A TDMA-based Scheduling and Routing Algorithm for Data-Centric Wireless Sensor Networks

Yean-Fu Wen
China University of Technology (CUTe)
Email: yeanfu@ieee.org

Frank Yeong-Sung Lin, and Hung-Shi Wang
National Taiwan University (NTU)
Email: {yslin,r93038}@im.ntu.edu.tw

***Abstract*—Recharging the batteries of a moribund sensor deployed as part of a wireless sensor network is often infeasible due to logistical considerations. With the purpose of prolonging sensor lifetime in such data-centric wireless sensor networks and with emphasis on TDMA-based routing and the efficient scheduling of sensor activities, we propose a mixed-integer nonlinear programming mathematical model, the objective of which is to minimize the total energy consumed by nodes and encompasses dynamic power range, collision free transmission, routing paths, and data aggregation tree constraints. Performing Lagrangean Relaxation, we find a near-optimal solution and verify that our proposed algorithm is energy efficient and bounds latency within a reasonable range. Our experiment results confirm improvement over data aggregation algorithms.**

I. INTRODUCTION

A wireless sensor network (WSN) is comprised of a number of small nodes, including sensor nodes. As recharging the batteries of a sensor is often an infeasible task, we focus on energy conservation from the physical layer up to the application layer for periodic applications. Our solution prolongs network lifetime by means of TDMA-based duty-cycle scheduling, dynamic adjustment of power range, and data aggregation routing. The objective is the minimization of the energy consumption, subject to the following conditions.

The aggregation of data, by which redundancy may be eliminated and the number of transmissions may be minimized, achieves significant energy savings for wireless routing in WSNs [1] [5] [7] [12]. Krishnamachari et al. proposed several data aggregation routing algorithms, namely Shortest Path Tree (SPT), Center Nearest Source (CNS), and Greedy Incremental Tree (GIT) [1] that solve the problem sub-optimally. Here, we propose a heuristic, developed by means of Lagrangean Relaxation (LR), that obtains a near-optimal solution.

Idle listening, the most energy wasteful of all the processes in the MAC protocol [3] [11] [12] [13]. It occurs when nodes listen even when no messages are coming in, but is significant reduced by duty cycling, scheduling non-active, battery-conserving sleep states for nodes when listening activity is unnecessary. Idle listening was reduced by S-MAC (with a fixed duty cycle) [13], the advanced T-MAC (with variable duty cycle by timeout) protocols [11], and D-MAC (with level-by-level scheduling approach). However, the protocol does not address such pertinent issues as data aggregation and dynamic radius.

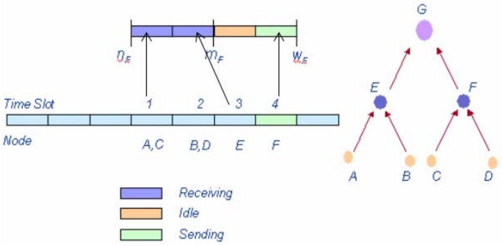


Fig. 1. Sensor activity time slot assignment

In our previous works, we have focused on CDMA-based collision avoidance solution [5] or near-optimal duty-cycle scheduling [12]. In this paper, the energy is conserved from the view point of TDMA to schedule a collision-free duty cycle. As shown in Fig. 1, we divide the time into several time slots. To achieve collision free transmissions, only a single pair of nodes is allowed to communicate within the interference area simultaneously. Each node is assigned a wake up slot begin aggregating data ($n_F = 1$) until complete aggregated data slot ($m_F = 3$), and then assign a slot ($w_F = 4$) that permits to relay the sensed or aggregated data to its next hop. By adjusting the receiving, sending, and sleeping behavior of all nodes, the network traffic is organized orderly.

Since the resources are limited in wireless sensor networks, a centralized algorithm may not be a bad choice in that it saves energy, especially for a regular application, such as temperature, ocean current, and atmospheric pressure. Each sensor node sends to the sink node its remaining battery life and the distance to all its neighbors. The sink node corrects the necessary data and then finds a near-optimal solution. Then, it broadcasts or multicasts the solution to the related nodes.

We use a mathematical formulation to construct an algorithm that builds a data-centric tree and schedules the activities of active sensors. Data-centric routing constructs an aggregation tree known as a Steiner tree, proven to be an NP-Hard problem [1] [4]. The LR-based approach, which has been used to solve many famous NP-complete problems [9], was used to determine near-optimal decisions. By this approach, we derive a number of multipliers and dual decision variables, both of which provide good hints that enable the construction of a primal feasible solution.

