國立交通大學

網路工程研究所

碩士論文

考量能源效益與傳輸量之無線感測網路中繼點佈建方式

On Energy-Efficient Traffic-Aware Deployment of a Wireless Sensor Network

研究生：高志偉

指導教授：曾煜棋　教授

中華民國 九十九 年六月

On Energy-Efficient Traffic-Aware Deployment of a Wireless Sensor Network

Student: **Chi-Wai Kou**

Advisor: **Prof. Yu-Chee Tseng**

Department of Computer Science

National Chiao-Tung University

1001 Ta Hsueh Road, Hsinchu, Taiwan 300, R.O.C.

考量能源效益與傳輸量之無線感測網路中繼點佈建方式

學生：高志偉　　　　　　　　　　指導教授：曾煜棋教授

國立交通大學網路工程研究所碩士班

摘　　要

在無線感測器網路中，由於感測器是依靠電池的電力維持的，因此，怎樣去減少耗電而延長無線感測器網路的壽命成爲很多人研究的議題之一。而在眾多延長網路壽命的研究中，有一種方法叫做“中繼點佈建”，我們可以透過佈建中繼點來減少感測器之間的通訊範圍，而因爲感測器的耗電與它的通訊範圍成正比，由此，我們可以使用中繼點來延長網路的壽命。而在早期的中繼點佈建研究中，大部份的研究人員都關注在研究於同種類或不同種類的感測器網路上，他們怎樣才能用最少數量的中繼點來使得網路達到連結性和存活性的需求。除此之外，某些研究人員開始察覺得當他們要在網路上佈建中繼點的時候，他們應該要同時考慮到網路流量的大小。而在某一篇考慮中繼點佈建的研究中，他們在佈建中繼點的時候，他們會同時考慮到網路流量的大小，並且顧及到網路的連結性而達到眞實網路的考量。可是，我們發現這篇研究當中存在某些缺點，這是因爲他們的網路拓樸是由Steiner tree建成的，而當中的Steiner tree是一種方法可保證用最短的距離把整個網路連起來。

因此，在這篇研究裏，首先，我們會介紹爲什麼建Steiner tree作爲有考慮流量大小的網路拓樸會產生問題。接下來，我們會針對在考慮網路流量大小的網路拓樸中進行中繼點佈建的這個問題上，提出一個“權重移動演算法”來解決這個問題，而這個演算法我們是透過非線性規劃(NLP)來處理。可是，因爲非線性規劃不能保證程式可以在多項式時間下結束，所以我們另外提出了兩個啓發式方法來處理這個佈建的問題。最後，

在我們的模擬結果中，我們會呈現我們提出的方法以及以前在中繼點佈建中提出的方法的數據。當中，我們察覺到我們的方法一定比以前的方法來得省電，因此網路的壽命也相對的比較長，而且我們的方法的改進效果在不同的情況下都可以維持在一個很好的範圍內。

關鍵字：省電、中繼點佈建、考慮流量、無線通訊與無線感測器網路。

On Energy-Efficient Traffic-Aware Deployment of a Wireless Sensor Network

Student: Chi-Wai Kou Advisor: Prof. Yu-Chee Tseng

Department of Computer Science

National Chiao Tung University

ABSTRACT

There are many different kinds of studies discussing how to achieve extending lifetime on sensor network. Among them, relay nodes deployment is a way to enlarge lifetime by reducing the communication range of sensor nodes. And in the early studies, they are usually concerned with placing minimum number of relay nodes into homogeneous or heterogeneous WSNs to meet certain connectivity and survivability requirements. However, some researchers start to consider the traffic volume in the network when they deploy relay nodes. And there is a study which the relay node deployment is related to the traffic volume and ensures the network connectivity. But, we have found some drawbacks on it since its network topology is based on Steiner tree, which can guarantee to connect the network with the minimum distance.

In this paper, we will first explain the problems about constructing Steiner tree as traffic-aware network topology. Then, we will propose a weighted moving algorithm using non-linear programming (NLP) to solve this traffic-aware relay node deployment problem. Since NLP can not guarantee to terminate in polynomial time, we also propose two heuristic approaches to solve this traffic-aware relay node deployment problem. In our simulation results, we have shown that both of our approaches must consume less energy than previous studies and obtain a pretty good improvement.

Keywords: energy efficient, relay node deployment, traffic-aware, wireless communication, wireless sensor networks.

誌謝

首先，誠摯的感謝曾煜棋教授對於我碩士生涯兩年來的指導與鼓勵，並且提供我優良的研究環境以及充足的實驗器材，讓我能夠順利的完成此篇論文並且順利取得碩士學位。同時也要感謝論文口試委員陳文村教授、張瑞雄教授、鍾葉青教授以及張西亞，在口試時給予我寶貴的意見，讓這篇論文變得更加完整。

此外，我亦要衷心的感謝葉倫武學長以及林政寬學長，感謝兩位學長在這篇論文研究上提供了不少的寶貴建議與指導。而當中，我要特別感謝葉倫武學長。因爲，在研究的過程中，當我遇到什麼阻礙難題的時候，他總是不厭其煩的協助我解決問題，並且在論文的寫稿中，葉學長細心地糾正我與指導我。另外，我也要感謝HSCC實驗室的全體成員，讓我可以在愉快的氣氛下進行研究。

最後，感謝我的父母以及我的女朋友對我無微不致的關懷，且在我做任何決定時給予的支持與鼓勵，使我能夠無虞地完成我的學業。

高志偉　於
國立交通大學網路工程研究所碩士班
中華民國九十九年六月

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid progress of wireless communication and embedded micro-sensing MEMS technologies has made wireless sensor networks (WSNs) possible. A WSN usually needs to configure itself automatically and support ad hoc routing. A lot of research works have been dedicated to WSNs, including power management [12], routing [3], sensor deployment and coverage issues. In WSNs, each sensor node has the capability to sense the environment (e.g., temperature, pressure, light, humidity, etc.) and process the data. And in recently, more and more researchers adopt WSNs to create a smart home, such as, intelligent lighting devices [13][14]. In general, WSNs have an ad hoc topology, and each node is capable of relaying the data toward the sink. Since most of the sensor nodes are battery-constrained, one of the design objective is to prolong the network lifetime. There is a lot of research studying how to extend the network lifetime, such as data aggregation, duty-cycle scheduling, energy-aware routing, relay node deployment, etc. And there are various ways to define the lifetime of a WSN. It can be defined as the time at which the first node runs out of energy. This time is equivalent to the time at which the first routing path is disconnected. It can be defined as the time at which a region within the WSN is not covered by any nodes.

In relay node deployment, the researchers have proposed to deploy lots of relay nodes which main function is to communicate with sensor nodes, other relay nodes and the sink node in a

WSN. And these studies can prolong network lifetime while meeting the certain network specifications. Most of the relay nodes deployment problem are focus on maintaining the network connectivity and survivability [6][8][9][11][15]. In connected relay node deployment, we place a minimum number of relay nodes to ensure that sensor nodes and the sink node are connected. In survivable relay node deployment, we place a minimum number of relay nodes to ensure that sensor nodes and the sink nodes are bi-connected. In the previous studies[6][8][9][11][15], they only consider the connectivity of the network but not traffic flows in a WSN. However, in relay node deployment, the network lifetime is limited by the relay node battery power. And this battery power consumption closely depends on the communication distance and traffic volume. So, [10] has brought up an idea that the location of the relay node assigned to the network should not only consider the distances but also the traffic volume. Unfortunately, the network topology of [10] is based on Steiner tree, which is the optimal solution for 1-connected relay node deployment. However, Steiner tree only considers the total length of entire network but not traffic rate in the network. Inspired by this situation, we believe that we can modify the topology of Steiner tree to find out some proper locations for Steiner points which retain connectivity and consider traffic rate as well. After constructing the new network topology, we can deploy the relay nodes on each edge depended on its traffic volume to get a better result than previous works. In Fig. 1.1, there are N relay node will be deployed. We consider two sensor nodes s_1 and s_2 with traffic rate 0.6 and 0.3, respectively, and the sink node s_0 which can collect the data from s_1 and s_2. In Fig. 1.1(a) only considers the connectivity for deployment. We can observe that the traffic distance from s_1 to v, s_2 to v and v to s_0 is less than the distance which the sensor nodes direct connected to the sink. Fig. 1.1(b) is the result of [10] which considers traffic-aware for deployment. It means that we can move some relay nodes from less traffic intensive section edge s_2, v to edge v, s_0 for achieving better performance. But, in Fig. 1.1(c), we consider both

