Distributed Flow Optimization Control for Energy-Harvesting Wireless Sensor Networks

Kiyoshi Nakayama, Nga Dang, Lubomir Bic, Michael Dillencourt, Elaheh Bozorgzadeh, Nalini Venkatasubramanian
Dept. of Computer Science, University of California, Irvine, USA
Email: {knakayam, ntdang, bic, dillenco, eli, nalini}@ics.uci.edu

***Abstract*—This paper proposes a distributed flow-based routing technique in energy-harvesting wireless sensor networks (EHWSNs) in order to balance the energy consumptions by sending packets assigned to routers that are sent from sensors to base stations. The objective of the flow optimization problem is to minimize the total load factors of all the nodes and wireless links, which leads to sustainable management of the sensor networks that exploit renewable power from energy harvesting systems. We propose a novel algorithm based on tie-set graph theory where the underlying graph of an EHWSN is divided into a set of independent loops to significantly reduce the topological complexity, which simplifies the flow optimization problem to be solved in a distributed manner. Simulation experiments against the shortest-path and multi-path algorithms demonstrate that optimized packet flows by the proposed method realize the sustainable EHWSNs and maintain the useful life of storage devices with modest increase in total energy consumption by routings.**

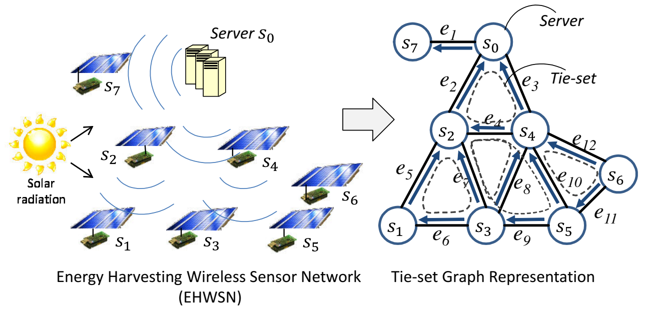


Fig. 1. Graph representation of an EHWSN and a routing example based on tie-sets.

I. Introduction

Development of sustainable routing in energy harvesting wireless sensor networks (EHWSNs) has been considered as an important issue in realizing a reliable wireless sensor networks that harvest power from the environment [1], [2]. In [3], [4], [5] the optimal routing problem for EHWSNs has been studied in terms of maximizing the workload that can be sustained by the network. [3] maximizes data rate for uniform monitoring using a flow algorithm. [4] maximizes the lexicographic of nodes' data rate. [5] solves maximal utility functions of data rate for tree topology networks. Given such EHWSNs with fixed data rate (e.g., maximum data rate in these works), we tackle the problem of guaranteeing and improving routing sustainability on the basis of tie-set graph theory.

The shortest-path distributed computing method as in [6] can also be applied to EHWSNs as it can compute the minimum energy-cost paths that minimize the total energy consumption by routings. However, utilization of shortest-path distributed algorithm may cause traffic congestion on the particular minimum-energy paths. Given that multiple routes to send traffic helps nodes to utilize resources more equitably, a distributed multi-path algorithm based on forwarding packets on different routes provides an easy way to use multiple paths without adding much complexity to a node [7]. Although a lot of efforts have been made in developing distributed multi-path algorithms, the decentralized nature of networked systems has made it difficult to drastically solve the flow-balancing problems.

The objective of flow optimization in this paper is to realize the sustainable EHWSNs by radically balancing the load factors in the network so that each node can maintain a reliable battery life without running out of energy. Therefore, we formulate the flow problem to minimize the maximum value of node-constrained load factor, which is the usage rate of the available energy at a node by outgoing packet flows.

An effective routing algorithm can be realized by exploiting tie-set graph theory that breaks a network into a set of loops and autonomously creating optimal routes using those loops. For instance, an EHWSN is abstracted into a graph representation by substituting sensors, servers, and base stations for nodes, and wireless connections for links of a graph as shown in Fig. 1. Then, the sustainable routing problem can be studied as a flow optimization problem that minimizes the total load balancing factor of all the nodes using the graph model. On the basis of the tie-set structure, optimum flows are calculated as in Fig. 1, which are directly applied to packet routings by the sensor devices.

The effectiveness of the autonomous distributed algorithm based on Tie-sets has been verified in various fields such as real-time optimal power flow control in smart power grid [8], traffic congestion improvement in information networks [9], and failure recovery by tie-set based fault tolerance (TBFT) technique [10]. On the basis of tie-sets, we propose a decentralized algorithm to calculate optimal flows in EHWSNs, which is iterated to realize the global optimization.

In the simulation section, we compare the load factors and energy consumptions optimized by the proposed method with those of using the shortest-path and multi-path algorithms. Then, the simulated behavior of energy battery level of the node that consumes the most energy by routings is shown at each time step to demonstrate reliable use of storage devices by balancing the workload of the nodes in the network.

II. Problem Formulation

In this section, we formulate a Flow Optimization problem in an Energy-Harvesting Wireless Sensor Network (EHWSN).

