MASTER THESIS

Optimal Energy-Efficient Routing for Wireless Sensor Networks

蕭至偉
Chih-Wei Hsiao
R91044@im.ntu.edu.tw

Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.

Advisor：Dr. Frank Yeong-Sung Lin

July 2004

論文摘要

論文題目：無線感測網路之節能最佳化路由選徑

作者：蕭至偉

指導教授：林永松 博士

無線感測網路的存活時間受限於電池壽命、能源使用效率等因素，此外由於感測網路往往不具同步化的機制，封包重傳所消耗的電力不可忽視。本篇論文中，我們強調路由選徑演算法對於網路存活時間的影響力，同時考慮封包重傳因子，用數學模型闡述封包重傳次數的期望值，以一個應變於各節點累積流量的凸性函數表示之。我們應用最佳化技術規劃這個無線感測網路上的節能最佳化路由問題，雖然該問題具有最小最大化目標式、非線性等特性造成數學解題上的難度所在，應用拉格朗日鬆弛法可以求出該凸性規劃問題的最佳解，基於拉格朗日鬆弛法，本文中提出兩種演算法：分別最佳化解決拉格朗日鬆弛後的對應問題，以及直接處理原始規劃問題的最佳化演算法。兩種方法各有其效率上的優缺點，將於內文中分析之。透過上述兩種方法的合併使用，我們得出一組最佳的路由選徑值，使得感測網路的存活時間最大化。經過實驗，其他以最短路徑為基礎的演算法相對於我們的最佳化演算法，存活時間只有最佳解的百分之四十八，證明我們的方法於解題品質上的優越性。

關鍵詞：感測網路、能源效率、網路存活時間、最佳化、拉格朗日鬆弛法、最佳繞路框架

Thesis Abstract

Optimal Energy-Efficient Routing for Wireless Sensor Networks

By Chih-Wei Hsiao

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT

NATION TAIWAN UNIVERSITY

July 2004

ADVISER：Dr. Frank Yeong-Sung Lin

The network lifetime for wireless sensor network plays an important role to survivability. It is constraint to battery capacity and energy-efficiency. Besides, being lack of synchronization mechanism in sensor network, the retransmission for each packet is non-neglected. In this thesis, we indicate the importance of routing protocol to network lifetime, and model the expected retransmission time as a convex function with respect to aggregate flow on each sensor node. Thus we formulate the optimal energy-efficient routing as a non-linear min-max programming problem with convex product form, which can be optimally solved by optimal routing framework. Based on the optimal routing framework, we propose Lagrangean-based algorithm and primal optimal algorithm. By the combination of these two algorithms, we can optimally and efficiently get the routing assignment to maximize the network life in the sensor network. From experiments, we observe that when the optimal network lifetime increases as the number of sensor nodes increase. While the shortest path-based heuristic algorithm can only achieve about 48% network lifetime compared with our solution approach.

Keywords: Sensor network, Energy-efficient, Network lifetime, Optimization, Lagrangean relaxation, Optimal routing framework

Contents

List of Tables

List of Figures

Chapter 1   Introduction

The gravity of energy in sensor networks is much more important than in conventional wire networks. It is necessary to optimize the energy consumption on all layers of the protocol stack, from PHY layer up to application layer. For example, on PHY layer, by switching passive mode on and off moderately, we can save unnecessary radio power consumption and lengthen the network life time. On MAC layer, a smart scheduling mechanism takes traffic loading and collision probability into account with the result that it reduces the chance of retransmission. The energy consumption will not waste on the packet retransmission. On application layer, we can apply content-based data aggregation to avoid redundant and duplicate packets in the network.

In wireless sensor networks, the routing behavior is inevitably multi-hop forwarding. It means that a node in wireless sensor network consumes energy not only on its self traffic flow but also on passing other nodes' traffic flow. Give an extreme example, a node locating on the traffic artery may spend all its energy on forwarding. As more and more nodes use up their energy, the connectivity or radio coverage decrease and the network partition will happen finally. In this case, the network no longer offers services for the original purposes. Because forwarding traffic consumes considerable energy in wireless sensor network, the energy aware routing (EAR) protocol was presented to extend the network life time [14][16]. The EAR belongs to the class of on-demand routing, where the energy utility is contained in the routing information. This concept requires hardware support which is the capability of knowing the battery status, e.g., how many Watt the node still remain. In [20], this concept was enhanced by introducing "altruist", the node having surplus energy to forward traffic, into wireless sensor networks. Through properly exchanging battery status between =neighbors, the routing policy is energy efficient and the network life time extends significantly.

There are two policies respect to route discovery phase, which are on-demand and table-driven. The former is suitable for the network with high topology change rate; the later is suitable for the network which topology and traffic pattern are quasi-static. In the wireless sensor network, the topology changes because nodes use up their energy, enter passive mode, or the communication channel is suffered form interference and signal fading. Note that the sensor is without mobility and this is the main difference between sensor networks and ad-hoc networks. In this paper, we propose an energy efficient routing algorithm to maximize the network life time in sensor networks. So as to avoid consume too much energy on broadcasting route discovery packets if the routing protocol belongs to the on-demand policy, we choose table-driven routing policy and apply distance vector based algorithm, e.g. distributed Bellman-Ford (DBF) algorithm.

1.1 Related Work

Take the advantage of the asynchronous convergence property of DBF [2], we build a routing protocol implemented with distributed fashion which is indispensable in practice for sensor networks. Using distance vector based algorithm, the routing information exchanged by nodes must contain link cost, e.g. hop counts, and the next hop in the path to the specified destination. In sensor networks, traffic flow are always sourcing form sensor nodes and to the only one destination, and there is no point-to-point traffic flow existing. Since there are only a few O-D pairs needed to be recorded in each node’s routing table, we can compute several candidate routing paths for each O-D pair. In this case, the only one path is to the specified destination.

The benefit of multiple candidate routing paths was argued in [10]. The reasons to do so are to enhance servility and to reduce the computation time in the route discovery phase. We can apply the features of optimal routing [9] when selecting the candidate paths and scaling traffic flow between paths until the optimality conditions are satisfied. The optimality conditions of optimal routing are described in Chapter 3. These features of optimal routing help to build our distributed routing implementation for stationary and quasi-static networks.

To apply optimal routing features on DBF, we have to define the link length as sophisticated parameters which are capable of affecting network life time as in [11][13]. The factors on the network layer having direct impact over the network life time include routing policy and retransmission. Jointly considering both impacts, the routing policy should prefer the next hop which aggregate traffic load is lighter than which communication channel is always busy. There are two reasons supporting the routing policy mentioned above: First, to forward packet to the node with lighter load implies less retransmission time and avoids wasting energy on retransmission. Second, to route through non-congested path consists of the load balancing principle, which is usually a good character of most networks. And it disperses the traffic flow to all the candidate paths but not centralize the energy consumption on a few number nodes. Based the analysis of the expected retransmission time and the collision probability in [4], we model the expected retransmission time as a convex function of aggregate traffic load on the node then take it into the proposed routing algorithm. We will give more detail explanation in chapter 3.

In order to optimize energy consumption utility, the well-designed architectures on all layers, form PHY layer up to Application layer, jointly contribute to energy consumption utility. We indicate the fact that the node in wireless sensor networks often uses most of its energy to forward the others' traffic, but not its self traffic. It is due to the nature of multi-hop routing and can not be averted. That is the most important motivation of this thesis. We build the routing architecture on Network layer to fully utilize the energy consumption in wireless sensor network. The energy efficient routing not only can be applied to sensor networks, but also on the networks among which energy and battery are critical resources, e.g. ad-hoc networks. The proposed Network layer framework is composite by two components which are sensor deployment and energy efficient routing algorithm. The sensor deployment component is for topology determination [15][17]. Note that the sensor network topology is non-regular and usually randomly spread as [5] [3].

1.2 Research Scope

The research scope and the proposed protocol stack are shown on Figure 1. We focus on the discussion of energy routing algorithm that is composed of four kernel modules: performance assurance & optimization, network monitoring & traffic analysis, capacity management and network servicing. Give the definition and more detail in the following:

(1) Performance assurance & optimization module: to optimize the system current performance. Based on DBF, this module is responsible to adjust the network configuration and parameters, such like link weight, to achieve better energy utility and satisfy with the given constraints and capacity constraints.

(2) Network monitoring & traffic analysis module: to monitoring the traffic in the network at real time. This module continually detects the system status, such as traffic load on each node, congestion and collision in order to catch any exceptions as the triggers of other modules.

(3) Capacity management module: to expand or shrink the traffic capacity of each node. In wireless sensor networks, this function can be operated by passive mode management. We model the fraction of node life time as a parameter between 0-1, which represents the ratio of maximal capacity the node applies.

(4) Network servicing module: to make decision how many resources are needed to resume the network application economically when the sensor network is unable to service routines normally. In wireless sensor network, it is usually done by re-spreading sensor nodes.