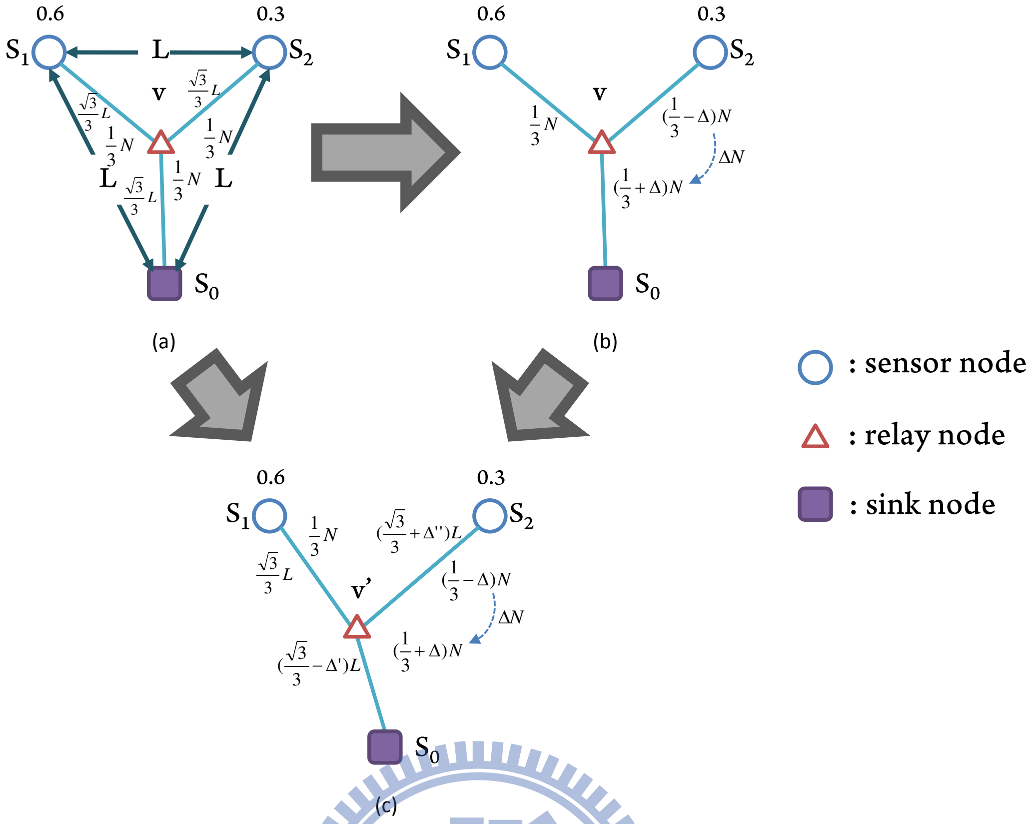


Figure 1.1: An example of relay node deployment: (a) connectivity (b) traffic-aware (c) connectivity and traffic rate.

connectivity and traffic rate for deployment. By the numerical analysis, we can observe that the topology of Fig. 1.1(c) can get better performance than Fig. 1.1(b) and Fig. 1.1(a) because the location of intersection v' is involved with the traffic volume.

In this thesis, we will propose one non-linear programming(NLP) algorithm and two heuristic approaches to solve the relay node deployment problem which must be considered network connectivity and traffic volume of the network. In NLP, we will find all proper locations for Steiner points at the same time. It considers the entire network topology at the same time so its complexity will be higher if the original network topology is complicated. Next, since the complexity of NLP may be high, we propose two heuristic approaches that is low complexity with acceptable performance. In heuristic approaches, we will divide the entire network into several blocks and solve them separately.

The rest of this thesis is organized as follows. In Section II, we will discuss the related works about relay node deployment. In Section III, we define the problem definition of the relay node deployment. In Section IV, we describe our relay node deployment algorithms. Simulation results are presented in Section V. Conclusions are given in Section VI.

Chapter 2

Related Works

In WSNs, the mainly energy consumption of a sensor node is wireless communication, which is proportional to the data rate and the communication distance. Since the communication distance can be adjustable, lots of research has been studied this property to achieve topology control with given node deployment, such as power-aware routing, relay node deployment, etc.

Relay node deployment for WSNs have been studied in various contexts. In [7], it was first shown that the connectivity for relay node deployment is NP-hard and it proposed a minimum spanning tree (MST) based 5-approximation algorithm. Later, the authors[4] also proposed a 3-approximation algorithm for this problem. In relay nodes deployment for WSNs, most of the studies were focused on the connectivity of the network. It meant that they had to guarantee that the given relay nodes should be connected to all sensor nodes and the sink node after their deployment. In [15], they not only considered the connectivity of the network in relay node deployment, but also proposed to create a network with fault-tolerant, which is k-connectivity network, by relay nodes. The connectivity problem had developed from 1-connected to k-connected. After that, the work[8] extended the fault-tolerant problem by considering that the relay nodes can only be placed at some given locations and considered the reality and the application requirement. Moreover, the work[6] provided fault-tolerance with higher network connectivity in heterogeneous wireless sensor networks, where sensor nodes posses different transmission radii. Work[9] not only considered deploying relay nodes into the network to mitigate network geometric deficiency and prolong network lifetime, but also provisioned additional energy on the existing nodes for a two-tier wireless sensor network. The deployment studies which we mentioned before were placed by their algorithm. But in [11], they developed three random deployment strategies for relay nodes in a heterogeneous WSN by impact of random device deployment on system lifetime not stressed enough.

But the strategy of the deployment in previous studies is evenly deployed relay nodes according to the length of the distance. So, in [10] has shown us that the strategy of the relay node deployment should be considered the traffic flows, and its performance is better than others. Unfortunately, the network topology of [10] is a Steiner tree, and Steiner tree only considers the total length of entire network but not traffic rate in the network. Hence, in our work through considering the traffic volume, we will move the Steiner points to some proper locations to satisfy minimum energy consumption, and we can ensure the new network connectivity as well.

Chapter 3

Problem Definition

We consider a wireless sensor network with three types of nodes: sensor node, relay node, and sink node. A sensor node is a device which can report environment data (such as temperature, humidity, and light intensity), but has no communication capability. A relay node is a device which has only communication capability, but no sensing capability. A sink node is a special relay node which is designed to collect all sensor nodes' data. We assume that there are one sink node, denoted by s_0, M sensor nodes, denoted by $s_1, s_2, \ldots, s_M$, and N relay nodes, denoted by $r_1, r_2, \ldots, r_N$, in the network, where $M \le N$. The locations of $s_0, s_1, \ldots, s_M$ are already known, but the locations of $r_1, r_2, \ldots, r_N$ are yet to be determined. The data generating rate of $s_j,\ j = 1 \ldots M$, is also known and is denoted by β_j.

Our goal is to design a deployment scheme to place these relay nodes to form an energy-efficient topology and forward sensing data. Since sensor nodes have no communication capability, each of them has to be accompanied by a relay node (this is why $M \le N$). We thus need to decide the locations of the rest of the $N - M$ relay nodes. Let l_i and t_i be the location and the transmission distance of r_i, respectively, $i = 1 \ldots N$. We assume that each t_i is controllable, but must satisfy

$$\mathbb{R}_{min} \le t_i \le \mathbb{R}_{max}.$$

For each sensor s_j, $j = 1 \ldots M$, we also need to find a path

$$p_j = r_{j_1} r_{j_2} \ldots r_{j_h} s_0,$$

such that r_{j_1} is located at s_j and the distance between r_{j_i} and $r_{j_{i+1}}$, denoted by $\text{dist}(r_{j_i}, r_{j_{i+1}})$, is $\leq t_{j_i}$, for $i = i \ldots h-1$, and $\text{dist}(r_{j_h}, s_0) \leq t_{j_h}$.