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

TABLE I
DEFINITIONS FOR INPUTS, VARIABLES, AND OUTPUT

Inputs	
$g_i(t)$	*Harvested Power*, the power harvested at node v_i at time t.
p_k	*Packet Energy,* the energy spent by the source node to process and send a packet across e_k.
Variables	
$a_i(t)$	*Available Energy* stored at node v_i at time t bounded by the size of its storage device.
$\tau_k(t)$	*Recovery Time* required by a node's energy scavenger to replenish the energy to send a packet, defined as $\tau_k(t) = \frac{p_k}{a_i(t)}$.
$c_k(t)$	*Channel Capacity*, the maximum packet rate across an edge e_k at time t.
Output	
$f_k(t)$	*Edge Flow*, a packet rate across an edge e_k imposed by the routing algorithm.

A. Input Model and Variables

We consider a directed connected graph $G = (V, E)$ with a set of nodes $V = (v_i, i = 1, ..., n)$ representing *sensors*, *routers*, and one or more *base stations* and a set of edges $E = \{e_k\} \subseteq V \times V$ representing wireless connectivity. A set of source nodes $\mathcal{S}$ and a set of sink nodes $\mathcal{T}$ are also given. Each link is directed with an arbitrarily defined direction, and a link from node v_i to v_j is denoted interchangeably by either $e(i, j) \in E$ or $i \rightarrow j$.

Table I defines the inputs, output, and other variables used in this paper. The energy stored at node v_i at time t follows the dynamics as

$$a_i(t) = a_i(t-1) + g_i(t), \quad 0 \leq a_i(t) \leq \overline{a}_i. \tag{1}$$

The channel capacity $c_k(t)$ of an edge e_k is the reverse of its recovery time defined as

$$c_k(t) = \frac{1}{\tau_k(t)} = \frac{a_i(t)}{p_k}. \tag{2}$$

B. Conditions

1) Channel Capacity Condition: An edge flow $f(i, j, t)$ is passing data quantity over a link $e(i, j)$ from vertex v_i to v_j at time t. When a flow $f(i, j, t)$ passes along the direction of an edge $e(i, j)$ then $f(i, j, t) \geq 0$ representing the flow going out from the node v_i; otherwise $f(i, j, t) \leq 0$ that is the flow coming from the node v_j. Therefore, the channel capacity $c_k(t)$ is

$$c_k(t) = \begin{cases} \frac{a_i(t)}{p_k}, & \text{if } f_k(i, j, t) \geq 0, \\ \frac{a_j(t)}{p_k}, & \text{if } f_k(i, j, t) < 0. \end{cases} \tag{3}$$

Then, the boundary condition of an edge flow $f_k(i, j, t)$ is

$$-\frac{a_j(t)}{p_k} \leq f_k(i, j, t) \leq \frac{a_i(t)}{p_k}, \quad \text{for } e_k \in E. \tag{4}$$

2) Flow Conservation Law: Let $j : i \rightarrow j$ and $h : h \rightarrow i$ denote the set of successors and predecessors of node v_i in the directed graph, respectively. For each node v_i, a *net flow export* at time t is defined as

$$F_i(t) = \sum_{j:i\rightarrow j} f_k(i, j, t) - \sum_{h:h\rightarrow i} f_k(h, i, t). \tag{5}$$

On the basis of the flow conservation law, the following holds true at node v_i at time t:

$$F_i(t) \begin{cases} > 0, & v_i \in \mathcal{S} \\ < 0, & v_i \in \mathcal{T} \\ = 0, & \text{otherwise.} \end{cases} \tag{6}$$

where $\sum_{v_i \in V} F_i(t) = 0$.

3) Power Budget: For outgoing edge flows from node v_i that satisfy

$$f_k(i, j, t) \geq 0 \text{ or } f_k(h, i, t) < 0,$$

an overall *power budget* $pF_i(t)$ of node v_i is defined as follows:

$$pF_i(t) = \sum_{j:i\rightarrow j} (p_k f_k(i, j, t)) - \sum_{h:h\rightarrow i} (p_k f_k(h, i, t)). \tag{7}$$

The power budget at a node is always $pF_i(t) \geq 0$. Then, the available energy $a_i(t)$ at t is updated to $a'_i(t)$ using the power budget $pF_i(t)$ as

$$a'_i(t) = a_i(t) - pF_i(t). \tag{8}$$

Since the available energy needs to sustain $a'_i(t) \geq 0$, the power budget $pF_i(t)$ must satisfy the following condition:

$$pF_i(t) \leq a_i(t-1) + g_i(t). \tag{9}$$

The equations (1), (8), and (9) suggest that the power budgets should be balanced; otherwise, particular node(s) run out of power quickly, which leads to unsustainable EHWSNs.

C. Objective Function

Here, we define the objective function to balance the workload on nodes (power budgets) in order to maintain sustainable battery life at each sensor device. As the balancing problem of power budgets is intrinsically the same problem as balancing edge flows, we formulate edge-load factors whose summation is to be minimized at all times.

1) Node-Constrained Flow Model: In order to realize the assignment of energetically sustainable workload (power budget) to each node, we want to consider the *node-constrained load factor* at node v_i below:

$$w_i(pF_i(t), t) := \frac{pF_i(t)}{a_i(t)}. \tag{10}$$

We define an overall *node-constrained network load factor* $W_v(pF_i(t), t)$ as

$$W_v(pF_i(t), t) := \sum_{v_i \in V} w_i(pF_i(t), t). \tag{11}$$

Next, we convert this node-constrained flow model into an edge-constrained flow model by focusing on the overall network load factor.