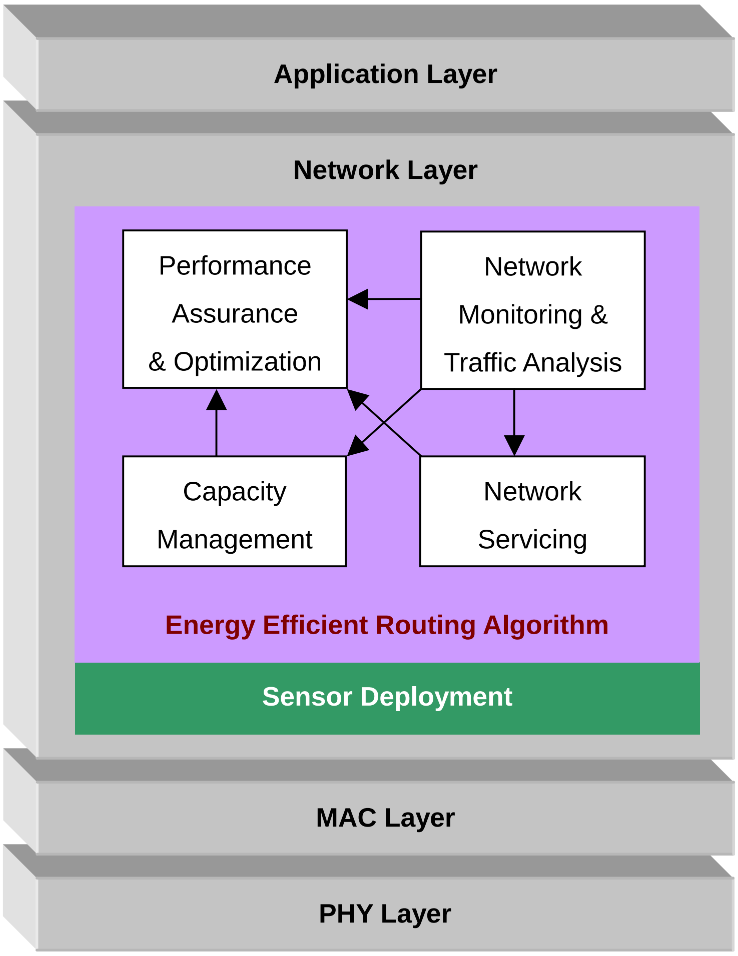

Figure 1: The protocol stack of wireless sensor networks

We formulate the energy efficient routing problem as a nonlinear optimization problem. To fulfill the timing and the quality of the optimal decisions, the solution approach to the mathematical problem is Lagrangean relaxation method. In the future computational experiments, our proposed routing algorithm is expected to be efficient and effective to deal with each complexity problems in this thesis.

1.3 Retransmission Model

In wireless sensor networks, interference is a significant effect on communication, which will affect bit-error rate and retransmission. We consider the influence of retransmission time on energy consumption and model the expected retransmission time as a function of traffic load. The optimal routing algorithm reacts with the retransmission time on each node, computes the corresponding energy consumption cost, and schedules the load-balancing routing assignment. Note that the retransmission model should be over-estimation.

We suggest the expected retransmission time is influenced by the aggregate flow on receiver in wireless communication, which is a convex function with respect to the routing assignment on the receiver node. To get an over-estimated retransmission model, consider the pure-aloha MAC formulation which can be taken as a performance lower-bound of those MAC layers in practice:

$$h = Ge^{-2G}$$

Notation S is the throughput of the transmitter node, defined as aggregate flow divided by wireless channel capacity. Notation G is the traffic load including retransmission for the transmitter node. Then the expected retransmission time R is:

$$R = E[retran\_time] = \sum_{k=1}^{\infty} kP_{success}(1 - P_{success})^{k-1} = e^{2G}$$

where $P_{success}$ is the probability that a transmission is successful. From the deduction in [18], the expected retransmission time in pure-Aloha system is e^{2G}. And what we need to know is the relation between throughput and expected retransmission time of the transmitter node. Give the deduction as following, first we apply Tyler expansion at $G=0$ (then $R=1$) on $f(R)=ln(R)$:

$$h = \frac{\ln R}{2R} = \frac{1}{2R}\left[F(1) + F'(1)(R-1) + \frac{1}{2!}F''(1)(R-1)^2\right]$$

$$= \frac{1}{2R}\left[\ln 1 + \frac{1}{1}(R-1) - \frac{1}{2}(R-1)^2\right] = -\frac{R}{4} + 1 - \frac{3}{4R}$$

After applying the quadratic equation formula, the expected retransmission time R

will be a function of throughput h:

$$R(h) = 2 - 2h - \sqrt{4h^2 - 8h + 1} \qquad \text{if} \quad h \le 0.133$$

$$R(h) = R(0.133) + R'(0.133)(h - 0.133) \qquad \text{if} \quad h > 0.133$$

Note that we only take the minus sign because the expected retransmission time is an increasing function with throughput S. Then this equation makes our retransmission function over-estimated which is inherited from pure-Aloha system and our algorithm will apply the result to achieve load-balancing routing assignment.

Chapter 2 Energy Efficient Routing

In this chapter, we intend to establish a model to discuss energy efficient routing problem for wireless sensor networks. We study how routing policy will influence the energy consumption which is a very critical and greatly restricted resource in wireless sensor networks. We develop a mathematical model to deal with the energy efficient routing problem in order to maximize the network life time in the system.

2.1 Problem Description

First we list the given system parameters as follows:

(1) Candidate paths of each O-D pair.

(2) The traffic arrival rate for each subscribed event.

(3) The capacity for each node.

(4) The energy configuration of initial level and consumption rate.

Given the wireless sensor network architecture mention above, we formulate the energy efficient routing problem as a complex nonlinear programming. The objective function in the program formulation is to maximize the network life time subject to:

(1) Bandwidth constraint.

(2) Call blocking probability constraint.

(3) Energy capacity constraint.

(4) Routing constraint.

Note that the routing constraint means that any traffic flow can route to the destination through active node as well as all the subscribed events are under the coverage of at least one active sensor. Before explaining the formulation of this routing problem, we show to flow chart in Figure 2 to describe how application layer provide services to system administrators then define the notations in Table 2.1, Table 2.2 and Table 2.3 as below.

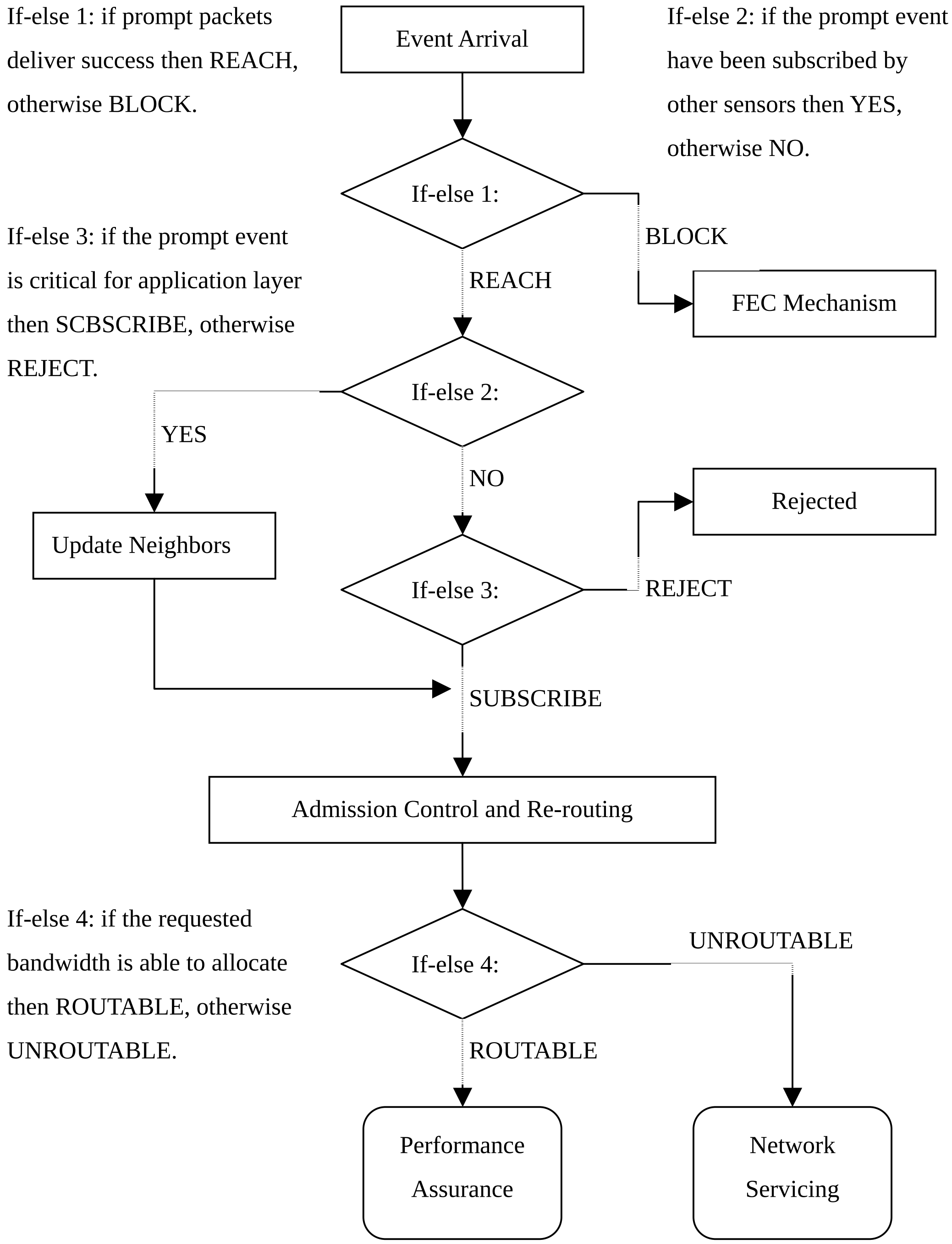

Figure 2: The flow chart describing how application layer provide services to system administrators.

All the sensor nodes in the network continuingly detect is there any critical events happen. Whenever sensor node discovers exceptional status and condition, it prompts a datagram packet [18], which records a short description to the events it detected, to route to the destination. This kind of prompt datagram packets route may reach destination or be blocked due to the node composite capacity is not enough, such as channel bandwidth capacity, energy capacity and contention capacity related to MAC layer. From the point of view to QOS, the network administrator has to guarantee the lower bound of call blocking probability of such prompt datagram packet.

If the prompt datagram packet is not blocked and reach the destination, the system will make decisions whether or not subscribed this event or not according to If-else 1-3 on Figure 2. If-else 1 indicates that a prompt datagram packet route may be success or be blocked. If-else 2 determine if the event requested have been requested by other sensor nodes. Note that the neighbors in sensor networks often observe the same or similar events because of their location. If-else 3 is dependent on application layer to make a decision if the event is critical enough to subscribe. A subscribed event is considered as a virtual circuit and is allocated fixed bandwidth to report the observed status periodically.

