

Constructing Near Optimal Backbones in Wireless Sensor Networks

Chih-Hsien Tang, Hsin-Wen Huang, and Chang Wu Yu

Department of Computer Science and Information Engineering, Chung Hua University
Taiwan, R.O.C.

{b9201053, b09302055, cwyu}@chu.edu.tw

摘要

無線感測網路的感測器除了感測的功能外，還必須利用骨幹(Backbone)上的 Router 將接收的資料轉傳給 Sink。因此如何建造一個最佳的骨幹，使得骨幹上所需 Router 數量最少且感測器到達 Sink 之間的 Hop 數(通訊延遲)最小，就成為一個相當重要的研究議題。本論文提出兩種有效率的無線感測網路上骨幹建造之演算法。此兩個演算法可建造出靠近最佳的骨幹使之兼顧省電與低延遲的目的，而其時間複雜度分別為 $O(n^3)$ 及 $O(n^4)$ ，此處 n 為部署感測器的個數。模擬實驗證實我們可以藉由增加些微的通訊延遲，就能有效降低所需 Router 的數量以達到節省電池電力的效果。

關鍵詞: Wireless Sensor Network, Router, Backbone, Delay.

一、簡介

感測網路可以透過一個 Sink(可視為一個 Gateway)來當作此感測網路和外界的橋樑。所有感測器收集到的資訊都會先匯集到 Sink，再由 Sink 將資訊以衛星、Internet 或是其它方式傳送給遠方的使用者或伺服器。因為感測器的體積小，它所能搭載的電池也就相對的小。而且此電池是無線感測器主要的電源，想更換電池以延長感測器的壽命在無線感測網路上是一個較難實現的方式。因此，如何妥善運用有限的電池電力，一直是無線感測網路上一個相當重要的研究議題。

許多感測器在有限的傳輸範圍內可能無法直接與 Sink 聯繫，而是必須將資料遞交給傳輸範圍內的鄰居，再由這些鄰居將資料傳送出去，直到資料抵達 Sink 便完成了傳送；這些進行轉傳動作的感測器被定義為 Router。我們定義包括 Sink 在內

的所有 Router 所構成之集合為骨幹(Backbone)，所有的感測器都透過骨幹來進行資料的傳送。利用骨幹的建造，不需要使得每個感測器都必須隨時待命以負責接收與轉傳來自鄰居的資料，達到節省整體電力資源消耗的效果。

傳統上最常用的網路架構為樹(tree)，利用建造一棵連接 Sink 的樹來當作整個傳送網路的骨架。其他的感測器則利用此骨架上的 Router 將感測的資料傳送到達 Sink。然而骨架上的 Router 除了自身的感測傳遞外，還必須負責來自鄰居的資料接收與轉傳，所以擔任 Router 的感測器要額外耗費許多電源，當感測器電源快速地耗盡會造成路徑的中斷，甚至造成網路無法連通，因此如何減少 Router 個數是相當重要的研究方向。此問題可容易被視為圖論上的連結支配點問題(connected dominating set problem)，而連結支配點問題早已被證明為 NP-complete。

另一個重要議題為延遲(delay)。感測器到 Sink 之間的距離就是這段路徑上所要經過的 Router 數，這段路上的 Router 數越多代表感測器的資料會越慢傳遞到 Sink，在無線感測網路的應用上，資料迅速地傳到 Sink 相當重要，例如火災的即時通報、竊賊入侵的警報系統。

一個好的網路連結架構，就必須在縮短與 Sink 之間的 Hop 數、節省 Router 個數、與減低時間複雜度取得一個平衡點。前人的成果大多研究如何建造一個減少 Router 個數的骨架或是建造一個最小延遲的骨架。很少有同時考慮兩個因素的研究。

本論文提出兩種有效率的無線感測網路上骨幹建造之演算法。此演算法可佈建出最佳的骨幹使之兼顧省電與低延遲的目的，而其時間複雜度分別為 $O(n^3)$ 及 $O(n^4)$ ，此處 n 為部署感測器的個數。模擬實驗證實我們可以藉由增加些微的通訊延遲，就能大量有效地降低所需 Router 的數量以達到節省電池電力的效果。

本篇論文的章節分配如下，第二節介紹相關論

文的成果，第三節定義最佳骨幹的問題，第四節詳細定義本篇論文的演算法，第五節進行本演算法的模擬與效能評比，第六節是最後的結論與未來展望。

二、相關研究與成果

在[3]中，Du 與 Lin 提出一個新的路由通訊協定，稱為 Multi-Class Routing Protocol。此協定利用異質感測器通訊能力比較強的感測器來做為 backbone nodes(B-nodes)。此路由的區域被細分好幾個方格(cells)，每個方格只能夠選擇一個 B-node，節點(node)與節點之間的資料傳送，就是利用 B-nodes 做為傳輸路徑。