The remainder of the paper is organized as follows. The next section presents a mixed-integer nonlinear programming

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

formulation of the TDMA-based data-centric routing problem, including the assignment of duty cycle schedules and radii. In Section III, the LR-based solution approach is brief described. In Section IV, the heuristics for getting good feasible solutions to these problems are addressed. In Section V, computational results are reported. Finally, in Section VI, our conclusions are presented.

II. PROBLEM FORMULATION

By assigning routings and time slots to active sensors by means of a single centralized sink node, we are able to avoid many of the difficulties that arise in a traditional decentralized TDMA environment [6]. For a network synchronized with the PEDAMACS scheme [10] that synchronizes the nodes by means of a high-powered access point, we make the reasonable assumption that propagation delays is ignored, and consider the link delays to be a constraint on the problem.

The problem for a data-centric WSN is modeled as a graph $\Gamma(V, L)$, where V represents the set of connected nodes and L the sets of direct links that enable nodes to communicate, summarized as follows:

Given:

· The set of sensor nodes V distributed on a specific area.
· The set of source nodes S.
· The sink node g.
· The set of candidate paths P_{sg} from the source nodes to the sink node.
· The set of links L.
· The distances $d_{(u,v)}$ between nodes (for all nodes $(u, v) \in L$).
· An arbitrary large number M.
· The maximum end-to-end delay t.

Object:

To minimize the energy consumption of the entire WSN.

Subject to the following constraints:

· The indicator function $\delta_{p(u,v)}$: which is equal to 1 for all links (u, v) on the path p; for all other links, to 0.
· Routing for each source node: only one routing path to the sink node.
· Tree: the combination of routing paths from all source nodes should be a tree; namely, a data aggregation tree.
· Scheduling for all nodes: the TDMA-based behaviors of sleeping, idleness, receiving and sending should be considered.
· Neighbors of each node: all neighbors, determined on the basis of transmission radius and timing, should be considered.
· Collision free transmissions: No nodes in a neighborhood transmit simultaneously.

To determine:

· A routing path x_p for each source node.
· The transmission radius r_u for each sensor node, where $r_u \in R_u$, where R_u is a discrete finite set available for radii.
· The data aggregation tree.
· A decision variable $y_{(u,v)}$ for each link: *true* if a particular link is on the data aggregation tree.
· The variable $\phi_{(u,v)}$ for link (u, v): *true* if the transmission radius of node u covers node v.
· A decision variable z_{uv} for each node: *true* if node v is within the transmission range in the same time slot of node u.
· A decision variable n_u for each sensor node in the data aggregation tree: the wake up time.
· A decision variable m_u for each sensor nodes $u \in V$: the time at which the aggregation time slot has been completed.
· A decision variable w_u for each sensor nodes $u \in V$: the time at which transmission has been completed.

The objective function (IP) is an expression of the total energy consumption, and as such includes all energy that is consumed within the network when data is received (the first term), when nodes are idle (the second term) and sleeping (third term), processing cost (the fourth term) and when data is sent (the fifth term). We seek to minimize the total energy consumption, and additionally work to ensure proper routing, link behavior, scheduling and transmissions.

$$Z_{IP} = \min \sum_{u \in V} \left(\begin{array}{l} (m_u - n_u)E_r + (w_u - 1 - m_u)E_{idle} \\ +(t - (m_u - n_u))E_{sleep} + \\ K \sum_{v \in V} y_{(u,v)} + V \cdot e_u(r_u) \end{array} \right) \tag{IP}$$

subject to:

(a) *Routing constraints*

Routing paths are determined by a set of binary decision variables x_p that indicate for a path p the state of the connection that lies between the source node s to the sink node g; $x_p = 1$ indicates that a candidate path p is used. To ensure that all data is transmitted to the sink node, each source node s is assigned one and only one routing path, as in (1).

$$\sum\nolimits_{p \in P_{sg}} x_p = 1, \quad \forall s \in S \tag{1}$$

(b) *Link constraints*

If a link (u, v) lies on a path p, for that link, the binary decision variable $y_{(u,v)}$ equals 1; otherwise, $y_{(u,v)}$ equals 0. No link can be used on a path more than once, as in (2).

$$\sum\nolimits_{p \in P_{sg}} x_p \delta_{p(u,v)} \leq y_{(u,v)}, \quad \forall s \in S; (u, v) \in L \tag{2}$$

To ensure that all data collected by the sensor nodes gets delivered to the sink node, the structure of the tree is restricted, as in the three link-constraints (3)-(5).