Table 1: Notation descriptions for given parameters

Given Parameters	
Notation	**Description**
N	The set of wireless sensor nodes;
V	The set of events subscribed by the application layer services in the sensor network;
W	The node set being capable of sensing subscribed events;
D	The only destination node in the sensor network;

R_i	The traffic flow source from subscribed event i, measured by kilobytes per second, assuming it is constant bit rate;
P_w	The candidate paths to the destination and origin from node w, $w \in W$;
E_n	The initial energy level of node n, measured by Watt;
e_n	The transmission energy required by node n to transmit an information unit, measured by Watt per kilobyte;
c_n	The capacity of traffic flow on node n, measured by kilobytes per second;
a_n	The energy required by node n to retain active mode, measured by Watt per second;
d_n	The energy required by node n to retain passive mode, measured by Watt per second;
$R(h_n)$	The expected retransmit time until success. It is a convex function related to the channel throughput of node n, measured by kilobytes per second. $R(h_n) = 2 - 2h_n - \sqrt{4h_n^2 - 8h_n + 1}$ if $h_n \le 0.133$ $R(h_n) = 1.6518 + 40.1924(h_n - 0.133)$ if $h_n > 0.133$
δ_{np}	Indicator function which is a 0-1 variable. If node n is in path p then set to 1, otherwise 0;

Table 2: Notation descriptions for decision variables

Decision variables	
Notation	**Description**
r_{iw}	The traffic flow source from subscribed event i and sensed by node w, measured by measured by kilobytes per second. $w \in W$;

f_{wp}	The traffic flow source from node w and route through path p, measured by measured by kilobytes per second per second;
g_n	Aggregate flow on node n;
h_n	Channel throughput of node n; $h_n = g_n / c_n$
q_n	The portion that node n is in passive mode of it self's node life time. It is in [0,1].
t_n	The time duration of node n to exhaust its energy;

2.2 Program Formulation

Objective function:

$$Z = \max \min_n t_n \qquad \text{(PB1)}$$

Subject to:

$$\sum_{w \in W} r_{iw} = R_i \qquad i \in V \qquad (1)$$

$$\sum_{p \in P_w} f_{wp} = \sum_{i \in V} r_{iw} \qquad \forall w \in W \qquad (2)$$

$$\sum_{w \in W} \sum_{p \in P_w} f_{wp} \delta_{np} = g_n \qquad \forall n \in N \qquad (3)$$

$$h_n = g_n / c_n \qquad \forall n \in N \qquad (4)$$

$$2g_n \le (1 - q_n) c_n \qquad \forall n \in N \qquad (5)$$

$$R(h_n) = 2 - 2h_n - \sqrt{4h_n^2 - 8h_n + 1} \qquad \forall n \in N, h_n \le 0.133 \qquad (6)$$

$$R(h_n) = 1.6518 + 40.1924(h_n - 0.133) \qquad \forall n \in N, h_n > 0.133 \qquad (7)$$

$$\frac{E_n}{g_n R(h_n) e_n + a_n (1 - q_n) + d_n q_n} = t_n \qquad \forall n \in N \qquad (8)$$

$$0 \le q_n \le 1 \qquad \forall n \in N \qquad (9)$$

$$f_{wp} \geq 0 \qquad \forall w \in W, p \in P_w \qquad (10)$$

$$f_{\overline{w}p} = 0 \qquad \forall \overline{w} \in \{N - W\}, p \in P_w \qquad (11)$$

$$r_{iw} \geq 0 \qquad \forall w \in W, i \in V \qquad (12)$$

$$r_{i\overline{w}} = 0 \qquad \forall \overline{w} \in \{N - W\}, i \in V \qquad (13)$$

The objective function is to maximize the network lifetime of the given wireless sensor network configuration. The network lifetime is related to the routing policy and passive mode management, which are the decision variables in our formulation. Constraint (1) ensures that the event-driven traffic can be fully dispatched to the corresponding sensors. Constraint (2) is the path-oriented routing requirement constraint. Constraint (3) calculates the aggregate flow on node *n*. Constraint (4) calculates the channel throughput according to the aggregate flow on node *n*. Constraint (5) is bandwidth constraint on wireless sensor nodes. Constraints (6) and (7) are both convex functions modeling the expected number of retransmission time related to the channel throughput of node *n*. Constraint (8) calculates the node lifetime concerning the aggregate traffic flow on node *n*, the energy consumption rate and the frequency that node *n* is in passive mode. Constraint (9) enforces the portion that node *n* is in passive mode of its node lifetime is between 0 and 1. Constraints (10)-(13) ensure the traffic flow are positive or zero.

Because node lifetime t_n must be positive, the original objective function can be rewritten as following:

$$Z = \max \min_n t_n = \min \max_n \frac{1}{t_n}$$
$$= \min \max_n \frac{g_n R(h_n) e_n + a_n(1 - q_n) + d_n q_n}{E_n}$$

Also, at the optimum, the passive mode must be fully utilize to achieve the best energy-efficient. Constraint (5) is active and $q_n = 1 - \frac{2g_n}{c_n}$. Thus an equivalent formulation of Problem (PB1) is:

Objective function:

$$Z = \min \max_n \frac{g_n R(h_n) e_n + (a_n - d_n)\frac{2g_n}{c_n} + d_n}{E_n} \qquad \text{(PB2)}$$

Subject to:

$$\sum_{w \in W} r_{iw} = R_i \qquad i \in V \qquad (14)$$

$$\sum_{p \in P_w} f_{wp} = \sum_{i \in V} r_{iw} \qquad \forall w \in W \qquad (15)$$

$$\sum_{w \in W} \sum_{p \in P_w} f_{wp} \delta_{np} = g_n \qquad \forall n \in N \qquad (16)$$

$$h_n = g_n / c_n \qquad \forall n \in N \qquad (17)$$

$$R(h_n) = 2 - 2h_n - \sqrt{4h_n^2 - 8h_n + 1} \qquad \forall n \in N, h_n \leq 0.133 \qquad (18)$$

$$R(h_n) = 1.6518 + 40.1924(h_n - 0.133) \qquad \forall n \in N, h_n > 0.133 \qquad (19)$$

$$f_{wp} \geq 0 \qquad \forall w \in W, p \in P_w \qquad (20)$$

$$f_{\overline{w}p} = 0 \qquad \forall \overline{w} \in \{N - W\}, p \in P_w \qquad (21)$$

$$r_{iw} \geq 0 \qquad \forall w \in W, i \in V \qquad (22)$$

$$r_{i\overline{w}} = 0 \qquad \forall \overline{w} \in \{N - W\}, i \in V \qquad (23)$$

This re-formulation eliminates Constraints (5), (8), and (9) as well as decision variables q_n by merging them into the objective function. Other Constraints are the same with Constraints (1)-(13) and the problem becomes a single decision variable programming problem.

2.3 Convex Programming Problem

To show our formulation is a convex programming problem, all the Constrains and the objective function in Problem (PB2) must be with convex forms. Except Constraint (20) and the objective function, the other Constraints are obviously with linear form which is convex. The following lemmas show *R(h)* and *g*R(h)* are both convex functions with respect to the decision variable *g*.

Lemma 2.1:

R(h) is a increasing and convex function when h is in [0,1].

Lemma 2.2:

*g*R(h) is a increasing and convex function when h is in [0,1].*

The following figures draw the curves of *R(h)* and *g*R(h)*. It is clear that their shapes are both increasing and convex.

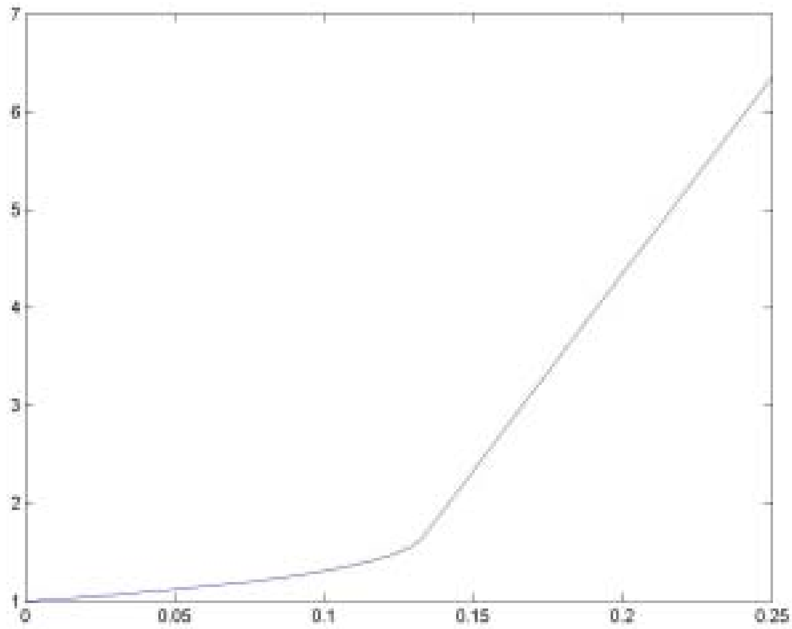

Figure 3: The graph plotting retransmissions function when throughput is in [0,0.25].