在[4]中，Ghosh 和 Basagni 提出休眠骨幹(Napping Backbone)架構以改善 Sensor-DMAC，做法就是以叢聚為一個單位進行 TDMA(Time Divide Multiple Access)的排程。雖然由於有休眠的要素存在，導致整體 Latency 增加，但是模擬數據指出 Napping Backbone 可以比 Sensor-DMAC 節省 70% 的電源消耗，大大的提升了網路的存活時間。

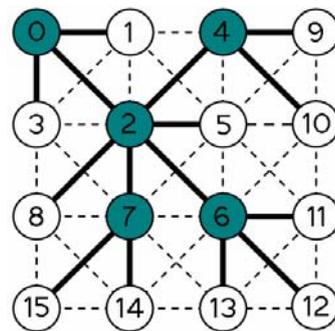
Wang 和 Berger 在[5]中，提出一個同質無線感測網路上的新架構 self-organizing Redundancy Cellular Architecture (SoRCA)。SoRCA 將覆蓋的區域切割成同樣大小的六角形稱為多餘細胞格(redundancy cells, 簡稱 RC)。並且每個 RC 中只保留一個感測點活著。他們認為此 SoRCA 提供一個數學分析及實驗的平台。

稍後在[6]中，Wang 和 Berger 更提出同質無線感測網路上建立於一個 SoRCA 上的之拓樸之分散式資源分配演算法(distributed resources allocation algorithm)，及最佳路徑演算法(the optimal path algorithm)，分散式機率路由演算法(the distributed probabilistic routing algorithm)。此分散式資源分配演算法可分配頻道(channel，也就是，time slots，frequency channels 或 CDMA spreading codes 等)給每一個六角形 RC。最佳路徑演算法計算從多餘細胞格到 Sink 的最佳路徑。分散式機率路由演算法則將封包導向亂數選取的最佳路徑以平衡工作負荷(workload)。

在[7]中，Tseng 和 Pan 針對 Zigbee 的 Scheduling 提出了幾種排法，如果網路拓樸以 Path 構成，Time Slot 的分配就由離 Sink 的 Hop 數最遠者開始依序遞增選擇，Ring 拓樸一樣是由最遠的 Node 開始，第一步會填完 Ring 一半邊的 Time Slot，另一半邊則由最接近 Sink 的 Node 開始，先選擇一個不與鄰居衝突並離 Sink 最近的 Time Slot，再依序向下遞減選擇；Path、Ring 的架構可以得到 optimal 的排程，第三種排程方法架構在

Breadth-First Search (BFS) Tree 拓樸下，又分別以 Centralized、Distributed 兩種方式來探討，Centralized 是先由 Leaf 開始選擇 Time Slot，再將 Leaf 中最大的 Time Slot+1 做為 Root 的 Time Slot，最後由 Root 宣布他的 Time Slot，讓 Leaf 以不與其他鄰居的衝突為前提調整為最接近 Root 的 Time Slot；Distributed 則是由 Root 以 Time Slot=0 廣播訊息，接收到廣播的 Node 開始倒數一段 Backoff Time，倒數過程中隨時紀錄從其他 Node 傳來的 Time Slot 資訊，當倒數完畢後，選擇一個不與鄰居衝突並離 Root 最近的 Time Slot，最後以兩倍 Original Transmission Range 的 Power 將 Time Slot 資訊廣播出去；這兩種方法為 Heuristic Algorithm。

感測器如果當作 Router 使用，電力會因為大量的封包存放與轉傳耗損，所以減少網路上的 Router 可以達到好的省電效果。Ma 等人在[8]中，設計了一種演算法 Ordinal Pruning，先從 Sink 傳送控制封包，讓所有感測器都知道鄰居資訊，再由離 Sink 最遠(Hop 數最大)、ID 最小的感測器開始判斷自己是否可以去除路由器的角色，如圖一所示，控制封包從 Sink 傳遍整個網路後，再從優先權最小的 Sensor 15 開始往回判斷，演算法的判斷條件是如果一個感測器去除 Router 的角色後，同 Hop 且擔任 Router 的鄰居可以涵蓋住感測器所有的子節點鄰居，此感測器就真正去除 Router 角色而成為一個 Leaf Node，以圖一的 Sensor 5 為例子，Sensor 5 的子節點鄰居(Sensor 9、10、11)可以被 Sensor 5 同 Hop 且擔任 Router 的鄰居(Sensor 4、6)所涵蓋，所以 Sensor 5 就去除 Router 的角色而成為 Leaf Node；Ordinal Pruning 雖然能使所有感測器到達 Sink 之間的 Hop 數減到最小，但是也使得為數不少的 Sensor 必須擔任 Router 的角色。



圖一：Ordinal Pruning

三、最佳骨幹問題的正式定義

一個幾何圖(geometric graph) $G=(V, E, r)$ 由放置在二維空間 R^2 上的 nodes 所構成，而且邊(edge)的集合 $E=\{(i, j) \mid d(i, j) \leq r, \text{ where } i, j \in V\}$ ，此處 $d(i, j)$ 代表兩個 node 之間的歐基里德距離(Euclidean distance)。顯然地，幾何圖很適合