- Only a single node receives the data sent from a source node s, as in (3):

$$\sum\nolimits_{v \in V} y_{(s,v)} = 1, \quad \forall s \in S \tag{3}$$

- There can be no more than one out-degree link for each node, as in (4):

$$\sum\nolimits_{v \in V} y_{(u,v)} \leq 1, \quad \forall u \in V \tag{4}$$

- For data to be delivered, the summation of in-degree links for sink node g is at least 1, as in (5).

$$\sum\nolimits_{u \in V} y_{(u,g)} \geq 1 \tag{5}$$

(c) *TDMA-based duty-cycle scheduling constraints*

The aggregation time slot of a node v begins at 0 and extends until all data from the leaf nodes is aggregated. The aggregation time slot completion decision variable m_v can be at maximum t. A node remains in receive mode until it has received transmissions from all nodes in its sub-tree, identically the time at which all nodes in its sub-tree complete their transmissions, seen as Constraint (6).

$$w_u - t(1 - y_{(u,v)}) \leq m_v, \quad \forall m_v \in \{0, 1, ..., t\}; u, v \in V \tag{6}$$

For a link (u,v) that is involved in the aggregation tree, to prevent losing the data from the leaf node u, node v must wake up before node u finishes aggregation and sends its data to node v, seen here as Constraint (7).

$$n_v \leq m_u + t(1 - y_{(u,v)}), \quad \forall n_v \in \{0,1,...,t\}; u,v \in V \quad (7)$$

The transmission time slot, during which node u begins aggregating data, also ranges from 0 to t. A node will never enter into sleep state before it receives all data coming to it from its offspring, seen as Constraint (8).

$$m_u + 1 \leq w_u, \quad \forall w_u \in \{0,1,2,...,t\}; u \in V \quad (8)$$

(d) *The number of collision nodes constraints*

The number of collision nodes for each communication node is calculated, as in Constraints (9)-(16). $\sum_{u \in V} z_{uv}$ stands for the total number of sensor nodes that interfered with by the transmission of sensor node u, or for the total number of sensor nodes whose transmission interfere with sensor node v.

If a node u is covered within the radius of node u (i.e., $r_u \geq d_{(u,v)}$), and there exists an overlap of time slots between the communication of node u and node v (i.e., $n_v \leq w_u \leq m_v$ or $w_u = w_v$), z_{uv} is set to 1, otherwise 0. By introducing z_{uv1} and z_{uv2}, we can model the relationship between the decision variables ϕ_{uv}, n_v, m_v, w_u, and z_{uv} properly.

By jointly enforcing Constraints (9) and (10), we can model the relationship between r_u, $d_{(u,v)}$, and ϕ_{uv}. These two constraints are complementary, as shown in Table I. If $r_u \geq d_{(u,v)}$, then ϕ_{uv} is dependent on (9) and is set to 1 (Row 1), otherwise 0 (Row 2).

$$r_u - d_{(u,v)}/M \leq \phi_{uv}, \quad \forall (u,v) \in L \quad (9)$$

$$d_{(u,v)}\phi_{uv} \leq r_u, \quad \forall r_u \in R_u; (u,v) \in L \quad (10)$$

Note that no source node can have a transmission radius of 0; namely, $r_s \neq 0$.

If $y_{(u,v)}$ is equal to 1, then ϕ_{uv} must be set to 1; otherwise, to 0 (11).

$$y_{(u,v)} \leq \phi_{uv}, \quad \forall (u,v) \in L \quad (11)$$

If $m_v \geq w_u$, the binary decision variable z_{uv1} should be equal to 1; otherwise, to 0. We can well model their relationship by (12) and (13).

$$m_v - w_u/t \leq z_{uv1}, \quad \forall u,v \in V \quad (12)$$

$$w_u - m_v/t \leq 1 - z_{uv1}, \quad \forall u,v \in V \quad (13)$$

By the same token, the decision variable z_{uv2} stands for whether w_u is larger than n_v or not. The relationships between w_u, n_v, and z_{uv2} can be modeled by (14) and (15). The above six constraints are complementary, as shown in Table II.