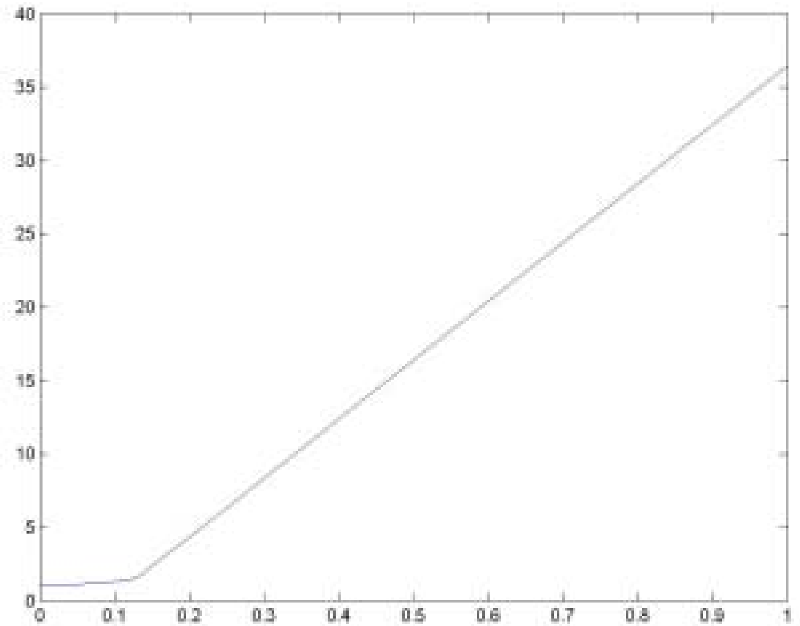

Figure 4: The graph plotting retransmissions function when throughput is in [0,1].

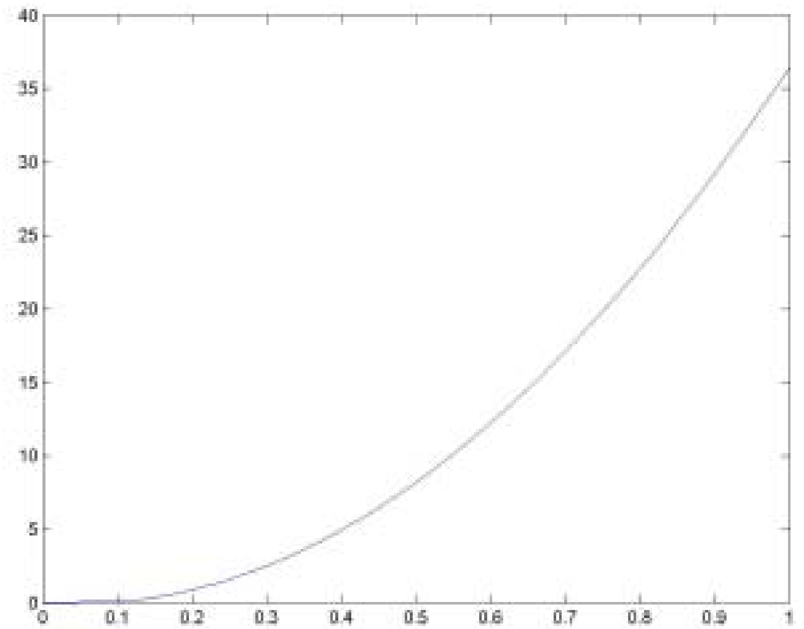

Figure 5: The graph plotting energy-consumption function when throughput is in [0,1].

Chapter 3 Solution Approach

The optimal energy-efficient routing problem (OEERP) in wireless sensor networks is a nonlinear programming problem with convex product form. We apply Lagrangean relaxation [8] to solve the optimal energy-efficient routing problem. By Lagrangean strong duality theorem, the tightest lower bound attained by Lagrangean dual problem is exactly the primal feasible objective function. In other words, because our problem is a convex programming problem, we could optimally solve the problem by Lagrangean relaxation method [6]. However, the decision variables which are routing assignment related to primal feasible solution do not guaranteed attainable by Lagrangean relaxation method. So we also conduct a primal algorithm to get the optimal routing assignment resulting to maximize network lifetime for the wireless sensor network.

3.1 Lagrangean Relaxation Methods

For large-scale network flow problems, Lagrangean relaxation method is a nice candidate mythology superior on efficiency and solution quality. Lagrangean relaxation (LR) based algorithm has been successfully adopted to solve many famous NP-complete problems [8]. We develop our LR-based algorithm and its subproblems in this chapter.

3.1.1 Lagrangean Subproblems

As a convention, we transform the maximization problem to minimization without loss of correctness.

Let $s = \max_{n \in N} \dfrac{g_n R(h_n) e_n + {}^{2g_n(a_n - d_n)}\!/_{c_n} + d_n}{E_n}$

An equivalent formulation of Problem (PB2) is:

Objective function:

$$Z = \min s \qquad \text{(PB3)}$$

Subject to:

Constraints (14)-(23)

$$s > 0 \qquad (24)$$

$$g_n R(h_n) e_n + {}^{2g_n(a_n - d_n)}\!\diagup_{c_n} + d_n \le sE_n \qquad \forall n \in N \qquad (25)$$

Constraint (24) ensures the equality with original problem (PB1). Constraint (25) defines the minimum node lifetime in the network. By using the Lagrangean relaxation method, the primal problem can be transformed into the following Lagrangean relaxation problem (LR) where Constraint (25) is relaxed. For a vector of non-negative Lagrangean multipliers, the Lagrangean relaxation problem is given by optimization problem (LR):

Objective function:

$$Z_{LR}(\alpha) = \min s + \sum_{n \in N} \alpha_n \left[g_n R(h_n) e_n + {}^{2g_n(a_n - d_n)}\!\diagup_{c_n} + d_n - sE_n \right]$$

Subject to:

$$s > 0 \qquad \text{(LR.1)}$$

$$\sum_{w \in W} r_{iw} = R_i \qquad i \in V \qquad \text{(LR.2)}$$

$$\sum_{p \in P_w} f_{wp} = \sum_{i \in V} r_{iw} \qquad \forall w \in W \qquad \text{(LR.3)}$$

$$\sum_{w \in W} \sum_{p \in P_w} f_{wp} \delta_{np} = g_n \qquad \forall n \in N \qquad \text{(LR.4)}$$

$$h_n = g_n / c_n \qquad \forall n \in N \qquad \text{(LR.5)}$$

$$R(h_n) = 2 - 2h_n - \sqrt{4h_n^2 - 8h_n + 1} \qquad \forall n \in N, h_n \le 0.133 \qquad \text{(LR.6)}$$

$$R(h_n) = 1.6518 + 40.1924(h_n - 0.133) \qquad \forall n \in N, h_n > 0.133 \qquad \text{(LR.7)}$$

$$f_{wp} \ge 0 \qquad \forall w \in W, p \in P_w \qquad \text{(LR.8)}$$

$$f_{\overline{w}p} = 0 \qquad \forall \overline{w} \in \{N - W\}, p \in P_w \qquad \text{(LR.9)}$$

$$r_{iw} \ge 0 \qquad \forall w \in W, i \in V \qquad \text{(LR.10)}$$

$$r_{i\overline{w}} = 0 \qquad \forall \overline{w} \in \{N - W\}, i \in V \qquad \text{(LR.11)}$$

In this formulation, α is the vector of { α_n}, which are Lagrangean multipliers and $\alpha_n \ge 0$. To solve this problem, we can decompose (LR) into the following two independent and solvable optimization Subproblems.

Subproblem (SUB1): related with decision variable s.

Objective function:

$$Z_{SB1}(\alpha) = \min(1 - \sum_{n \in N} \alpha_n E_n)s$$

Subject to:

$$0 < s \le \overline{s} \qquad \text{(SUB1.1)}$$

We add a restricted upper-bound of s in Constraint (SUB1.1) and it should not change the optimal solution value in (SUB3.1). The meaning of $\overline{s}$ is the upper bound above the reciprocal of node lifetime, equal to the lower bound on node lifetime. And in practice the network designer will not make the sensor network short-lived purposely, so we simply set $\overline{s} = 10.0$ in our algorithm. Then Subproblem (SUB1) becomes a bang-bang problem which is easily solvable.

Subproblem (SUB2): related with decision variables r_{iw} and f_{wp}.

Objective function:

$$Z_{SB2}(\alpha) = \min \sum_{n \in N} \alpha_n \left[g_n R(h_n) e_n + {}^{2g_n(a_n - d_n)}\!\big/_{c_n} + d_n \right]$$

Subject to:

$$\sum_{w \in W} r_{iw} = R_i \qquad i \in V \qquad \text{(SUB2.1)}$$

$$\sum_{p \in P_w} f_{wp} = \sum_{i \in V} r_{iw} \qquad \forall w \in W \qquad \text{(SUB2.2)}$$

$$\sum_{w \in W} \sum_{p \in P_w} f_{wp} \delta_{np} = g_n \qquad \forall n \in N \qquad \text{(SUB2.3)}$$

$$R(h_n) = 2 - 2h_n - \sqrt{4h_n^2 - 8h_n + 1} \qquad \forall n \in N, h_n \le 0.133 \qquad \text{(SUB2.4)}$$

$$R(h_n) = 1.6518 + 40.1924(h_n - 0.133) \qquad \forall n \in N, h_n > 0.133 \qquad \text{(SUB2.5)}$$

$$f_{wp} \ge 0 \qquad \forall w \in W, p \in P_w \qquad \text{(SUB2.6)}$$

$$f_{\overline{w}p} = 0 \qquad \forall \overline{w} \in \{N - W\}, p \in P_w \qquad \text{(SUB2.7)}$$

$$r_{iw} \ge 0 \qquad \forall w \in W, i \in V \qquad \text{(SUB2.8)}$$

$$r_{i\overline{w}} = 0 \qquad \forall \overline{w} \in \{N - W\}, i \in V \qquad \text{(SUB2.9)}$$

Constraints (SUB2.1) to (SUB2.9) are the same as Constraints (LR.2) to (LR.11). Problem (SUB2) is the bottleneck of all sub-problems. To apply standard nonlinear optimization techniques, we reformulate Problem (SUB2) as (SUB2-2) with single path-oriented decision variable x_{ip} modeling the routing assignment form subscribed event i to destination through path p. The objective function and constraints are as follow:

Sub-problem (SUB2-2): related with decision variables x_{ip}.