To calculate the energy cost, we denote energy incurred to r_i by path p_j by

$$\hat{E}(r_i, p_j) = \hat{E}_{tx}(r_i, p_j) + \hat{E}_{rx}(r_i, p_j),$$

where $\hat{E}_{tx}(r_i, p_j)$ and $\hat{E}_{rx}(r_i, p_j)$ are the transmit and receive costs, respectively,

$$\hat{E}_{tx}(r_i, p_j) = \begin{cases} \beta_j(t_i^{\alpha} + e_{tx}), & \text{if } r_i \in p_j \\ 0, & \text{otherwise,} \end{cases}$$

$$\hat{E}_{rx}(r_i, p_j) = \begin{cases} \beta_j e_{rx}, & \text{if } r_i \in p_j \text{ and } r_i \text{ is not the head of } p_j \\ 0, & \text{otherwise,} \end{cases} ,$$

where α is the power attenuation factor ($2 \leq \alpha \leq 6$), and e_{tx} and e_{rx} are manufacture-dependent constant of transmit and receive factors, respectively. The total energy incurred on r_i is:

$$\hat{E}(r_i) = \sum_{\forall j} \hat{E}(r_i, p_j).$$

Our relay node deployment problem is to find l_i, t_i, and p_j for $i = 1 \ldots N$ and $j = 1 \ldots M$ with the following objective function:

$$\min\{\max_{\forall i}\{\hat{E}(r_i)\}\}.$$

Chapter 4

Deployment Algorithms

Below, we first make some observations based on an existing work [10]. Then we show how to improve it by presenting our non-linear programming algorithm and some heuristics.

4.1 Basic Ideas

The work [10] shows that an efficient deployment should find a Steiner tree to connect all sensor nodes and the sink node and then evenly place relay nodes along each Steiner edge according to some weighting mechanism. The following lemma shows how relay nodes should be deployed along a Steiner edge.

Lemma 1. *Given two relay nodes r_a and r_b, suppose that r_a has received a certain amount of data to be delivered to r_b. If there are k extra relay nodes available, the best way to deploy these k relay nodes is to evenly distribute them along the line connecting r_a and r_b.*

Proof. There are two ways for us to deploy k extra relay nodes. At first, we can divide the data volume of r_a into q parts and relay these q traffic flows separately by assigning n_i relay nodes, where $i = 1 \ldots q$ and $\sum_{i=1}^{q} n_i = k$. Secondly, we can just relay all the data of r_a by using only one traffic flow with k relay nodes. By the result of [10], we know that we would rather to use the latter one, and must deploy k relay nodes evenly on this flow. □

Fig. 4.1 shows an example. There are q traffic flows and k relay nodes. The length of inflow

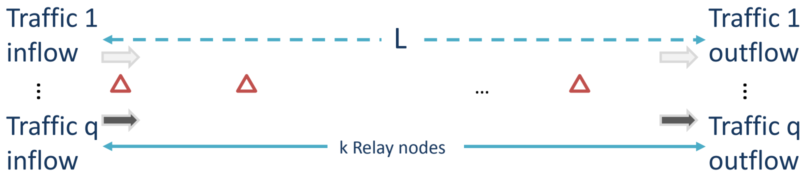


Figure 4.1: An illustration of relay node deployment for single source q traffic flows with length of edge L and k extra nodes .

and outflow are L. The best way to deploy relay nodes is to evenly distribute them among edge L.

Based on the above Lemma 1, the work[10] proposes to solve the deployment problem as follows:

1. Construct a graph of Steiner tree $G = (V, E)$ to connect all sensor nodes and the sink node because Steiner tree can guarantee to interconnect a network with the shortest length where the length is the sum of lengths of all edges, by adding some new vertices which are known as Steiner points. And it is the optimal solution for 1-connected relay node deployment.

2. After the Steiner tree G has constructed, we have to calculate the traffic rate λ_{e_k} of each Steiner edge k, where λ_{e_k} is the sum of data generating rates of the traffic flows passing through the edge k.

3. In Lemma 1, we understand we should deploy relay nodes evenly on each edge k. Hence, the energy of each relay node in edge k must be the same. We can formalize the energy model as

$$\hat{E}_{e_k} = \hat{E}(r_i),$$

and since both terms e_{rx} and e_{tx} are quite small than t_i^{α}, we can simplify the above equation as

$$\hat{E}_{e_k} \approx \lambda_{e_k} t_i^{\alpha} = \lambda_{e_k} (\frac{L_{e_k}}{n_{e_k}})^{\alpha}.$$

And our target is to $\min\{\max_{\forall i}\{\hat{E}(r_i)\}\}$, it can also mean to $\min\{\max_{e_i \in E}\{\hat{E}_{e_k}\}\}$ by given N relay nodes. Therefore, we can let $\hat{E}_{e_1} = \hat{E}_{e_2} = ... = \hat{E}_{e_k}$, and it can be expressed as follows,

$$\lambda_{e_1}(\frac{L_{e_1}}{n_{e_1}})^{\alpha} = \lambda_{e_2}(\frac{L_{e_2}}{n_{e_2}})^{\alpha} = \ldots = \sum_{e_k \in E} \lambda_{e_k}(\frac{L_{e_k}}{n_{e_k}})^{\alpha}. \tag{4.1}$$

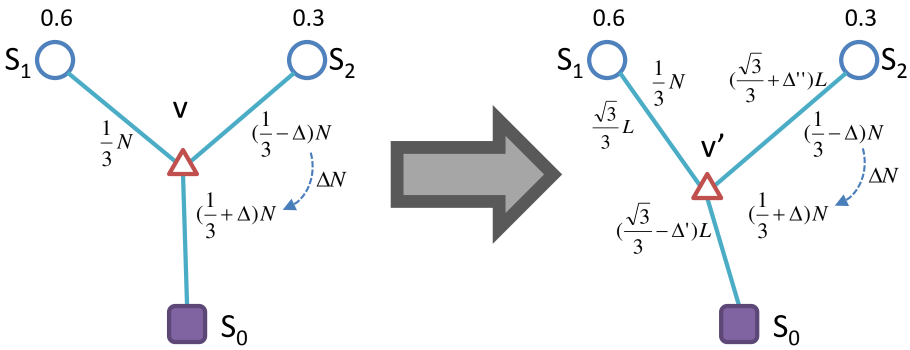
Though Eq. (4.1), we can calculate n_{e_k} for each edge k.

4. After calculating n_{e_k}, we deploy n_{e_k} relay nodes on edge k evenly with communication range $t_{e_k} = \dfrac{L_{e_k}}{n_{e_k}}$. And the first relay node must be placed on the location of the head node of edge. The head node of edge must be sensor node or Steiner point.

Although [10] proposed a pretty good algorithm for traffic-aware relay node deployment, it still has some drawbacks. Fig. 4.2 shows an example. In Fig. 4.2(a), we deploy two sensor nodes and one sink node by [10]. As shown in Fig. 4.2(b), we move the Steiner point v to the new location v'. By numerical analysis, we observe that the energy consumption of each relay node in Fig. 4.2(b) is smaller than Fig. 4.2(a). From the example, we know that there must exist a better network topology for a traffic-aware WSN. Hence, we try to move all Steiner points to proper locations in order to obtain the better topology. In the following, we will show how to move all Steiner points to proper locations and assign appropriate relay nodes on each edge to prolong network lifetime.

4.2 Weighted Moving Algorithm

From above section, we know that moving Steiner points may prolong network lifetime. Here, we propose a weighted moving method which can move the Steiner points according to the nodes' traffic rates. The moving scheme is based on a non-linear programming. Through the results of non-linear programming, we can get the better locations for Steiner points.



(a) (b)

Figure 4.2: An example of improving case for relay node deployment.

In the beginning of relay node deployment problem, we just know the locations of sensor nodes and the sink node, and they are not connected. So, our objective is to connected all sensor nodes and the sink node with minimum energy consumption. And our xxxx algorithm are organized as follows:

1. Construct a graph of Steiner tree $G = (V, E)$ to connect all sensor nodes and the sink node by using heuristic method proposed in [1], where $V = \{v_0, v_1, \ldots, v_M, v_{M+1}, \ldots\}$, and $E = \{e_1, e_2, \ldots\}$. Here, we define vertices $v_i = s_i$ for $i = 0 \ldots m$, and v_j is called Steiner point, $j \geq m + 1$. Let $e_1, e_2, \ldots$ represent the edges connect the vertices in V, where traffic flows are unidirectional.

2. After the Steiner tree G has constructed, we have to calculate the traffic rate λ_{e_k} of each edge k, and λ_{e_k} is the sum of data generating rates of the traffic flows passing through the edge k.