$$w_u - n_v/t \leq z_{uv2}, \quad \forall u,v \in V \quad (14)$$

$$n_v - w_u/t \leq 1 - z_{uv2}, \quad \forall u,v \in V \quad (15)$$

TABLE I
EXPLANATION OF THE RELATIONSHIP BETWEEN (9) AND (10)

Row	relationship	Constraint (9)	Constraint (10)
1	$r_u \geq d_{(u,v)}$	$\phi_{uv} = 1$	$\phi_{uv} = 0 or 1$
2	$r_u < d_{(u,v)}$	$\phi_{uv} = 0 or 1$	$\phi_{uv} = 0$

TABLE II
EXPLANATION OF THE RELATIONSHIP BETWEEN (12)-(15)

Row	relationship	Constraint (12)	Constraint (13)
1	$m_v \geq w_u$	$z_{uv1} = 1$	$z_{uv1} = 0 or 1$
2	$m_v < w_u$	$z_{uv1} = 0 or 1$	$z_{uv1} = 0$
	z_{uv2}	Constraint (14)	Constraint (15)
3	$w_u \geq n_v$	$z_{uv2} = 1$	$z_{uv2} = 0 or 1$
4	$w_u < n_v$	$z_{uv2} = 0 or 1$	$z_{uv2} = 0$

TABLE III
EXPLANATION OF CONSTRAINT (16)

Row	z_{uv}	Constraint (16)
1	$z_{uv1} + z_{uv2} + \phi_{uv} = 0$	$z_{uv} = 0 or 1$
2	$z_{uv1} + z_{uv2} + \phi_{uv} = 1$	$z_{uv} = 0 or 1$
3	$z_{uv1} + z_{uv2} + \phi_{uv} = 2$	$z_{uv} = 0 or 1$
4	$z_{uv1} + z_{uv2} + \phi_{uv} = 3$	$z_{uv} = 1$

Accordingly, Constraint (16) confines that if node u is covered within the transmission range and overlap slot time of node v, and there is any overlap between the communication of node u and node v, z_{uv} shall be one, as shown in Table III. In another word, If $\phi_{uv} = 1$, $z_{uv1} = 1$, and $z_{uv2} = 1$, z_{uv} must be equal to 1 (as in row 4), otherwise these constraints are not affected by z_{uv}.

$$(z_{uv1} + z_{uv2} + \phi_{uv}) - 2 \leq z_{uv}, \forall u,v \in V \quad (16)$$

Up to present, we have formulated if $n_v \leq w_u \leq m_v$ and $\phi_{uv} = 1$, then z_{uv} is set to 1. However, if $w_u = w_v$ and $\phi_{uv} = 1$, then $z_{uv1} = 0$, $z_{uv2} = 1$, and Constraint (16) can not enforce z_{uv} to be 1 (i.e., z_{uv} can be 0 or 1). Hence, we need some extra constraints to restrict that when $w_u = w_v$ and $\phi_{uv} = 1$, z_{uv} is equal to one.

By jointly enforcing constraints (17), (18), and (19), we can suitably model the relationship described above.

$$w_u - w_v \leq D_{uv}, \quad \forall D_{uv} \in \{0,1,2,...,t\}; u,v \in V \quad (17)$$

$$w_v - w_u \leq D_{uv}, \quad \forall D_{uv} \in \{0,1,...,t\}; u,v \in V \quad (18)$$

$$1 - z_{uv} \leq D_{uv}, \quad \forall D_{uv} \in \{0,1,...,t\}; u,v \in V \quad (19)$$

where D_{uv} is the difference between w_u and w_v.

(e) *Collision free constraints*

To ensure that no collisions occur while a node is communicating with the sensors within its neighborhood, a communicating node v is constrained to at most one communication node, as seen in Constraint (20).

$$\sum_{u \in V} z_{uv} \leq 1, \quad \forall v \in V \quad (20)$$

III. SOLUTION APPROACH

An LR-based approach was widely used for solving integer programming problems in the 1970s, because it is flexible and provides excellent solutions for problems [9]. In this research problem (IP), by introducing Lagrangean multiplier vectors $\mu^1_{suv}, ..., \mu^{15}_{uv}$, Constraints (2), (6), (8), (9), (10), (11), (12),

1930-529X/07/$25.00 © 2007 IEEE

(13), (14), (15), (16) (17), (18), and (19) are relaxed as the following Lagrangean dual problem (LR).