Objective function:

$$Z_{SB2-2}(\alpha) = \min \sum_{n \in N} \alpha_n \left[g_n R(h_n) e_n + {2g_n(a_n - d_n)}\big/{c_n} + d_n \right]$$

Subject to:

$$\sum_{p \in P} x_{ip} = R_i \qquad i \in V \qquad \text{(SUB2-2.1)}$$

$$\sum_{i \in V} \sum_{p \in P_w} x_{ip} \delta_{np} = g_n \qquad \forall n \in N \qquad \text{(SUB2-2.2)}$$

$$R(h_n) = 2 - 2h_n - \sqrt{4h_n^2 - 8h_n + 1} \qquad \forall n \in N, h_n \leq 0.133 \qquad \text{(SUB2-2.3)}$$

$$R(h_n) = 1.6518 + 40.1924(h_n - 0.133) \qquad \forall n \in N, h_n > 0.133 \qquad \text{(SUB2-2.4)}$$

$$x_{ip} \geq 0 \qquad \forall i \in V, p \in P \qquad \text{(SUB2-2.5)}$$

Problem (SUB2-2) is a minimum cost flow problem with a convex cost function and multi-commodities routing requirement. We solve this problem with optimal routing framework which is a variation of projection methods for convex cost routing problem [9]. Note that optimal routing framework is embedded with standard nonlinear programming methodology such as steepest descent and Newton's method. We will give detail description how we apply optimal routing framework in next sections.

3.1.2 The Dual Problem and the Subgradient Method

According to the weak Lagrangean duality theorem [1], for any $\alpha_i \geqq 0$, $Z_{LR}(\alpha_i)$ is a lower bound on Z_{IP}. The following dual problem (D) is then constructed to calculate the tightest lower bound.

$$Z_D = \max_{\alpha_i \geq 0} Z_{LR}(\alpha_i)$$

There are several methods to solve the dual problem, among which the subgradient method is the most popular and is employed here. Computational performance and theoretical convergence properties of the subgradient method are discussed in [12]. In this dual problem, let a vector χ be a subgradient of problem $Z_{LR}(\alpha_i)$. In iteration k of the subgradient optimization procedure, the multiplier vector π is updated by $\pi^{k+1} = \pi^k+\xi^k\chi^k$. The step size ξ^k is determined by $\zeta^k = \zeta \frac{Z^h - Z_D(\pi_k)}{\left\|\chi^k\right\|^2}$, where Z^h is the primal objective function value for a heuristic. It is an upper bound on Z_D.

3.2 Optimal Routing Framework

Optimal routing framework was first proposed in [2], it used to solve problems that minimize sum of total link delay or sum of total blocking rate. The superiority of optimal routing framework is capable of dealing with multi OD-pairs and convex cost function simultaneously. For Subproblem (SUB2) of optimal energy-efficient routing, we have to jointly consider the aggregate flow on each node came form multiple OD-pairs, just like the bundle constraint in multi-commodities network flow problem.

3.2.1 Features of Optimal Routing Framework

Although the primal objective function in optimal energy-efficient routing problem is not differential. The non- differential property is come from the behavior of “max-min node lifetime”. But after Lagrangean relaxation method, Subproblem (SUB2) becomes a differential convex programming problem which minimizes the sum of node lifetime weighed by Lagrangean multipliers. Then we can apply optimal routing framework to optimally solve Subproblem (SUB2) and get the effective objective value of the dual problem. Optimal routing framework is based on 2 lemmas:

Lemma 3.1:

Optimal path flow is positive only on paths with a minimum first derivative length.

Lemma 3.2:

At an optimum, the paths along which the input flow of each OD-pair is split must have equal first derivative length.

The physical meaning of first derivative length (FDL) is as the margin cost or utility in Economy. Consider two paths of the same OD-pair with positive flow both and different first derivative length: if we can shift a small amount flow form the path with larger first derivative length to another, the total cost will decrease. In words, at an optimum, only those paths with minimum and equal length have positive flow consisting with lemma 3.1 and 3.2. Optimal routing framework is an iteration-based algorithm. Iteration by iteration, we continuously adjust flow between paths of each OD-pair according to the first derivative length. Until the condition of Lemma 3.1 and Lemma 3.2 are satisfied, we get the necessary condition of optimality. It can also be shown to be sufficient for optimality if the cost functions are convex [2].

For Subproblem (SUB2), we apply optimal routing framework to solve this minimum sum of convex cost programming problem. We describe the solution procedure of optimal energy-efficient routing as follows:

1. Set the iteration counter k to be 1. Pre-calculate all the candidate paths of each OD-pairs and init any arbitrary one of feasible routing assignment set.
2. If k is greater than a pre-specified counter limit then stop.
3. Update the aggregate flow on each node in the network according to the current routing assignment.
4. Compute node-FDL and path-FDL according to the up-to-date aggregate flow.

5. Shift flow between paths according to path-FDL.
6. Increase k by 1 and go to Step 2.

We give detailed description of each procedure in the next sections.

3.2.2 Computing First Derivative Length

By definition, the cost function of Subproblem (SUB2-2) is:

$$\sum_{n \in N} \alpha_n \left[g_n R(h_n) e_n + \frac{2 g_n (a_n - d_n)}{c_n} + d_n \right]$$

It means that the object is to minimize the sum of reciprocal of node lifetime weighted by Lagrangean multipliers, and the FDL is the first derivation with path flow. Note that to change one unit of path flow equals to change one unit aggregate flow on nodes of the path, we can write first derivative length as following:

$$\sum_{n \in N} \alpha_n \left[R(h_n) e_n + g_n R'(h_n) \frac{dh_n}{dg_n} e_n + \frac{2(a_n - d_n)}{c_n} \right]$$

Given the valuation of first derivative length as mentioned above, at every iteration we compute each node’s lifetime according the current routing assignment, and recognize the bottleneck node in the network. Then update the first derivative length of all the paths. Finally we shift flow between paths based on the first derivative length computed earlier. Repeat these steps until algorithm converge.

3.2.3 Finding MFDL Path

The main idea of optimal routing framework is to shift flow form non-economic paths to economic ones which are the minimum first derivative length path. Theoretically we need to pre-calculate all the candidate paths of each OD-pair before running of optimal routing framework. The algorithm for finding all the candidate paths can be depth first search or branch first search. But it is clear that the time complexity is extreme high when number of nodes in the network work is high enough. In reality, the number of nodes in wireless sensor network is usually under hundreds or even thousands.

By the observation that there are many cases which can not be possibly the optimal routing paths. E.g. the path passes through the bottleneck node and is with longer number of hops than other paths. Thus we can only find the path with minimum first derivative length when running optimal routing framework according the up to date information and routing assignments. Our goal is to find the most economic path to shift positive flow iteration by iteration. By the lemmas of optimal routing framework, the most economic and energy-efficient path must with the minimum sum of node-FDL. Thus we directly apply Dijkstra shortest path algorithm to find the MFDL path for each OD-pair iteration by iteration, where the length computed for shortest path is the node-FDL.

3.2.4 Path Flow Adjustment

By the procedure mentioned above, we find the maximum and minimum FDL path for each OD-pair every iteration. We next shift positive amount of flow form the maximum FDL path to the minimum one. The determination of the positive amount of flow Δx must satisfy the following properties:

- The amount of flow Δx should be a *feasible direction* in the sense that small changes along Δx maintain the feasibility of the path flow vector x. Mathematically, the flow shift is feasible, which implies that

  $$\sum_{p \in P_w} \Delta x_p = r_w$$

  It simply expresses that all increases of flow along some paths must be compensated by corresponding decreases along other paths of the same OD-pair.

- The amount of flow Δx should be a *decent direction* in the sense that the cost function can be decreased by making small movements along the direction Δx starting form *x*. This implies flow shift should be from the maximum FDL paths to the minimum FDL paths.

Here that the difference between the maximum FDL and the minimum FDL affects on the amount of flow shift. To guarantee the flow shift is along a decent direction, we have to choose line search techniques for choosing the positive stepsize β and simultaneously satisfy the feasible direction. Their basic iteration is given by:

$$x := x + \beta \Delta x$$

In our algorithm, we apply Newton method on line search. Let β^* be the stepsize that minimizes D[x+β(x'-x)] over all β is between 0 and 1, that is,

$$D[x + \beta^* (x' - x)] = \min_{\beta \in [0,1]} D[x + \beta^* (x' - x)]$$

The procedure above is a special case of the so-called Frank-Wolfe method for solving nonlinear programming problems with convex constraint sets.

3.3 Primal Optimal Algorithm

One difficulty of optimal energy-efficient routing problem came form the behavior of min-max objection. To minimize the maximum reciprocal of node lifetime, only these bottleneck nodes will influence the objective function. And the aggregate flow trough non-bottleneck nodes do not affect the network lifetime, no matter what the node lifetime is quite long or just slightly lager than the bottleneck node. This feature makes the objective function non-differential. For example, consider if we shift a small amount of flow from the path through bottleneck node *b*, the node lifetime of node *b* increases. Now node *b* may not still the bottleneck node in the sensor network. For the respect to cost function, whenever the adjusting of path flow makes the bottleneck node change form *b* to *b'*, the first derivative length of the paths though node b switch form sensitive state to non-sensitive state and retain zero as if node b is not the bottleneck node.

The non-differential property causes the hardness of optimal energy-efficient routing problem, and we can not apply standard solution approach for convex programming problem to optimally solve our problem. However, by the observation that as long as we know the exact bottleneck node at optimum, which results to optimal maximum network lifetime, the optimal energy-efficient routing problem reduces to convex programming problem. And hence we can apply optimal routing framework to optimally solve the routing problem with convex cost function.

To decompose the original optimal energy-efficient routing problem into $|N|$ subproblems and each one represents of fixing node b as the bottleneck node, the formulation will be:

Objective function:

$$Z = \min t_b \qquad \text{(SUB)}$$

Subject to:

Constraints (15)-(25)

$$t_b \geq t_n \qquad \forall n \in N \qquad (28)$$

It is obvious that to solve $|N|$ subproblems and to apply optimal routing framework $|N|$ times is time-consuming. And the global information such as candidate paths for each OD-pair, the selected bottleneck node in the whole sensor network, is not easily collectable in a distributed environment. So we propose an primal optimal algorithm to cope with our problem. We expect our algorithm can optimally solve our routing problem and can be implemented in a distributed system.