2) Edge-Constrained Flow Model: An *edge-constrained load factor* $w_k(f_k(t))$ for each edge $e_k \in E$ and an overall *edge-constrained network load factor* $W_e(f_k(t), t)$ with regard to $f_k(t)$ are defined as follows:

$$w_k(f_k(t), t) := \frac{|f_k(t)|}{c_k(t)}, \tag{12}$$

$$W_e(f_k(t), t) := \sum_{e_k \in E} w_k(f_k(t), t), \tag{13}$$

Then, we have the following theorem.

Theorem: The node-constrained network load factor is the same as the edge-constrained network load factor as in

$$W_v(pF_i(t), t) = W_e(f_k(t), t) \tag{14}$$

The proof can be found in Appendix A. Therefore, we want to balance the edge-constrained load factors.

Now,we define a convex *edge-load function* $\psi_k(f_k(t), t)$ as

$$\psi_k(f_k(t), t) := w_k^{2\gamma}(f_k(t), t). \tag{15}$$

with some positive integer $\gamma > 0$. In this paper, we simplify the convex function by deciding the value as $\gamma = 1$.

For each edge $e_k \in E$, an *edge-energy function* $\pi(f_k(t), t)$ represents the energy consumption by the edge flow on e_k at time t is defined as

$$\pi(f_k(t), t) := p_k|f_k(t)|. \tag{16}$$

According to (15), since the edge-load function contains the edge-energy function as in

$$\psi_k(f_k(t), t) = \begin{cases} \left(\frac{\pi(f_k(t),t)}{a_i(t)}\right)^{2\gamma}, & \text{if } f_k(i, j, t) \geq 0, \\ \left(\frac{\pi(f_k(t),t)}{a_j(t)}\right)^{2\gamma}, & \text{if } f_k(i, j, t) \leq 0. \end{cases} \tag{17}$$

minimizing the sum of edge-load functions also leads to balancing the energy consumption by the edge flows.

In summary, we define a *Flow Optimization Problem in an EHWSN (FOP-EHWSN)* as follows:

FOP-EHWSN: Given the initial net flow export $F_i(1)$, the initial available energy $a_i(0)$, and the randomized harvested power $g_i(t)$ at time t for each node $v_i \in V$, the packet energy p_k for each edge $e_k \in E$, and a given time sequence $\Gamma = \{t\}$, the FOP-EHWSN is

$$\begin{aligned} \min \quad & \sum_{t\in\Gamma} \sum_{e_k\in E} \psi_k(f_k(t), t), \\ \text{over} \quad & f, F, pF, \\ \text{s. t.} \quad & (1) - (9). \end{aligned} \tag{18}$$

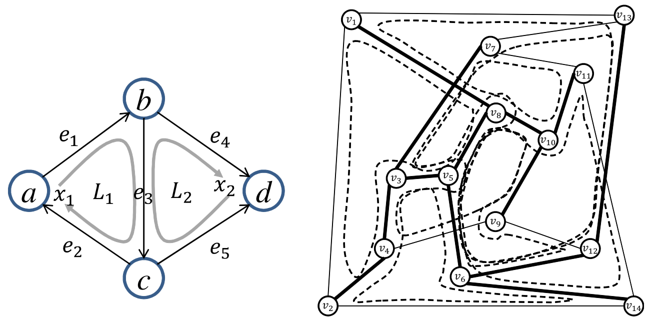
III. Flow Optimization based on Tie-set Graph

Tie-set graph theory divides a network into a set of loops in which optimization is conducted in a distributed manner. In each loop, sustainable routes of workload are constantly calculated. Iterative calculation of sustainable routes in individual loops also leads to global sustainability on the basis of the notion of tie-sets described as follows.

A. Tie-Set Graph Theory

As the tie-set graph theory is described in [11], [12] in detail, we provide the basis for the unfamiliar reader.

For a given connected graph $G = (V, E)$, let $L_\lambda = \{e_1^\lambda, e_2^\lambda, ...\}$ be a set of all the edges that constitutes a loop in G called a *tie-set* [13]. Let T and $\overline{T}$ respectively be a spanning tree and a cotree of G, where $\overline{T} = E - T$. $\mu = \mu(G) = |\overline{T}|$ is called the *nullity* of a graph. Focusing on a subgraph $G_T = (V, T)$ of G and an edge $l_\lambda = e_\lambda(a, b) \in \overline{T}$, there exists only one elementary path $P_T(b, a) \subseteq T$ whose origin is b and terminal is a in G_T. Then, a *fundamental tie-set* that consists of the path P_T and the edge l_λ is uniquely determined as $L_\lambda(l_\lambda) = \{l_\lambda\} \cup P_T(b, a)$. We often refer to a fundamental tie-set as a tie-set. There are μ fundamental tie-sets in G and they are called a *fundamental system of tie-sets* denoted as $\mathbf{L}_B = \{L_1, L_2, ...L_\mu\}$. If a graph G is bi-connected, a fundamental system of tie-sets covers all the vertices and edges as shown in both the planar graph of Fig. 2(a) and the non-planar graph of Fig. 2(b).