Objective function:

$$
\begin{gathered}
Z_{LR}\begin{pmatrix} \mu_{suv}^{1}, \mu_{uv}^{2}, \mu_{uv}^{3}, \mu_{uv}^{4}, \mu_{uv}^{5}, \mu_{uv}^{6}, \mu_{uv}^{7}, \mu_{uv}^{8}, \\ \mu_{uv}^{9}, \mu_{uv}^{10}, \mu_{uv}^{11}, \mu_{uv}^{12}, \mu_{uv}^{13}, \mu_{uv}^{14}, \mu_{uv}^{15} \end{pmatrix} = \min \\
\left\{ \sum_{u\in V} \begin{bmatrix} (m_u - n_u)E_r + (w_u - 1 - m_u)E_{idle} + \\ (t - (w_u - n_u))E_{sleep} + K \sum_{v\in V} y_{(u,v)} + \\ V \cdot e_u(r_u) \end{bmatrix} \right. \\
\sum_{s\in S} \sum_{(u,v)\in L} \mu_{suv}^{1} \left[\sum_{p\in P_s} x_p \delta_{p(u,v)} - y_{(u,v)} \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{2} \left[w_u - t(1 - y_{(u,v)}) - m_v \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{3} \left[n_v - t(1 - y_{(u,v)}) - m_u \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{4} \left[m_u + 1 - w_u \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{5} \left[(r_u - d_{(u,v)}) - M\phi_{uv} \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{6} \left[(d_{(u,v)}\phi_{uv} - r_u) \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{7} \left[(y_{(u,v)} - \phi_{uv}) \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{8} \left((m_v - w_u) - tz_{uv1} \right) + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{9} \left((w_u - m_v) - t + tz_{uv1} \right) + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{10} \left[(w_u - n_v) - tz_{uv2} \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{11} \left[(n_v - w_u) - t + tz_{uv2} \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{12} \left[(w_u - w_v) - D_{uv} \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{13} \left[(w_v - w_u) - D_{uv} \right] + \\
\sum_{u\in V} \sum_{v\in V} \mu_{uv}^{14} \left[1 - z_{uv} - D_{uv} \right] + \\
\left. \sum_{u\in V} \sum_{v\in V} \mu_{uv}^{15} \left[(z_{uv1} + z_{uv2} + \phi_{uv}) - 2 - z_{uv} \right] \right\}
\end{gathered}
\tag{LR}
$$

subject to: (1), (3), (4), (5), and (20).

By the process of LR, the objective problem can be decomposed into eleven sub-problems. Due to the limited paper length, the explicit details of the optimal algorithms by which these sub-problems may be solved are not elaborated here. For an excellent text on Lagrangean decomposition, please refer to [9].

Based on the weak Lagrangean duality theorem, the optimal objective function value of the corresponding LR problem is a lower bound (LB) on the objective function value of the primal problem for any given set of nonnegative multipliers (i.e., Z_{LR} is a LB on Z_{IP}) [9]. We construct the following dual problem to calculate the tightest LB and solve the dual problem by the Subgradient method.

Lagrangean dual problem Z_D
Objective function

$$
Z_D = \min Z_{LR}\left(\mu_{suv}^{1}, \mu_{uv}^{2}, ..., \mu_{uv}^{15} \right) \tag{D}
$$

subject to: $\mu_{suv}^{1}, \mu_{uv}^{2}, ..., \mu_{uv}^{15} \geq 0$

Let the decision variable vectors (x_p, $y_{(u,v)}$, r_u, n_u, m_u, and w_u) be subgradients of Z_D. Then, to derive iteration $k+1$ of the subgradient optimization procedures, the multiplier vector, $\pi^k = (\mu_{suv}^{1}, \mu_{uv}^{2}, ..., \mu_{uv}^{15})$, is updated, which gives us $\pi^{k+1} = \pi^k + \tau^k \xi^k$. The step size τ^k is determined by $\tau^k = \sigma \cdot (Z_{IP}^h - Z_D(\pi^k)) \big/ \left\| g^k \right\|^2$, where Z_{IP}^h is the primal objective function value for a heuristic solution, which is an upper bound (UB) on Z_{IP}, and is a constant, $0 < \sigma \leq 2$.