The non-differential property is no longer existed when all candidate paths are adjacent or passing through the bottleneck node *b*. But in fact there must be some paths on which amount of flow are not active to the bottleneck node lifetime. So we artificially assign each path the local bottleneck node *b’* and compute MFDL according to *b’* then make the sensitivity of non-critical paths meaningful. To decide the virtual bottleneck node *b’*, we only have to record the node with minimum lifetime during shortest path computation.

The primal optimal algorithm is similar with that we use to solve Lagrangean Subproblem (SUB2). However, the procedures including computing path FDL phase, finding path phase, and flow adjustment phase are different. This difference is resulted from the min-max behavior and the non- differential property. We will give a detailed description of min-max Dijkstra algorithm in the next section.

3.3.1 Finding Minimum Capacity Cut Path

Because the objective function in our primal problem is a min-max objective function, but not traditional min-sum function, standard Dijkstra shortest path algorithm is not suit for finding MFDL path. In this section, we modify Dijkstra algorithm to find such a min-max shortest path and the pseudo code is as following:

Algorithm Single_Source_Minimum_Cut_Paths(*G,s,d*)

Input: G=(V,E) (a weighted directed graph), *s* (the source node), and *d* (the destination node).

Output: for destination node, *d*.SDC is the capacity of the minimum cut from *s* to *d*;

{ all capacities are assumed to be nonnegative. }

```
begin
    for all nodes w do
        w.mark := false;
        w.SDC := Infinite;
        s.SDC := 0;
```

```
      while the vertex d is unmarked do
          let w be an unmarked vertex such that w.SDC is minimal;
          w.mark := true;
          for all edges (w,z) such that z is unmarked do
              if   maximum(w.SDC, z.capacity) < z.SDC   then
                  SDC := maximum(w.SDC, z.capacity);
end
```

Figure 6: Single_Source_Minimum_Cut_Path algorithm

The complexity is as the standard Dijkstra algorithm which is O(|N|^2). By applying heap data structure on searching the minimum unmarked vertex the complexity reduce to O(|N|log|N|).

The Single_Source_Minimum_Cut_Paths (SSMCP) algorithm finds the path with minimum capacity cut form the given OD-pair. However, this algorithm only returns the minimum cut capacity value but does not record the path. This is because the algorithm tend to exhaustively search all the possible paths in the whole network, thus the returned minimum capacity cut path is usually with large number of hops. Considering the following two examples:

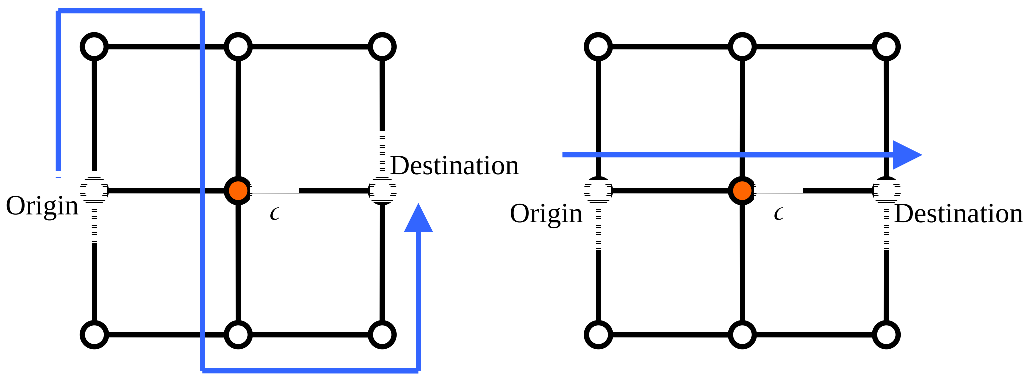

Figure 7: Example for two paths with the same capacity cut

The two paths are with the same minimum capacity cut which value is c. In general, we choose the right path with fewer hops as a better energy-efficient path than the left. But the SSMCP algorithm may still possible return the left path. This is because they have the same value form the view of minimum capacity cut. It is clear that there are several paths all with the same minimum cut value and our algorithm do not guarantee returning the shortest among them. So our algorithm does not record the pre-successors in the returned path. We only take the value of minimum capacity cut at this stage. Based on the output of SSMCP algorithm, we apply modified Dijkstra algorithm to find the shortest and minimum cut path. The following pseudo-code is for this purpose:

```
    Algorithm Capacity_Constraind_Shortest_Paths(G,s,d)
Input: G=(V,E) (a weighted directed graph), s (the source node), d (the destination node), and UC
        (the upper bound of constrained capacity).
Output: for destination node, d.SP is the length of the shortest path with the given minimum
        capacity cut from s to d;
      { all lengths are assumed to be nonnegative. }
begin
      for all vertices w do
            w.mark := false;
            w.SP := Infinite;
            s.SP := 0;
      while the node d is unmarked do
            let w be an unmarked node such that w.SP is minimal;
            w.mark := true;
            for all edges (w,z) such that z is unmarked do
                  if   w.SP+z.length < z.SP and z.capacity < UC   then
                        z.SP := w.SP+z.length;
end
```

Figure 8: Capacity_Constrained_Shortest_Paths algorithm

The given parameter UC is computed form SSMCP, which is the minimum capacity cut of OD-pair (s,d) in the network. The complexity is as the standard Dijkstra algorithm which is $O(|N|^2)$. By applying heap data structure on searching the minimum unmarked vertex the complexity reduce to $O(|N|\log|N|)$.

The Capacity_Constrained_Shortest_Path (CCSP) algorithm reroutes the minimum capacity cut path to a shortest path which capacity cut is still the same. Note that the length used for CCSP algorithm has many choices, which can be number of hops, energy consumption rate, reciprocal of residual battery capacity, or any other subject reacted with our sub-objective function. We recommend take the energy consumption rate as the length parameter of CCSP to achieve better average node lifetime in sensor network. In [19], the author proposed that min-max node lifetime objective function tend to find longer path resulting to decrease average node lifetime. Our algorithm contributes to keep balance between minimum node lifetime and average node lifetime in this stage. It also confirms multi objective functions optimization and the sub-objective function is very flexible to adjust according different application requirement.

3.3.2 Polya's Method on Path Flow Adjustment

We now propose an approach to adjust flow between each path with the same OD-pair. The basic idea is to adjust path flow according to the current bottleneck node lifetime of each path. More precisely, if the bottleneck node lifetime of path p is the minimum in the sensor network, then we decrease the fraction of path flow on p according to specific stepsize and add back the same amount of flow to another path with larger bottleneck node lifetime. It is clear that the bottleneck node lifetime of path p does not decrease when the amount of flow we shift is positive.

The step of flow adjustment procedure is given below:

1. Set the iteration counter k to be 1. Compute the path bottleneck lifetime according to any given feasible routing assignment.
2. If k is greater than a pre-specified counter limit then stop.
3. Find the paths with minimum and maximum bottleneck lifetime.
4. Select path with minimum bottleneck lifetime and denote its flow *f*. Shift the fraction of *f* by a positive stepsize t_f^k. More precisely, we shift flow with amount of $f * t_f^k$ form the path with minimum bottleneck lifetime to the path with minimum bottleneck lifetime.
5. Calculate the bottleneck node lifetime of each path.
6. Increase k by 1 and go to Step 2.

$f * t_f^k$ can be chosen by different ways. However, the following two properties, from Polya's method, of $\{t_f^k\}$ must be satisfied: (i) $\sum_{k=1}^{\infty} t_f^k \to \infty$ and (ii) $t_f^k \to 0$ as $k \to \infty$. The first property is meant to prevent the algorithm form being stalled, while the second property decreases the possibility of oscillation. If a sequence of t_f^k satisfies the first property, then each path flow *f* will be unbounded when $k \to \infty$. In our algorithm, we set $\{t_f^k\}$ as harmonizing so $t_f^k = \dfrac{1}{1+k}$. Since our problem is a convex programming problem, by applying Polya's method we can optimally solve it after sufficient large number of iterations.

Chapter 4   Experiment

In wireless sensor networks, sensor node deployment is a considerable issue. The deployment should achieve connectivity and coverage under certain budget. In practical, the sensor node deployment is usually by random. Only parameter we can control is the total number of sensor nodes. To deploy more sensors is with higher probability satisfying connectivity and coverage requirement, while it takes higher cost. It is obvious a trade-off between performance and cost. Thus the optimization technique of resource allocation plays an important role in this researching field.

4.1 Experiment Environments

Before all the experiments, we need know the minimum number of sensor nodes to assure the wireless sensor network is connected. In real world, the number of sensors deployed by network designers must larger than the minimum number. In this section, besides the description of the major assumptions and parameters, we first experiment this minimum number of sensors in Table 4.

4.1.1 Assumptions

The assumptions we make in this study are as follows:

1. In our model, sensors are concepts of location-based wireless nodes. The mobility of sensor nodes is ignored.
2. Traffic demand is initialed from the subscription of the application layer and we assume it is constant bit rate.
3. For the general propose, we do not apply any synchronize mechanism.
4. We derive the expected retransmission time from pure-Aloha system, which is the lower-bound on the expected retransmission time.

5. The factors affecting energy consumption are data communication, passive mode usage, and initial battery voltmeter.
6. The energy consumption on data communication is a convex function with respect to the aggregate flow.

To find the minimum number of sensor nodes achieving 1-connectivity, the experimentation variable δ is defined as the ratio between the edge lengths of grid area and the sensor's communication radius. The experiment result is given below. In wireless sensor networks, the communication radius is about 12.5 meters and when δ =8 the area size is 100╳100 meters.