(a) In a planar graph (b) In a non-planar graph

Fig. 2. Examples of a fundamental system of tie-sets

B. FOP-EHWSN with Tie-Set Flows

A flow $x_\lambda(t)$ circulating along a tie-set $L_\lambda \in \mathbf{L}_B$ at time t is defined as a *tie-set flow* with respect to a tie-set L_λ. A tie-set flow $x_\lambda(t)$ has its direction as shown in Fig. 2(a). Then, a set of tie-set flows with respect to $\mathbf{L}_B$ at time t is defined as $X(t) = \{x_1(t), x_2(t), ..., x_\mu(t)\}$.

Let $B_\lambda = \{b_{\lambda k}\}$ be a set of edge directions with respect to a tie-set flow $x_\lambda(t)$, where

$$b_{\lambda k} = \begin{cases} 1 & e_k \in L_\lambda, \text{ the same direction as } x_\lambda(t) \\ -1 & e_k \in L_\lambda, \text{ the opposite direction to } x_\lambda(t) \end{cases} \tag{19}$$

For example, B_1 and B_2 in Fig. 2(a) are determined as follows:

$B_1 = \{b_{11}, b_{12}, b_{13}\} = \{1, 1, 1\}, B_2 = \{b_{23}, b_{24}, b_{25}\} = \{-1, 1, -1\}.$

Once a tie-set flow $x_\lambda(t)$ in L_λ is decided, the edge flow $f_k(t)$ at time t is updated as follows:

$$f_k(t) = f_k(t-1) + b_{\lambda k} x_\lambda(t), \quad \text{for } e_k \in L_\lambda, \tag{20}$$

where $f_k(t-1)$ is the edge flow on e_k from a previous time step.

On the basis of (12) and (15), we now define a *Tie-set Flow Optimization (TFO)* function $\phi_\lambda(f_k(t), t)$ at time t as

$$\phi_\lambda(f_k(t), t) = \sum_{e_k\in L_\lambda} \psi_k(f_k(t), t). \tag{21}$$

The TFO function is converted into the function with the tie-set flow (*TFO-x* function) using (20). Here, γ in $\psi_k(f_k(t), t)$ is 1.

$$\phi_\lambda(x_\lambda(t), t) = M_\lambda x_\lambda^2(t) + N_\lambda x_\lambda(t) + Q_\lambda. \tag{22}$$

where $M_\lambda := \sum_{e_k\in L_\lambda} \frac{1}{c_k^2(t)}$, $N_\lambda := \sum_{e_k\in L_\lambda} \left(\frac{2b_{\lambda k} f_k(t-1)}{c_k^2(t)}\right)$, and $Q_\lambda := \sum_{e_k\in L_\lambda} \left(\frac{f_k^2(t-1)}{c_k^2(t)}\right)$, since $b_{\lambda k}^2 = 1$.

As the TFO-x function is a convex function with respect to $x_\lambda(t)$, the optimal tie-set flow $x_\lambda^*(t)$ is $\frac{\partial\phi_\lambda(x_\lambda^*(t),t)}{\partial x_\lambda^*(t)} = 0$, i.e.,

$$x_\lambda^*(t) = -\frac{N_\lambda}{2M_\lambda}. \tag{23}$$

By iteratively optimizing tie-set flows among a fundamental system of tie-sets, edge flows are updated in order to satisfy

$$\left(\frac{\partial \phi_1(x_1(t), t)}{\partial x_1(t)}, ..., \frac{\partial \phi_\mu(x_\mu(t), t)}{\partial x_\mu(t)}\right) \to 0. \tag{24}$$

Since edge flows are gradually optimized, each tie-set flow also converges on 0 at the same time.

$$X(t) = (x_1(t), x_2(t), ..., x_\mu(t)) \to 0. \tag{25}$$

IV. Distributed Control Model for FOP-EHWSN

Section III has discussed that solving the FOP-EHWSN based on tie-sets leads to the global optimization. Now we show how to realize the tie-set based routing with decentralized algorithms where each node/sensor/base station communicates with each other independently.

A. Distributed Algorithm for Flow Optimization

In this section, we describe a *Distributed Algorithm for Flow OPtimization (DAFOP)* that is conducted in each tie-set.

Let V_λ be the set of nodes included in a tie-set L_λ. Every tie-set L_λ has a *leader node* $v_l^\lambda \in V_\lambda$ that holds the topological information of L_λ including the routing table to each node $v_i \in V_\lambda$. At each time step t, a *Tie-set Agent (TA)*[1] that autonomously navigates a tie-set obtains data of nodes V_λ in L_λ using *Measurement Vector (MV)*[2] $y_\lambda(t)$ and reports them to the leader node v_l^λ. The MV contains the data on previous available energies $a_i(t-1)$, harvested powers $g_i(t)$, and previous edge flows $f_k(t-1)$. Based on the information above, the leader node of a tie-set conducts the following procedure, which is written in Algorithm 1.