3. Now, we need to find a proper network topology and relay node deployment. In the following, we will propose a non-linear programming to solve this problem. Since the locations of v_i, $i = 0 \ldots m$, i.e., sensor nodes and the sink node, are fixed, we can only modify the locations of v_i, $i = M+1 \ldots$, i.e., Steiner points, to achieve our target. Hence,

we can modify Eq. (4.1) into

$$\lambda_{e_1}(\frac{\sqrt{(X_{A_{e_1}} - X_{B_{e_1}})^2 + (Y_{A_{e_1}} - Y_{B_{e_1}})^2}}{n_{e_1}})^\alpha = \dots$$

$$= \sum_{e_k \in E} \lambda_{e_k}(\frac{\sqrt{(X_{A_{e_k}} - X_{B_{e_k}})^2 + (Y_{A_{e_k}} - Y_{B_{e_k}})^2}}{n_{e_k}})^\alpha \tag{4.2}$$

where $X_{A_{e_i}}$, $Y_{A_{e_i}}$, $X_{B_{e_i}}$ and $Y_{B_{e_i}}$ represent the x-axis and y-axis location of two vertices, i.e., A_{e_i} and B_{e_i} of edge e_i, and the vertices can be sensor nodes, Steiner points or the sink node. In the following, we call the Steiner point as *merge vertex*. According to Eq. (4.2), we model this problem, which can guarantee both the minimum energy for transmitting traffic flows and the network connectivity, as follows.

$$\begin{aligned}
\text{minimize} \quad & \sum_{e_k \in E} \lambda_{e_k}(\frac{\sqrt{(X_{A_{e_k}} - X_{B_{e_k}})^2 + (Y_{A_{e_k}} - Y_{B_{e_k}})^2}}{n_{e_k}})^\alpha \\
\text{subject to} \quad & \sum_{e_k \in E} n_{e_k} = N \\
& \min(X_{v_0}, \dots, X_{v_m}) \le X_{v_i} \le \max(X_{v_0}, \dots, X_{v_m}), \ i = M+1, M+2, \dots \\
& \min(Y_{v_0}, \dots, Y_{v_M}) \le Y_{v_i} \le \max(Y_{v_0}, \dots, Y_{v_m}), \ i = M+1, M+2, \dots \\
& \mathbb{R}_{min} \le \frac{\sqrt{(X_{A_{e_k}} - X_{B_{e_k}})^2 + (Y_{A_{e_k}} - Y_{B_{e_k}})^2}}{n_{e_k}} \le \mathbb{R}_{max}, \ \forall e_k \in E \\
& \lambda_{e_j}(\frac{\sqrt{(X_{A_{e_j}} - X_{B_{e_j}})^2 + (Y_{A_{e_j}} - Y_{B_{e_j}})^2}}{n_{e_j}})^\alpha = \\
& \lambda_{e_{j+1}}(\frac{\sqrt{(X_{A_{e_{j+1}}} - X_{B_{e_{j+1}}})^2 + (Y_{A_{e_{j+1}}} - Y_{B_{e_{j+1}}})^2}}{n_{e_{j+1}}})^\alpha, \ j = 1 \dots k-1
\end{aligned} \tag{4.3}$$

where k is amount of total edges

The constraint 1, it ensures that given N relay nodes must be used completely and assigned to each edge. Constraint 2 and 3 can guarantee the locations of merge vertices must be inside a boundary. Constraint 4 guarantees that the communication range of each relay node must not be over the boundary and also guarantees the connectivity. The last constraint guarantees that the energy of each relay node must be balance. Through the result of the NLP, we can find out all the proper locations for merge vertices and the number of relay nodes n_{e_k} assigned on each edge k.

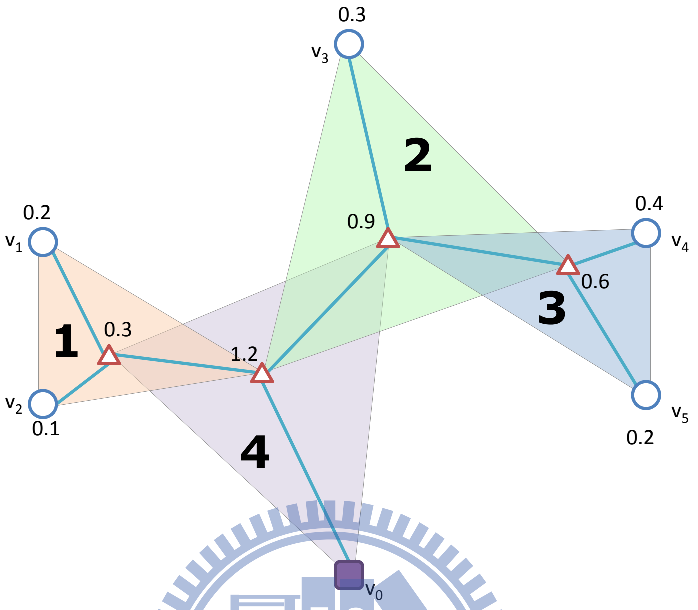
4. After we have decided n_{e_k}, and the locations of merge vertices, we can deploy n_{e_k} relay nodes on edge k evenly with communication range $t_{e_k} = \dfrac{L_{e_k}}{n_{e_k}}$. And the first relay node must be placed on the location of the head node of edge. The head node of edge must be sensor node or Steiner point. Our algorithm can be summarized as Algorithm 1.

Algorithm 1 Weighted Moving algorithm

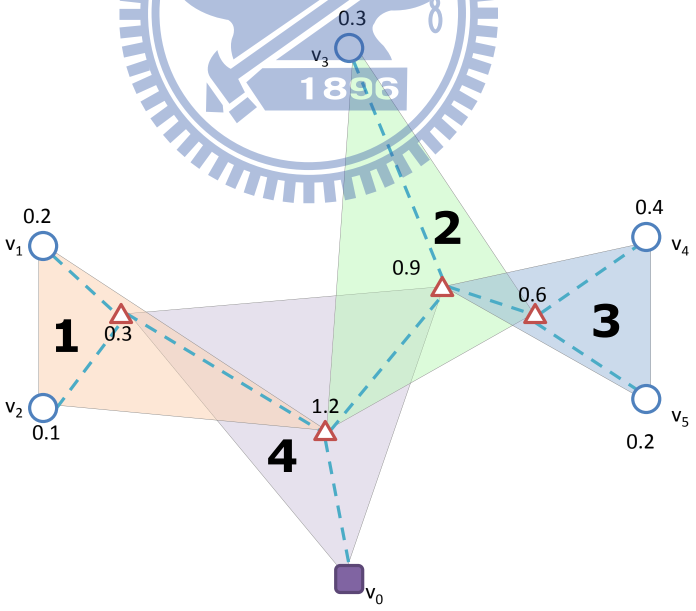
```
Input: $s_0, \ldots, s_m$, $X_{s_0}, \ldots, X_{s_M}$, $Y_{s_0}, \ldots, Y_{s_M}$,
    $\beta_1, \ldots, \beta_M$, $N$ relay nodes, $\mathbb{R}_{min}$ and $\mathbb{R}_{max}$.
Output: $G'(V', E')$

1. Construct a Steiner tree $G(V, E)$ with heuristic approach.
2. Calculate the traffic rate $\lambda_{e_k}$ based on the $G(V, E)$, where $e_k \in E$
3. By Eq. (3), we can calculate the approximative solution $G'(V', E')$ with $n_{e_k}$, $X_{v_{m+1}}, X_{v_{M+2}} \ldots$, $Y_{v_{M+1}}, Y_{v_{M+2}} \ldots$ to decide where the merge vertices are so as to minimize and balance the relay node energy, $e_k \in E'$.
4. Deploy $n_{e_k}$ relay nodes on edge $k$ evenly with communication range $t_{e_k} = \frac{L_{e_k}}{n_{e_k}}$.
```

4.3 Heuristic Approaches

In above section, we use non-linear programming algorithm to model our problem and also find out the proper locations of merge vertices. From the Algorithm 1, we realize that it can find approximative solution. However, since it considers the entire topology to find the proper locations of merge verteices, the complexity of it is depended on how many variables are there. If there are m merge vertices and *n* edges, the variables are *2m+n*. It means that the more merge vertices and Steiner edges there are, the higher complexity it is. Therefore, in this section, we will propose some heuristic approaches that are lower complexity with acceptable improvement. And our heuristic approaches are organized as follows:



(a) Original topology



(b) After finding all proper locations of merge vertices

Figure 4.3: An example of the block-by-block network topology.