Table 3: Minimum number of sensor nodes to achieve *1*-connectivity by different ratio δ
(δ = edge lengths of a grid / sensor's communication radius)

Ratio δ	4	8	12	16
Number of sensors	18	131	477	881

4.1.2 Parameters

We adopt the energy consumption parameter of EYES-nodes [7] in our study. For each sensor node, the parameters are as follows:

1. Sensing range and communication radius are both 12.5 meters.
2. Wireless channel capacity is 10 kbps.
3. Initial battery capacity of each sensor node is between 1300 and 1600 Watts.
4. Energy consumption rate on receiving (transmitting) is 0.2 Watts per byte.
5. Energy consumption rate to retain in active mode (passive mode) is 50 (10) Watts per second respectively.

4.2 Scenarios

In our computational experiments, we generate several system scenarios with different (1) average package length, (2) traffic demand, (3) area size ratio δ, and (4) sensor node density. Then we apply the primal optimal algorithm introduced in Chapter 3 to compute the maximum network lifetime.

To experiment (1) average package length, we set up two cases with different parameters. In Case 1, the area size ratio δ is 4. Here we set the number of nodes in Case 1 is 27, which is 1.5 times the minimum number of sensor nodes from Table4. And traffic demand is set as 5 which is 0.2 times the number of sensor nodes. In Case 2, the area size ratio δ is 8, and we set up parameters according to the same logic as in Case 1. The parameters in both Cases is list below, and the experiment results are given in Table 6 and Table 7.

Table 4: Test cases of average packet length experiment

Case	Size ratio δ	Sensor Nodes	Traffic Demand	Packet Length
1	4	27	5	100~600
2	8	196	40	100~600

Table 5: Experiment results for average packet length of Case 1

Packet Length	Network Lifetime	Running Time
100	14925	32
200	7089	31
300	4075	32
400	1653	32
500	844	33
600	516	33

Table 6: Experiment results for average packet length of Case 2

Packet Length	Network Lifetime	Running Time
100	665.9	1445
200	123.2	1669
300	50.4	1598
400	27.2	1892
500	17.0	1913
600	11.6	1870

To experiment (2) traffic demand, we set up two cases with different parameters. In Case 3, the area size ratio δ is 4. Here we set the number of nodes in Case 1 is 27, which is 1.5 times the minimum number of sensor nodes from Table4. And we fix up average packet length to 200 bytes. Finally, traffic demand is set as 0.1$|N|$ 0.2$|N|$, 0.3$|N|$, 0.4$|N|$, 0.5$|N|$, and 0.6$|N|$, where $|N|$ is the number of sensor nodes. So they are 3, 5, 8, 10, 13, and 16 respectively. In Case 2, the area size ratio δ is 8, and we set up parameters according to the same logic as in Case 1. The parameters in both Cases is list below, and the experiment results are given in Table 9 and Table 10.

Table 7: Test cases of average packet length experiment

Case	Size ratio δ	Sensor node	Traffic demand	Packet length
3	4	27	3~16	200
4	8	196	20~120	200

Table 8: Experiment results for traffic demand of Case 3

Traffic Demand	Network Lifetime	Running Time
3	14244	19
5	7089	31
8	3527	42
10	2053	50
13	1026	55
16	649	64

Table 9: Experiment results for traffic demand of Case 4

Traffic Demand	Network Lifetime	Running Time
20	863	1208
40	123.2	1669
60	38.2	2430
80	23.4	3622
100	13.4	3869
120	8.8	3906

To experiment (4) sensor node density, we set up two cases with different parameters. In Case 5, the area size ratio δ is 4. Here traffic demand is fix at 6 and average packet length is 200 bytes. The numbers of nodes are 1.5, 2.5, 3.5, and 4.5 times the minimum number of sensor nodes satisfying 1-connectivity. So they are 27, 45, 63, and 81 respectively. In Case 6, the area size ratio δ is 8, and we set up parameters according to the same logic as in Case 5. The parameters in both Cases is list below, and the experiment results are given in Table 12 and Table 13.

Table 10: Test cases of sensor node number experiment

Case	Size ratio δ	Sensor Nodes	Traffic Demand	Packet Length
5	4	27~81	5	200
6	8	196~591	40	200

Table 11: Experiment results for sensor node number of Case 5

Sensor Nodes	Network Lifetime	Running Time
27	23194	31
45	23901	33
63	28544	34
81	28714	37

Table 12: Experiment results for sensor node number of Case 6

Sensor Nodes	Network Lifetime	Running Time
196	123	1669
327	418	23303
458	1035	64983
591	1388	107236

4.3 Discussion

Here we note some specific experiment results from these tables above. First we evaluate the effect of the expected retransmission time function. Because the function is convex with respect to the aggregate flow on each sensor node, this indicates if we add number of OD-pairs or average traffic density, the network lifetime will degrade greatly. This statement is consistent with Table 5, 6, 8, and 9.

Second, we observe how the network lifetime is affected. It is clear that the network lifetime increase as the network connectivity increases. The factor affecting network connectivity in our experiment is the number of sensor nodes. In words, if the number of sensor nodes increases, it is with higher probability the network topology has higher connectivity. Thus the average network lifetime increase. We specifically plot the datum in Table 11 with the following figure.

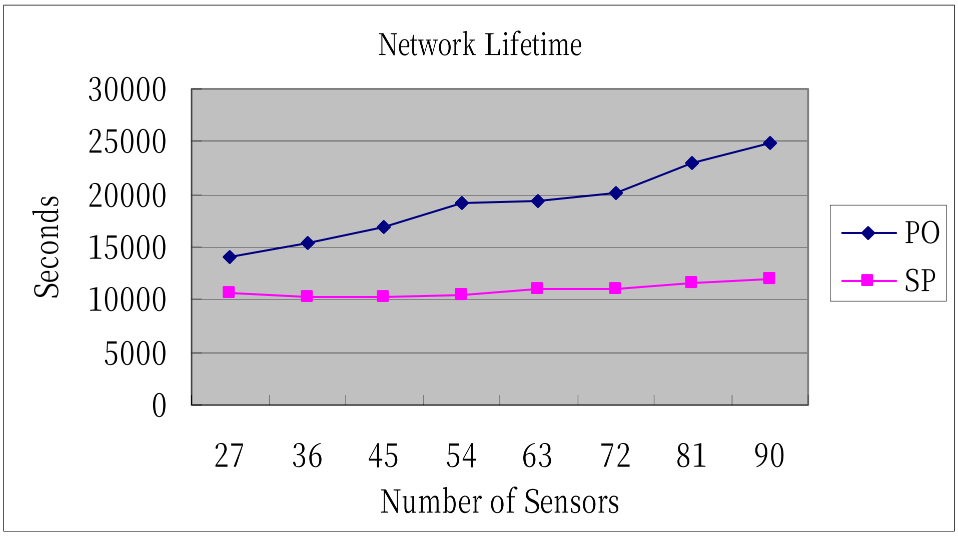

Figure 9: Comparing network lifetime between PO and SP

PO is our primal optimal algorithm while SP is a simple heuristic algorithm for comparison. In SP, we apply the reciprocal of node lifetime as length parameter in shortest path algorithm. And use Dijkstra to find the shortest path of each OD-pair in sequence. It is obvious that each traffic origin node routes the entire traffic requirement through only one path. Even though the path is the shortest path from the point of view with node lifetime and energy-efficiency, it is still fragile if any one bottleneck node exhaust it battery life. While our PO algorithm use multiple paths to route traffic between every OD-pairs, and the network lifetime increase as the number of sensor nodes increases.

Finally, to observe the trend of algorithm running time under different number of sensor nodes and OD-pairs, we plot the datum of Table 9 and 12 in following figures.

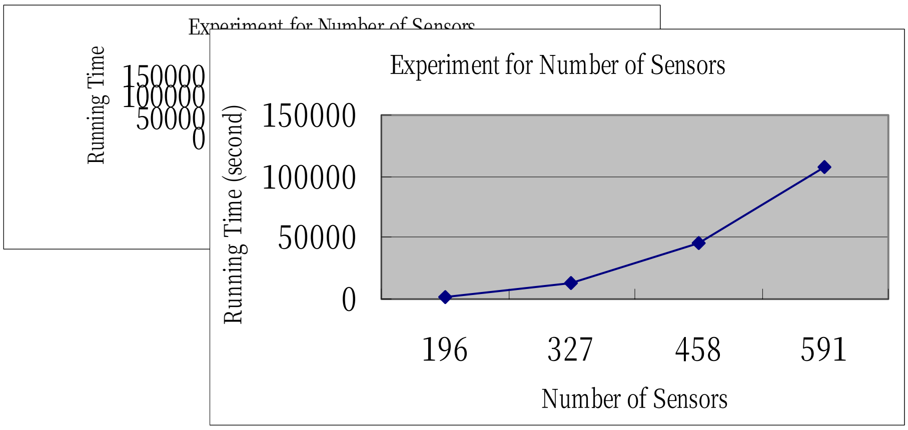

Figure 10: The graph plotting running time to number of sensors

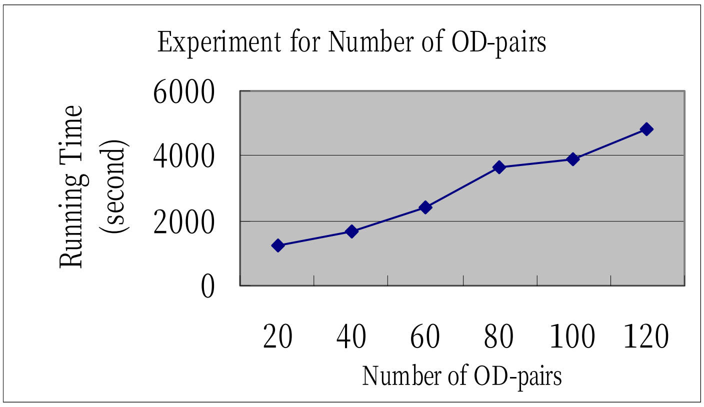

Figure 11: The graph plotting running time to number of OD-pairs

The results are consistent with the computation complexity we analyze in the next section. If the number of sensor nodes increases, the finding MFDL path procedure in every iteration takes longer time. From Figure 10, its trend is between $O(|N|)$ and $O(|N|^2)$. While from Figure 11, the computation overhead is portion to the number of OD-pairs, which trend is approximately linear.