1) Initialization: Initialization step starts from line 1 to 7 in Algorithm 1. First, v_l^λ initializes the value of its tie-set flow as $x_\lambda(t) = 0$. As the data of $a_i(t-1)$ and $g_i(t)$ of V_λ have already been sent to v_l^λ by MV $y_\lambda(t)$ of TA, v_l^λ calculates $a_i(t) = a_i(t-1) + g_i(t)$ for each $v_i \in V_\lambda$. v_l^λ also obtains the data of $f_k(t-1)$ for each $e_k \in L_\lambda$.

2) Calculating Tie-set Flow: From line 8 to 33 in Algorithm 1, tie-set flow is calculated. For $e_k \in L_\lambda$, we define $r_{\lambda k}^j$, where $r_{\lambda k}^j = \{1, -1\}$. The set of $r_{\lambda k}^j$ is denoted as $R_j = \{r_{\lambda k}^j\}$. As $r_{\lambda k}^j$ is either -1 or 1, there are $2^{|L_\lambda|}$ combinations for R_j. The set of all the combinations of R_j is expressed as $\mathbf{R}_\lambda$ where $R_j \in \mathbf{R}_\lambda$. The $flag$ in this procedure is used to check if $R_j = \{r_{\lambda k}^j\}$ satisfies the following rule:

$$r_{\lambda k}^j = \begin{cases} 1 & \text{if } f_k(t-1) + b_{\lambda k} x_\lambda(t) \geq 0, \\ -1 & \text{if } f_k(t-1) + b_{\lambda k} x_\lambda(t) < 0. \end{cases} \tag{26}$$

If all the elements $\{r_{\lambda k}^j\}$ of R_j meet (26), then $flag = 1$, otherwise $flag = 0$. The $flag$ is initially set as 0. In the *while* sentence in Algorithm 1, the optimal $x_\lambda^*(t)$ with proper $c_k(t)$ is calculated. $x^*(t)$ is first set as 0, and then v_l^λ selects R_j from $\mathbf{R}_\lambda$. As the channel capacity $c_k(t)$ changes according to (3), v_l^λ sets $c_k(t) = a_i(t)/p_k$ if $r_{\lambda k}^j = 1$, otherwise $c_k(t) = a_j(t)/p_k$. Then, v_l^λ calculates the optimal tie-set flow $x_\lambda^*(t)$ according to (23). After calculating $x_\lambda^*(t)$, v_l^λ checks whether or not each $r_{\lambda k}^j$ satisfies (26). If all of $\{r_{\lambda k}^j\}$ satisfy (26) $flag = 1$, otherwise $flag = 0$. In case that $flag = 1$ after checking (26), the *while* sentence finishes by setting the tie-set flow at time t as $x_\lambda(t) = x_\lambda^*(t)$, otherwise it is repeated by removing R_j from $\mathbf{R}_\lambda$.

[1] An autonomous agent that constantly navigates a tie-set to bring the current state information of L_λ to its leader node.

[2] MV $y_\lambda(t)$ contains various information of node $v_i \in V_\lambda$ at time t such available energies, harvested powers, edge flows, etc.

Algorithm 1 Distributed Algorithm for FOP-EHWSN

1: Initialize tie-set flow $x_\lambda(t)$ of L_λ as 0.
2: **for** each $v_i \in V_\lambda$ **do**
3: Obtain $a_i(t) = a_i(t-1) + g_i(t)$.
4: **end for**
5: **for** each $e_k \in L_\lambda$ **do**
6: Obtain $f_k(t-1)$.
7: **end for**
8: Set $flag = 0$.
9: **while** $flag \neq 1$ **do**
10: Set $x_\lambda^*(t) = 0$.
11: Select $R_j = \{r_{\lambda k}^j\}$ from $\mathbf{R}_\lambda$.
12: **for** each $e_k(i,j) \in L_\lambda$ **do**
13: **if** $r_{\lambda k}^j = 1$ **then**
14: Set $c_k(t) = a_i(t)/p_k$.
15: **else**
16: Set $c_k(t) = a_j(t)/p_k$.
17: **end if**
18: **end for**
19: $x_\lambda^*(t) = -\frac{N_\lambda}{2M_\lambda}$.
20: Set $flag = 1$.
21: **for** each $e_k \in L_\lambda$ **do**
22: **if** $f_k(t-1) + b_{\lambda k} x_\lambda^*(t) \geq 0$ & $r_{\lambda k}^j = -1$ **then**
23: Set $flag = 0$.
24: **else if** $f_k(t-1) + b_{\lambda k} x_\lambda^*(t) < 0$ & $r_{\lambda k}^j = 1$ **then**
25: Set $flag = 0$.
26: **end if**
27: **end for**
28: **if** $flag = 1$ **then**
29: $x_\lambda(t) = x_\lambda^*(t)$.
30: **else**
31: Remove R_j from $\mathbf{R}_\lambda$.
32: **end if**
33: **end while**
34: **for** each $e_k \in L_\lambda$ **do**
35: Update edge flow $f_k(t) = f_k(t-1) + b_{\lambda k} x_\lambda(t)$.
36: **end for**