1. We construct a Steiner tree $G(V, E)$ to connect all sensor nodes and the sink node with heuristic approach.

2. Then, we can calculate the traffic rate λ_{e_k} based on $G(V, E)$ for each Steiner edge.

3. For decreasing the complexity of Algorithm 1, we need to reduce the number of variables of it. Hence, we firstly divide the network topology into blocks and are going to solve them separately in the following two approaches. Each block is a triangle and composed of four nodes. The merge vertex is a node which is connected three neighbor nodes in the same block, and other three nodes can be sensor nodes, *merge vertex* or sink node. Each block can be considered as an independent 4-points tree, and we can only modify the location of merge vertex in each block. The traffic-rate of merge vertex can represent its block's rate, and we will cope with the blocks in order by pre-defined priorities sequence. The higher rate the block is, the higher priority it is. Fig. 4.3 shows an example of five sensor nodes with different data generating rate and one sink node in the network. In Fig. 4.3(a), we divide the network into four blocks, and each block is a triangle with 4-points tree. We can observe that the block with the highest rate is 4, thus priority of block 4 is the highest priority. And the next highest is block 2. Finally, we can set the order as $(4, 2, 3, 1)$ in this network. Then, we will find the proper location of each merge vertex of each block in orderly.

4. After dividing the network into blocks, we propose two methods to find the proper location of each merge vertex. The first method is "constrained heuristic" which can be finished in polynomial time but its performance worse than Algorithm 1, and the second one is "small-scale nonlinear programming" which can get better performance than the former one with acceptable complexity but the performance also worse than Algorithm 1.

And we will explain these two methods in detail later. Fig. 4.3(b) shows the result after we have solved all blocks in Fig. 4.3(a).

5. In these two methods, we only find the proper location of each merge vertex of each block. Since we solve the relay node deployment problem locally do, not actually calculate how many nodes should be assigned on each Steiner edge. Hence, we can not ensure the communication range of each relay node is in the boundary and also guarantee the network connectivity. Therefore, we must check the network connectivity. We use Eq. (4.1) to calculate $n_{e_1}, n_{e_2}, \ldots$ for each Steiner edge and deploy n_{e_k} relay nodes on Steiner edge k evenly with communication range $r = \frac{L_{e_k}}{n_{e_k}}$.. If the network is connected, the communication range of each relay node must be satisfied the boundary, $\mathbb{R}_{min} \leq \frac{L_{e_k}}{n_{e_k}} \leq \mathbb{R}_{max}$, where $e_k \in E$. And the result can be the candidate of our final solution.

6. In Fig. 4.3(b), we can observe that the boundary of each block is different from the boundary in Fig. 4.3(a). This is because the boundary of a block can be influence by the locations of the merge vertices in close blocks of that block. Therefore, we realize that if all merge vertices have been changed once, the boundaries of each block may be enlarged or reduce. It means that we may find a better location of each merge vertex in the second round. So, we will repeat step $3, 4, 5$ in T iterations to find a better solution, where T is a user-setting parameter.

7. After executing T iterations, we will choose the candidate with minimum energy consumption and connected network as our final solution.

Next, we will explain the two methods which we use to find out the proper location of merge vertex of each block in step 4 in detail.

4.3.1 Constrained Heuristic (CH)

In Algorithm 1, the complexity is very high if there are many merge vertices or Steiner edges. It may not be terminated in polynomial time. Hence, we show the first method which can guarantee to be finished in polynomial time.

In the object function of Eq. (4.3), since there are only three edges in one block, we can modify the object function as,

$$\min \quad \sum_{k=1}^{3} \lambda_{e_k} \left(\frac{\sqrt{(X_{A_{e_k}} - X_{MV})^2 + (Y_{A_{e_k}} - Y_{MV})^2}}{n_{e_k}}\right)^{\alpha}, \tag{4.4}$$

where $e_1,\ e_2,\ e_3$ represent the three edges connected to the merge vertex, X_{MV} and Y_{MV} are location of merge vertex, $X_{A_{e_k}}$ and $Y_{A_{e_k}}$ represent the locations of three vertices connected with the merge vertex, and these three vertices can be sensor nodes, sink node or the merge vertices of other blocks. In this approach, we assume $\alpha = 2$, $n_{e_1} : n_{e_2} : n_{e_3} = 1 : 1 : 1$, and $n_{e_1} + n_{e_2} + n_{e_3} = C$, where C is a constant. Hence, $n_{e_1} = n_{e_2} = n_{e_3} = \frac{C}{3}$. The Eq. (4.4) can be modified as,

$$\min \quad \sum_{k=1}^{3} \lambda_{e_k} \left(\frac{\sqrt{(X_{A_{e_k}} - X_{MV})^2 + (Y_{A_{e_k}} - Y_{MV})^2}}{\frac{C}{3}}\right)^{2} \tag{4.5}$$

Since we want to minimize Eq. (5), we can omit constant $\frac{C}{3}$. And when we extend Eq. (5), we can observe that the solution of it is center of gravity of the triangle of $A_{e_1}A_{e_2}A_{e_3}$. It is proved by following equations,

$$\min \quad \sum_{k=1}^{3} \lambda_{e_k}\left((X_{A_{e_k}} - X_{MV})^2 + (Y_{A_{e_k}} - Y_{MV})^2\right) \tag{4.6}$$

$$\min \quad \sum_{k=1}^{3} \lambda_{e_k}(X_{MV}^2 + Y_{MV}^2) - 2X_{MV}\left(\sum_{k=1}^{3} \lambda_{e_k} X_{A_{e_k}}\right) - 2Y_{MV}\left(\sum_{k=1}^{3} \lambda_{e_k} Y_{A_{e_k}}\right) + K \tag{4.7}$$

$$\min \quad \left(X_{MV} - \frac{\sum_{k=1}^{3} \lambda_{e_k} X_{A_{e_k}}}{\sum_{k=1}^{3} \lambda_{e_k}}\right)^2 + \left(Y_{MV} - \frac{\sum_{k=1}^{3} \lambda_{e_k} Y_{A_{e_k}}}{\sum_{k=1}^{3} \lambda_{e_k}}\right)^2 + K' \tag{4.8}$$

where K and K' are constant. From Eq. (8), when X_{MV} and Y_{MV} are equal to $\dfrac{\sum_{k=1}^{3}\lambda_{e_k}X_{A_{e_k}}}{\sum_{k=1}^{3}\lambda_{e_k}}$ and $\dfrac{\sum_{k=1}^{3}\lambda_{e_k}Y_{A_{e_k}}}{\sum_{k=1}^{3}\lambda_{e_k}}$, Eq. (8) will get minimum.

Theorem 2. *Considering one block in relay deployment problem, if* $\alpha = 2$, $n_{e_1} : n_{e_2} : n_{e_3} = 1 : 1 : 1$, *and* $n_{e_1} + n_{e_2} + n_{e_3} = C$, *where C is a constant. The best location of merge vertex must be the center of gravity of the triangle* $A_{e_1}A_{e_2}A_{e_3}$.