4.4 Computational Complexity

We denote the number of sensor nodes, the number of OD-pairs, the number of candidate paths of each OD-pair as $|N|$, $|W|$, and $|P|$ respectively. The number of decision variable f_{wp} is equals to $|W|*|P|$. Note that $|P|$ is not simply a polynomial function of $|N|$. By experiment, we observe that $|P|$ is approximated to $|N|^{1.5}$. The numbers of decision variables, including traffic assignment f_{wp} and passive mode usage q_n, in our LR-based algorithm and in the primal optimal algorithm are both $|N|+|P|$. The required number of total Lagrangean multipliers is $|N|$. For the most extensive experiment of our testing network, the problem size is $|N|$=862, and $|W|$=40. Therefore, we can pre-calculate the total number of decision variables is 1,013,189.

To measure the time-complexity of our LR-based energy-efficient routing and the primal optimal algorithm, we first analyze each subproblem of the LR-based solution approach. Considering that Subproblem (SUB2) has similar structure with our primal optimal algorithm, first we analyze each procedure among the optimal routing framework embedded them. Then we analyze the getting primal feasible procedure and these Subproblems. For one iteration, the computational complexities of the optimal energy-efficient routing problem are listed in Table 14.

Table 13: The time complexity of optimal routing framework

Optimal Routing Frame Work		
Procedure	**Number of Operations Required**	**Time Complexity**
Updating node flow	$\|N\|+\|W\|\|P\|\|N\|$	$O(\|W\|\|P\|\|N\|)$
Finding MFDL path	$\|W\|\|N\|^2$	$O(\|W\|\|N\|^2)$
Computing path FDL	$\|W\|\|P\|\|N\|$	$O(\|W\|\|P\|\|N\|)$
Flow adjustment	$\|W\|*(\|P\|+1)$	$O(\|W\|\|P\|)$

As mentioned above, the number of candidate path $|P|$ is approximated to $|N|^{1.5}$. This is true for a highly connected network. At this situation, there are many candidate paths of the OD-pair existing. So the time complexity relies on the Computing path FDL procedure and the Updating node flow procedure. On the other hand, when the network is low connected, all the paths of any OD-pair must be through a few bottleneck nodes. This greatly decreases the number of candied paths $|P|$, and $|P|$ is considered as a constant number at this situation. So the time complexity relies on the Finding FDL path procedure.

Table 14: The time complexity of LR-based solution approach

LR-based Solution Approach	
Subproblem / Procedure	**Time Complexity**
Subproblem (SUB1)	$O(\|N\|)$
Subproblem (SUB2)	$\max(O(\|W\|\|P\|\|N\|),O(\|W\|\|N\|^2))$
Lagrangean dual problem	$O(\|N\|)$

Table 15 shows that the time complexity of our LR-based solution approach is dominated by Subproblem (SUB2). It is a minimum sum of convex cost flow problem which can be solved by the optimal routing framework. This framework is applied on our primal optimal algorithm too. Its time complexity for a highly or low connected network are $O(|W||P||N|)$ and $O(|W||N|^2)$ respectively.

All the experiments are performed on a PC with two 1.3 GHz CPUs and 2.0 GB DRAM. The operating system running on this computer is Linux Red Hat 9.0 with kernel version 2.4.20. The code is written in C language complied by GNU gcc version 3.2.2.

Chapter 5 Conclusion

In this study, we address the importance of routing protocol on energy efficiency, and contribute to:

- Model expected retransmission time under the channel throughput.
- Formulate OEERP as a convex programming problem and optimally solve it by our LR-based and PO algorithm.
- Extend optimal routing framework to attack nonlinear min-max routing problem.

5.1 Summary

First we apply the energy efficient routing protocol to extend the lifetime of wireless sensor networks. However, the sensor network is one class of distributed system. In lack of any synchronize mechanism, the retransmission time highly influence the energy consumption. We model the expected packet retransmission time as a convex function of aggregate flow, and apply this factor on our routing protocol.

Then we formulate the optimal energy-efficient routing as a nonlinear programming problem, and propose two algorithms: Lagrangean relaxation based algorithm can efficiently get the near-optimal solution; the primal algorithm can optimally solve the problem but spending much more time. Both of them are variations of optimal routing framework, which is to optimally solve the multi-commodities routing problem.

From the inspiration of optimal routing framework, iteration by iteration, we appraise the marginal cost of candidate paths by MFDL in LR-based algorithm and by capacity cut in primal optimal algorithm respectively. Using the quantity we adjust flow between paths for each OD-pair until the two optimally conditions are satisfied. Then we eventually achieve the optimal routing which is energy-efficient weighted load-balancing.

As in [7], we apply the EYES nodes as our experiment parameters. We generate several system scenarios with different (1) average package length, (2) number of OD-pairs, (3) area size, and (4) network density. The total time complexity is max($O(|W||P||N|), O(|W||N|^2)$).

5.2 Future Work

There still are several difficulties to be overcome. First we expect our routing algorithm will be implemented in distributed architecture. The mathematic model and formulation are used for performance evaluation and theoretic analysis. For this purpose, the decision variable and Lagrangean multipliers we used in mathematic formulation should combine with the parameters in distributed Bellman-Ford (DBF) algorithm. In DBF, each sensor nodes exchange routing information with their neighbors in asynchronized fashions. The routing information required by optimal energy-efficient routing is its MFDL and node lifetime. The MFDL computing procedure is described in Chapter 3 in detail. However, for simpler implementation, we also can get MFDL by measurement. So how to design an un-bias and easy-attainable measurement scheme is needed in the future.

The second issue is about distributed implementation too. For wireless sensor network, all the traffic eventually route to the only one destination. If we add some routing information as overhead appending onto packets, the destination will able to know the whole topology of this network. Sensor network is not fully distributed from this point of view. We should evaluate the overhead which is to discover the sensor network topology by simulation.

Finally, practicality the running of our algorithm must be smaller than the network lifetime. But we note that when the number of sensor nodes is huge, this statement is false. This is because the convergence rate in primal optimal algorithm is slow as mentioned in Chapter 4. To overcome this critical problem, Subgradient method is a good candidate solution. Though the objective function in primal problem is non-differential, we still apply the value of Subgradient for every path to adjust flow. It achieves faster convergence rate but loss the necessary condition which guarantees the flow adjustment is on decent directions. The outcome of importing Subgradient method is needed to further evaluate.

References

[1] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, “Network Flows: Theory, Algorithms, and Applications,” *Prentice Hall*, 1993.b

[2] D. Bersekas and R. Gallager, “Data Networks – second edition,” pp.404, *Prentice-Hall*, 1992.

[3] E. Biagioni and K. Bridges, “The Application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species,” *Special issue on Distributed Sensor Networks for the International Journal of High Performance Computing Applications*, vol.16, no.3, 2002.

[4] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE J. Select. Areas Comm.*, vol.18, pp.318-320, 2000.

[5] V. Bychkovskiy, S. Megerian, D. Estrin and M. Potkonjak, “A Collaborative Approach to In-Place Sensor Calibration,” *Proc. of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03), vol. 2634 of Lecture Notes in Computer Science*, pp.301-316, 2003.

[6] R. L. Cruz, and A. V. Santhanam, “Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks”, *IEEE INFOCOM* 2003.

[7] T. V. Dam, and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks”, *The First ACM Conference on Embedded Networked Sensor Systems*, November, 2003.

[8] M. L. Fisher, “The Lagrangian Relaxation Method for Solving Integer Programming Problems,” *Management Science*, vol.27, no.1, pp.1-18, 1981.

[9] R. Gallager, “A Minimum Delay Routing Algorithm Using Distributed Computation,” *IEEE Trans. Comm.*, vol.25, pp.73-84, 1977.

[10] J. J. Garcia-Luna-Aceves, S. Vutukury and W. T. Zaumen, “A Practical Approach to Minimizing Delays in Internet Routing,” *IEEE ICC'99*, 1999.

[11] A. Hac, “Wireless Sensor Network Designs”, *John Wiley & Sons, Ltd*. 2003.

[12] M. Held, P. Wolfe and H. P. Crowder, “Validation of Subgradient Optimization,” *Math. Programming*, vol.6, pp.62-88, 1974.

[13] R. Kannan, S. Sarangi, S. S. Iyengar and L. Ray, “Sensor-Centric Quality of Routing in Sensor Networks,” *INFOCOM* 2003.

[14] C. E. Perkins, “Ad Hoc Networking”, *Addison-Wesley*. 2000.

[15] C. Schurgers, V. Tsiatsis, S. Ganeriwal and M. Srivastava, “Optimizing Sensor Networks in the Enegy-Latency-Density Design Space,” *IEEE Transactions on Mobile Computing*, vol.1, no.1, 2002.

[16] R. C. Shah and J. M. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks,” *Proc. IEEE Wireless Comm. and networking Conference (WCNC)*, 2002.

[17] S. Shakkottai, R. Srikant and N. Shroff, “Unreliable Sensor Grids: Coverage, Connectivity and Diameter,” *IEEE INFOCOM* 2003.

[18] A. S. Tanenbaum, "Computer Networks", Fourth Edition, *Prentice-Hall*, 2002

[19] C. K. To, “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad hoc Networks”, *IEEE Communications Magazine*, June 2001.

[20] A. Willig, R. Shah, J. Rabaey and A. Wolisz, “Altruists in the PicoRadio Sensor Network,” *International Workshop on Factory Comm. Systems (WFCS)* 2002.

[21] G. Zussman and A. Segall, “Energy Efficient Routing in Ad Hoc Disaster Recovery Networks,” *Proc. IEEE INFOCOM* 2003.