IV. OBTAINING THE PRIMAL FEASIBLE SOLUTION

To obtain the primal feasible solution, we proposed to solve the primal problem with a two-phase heuristic LRA, an LR-based approach. This solution includes two phases: Phase I, a routing scheme, and Phase II, a TDMA-based duty-cycle scheduling scheme. These two phases help us determine each variable efficiently. However, it may potentially lead to the transmission latency violate the maximum end-to-end delay requirement. In order to reduce the maximum end-to-end delay within a reasonable range, we proposed a rerouting heuristic to avoid this situation.

Initially, the six major decision variables x_p, $y_{(u,v)}$, r_u, n_u, m_u, and w_u have not been determined. In Phase I, once the routing scheme has been determined, x_p has been determined, and $y_{(u,v)}$ and r_u are handily derived. With the values of x_p, $y_{(u,v)}$ and r_u, a data-aggregation tree is constructed. Then in Phase II, the variables n_u, m_u, and w_u are determined by a Phase II scheme.

- Phase I: Heuristic for routing scheme
  The basic premise behind Phase I is that when a path has been involved in the data-aggregation tree, then the other source nodes attempt to route to the selected nodes along the shortest paths. The detailed procedures are shown as follows. In the beginning of routing scheme, the arc weight of link (u, v) is regenerated by $\sum_{s\in S} \mu_{suv}^{1} + \mu_{uv}^{2} + e_u(d_{(u,v)}/\overline{R})$, where $\overline{R}$ denotes the maximum radius. Then, the Dijkstra's algorithm is used to get the shortest path from source node to the sink node. Here, the function $e_u(d_{(u,v)}/\overline{R})$ stands for the energy conservation from node u to node v, which is an exponential function of the distance $d_{(u,v)}$ [8]. The reason why we divide it by is for normalization purpose such that the arc weight will not be only dominated by $d_{(u,v)}$.
  After obtained the set of routing path and path cost, we select a path with the minimum cost and set the corresponding $y_{(u,v)}$ to one, and then adjust the arc weights along the path to be zero. Then, run Dijkstra's algorithm once again. The arc weights adjusting procedure and Dijkstra's algorithm are repeated until all source nodes find a path route to the sink node.

```
Step 1 Set the arc weight for link $(u,v)$ to be $\sum_{s\in S}\mu^{1}_{suv} + \mu^{2}_{uv} + e_u(d_{(u,v)}/\overline{R})$, and run Dijkstra's algorithm to get the shortest paths for all source nodes. Select the minimum cost path.
Step 2 Set link ($y_{(u,v)} = 1$) to one and mark the node along the path. Then, adjust the arc weight of these selected links to zero.
Step 3 Find a shortest path by Dijkstra's algorithm for the unmarked source node.
Step 4 Repeat Step 2-3 until a path to the sink node has been found for all source nodes.
Step 5 The set power range $r_u$, the energy required for node $u$ to reach its next hop node $v$, is calculated based on the selected links $y_{(u,v)}$.
Step 6 $y_{(u,v)}$ is used to construct a data-aggregation tree.
```

- Phase II: Heuristic for scheduling scheme

  The basic premise behind Phase II is the minimization of the total number of slots used in transmission, which means assigning as many of the same time slots as possible for the active nodes to shorten the duty-cycle. The out-degree of a particular node stands for the possibility that this node potentially influences the transmission of its neighbors. If we assign a higher priority for nodes with lower numbers of out-degrees to transmit data out, more nodes will transmit data simultaneously without collision. Thus, this procedure potentially reduces the total number of assigned slots. The detailed procedures are shown as follows.

```
Step 1 Initialize $current\_slot = 1$. By executing a topology-sort algorithm, the outliers of the data-aggregation tree can be derived. These outliers are put into $stack\_1$.
Step 2 Sort the nodes in $stack\_1$ on the basis of the number of out-degrees. Pop the node with the minimum number of out-degrees. If this popped node does not interfere with the transmissions of the node in the previous slot, we go to Step 3; otherwise, the node is put into $stack\_2$, and we go to Step 1.
Step 3 Set the decision variable $w_u$ to $current\_slot$. Set $m_u$ to the maximum value of $w_u$ from its sub-tree. Set $n_u$ to the minimum $m_u$ from its sub-tree. Repeat Steps 1-3 until $stack\_1$ is empty.
Step 4 Swap the values in $stack\_1$ and $stack\_2$. Set $current\_slot$ to $current\_slot + 1$.
Step 5 Repeat Steps 1-4 until $stack\_2$ is empty.
```

