) - \alpha^r_m \left[ \sum_{e \in N_e(v)} f_e^d(m) - T_{v}^d \right] \right\}^+ ,
$$

where the $\{\alpha^\mu_m\}$, $\{\alpha^q_m\}$ and $\{\alpha^r_m\}$ are the step sizes.

In order to map the solution to various network functions, we further discuss how to solve (15). Since $f_e^d(m)$ is the average data rate for destination $d$ on link $e$ using $t_e(m)$ fraction of the time for transmission, we can obtain the following equation

$$
f_e^d(m) = R_e^d(m) t_e(m), \quad (16)
$$

where $R_e^d(m)$ is the instantaneous data rate allocated on link $e$ for destination $d$. Hence, we can rewrite $c_e(f_e, t_e, m)$ as follows: (We omit the time index for the ease of notation and let $R_e = [R_e^d]_{d \in D}$)

$$
c_e(f_e, t_e) = t_e l_e(R_e), \quad (17)
$$

where

$$
l_e(R_e) = h(\sum_d R_e^d) + (\mu_x(e) + \mu_r(e)) \cdot \left[ \sum_{d \neq d'} \left( \sum_{e \in N(x(e))} q_{d',e}^d - \min_{\sigma} \left( q_{d',e}^d, q_{d',e}^d \right) \right) \right] .
$$

Since $t_e \geq 0$, to minimize (17), we should first minimize $l_e(R_e)$. Note that the input of $h(\cdot)$ is the total amount of data rate allocated for all destinations on this link. Since we now consider the total power consumption, it does not matter how we assign data rate to the destinations. We can assign all data rates to one destination as long as the total data rate is the same. As a result, the minimum of $l_e(R_e)$ is attained when all the data rates are allocated to the destination, $\hat{d}$, with the maximum positive difference, i.e.,

$$
\hat{d} = \arg\max_d \left( \sum_{e \in N(x(e))} q_{d,e}^d - \min_{\sigma} \left( q_{d,e}^d, q_{d,e}^d \right) \right). \quad (18)
$$

Hence, $R_{\hat{d}} > 0$ if $(\sum_{e \in N(x(e))} q_{d,e}^d - \min_{\sigma} \left( q_{d,e}^d, q_{d,e}^d \right) > 0)$. Otherwise, $R_{\hat{d}} = 0$ and $R_{\hat{d}} = 0$ for all $d \neq d$. We have determined $\hat{R}_e$ to minimize $l_e(R_e)$. The following is to determine the value of $t_e$. Since $t_e$ is in the interval $[0, \beta]$, in order to minimize $c_e(f_e, t_e) = t_e l_e(R_e)$, the optimal value of $t_e$ is

$$
t_e = \begin{cases} 
\beta, & \text{if } \min_{R_e} l_e(R_e) \leq 0; \\
0, & \text{if } \min_{R_e} l_e(R_e) > 0. \quad (19)
\end{cases}
$$

Now we map the above solution to network functions in various protocol layers:

• **Routing**: In each link, we choose flow $\hat{d}$ to be the candidate for transmission.

• **Power control**: The transmission power on link $e$ is determined by the data rate, $R_e$. That is, link $e$ uses the power which supports the data rate $R_e$.

• **Link scheduling**: $t_e$ determines the amount of time in which link $e$ should be active. However, the exact time-slots in which link $e$ should be active should satisfy the node-exclusiveness constraint. Specifically, given the link assignment $U(m)$, a scheduling solves the conflict by delaying transmission of some links for some time-slots. Clearly, by choosing $\beta = 1/2 - \eta$, a scheduling based on distributed maximal matching can make the system stable [10], [11], [12], [13].

### IV. NUMERICAL RESULT

We demonstrate the performance of the distributed algorithm by an example. The example contains eight nodes with two source nodes and two destination nodes. The topology is shown in Fig. 2. Node 1 and Node 3 are chosen as the source nodes. Node 7 and Node 8 are chosen as the destination nodes. Each source node has two kinds of packets destined to the different destinations, Node 7 and Node 8. Hence, when the packets which are from different sources and destined to the same destination traverse a common node, they can be aggregated. The power-rate function is given in the follow form

$$
\hat{R}_e = W \log_2 \left[ 1 + \sigma_e p_e N_0 W \right],
$$

where $\hat{R}_e$ is the resultant instantaneous data rate of link $e$, $W = 1.0$ MHz is the available bandwidth, $\sigma_e = 1.6 \times 10^{-12}$ is the channel gain of link $e$, $p_e$ is the transmission power of line $e$, and $N_0 = 1.6 \times 10^{-18}$ mW/Hz is the noise spectral density. The details of paths and parameters are shown in Table I, where the flow $xy$ denotes the flow from the source node $x$ to the destination node $y$. In this study we assume that $\beta = (0.5 - 10^{-4})$. This value ensures that a schedule can be found with the node-exclusive interference constraint.
The average result is taken over a moving time window of length 300 time slots. Flow 18 and flow 38 change their data rate from 500 kbps to 250 kbps at the 1200th time-slot. The total simulation time is 24000 slots. The average total power consumption is shown in Fig. 3. From Fig. 3, we show the total power consumption of the network as Algorithm 2 is executed. From this figure, we see that the distributed algorithm can adapt to the change of data rate, and it converges to the solutions calculated off-line. Fig. 3 also shows the total power consumption by the distributed algorithm without data aggregation proposed in [10] for problem A. This figure shows that one achieves substantial power reduction by aggregating data at internal nodes.

It is interesting to examine the distribution of flows in the optimal solution of Problem (B). Take destination 7 in the network shown in Fig. 2 for example. According to Table I, flow 17 (resp. flow 37) can be split and carried along paths 1 → 4 → 7 and 1 → 2 → 5 → 7 (resp. paths 3 → 2 → 5 → 7 and 3 → 6 → 8 → 7). Flows 17 and 37 can be aggregated on link (2,5). If $x \geq y$, $\max(x, y) = x$ no matter how small $y$ is. It is intuitive that the split of flows 17 and 37 is such that flows 17 and 37 contribute an equal amount of data rates on link (2,5). Hence, we simulate the average data rate on link (1,4), (2,5) and (3,6). The results are shown in Fig. 4. Indeed, according to Fig. 4 flows 17 and 37 (resp. flows 18 and 38) contribute an equal amount of data flow to be aggregated on link (2,5). After time slot $1.2 \times 10^4$, the source rate of flows 18 and 38 are reduced. However this observation on flow splitting still holds after time slot $1.2 \times 10^4$. This result also indicates that not all data rate is aggregated with the other before arriving destination nodes.

![Fig. 3. Total power consumption](image)

![Fig. 4. Average data rate](image)

**V. CONCLUSION**

In this paper, we proposed a cross layer algorithm to deal with power control, scheduling and routing problems for wireless sensor networks with data aggregation. Our proposed algorithm is distributed and very suitable for practical implementation. From the numerical results, we see that data rates are spread on multiple links instead of being concentrated on an aggregation tree.

**REFERENCES**