4.3.2 Small Scale Non-linear Programming (SSNLP)

In the first approach, although we can solve the problem in polynomial time if $\alpha = 2$ and $n_{e_1} : n_{e_2} : n_{e_3} = 1 : 1 : 1$, the performance may not improved more because the proper location is only related to the traffic rate and the length of Steiner edge. What we really need is to find a proper location of merge vertex which is involved with the traffic rate, the length of Steiner edge and the ratio among n_{e_1}, n_{e_2} and n_{e_3}. And it is suitable for any cases. As can be seen that the reason why the complexity of Algorithm 1 is high because it considers the entire topology at the same time. But, in the previous heuristic approach, we show that we can divide the whole topology into blocks and solve them orderly. Therefore, we can modify Eq. (4.3) into

$$\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{3}\lambda_{e_k}\Big(\frac{\sqrt{(X_{A_{e_k}}-X_{MV})^2+(Y_{A_{e_k}}-Y_{MV})^2}}{n_{e_k}}\Big)^{\alpha} \\
\text{subject to} \quad & n_{e_1}+n_{e_2}+n_{e_3}=C \\
& X_{MV}, Y_{MV} \in triangle(X_{A_{e_k}}, Y_{A_{e_k}}),\ k=1,2,3 \\
& \lambda_{e_j}\Big(\frac{\sqrt{(X_{A_{e_j}}-X_{MV})^2+(Y_{A_{e_j}}-Y_{MV})^2}}{n_{e_j}}\Big)^{\alpha} = \\
& \lambda_{e_{j+1}}\Big(\frac{\sqrt{(X_{A_{e_{j+1}}}-X_{MV})^2+(Y_{A_{e_{j+1}}}-Y_{MV})^2}}{n_{e_{j+1}}}\Big)^{\alpha}, j=1,2\,,
\end{aligned} \tag{4.9}$$

where C is a constant, $e_1,\ e_2,\ e_3$ represent the three edges connected with the merge vertex, $X_{A_{e_k}}$ and $Y_{A_{e_k}}$ represent the locations of three vertices connected with the merge vertex, and these three vertices can be sensor nodes, sink or the merge vertices of other blocks, and where X_{MV} and Y_{MV} are the location of merge vertex. Since we solve the relay node deployment locally, we can only calculate the ratio among n_{e_1}, n_{e_2} and n_{e_3} in constraint 1. Constraint 2 ensures the proper location of merge vertex must be inside the block. Constraint 3 can guarantee all energy consumption of relay nodes on three Steiner edges of the block must be the same.

In Eq. (4.9), we only find the proper location of merge vertex, i.e., X_{MV} and Y_{MV}, and the ratio among n_{e_1}, n_{e_2} and n_{e_3}, so the number of variables of Eq. (4.9) is only 5. The number of variables is fixed and will not change due to different network topology. Although the complexity of second method is little higher than the first one, the performance of it must be better.

Now, let us summarize our heuristic approaches briefly. Before we start to execute the heuristic approaches, we first check whether the given N relay nodes can connect the Steiner tree or not by Eq. (4.1). If N relay nodes are enough to connect the Steiner tree, the heuristic approaches can be executed. Otherwise, since the given N relay nodes is less than minimum requirement, we can not find a solution for this problem. Then, after finding out all proper locations of merge vertices, we will check whether the new network topology can be connected by N relay nodes or not. If the new topology can be connected, it will be the candidate of our final solution. Then we will repeat the above steps in T times. After that, we choose the candidate with minimum energy consumption and connected network as our final solution. Our heuristic approaches can be summarized as Algorithm 2.

Algorithm 2 Heuristic

```
Input: $s_0, \ldots, s_M$, $X_{S0}, \ldots, X_{SM}$, $Y_{S0}, \ldots, Y_{SM}$,
    $\beta_1, \ldots, \beta_M$, $N$ relay nodes, $\mathbb{R}_{min}$ and $\mathbb{R}_{max}$, method.
Output: $G(V, E)$

1. Construct a Steiner tree $G(V, E)$ with heuristic approach.
2. Calculate the traffic rate $\lambda_{e_k}$ based on the $G(V, E)$, where $e_k \in E$
if CheckConnectivity(N, $G(V, E)$) then
  Calculate $\hat{E}$.
  for $i := 1$ to $T$ do
    Partition the network topology into blocks.
    switch (method)
    case 1: Execute constrained heuristic
    case 2: Execute small scale non-linear programming
    end switch
    Use Eq. (1) to calculate $n_{e_1}, n_{e_2}, \ldots$ and calculate $\hat{E}'$.
    if ($\hat{E}' < \hat{E}$) and CheckConnectivity(N, $G'(V', E')$) then
      $\hat{E} = \hat{E}'$;
      $G(V, E) = G'(V', E')$;
    end if
  end for
  Use Eq. (1) to calculate $n_{e_k}$ for edge $k$, $e_k \in E$
  Deploy $n_{e_k}$ relay nodes on edge $k$ evenly with communication range $t_{e_k} = \frac{L_{e_k}}{n_{e_k}}$.
else
  The given $N$ relay nodes can not connect the Steiner tree.
end if
```

Chapter 5

Simulation

In this section, we will evaluate the performance of our algorithms by numerical analysis. Since NLP can not guarantee to solve this problem in polynomial time, we just evaluate our two heuristic approaches, i.e., CH and SSNLP. We have deployed 5 to 25 sensor nodes by random distribution in a field of $5000m \times 5000m$ with the sink node settled at the center $(2500, 2500)$. And the normalized data rate of each sensor node is randomly chosen from $(0, 1]$. For each number of sensor node, we generated 10 topologies for analysis. In this network, there are only 250 relay nodes for us to deploy. We set the power attenuation of relay nodes $\alpha = 2$, the maximum communication range of relay nodes $\mathbb{R}_{max} = 500m$, and minimum one $\mathbb{R}_{min} = 10m$. The above simulation environment is shown in Table 5.1.

For comparison, we implement two deployment approaches from [10][15], namely *Connectivity-only*, denote as CO, and *Traffic-aware*, denote as TA. CO is chosen from a state-of-the-art scheme proposed in [15], which optimizes the system performance by considering connectivity-

Table 5.1: Simulation Environment

Sensor nodes	$5, 10, 15, 20, 25$
Data generating rate	$[0.1, 0.2, \ldots, 1]$
Maximum communication range of relay node	$500m$
Minimum communication range of relay node	$10m$
Deployment area	$5000 * 5000m^2$
Location of sink node	$(2500, 2500)$
Strategy of sensor deployment	Random distribution
Numbers of topology of each sensor	10

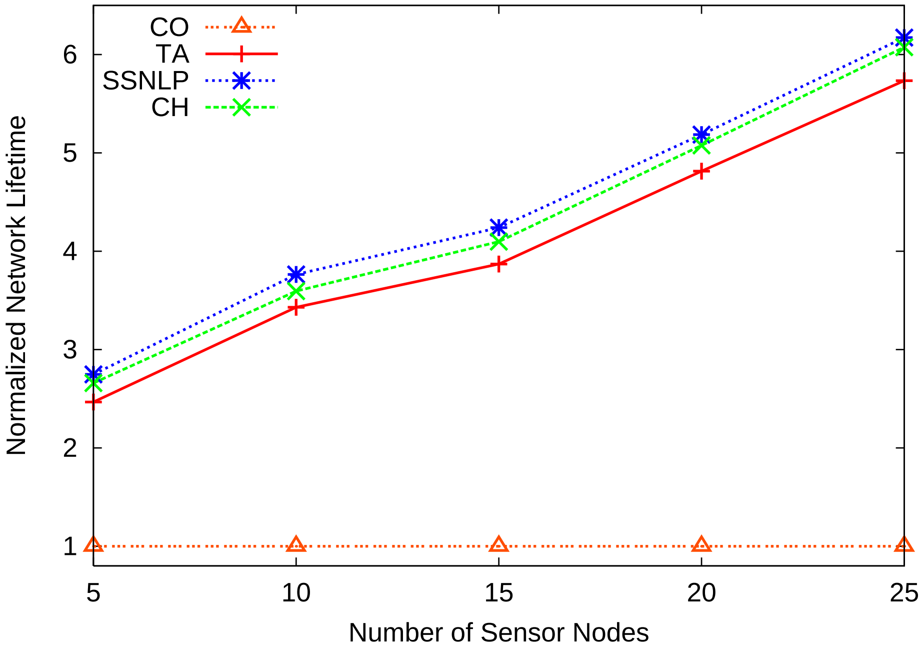


Figure 5.1: Normalized network lifetime.

only. In [15], there are multiple versions of scheme. Here, we use the 1-connectivity version and construct a Euclidean Steiner minimum tree [1] for the network topology, and CO serves as a baseline. TA is chosen from [10] which is the first study accommodate the heterogeneous traffic flows in relay node deployment for WSN.