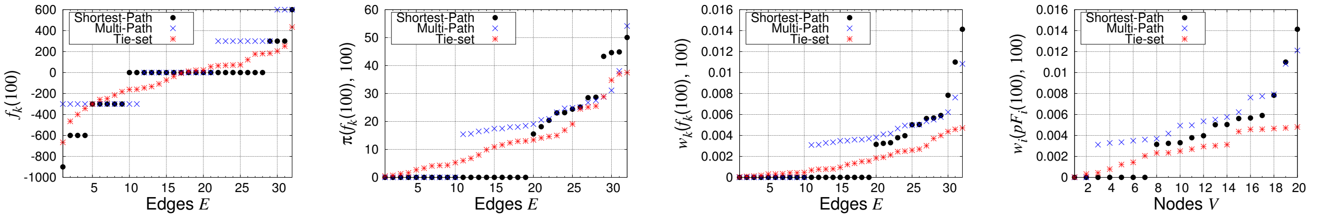
3) Updating Flows: After the value of the tie-set flow $x_\lambda(t)$ at time t has been decided, v_l^λ updates the value of each edge flow as $f_k(t) = f_k(t-1) + b_{\lambda k} x_\lambda(t)$ for each edge $e_k \in L_\lambda$ as in the steps 34 to 36 in Algorithm 1, and sends the information of the edge flows to the all the nodes V_λ in L_λ.

B. Tie-set Based Autonomous Distributed Control (TADiC)

A *Tie-set based Autonomous Distributed Control (TADiC)* is the method conducted in a leader node v_l^λ that realizes completely parallel optimizations among tie-sets, which has been proposed in [14], [8]. In TADiC, the leader node v_l^λ in each tie-set exchanges *Tie-set Evaluation Function (TEF)*[3] with adjacent tie-sets to decide process priority for overlapping resources at every time step. Here, *adjacent tie-sets* $\mathbf{L}(L_\lambda)$ of a tie-set L_λ are defined that if $L_\lambda \cap L_j \neq \emptyset$, L_j is an adjacent tie-set of L_λ. A leader node also has information of adjacent tie-sets and the routing table to the leader nodes of $\mathbf{L}(L_\lambda)$. The TEF in this paper is defined as

$$\Phi(L_\lambda, t) = \left|\frac{\partial \phi_\lambda(x_\lambda(t), t)}{\partial x_\lambda(t)}\right| = |2M_\lambda x_\lambda(t) + N_\lambda|. \tag{27}$$

[3] A function that evaluates a tie-set based on the current MV $y_\lambda(t)$ with certain predefined criteria.



(a) Edge Flow. (b) Energy Consumption. (c) Edge-Constrained Load Factor. (d) Node-Constrained Load Factor.

Fig. 3. The edge flow, energy consumption, and load factor on edges $e_k \in E$ and the load factor on nodes $v_i \in V$ by Shortest-path, multi-path, and Tie-set based algorithms at time $t = 100$ min.

TABLE II

COMPARISON OF EDGE-CONSTRAINED LOAD FACTOR, NODE-CONSTRAINED LOAD FACTOR, AND ENERGY CONSUMPTION AT $t = 100$ MIN

	Edge-Constrained Load Factor			Node-Constrained Load Factor			Energy Consumption		
	Total	Max	Min	Total	Max	Min	Total	Max	Min
Shortest-Path	0.0777	0.0141	0.0	0.0777	0.0141	0.0	388.31 mJ	50.06 mJ	0.0 mJ
Multi-Path	0.1037	0.0108	0.0	0.1037	0.0121	0.0	518.50 mJ	54.15 mJ	0.0 mJ
Tie-set	0.0529	0.0047	0.000044	0.0529	0.0048	0.00023	434.56 mJ	37.50 mJ	0.48 mJ

If TEF of a tie-set L_λ is the largest among those of adjacent tie-sets $\mathbf{L}(L_\lambda)$, then v_l^λ sets its *Tie-set Flag (TF)*[4] as $\zeta(L_\lambda) = 1$, otherwise $\zeta(L_\lambda) = 0$. When a tie-set L_λ gains process priority over the shared resources, v_l^λ executes DAFOP described in Algorithm 1. After conducting DAFOP, v_l^λ sets its TF as 0. Then v_l^λ stands by for Δt and iterates TADiC again.

V. SIMULATION AND EXPERIMENTS

We conducted simulations and experiments to testify the DAFOP with TADiC for solving the FOP-EHWSN problem as well as analyze the solution and its behavior. The following simulation conditions are based on the Qualnet parameters when implemented with all the distributed functions introduced in this paper.

The graph is given with $|V| = 20$ and $|E| = 32$ connecting links at random. The number of tie-sets is $\mu(G) = |E| - |V| + 1 = 13$ where the height of the tree is 5. We have multiple sources $|\mathcal{S}| = 5$ and single sink $|\mathcal{T}| = 1$. Packet energy p_k at each edge is randomly given between 0.05 mJ to 0.1 mJ. Each node has a function that produces renewable power $g_i(t)$ between 20 mW to 100 mW at random. Initial available energy at each node is set as $a_i(0) = 5000$ mJ. The size of the storage device of each node is $\overline{a}_i = 10000$ mJ. The net flow export rate at each source node is fixed as $F_i(t) \equiv 300$. Time Interval Δt of conducting TADiC at each tie-set is 1 second.