- Rerouting scheme

  Based on the above 2-phase heuristic, we focus on a feasible solution for routing and scheduling that focuses only on low-energy conservation. However, the latency of the network may be in violation of the required maximum end-to-end delay. In order to improve the solution quality and decrease the end-to-end delay to within a reasonable range, we proposed the rerouting heuristic shown as follows.

```
Step 1 Identify the path (denoted as $p'$) that incurs the highest end-to-end delay. Note that this path can find the sink node.
Step 2 Investigate the nodes on $p'$ one by one. For each checked node (denoted as $n$), each node (denoted as $k$) is investigated. If the end-to-end delay of node $u$ plus one unit of delay is smaller than that of $k$, then reroute the path from $n$ to $k$.
Step 3 Update the corresponding decision variables. Reconstruct the data aggregation tree.
Step 4 Repeat Steps 1-3 until the value of $m_g$ is less than the maximum allowable end-to-end delay.
```

The worst case scenario for these problems is $O(|S||V|^2)$ by Dijkstra's algorithm for each source node per iteration. The time complexity is the same as SPT and CNS algorithms per iteration. However, the number of iterations I is required to adjust multipliers with our algorithms.

V. EVALUATION AND EXPERIMENTAL RESULTS

In order to test the quality of our proposed heuristics, we conducted several experiments that used two data aggregation algorithms, SPT and CNS [1], as a basis for comparison. We denote the value obtained by the LR problem as LB, and the value get from LRA as UB. Besides, two metrics, 'gap' (calculated by $(UB - LB)/UB * 100\%$)is used to evaluate measures. The experimental scenarios are:

- Random network with different number of sensor nodes;
- Grid network with different number of sensor nodes; and
- Random network with different density of source nodes

In a random network, all sensors are scattered disorderly in the field. Both the position and density of the sensor nodes are haphazard. In a grid network, sensors are placed in uniformly.

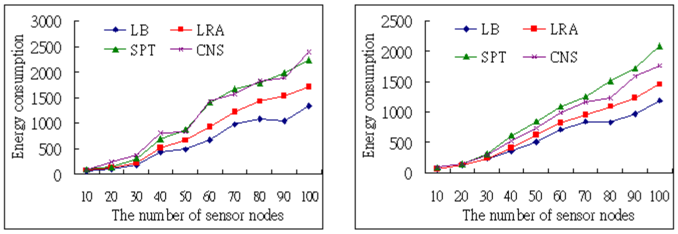
Furthermore, two different sensor placement manners, namely (i) random sources and (ii) congregated sources are tested in the random network. In (i), the sink node is placed in the center of the topology, and all source nodes are dispersedly scattered around the sink. In (ii), the sink node is placed in the corner of the topology, and all source nodes are scattered in another corner. In the last part, in order to observe the impact effect when the number of source nodes increases, we carry an experiment out in random network with different number of source nodes.

The energy savings of CNS was better than SPT as a general rule. In Fig. 2(a), when all source nodes are dispersedly scattered around the sink node, some energy is wasted by CNS on transmitting the information to a remote source. However, the LRA does not lead to the side effect like CNS. So it is eminently superior to these simple algorithms by 11.1%-51.6%. In networks shown in Fig. 2(b), all data was first aggregated by the nearest source node, before being sent to the sink node with CNS algorithm. Thus, the algorithm is a great improvement over SPT. Nevertheless, the algorithm we proposed still outperformed both SPT and CNS by 13.8%-49.8%. The duality gap was less than 36.8%.

Fig. 3 shows the energy consumption rate for different numbers of sensor nodes in grid networks. SPT has the worst performance of the heuristics due to its intuitive selection process. As was mentioned, the energy consumption incurred by CNS is slightly lower than that by SPT when all source nodes are congregated. The performance of LRA, the proposed algorithm, is significantly better than others, especially in cases when there are a large number of nodes. Fig. 4 shows that increasing the density of sources gives the same results as Fig. 3.

When nodes are deployed randomly, the end-to-end delay is dependent on the quantity of sensor nodes and the density of source nodes. Our proposed algorithm focuses on achieving minimum energy in conjunction with a shorter delay. The experimental results shown in Fig. 5 demonstrate that it obtains significantly shorter end-to-end delay.