We use network lifetime, defined as the lifetime of the first depleted relay node, as our metrics for evaluation. The first depleted relay node can serve as a good indicator for the end of the network lifetime because if the first relay node is out-of-battery, the data of some sensor nodes can not be relayed to the sink node in 1-connected WSN. In the following simulation results, they are normalized by the base-line scheme CO. We use MATLAB [2] as our simulation tools, and the interior point methods[5] to solve non-linear problems. Fig. 5.1 shows the results of the network lifetime with different number of sensor nodes. When the number of sensor nodes increases, the lifetime of both SSNLP and CH increases and is higher than lifetime of TA. Since SSNLP has considered traffic rate, length of edge and ratio of relay nodes among three edges, we can observe that the lifetime of SSNLP is always better than CH's in Fig. 5.1. We

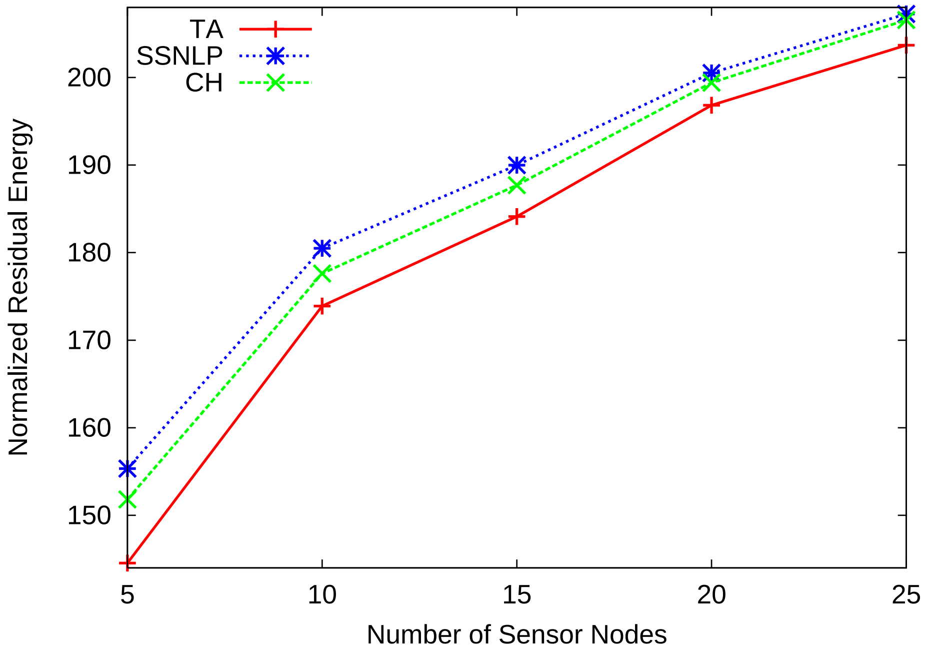


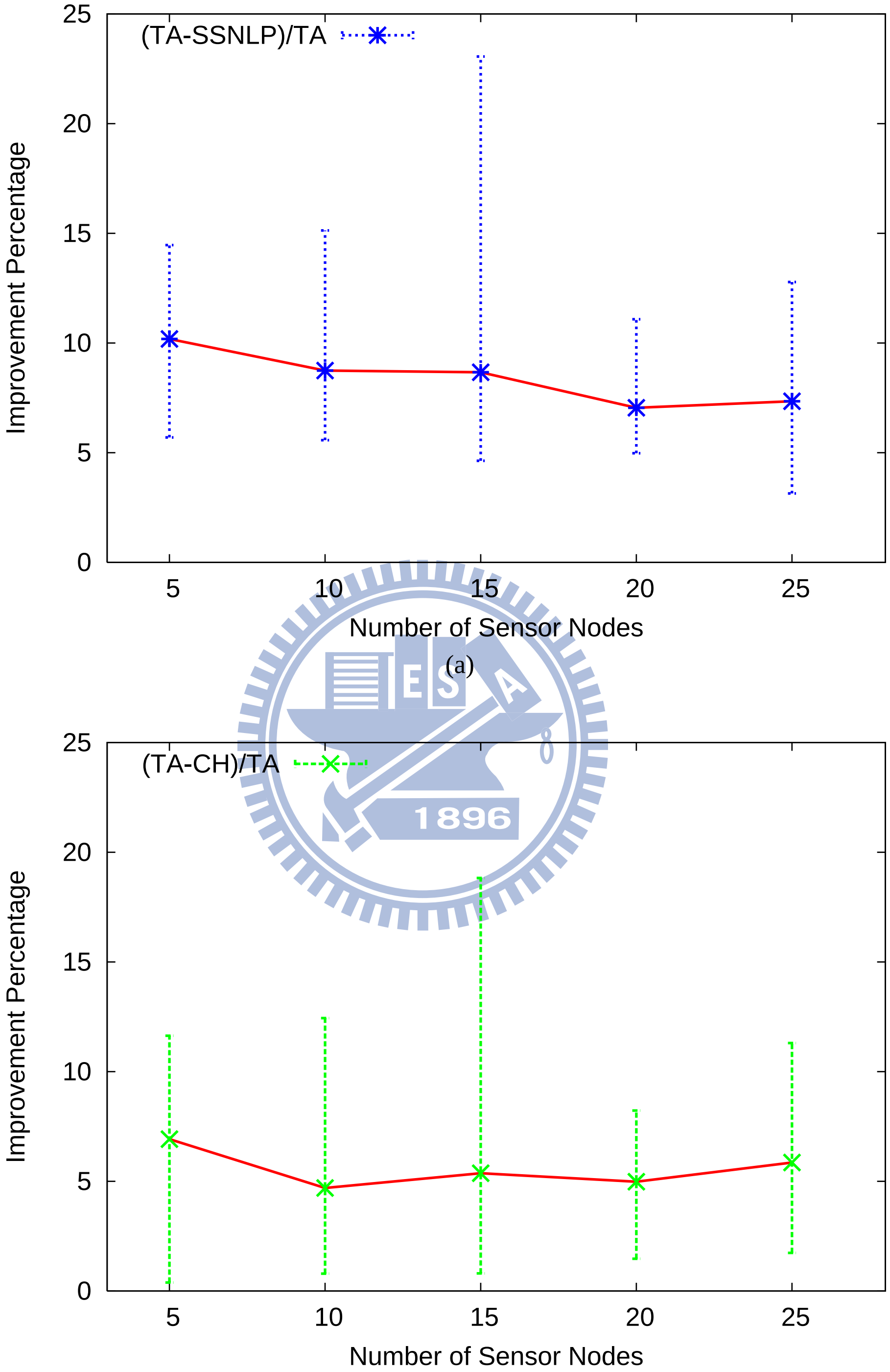
Figure 5.2: Normalized residual energy.

have also evaluated the residual energy among these approaches. Since CO can not guarantee the energy consumption of each relay node is the same, we only evaluate the residual energy of TA, SSNLP and CH when the lifetime of CO is finished. And the total residual energy can be estimated as

$$\hat{E}_{residual} = \sum_{e_k \in E} [\hat{E} - \lambda_{e_k} (\frac{L_{e_k}}{n_{e_k}})^{\alpha} T] \cdot n_{e_k},$$

where T is the network lifetime of CO, and the result is shown in Fig. 5.2. When the number of sensor increases, the normalized residual energy of TA, SSNLP and CH increases. And because the energy consumption of SSNLP and CH is less than TA, the normalized residual energy of them is higher than TA's.