A. Experimental Results with Snapshot Flows

We first analyze the optimized edge flow $f_k(100)$, energy consumption $\pi(f_k(100), 100)$, edge load factor $w_k(f_k(100), 100)$ of every link, and the node load factor $w_i(pF_i(100), 100)$ of every node with the snapshot result at time $t = 100$ min.

When $t = 100$ min, the average TEF of all the tie-sets is $\frac{\sum_{L_\lambda \in \mathbf{L}_B} \left| \frac{\partial \phi_\lambda(x_\lambda(t), t)}{\partial x_\lambda(t)} \right|}{|\mathbf{L}_B|} = 1.34 \times 10^{-15}$, and the average value of tie-set flows is $\frac{\sum_{L_\lambda \in \mathbf{L}_B} |x_\lambda(t)|}{|\mathbf{L}_B|} = 9.93 \times 10^{-9}$, so that (24) and (25) are almost satisfied.

[4]When TF $\zeta(L_\lambda) = 0$, a tie-set L_λ is stand-by; otherwise L_λ is in process ($\zeta(L_\lambda) = 1$).

As shown in Fig. 3(a), since Tie-set Based Algorithm (TBA) constantly optimizes the edge flows with 1-second time interval to satisfy (24) and (25), all the flows are allocated in a balanced manner. With optimized edge flows at $t = 100$ min, energy consumptions and load factors of all the edges and nodes are also balanced as in Fig. 3(b) - 3(d) where the maximum value of those factors is minimized as in TABLE II.

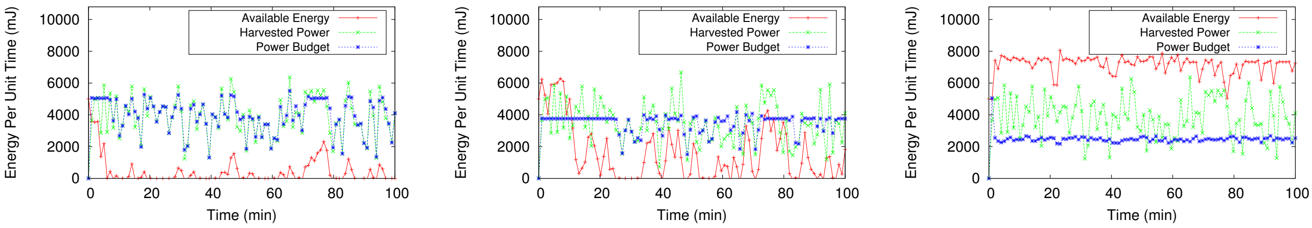
Shortest Path Algorithm (SPA) always allocates the flows on the minimum energy-cost paths from sources to a sink, where the summation of packet energies by routings is minimum. Therefore, the total energy consumption of all the packet energies by TBA increases with the optimized flows compared with SPA as in TABLE II. Although the total energy consumption by routings with shortest paths is assured to be minimum, flows are frequently concentrated on particular path(s) indicated in Fig. 3.

To reduce the load factors of nodes and edges, multiple-path algorithm (MPA) allocates the flows in different minimum cost paths with minimum overlaps of those paths (except the bottleneck around the sink node) as in Fig. 3. By using MPA, the maximum edge-load and node-load factor get slightly lightened than SPA whereas the total load factor increases. This result indicates that TBA with μ-dimensional optimization is a radical improvement for the network traffic congestion compared with MPA, whose concept has been applied to many techniques as in Multi-Protocol Label Switching (MPLS).

In the next section, we discuss that minimizing the maximum load factor is important in terms of realizing the assignment of energetically sustainable workload even thought the total consumption modestly increases.

B. Behavior of Available Energy at Node

This section analyzes the simulated behavior of the changing process of the available energy $a_i(t)$, harvested power $g_i(t)$, and power budget $pF_i(t)$ at a certain node from $t = 0$ to 100 (min) where SPA, MPA, and TBA are compared against each other. The same conditions as those of previous experiment are also adopted in this section. The simulation data are shown with 1 minute intervals where $\Gamma = \{0, 1, ..., 100\}$ (min).



(a) By Shortest-Path Algorithm (SPA). (b) By Multi-Path Algorithm (MPA). (c) By Tie-set Based Algorithm (TBA).

Fig. 4. The simulated behavior of available energy $a_i(t)$, harvested power $g_i(t)$, and power budget $pF_i(t)$ form $t = 0$ to 100 (min) at a node $v_i \in V$.

We pick up a node $v_i, (i = 18)$ that has the largest power budget. In Fig. 4(a) and 4(b), the available energy $a_i(t)$ frequently runs out because of the large power budget assigned to v_i by employing SPA and MPA. On the other hand, in Fig. 4(c), as the net flow export from v_i has been distributed to other peripheral nodes by TBA, v_i maintains useful storage life. Therefore, even though the total amount of energy consumptions modestly increases, it is important to balance the edge flows to realize the sustainable flow network that exploits intermittent renewables by energy harvesting systems.