Despite the energy savings that have been achieved here, after an extended period, some sensors in a data aggregation tree may perish prematurely due to undue traffic conditions; therefore, future work can concern load-balancing and the permutations of an aggregation tree.



(a) Random source (b) Congregated sources

Fig. 2. Effect of sensor node quantity on energy consumption for heuristics with random network.

VI. CONCLUSION

In this paper, primarily regarding to data-centric WSNs, we emphasized routing and the scheduling of sensor activity. Beginning from a mathematical formulation for energy conservation, delay, scheduling, and data-centric routing, we developed a comprehensive optimization-based solution. We developed an LR-based heuristic; furthermore, we proposed a rerouting heuristic that ensures that the end-to-end delay is bound within a reasonable range. In order to evaluate the quality of our solution, we carried out a number of experiments that demonstrated that our algorithm significantly outperformed SPT and CNS both in random and grid network by 11.1%-51.6% with a duality gap of less than 36.8%. In our future work, we will focus on sensibility analysis to discuss how the link weight are obtained.

REFERENCES

[1] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling Data-Centric Routing in Wireless Sensor Networks," *in USC Computer Engineering Technical Report CENG 02-14*, 2002.

[2] C. Intanagonwiwat, R. Govidan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *in Proc. ACM/IEEE MobiCom*, Aug. 2002, pp. 56-67.

[3] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay Efficient Sleep Scheduling in Wireless Sensor Network," *in Proc. IEEE INFOCOM*, Mar. 2004, pp. 2470-2481.

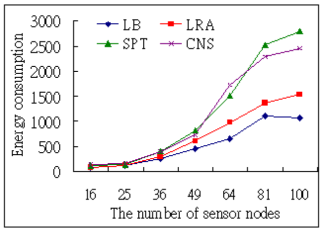


Fig. 3. Energy consumption by quantity of sensor nodes (Grid Network).

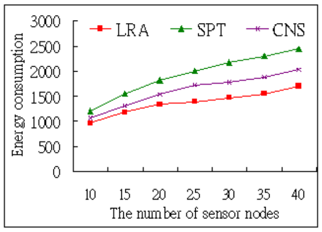
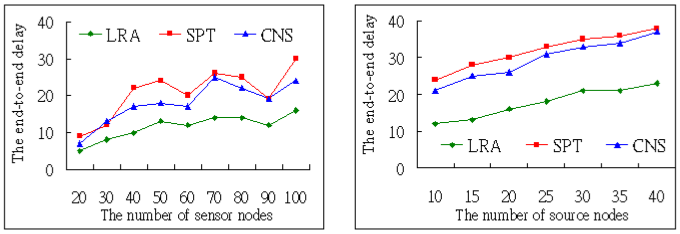


Fig. 4. Energy consumption with varying density of source nodes (Random Network).



(a) Number of sensor nodes (b) Density of source nodes

Fig. 5. Effect on end-to-end delay (Random Network).

[4] Garey M. R. and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[5] H. H. Yen, F. Y. S. Lin, and S. P. Lin, "Energy-Efficient Data-Centric Routing in Wireless Sensor Networks," *IEICE Trans. on Communications*, E88-B(12), Dec. 2005, pp. 4470-4480.

[6] I. Demirkol, C. Ersoy, F. Alagoz, "MAC Protocols for Wireless Sensor Networks: A Survey," *IEEE Communications Magazine*, 44(4), Apr. 2006, pp. 115-121.

[7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Network: A Survey," *Computer Network*, 38(4), Mar. 2002, pp. 393-422.

[8] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized Minimum-Energy Broadcasting in Ad-Hoc Networks," *in Proc. IEEE INFOCOM*, Mar. 2003, pp. 2210-2217.

[9] M. L. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science*, 27(1), Jan. 1981.

[10] S. C. Ergen, P. Varaiya, "PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks," *IEEE Transactions on Mobile Computing*, 5(7), Jul. 2006, pp. 920-930.

[11] T. V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *in Proc. ACM SenSys*, Nov. 2003, pp.171-180.

[12] Y. F. Wen and Frank Y. S. Lin, "Cross-Layer Duty Cycle Scheduling With Data Aggregation Routing in Wireless Sensor Networks," *in Proc. IFIP EUC*, Seoul Korea, Aug. 2006, pp. 894-903.

[13] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *in Proc. IEEE INFOCOM*, Jun. 2002, pp. 1567-1576.