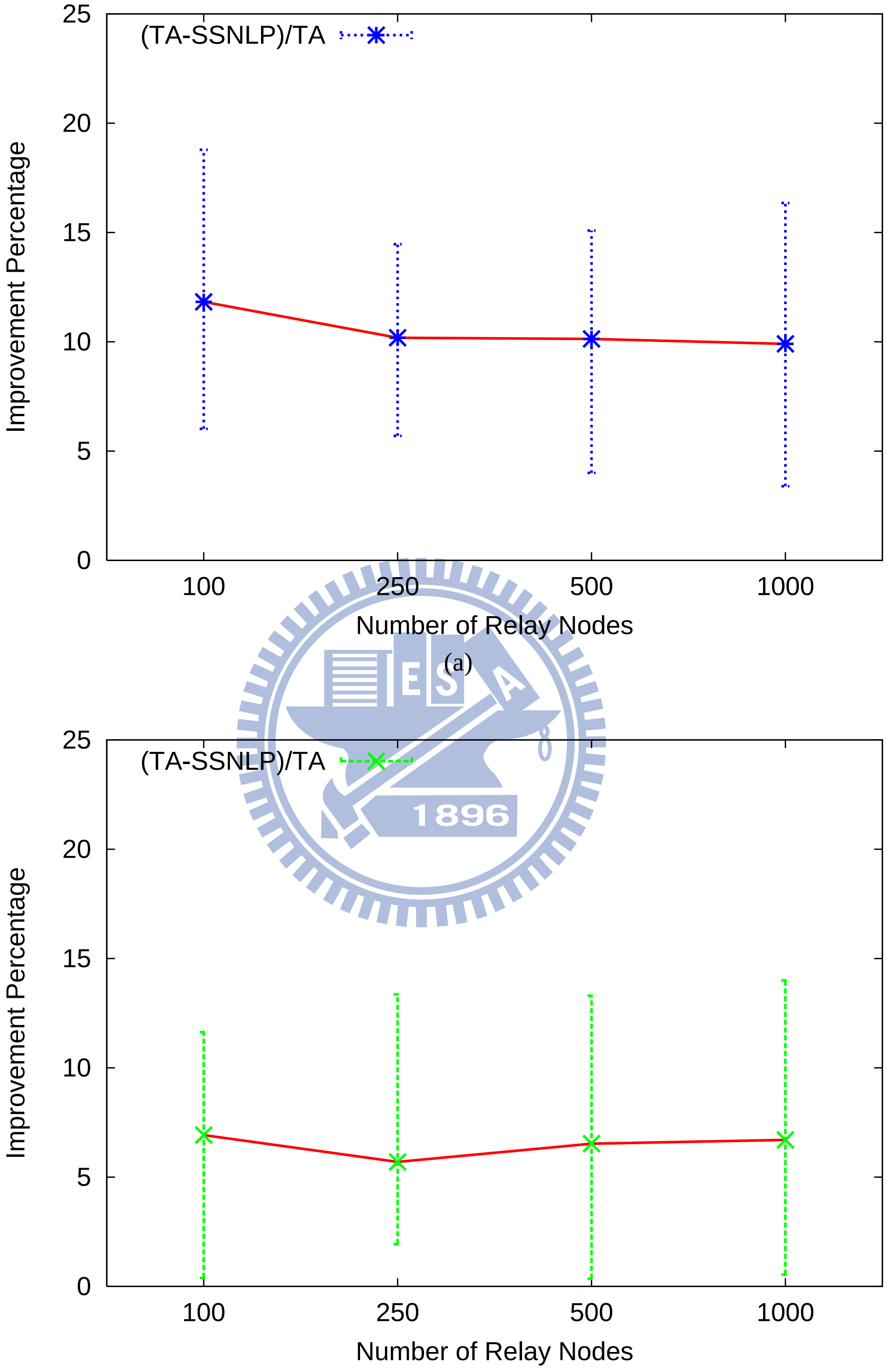
Fig. 5.3(a) and Fig. 5.3(b) show that the improvement from TA to SSNLP and CH, respectively. We observe that SSNLP can always obtain $8 \sim 10\%$ improvement with different number of number of sensor nodes, and CH can obtain $5 \sim 7\%$ improvement. Although the averages of the improvement percentage of both SSNLP and CH are not very high, we can notice that the difference between maximum and minimum improvement is large, such as, the maximum can



(a)

(b)

Figure 5.3: The performance improvement of CH and SSNLP with different numbers of sensor nodes.



(a)

(b)

Figure 5.4: The performance improvement of CH and SSNLP with different numbers of relay nodes.

be 23% and the minimum can be 5% in SSNLP. How many improvement percentages obtained by SSNLP and CH is decided by the configuration of the network topology.

Next, we fix the network topology with only 5 sensor nodes, and set the number of relay nodes from 100 to 1000. Fig. 5.4(a) and Fig. 5.4(b) show that the improvement from TA to SSNLP and CH, repsectively. We can observe that no matter how many relay nodes there are, CH and SSNLP can also retain a pretty good improvement. SSNLP can maintain the improvement percentage around $10 \sim 13\%$, and CH can keep the improvement percentage around $6 \sim 8\%$.

From the above results, we realize that only considering connectivity is the worst case for relay node deployment, and only considering traffic volume based on Euclidian Steiner tree also is not the best way. The results show us that we should construct a traffic-aware network topology and the strategy of the relay node deployment must be according to the traffic volume of each edge as well.

Chapter 6

Conclusions

In this thesis, we showed some problems about deploying relay nodes on Euclidian Steiner tree. We proposed some algorithms to modify the existing Euclidian Steiner tree to adapt the traffic volume in reality. Firstly, we proposed a weighted moving algorithm. In this algorithm, we modeled the relay node deployment problem as a non-linear programming. In this programming, we could guarantee that the network is connected and also accommodate the traffic volume. But the non-linear programming could not ensure to solve the problem in polynomial time, so we had proposed two heuristic approaches, which could finish in polynomial time, to compare with the previous studies by MATLAB. According to the simulation results, we observed that the performance of these two heuristic approaches was better than all previous studies. And no matter what circumstances, they could obtain a pretty good improvement.

Bibliography

[1] Geosteiner. http://www.diku.dk/hjemmesider/ansatte/martinz/geosteiner/.

[2] Matlab. http://www.mathworks.com/.

[3] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proc. of ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[4] D. Chen, D. Du, X. Hu, G. Lin, L. Wang, and G. Xue. Approximations for steiner trees with minimum number of steiner points. *Journal of Global Optimization*, 18:17–33, 2000.

[5] E. K. P. Chong and S. H. Zak. *An Introduction to Optimization, 2nd Edition*. Wiley, 2001.

[6] X. Han, X. Cao, E. L. Lloyd, and C. C. Shen. Fault-tolerant relay node placement in heterogeneous wireless sensor networks. In *IEEE Trans. on Mobile Computing*, 2009.

[7] G. Lin and G. Xue. Steiner tree problem with minimum number of steiner points and bounded edge-length. *Information Processing Letters*, 69:53–57, 1999.

[8] S. Mistra, S. D. Hong, G. Xue, and J. Tang. Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements. In *Proc. of IEEE INFOCOM*, 2008.

[9] F. Wang, D. Wang, and J. Liu. Prolonging sensor network lifetime with energy provisioning and relay node placement. In *Proc. of IEEE Sensor and Ad Hoc Communications and Networks Conference (SECON)*, 2005.

[10] F. Wang, D. Wang, and J. Liu. Traffic-aware relay node deployment for data collection in wireless sensor networks. In *Proc. of IEEE Sensor and Ad Hoc Communications and Networks Conference (SECON)*, 2009.

[11] K. Xu, H. Hassanein, and G. Takahara. Relay node deployment strategies in heterogeneous wireless sensor networks: Multiple-hop communication case. In *Proc. of IEEE Sensor and Ad Hoc Communications and Networks Conference (SECON)*, 2005.

[12] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. of IEEE INFOCOM*, 2002.

[13] L. W. Yeh, C. Y. Lu, C. W. Kou, Y. C. Tseng, and C. W. Yi. Autonomous light control by wireless sensor and actuator networks. *IEEE Sensors Journal*, To appear.

[14] L. W. Yeh, C. Y. Lu, Y. H. Lin, J. L. Liao, Y. C. Tseng, C. Chen, and C. W. Yi. ilamp: A sensor-enhanced lamp with surface-tracking capability based on light intensity. In *IEEE Int'l Conf. on Pervasive Computing and Communications(PerCom)*, 2009.

[15] W. Zhang, G. Xue, and S. Misra. Fault-tolrant relay node placement in wireless sensor networks: Problems and algorithms. In *Proc. of IEEE INFOCOM*, 2007.

Vita

Chi-Wai Kou

Contact Information

Department of Computer Science

National Chiao Tung University

1001 Ta Hsueh Road, Hsinchu, Taiwan 300

Email: kaocw@csie.nctu.edu.tw, chiwai319@gmail.com

Education

M.S.: Network Engineering, National Chiao Tung University (2008.9 ~ 2010.7)

B.S.: Computer Science, National Tsing Hua University (2004.9 ~ 2008.6)

Awards

1. 旺宏金矽獎 銅獎, “以無線感測器網路爲基礎之智慧型節能燈光調控系統”, with 葉倫武, 林育萱, 呂哲彥, 曾煜棋, 2009
2. RFID 校園創意競賽 佳作, “智慧型展場導覽系統”, with 葉倫武, 許藍尹, 羅棨鐘, 曾煜棋, 2009

Publication Lists

Journals

1. L. W. Yeh and C. Y. Lu and C. W. Kou and Y. C. Tseng and C. W. Yi, “Autonomous Light Control by Wireless Sensor and Actuator Networks”, *IEEE Sensors Journal*, To appear.