VI. Conclusion

In this paper, we presented a novel distributed flow-based routing method that solves a flow optimization problem in energy-harvesting wireless sensor networks (EHWSNs) to realize the assignment of energetically sustainable workload on every node. We introduced a distributed algorithm for flow optimization problem (DAFOP) in EHWSNs where the packet flows within a tie-set are allocated so that heavily loaded power budgets are distributed to the nodes with lightly loaded power budges. DAFOP is repeated among the fundamental system of tie-sets with the scheme called Tie-set based Autonomous Distributed Control (TADiC) until the iteration of local optimization makes the entire edge flows optimized.

The experimental results at a certain point of time show that globally balanced assignment of link flows radically reduces the maximum load factor in EHWSNs. The result of comparison experiment at a particular node against the shortest-path and multi-path algorithms also suggest that the proposed method achieves sustainable allocation of packet flows by maintaining the reliable life of storage devices.

Appendix A
Analysis on Node-Constrained and Edge-Constrained Network Load Factors

Here, we provide the proof that the overall node-constrained network load factor and edge-constrained network load factor are the same as in (14). Let m be the number of edges $|E|$ of a graph G.

Proof: We first look at the node-constrained load factor $w_i(pF_i(t), t) = \frac{pF_i(t)}{a_i(t)}$ at node v_i. By the definition of $pF_i(t)$,

$$\frac{pF_i(t)}{a_i(t)} = \sum_{j:i\to j} \left(\frac{p_k f_k(i,j,t)}{a_i(t)} \right) - \sum_{h:h\to i} \left(\frac{p_k f_k(h,i,t)}{a_i(t)} \right).$$

As $f_k(h,i,t)$ is negative, $f_k(i,h,t)$ becomes positive. Since $f_k(i,j,k)$ and $f_k(i,h,t)$ are positive, the capacity of those flows is $c_k(t) = a_i(t)/p_k$ according to (3). Namely,

$$\frac{pF_i(t)}{a_i(t)} = \sum_{j:i\to j} \left(\frac{f_k(i,j,t)}{c_k(t)} \right) + \sum_{h:h\to i} \left(\frac{f_k(i,h,t)}{c_k(t)} \right).$$

By the definition of power budget, if edge flow $f_k(i,j,t) \geq 0$, the flow is included in $pF_i(t)$, otherwise it is included in $pF_j(t)$. Therefore, the following equation holds true:

$$\sum_{v_i \in V} \frac{pF_i(t)}{a_i(t)} = \sum_{e_k \in E} \frac{|f_k(t)|}{c_k(t)} = W_e(f_k(t), t).$$

References

[1] E. Lattanzi, E. Regini, A. Acquaviva, and A. Bogliolo, "Energetic sustainability of routing algorithms for energy-harvesting wireless sensor networks," *Computer Communications*, vol. 30, no. 1415, pp. 2976 – 2986, 2007.

[2] A. Bogliolo, E. Lattanzi, and A. Acquaviva, "Energetic sustainability of environmentally powered wireless sensor networks," in *ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks*, PE-WASUN'06, pp. 149–152, 2006.

[3] K.-W. Fan, Z. Zheng, and P. Sinha, "Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks," in *ACM Conference on Embedded Network Sensor Systems*, SenSys'08, pp. 239–252, 2008.

[4] V. Pryyma, L. Bölöni, and D. Turgut, "Uniform sensing protocol for autonomous rechargeable sensor networks," in *International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM'08, pp. 92–99, 2008.

[5] B. Zhang, R. Simon, and H. Aydin, "Maximum utility rate allocation for energy harvesting wireless sensor networks," in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM'11, pp. 7–16, 2011.

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *ACM conference on communications architectures, protocols and applications*, SIGCOMM'94, pp. 234–244, 1994.

[7] R. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *IEEE Wireless Communications and Networking Conference*, WCNC'02, pp. 350–355, 2002.

[8] K. Nakayama, C. Zhao, L. Bic, M. Dillencourt, and J. Brouwer, "Distributed real-time power flow control with renewable integration," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 516–521, 2013.

[9] K. Nakayama and T. Koide, "A decentralized algorithm for network flow optimization in mesh networks," in *IEEE Global Communications Conference (Globecom)*, pp. 1554–1559, 2013.

[10] K. Nakayama, K. Benson, V. Avagyan, M. Dillencourt, L. Bic, and N. Venkatasubramanian, "Tie-set based fault tolerance for autonomous recovery of double-link failures," in *IEEE Symposium on Computers and Communications (ISCC)*, 2013.

[11] N. Shinomiya, T. Koide, and H. Watanabe, "A theory of tie-set graph and its application to information network management," *Int. Journal of Circuit Theory and Applications*, vol. 29, no. 4, pp. 367–379, 2001.

[12] T. Koide, H. Kubo, and H. Watanabe, "A study on the tie-set graph theory and network flow optimization problems," *International Journal of Circuit Theory and Applications*, vol. 32, no. 6, pp. 447–470, 2004.

[13] M. Iri, I. Shirakawa, Y. Kajitani, and S. Shinoda, *Graph Theory with Exercises*. CORONA Pub: Japan, 1983.

[14] K. Nakayama, K. Benson, L. Bic, and M. Dillencourt, "Complete automation of future grid for optimal real-time distribution of renewables," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 418–423, 2012.