用來表示一個已經部署好的感測網路，在圖 G 中，某一個集合 $S \subseteq V(G)$ ，若其他所有不在此集合 S 內的每 vertex 都至少有一個鄰居(neighbor)在此集合 S 之中，則我們將此集合 S 稱為支配集合(dominating set)。

一個由一些邊所構成的序列 $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)$ 稱為一條路徑(path)，且這條路徑是從 v_1 開始，到 v_n 結束，長度(length)為 $n-1$ ；此外，我們稱 v_1 和 v_n 為這條路徑的兩個端點(end vertices)。一個圖 G 中，如果每一對 nodes 之間都存在有一條路徑，則我們稱此圖為連通的(*connected*)；否則，稱此圖為非連通的(*disconnected*)。一個沒有循環(*cycles*)的圖我們稱作無環路的(*acyclic*)。一座森林(*forest*)是一個 *acyclic graph*。一棵樹(*tree*)是一個 *connected acyclic graph*。在圖 G 中，我們將任意兩個 vertex u 和 v 之間的距離 *distance* $d_G(u, v)$ 定義為圖 G 中最短路徑的 length。在圖 $G=(V, E)$ 中，一個離 vertex v 最遠的 node 到 v 之間的 distance，稱作 *eccentricity* $e(v)$ ；也就是說 $e(v)=\max\{d(u, v), \text{ where } u \in V\}$ 。

四、建造近似最佳骨幹的演算法

在本節中，我們提出兩種貪婪演算法，在無線感測網路上建造一個不僅能節省電池電力，並且也要求有最小通訊延遲時間的骨幹。首先，在 4.1 節正式地定義最佳骨幹問題。然後分別在 4.2 節和 4.3 節描述最佳覆蓋優先演算法(the best coverage first algorithm)與最少延遲優先演算法(the least delay first algorithm)。

4.1 最佳骨幹問題

這篇論文所討論的無線感測網路，由三種 sensor 構成，分別是 sink node、backbone nodes 和 normal nodes。sink node 回應整個感測網路之中，從其他所有 sensor 傳送過來的他們所蒐集與接收的監控資訊。backbone nodes 連接在一起，接收 normal nodes 的資訊並轉傳到 sink node 以 hop by hop 的方式。

一個在感測網路之中最佳的骨幹，不僅要把每個 sensor 所蒐集的資訊以最少的 hop 數傳送到 sink node，而且要求使用最少數量的 sensor 去建造出可以蒐集到每個 normal node 所送來的資訊的 backbone。這表示每一個 normal node 的通訊範圍內必定至少存在一個以上的 backbone node。

我們可以用一個幾何圖來正式地定義一個最佳骨幹問題。定義描述如下：

給一個幾何圖 $G=(V, E, r)$ ，並且令 sink node 為 node s ，最佳骨幹問題就是要從圖 G 之中找出一

個子圖 T 並且滿足以下條件：

Condition (1): $s \subseteq T$.

Condition (2): T is a tree.

Condition (3): T is a minimum dominating set of G .

Condition (4): The value $\sum_{u \in V} d_T(u, s)$ of T is minimized.

Condition (1)確保 backbone 裡面一定包含 sink node，以接受從所有 normal nodes 蒐集到的資訊。

Condition (2)保證我們想要建造的 backbone 的拓撲結構為一棵 tree，tree 是在 sensor networks 中常被採用的一種結構。

Condition (3)確保所有不是 backbone nodes 的 sensor 在其通訊範圍內可以連接到至少一個 backbone node。

最後一個條件，用來使每個 sensor 蒐集的資訊傳到 sink node 的路徑上，被要求必須經過的 total hop number (即 delay time)，減到最小。

如果我們設計並建造一個只滿足 condition (4) 的 backbone，我們可以在 G 中找出一條從某個 sensor 到 sink node 的最短路徑。因此，我們發現到，eccentricity $e(s)$ of T 必須要大於或等於 $e(s)$ of G 。另一方面，我們也嘗試去獲得一個 small size of $T(|T|)$ ，以便剩下的 sensors (當作 normal nodes)，能節省寶貴的電池電力。

基於上述討論，我們可以用下面列舉的一個多目標線性規劃(multi-objective linear programming)來表示最佳骨幹問題。

Objective (1): Minimize $|T|$.

Objective (2): Minimize $\sum_{u \in V} d_T(u, s)$.

Subject to:

1. T is a subgraph of G .

2. T is a tree including s .

3. T is a connected dominating set of G .

光是尋找一個 minimum amount of connected dominating set 就是一個 NP-hard 問題。很明顯地我們無法建造同時滿足這兩個目標的一棵樹。因此，在下面兩小節之中，我們設計了兩個建造 tree 的 near optimal heuristics，期望可以接近最理想的其中一個目標。

4.2 最佳覆蓋優先演算法(The best coverage first algorithm)

首先提出的是最佳覆蓋優先演算法(The best coverage first algorithm)簡稱 BCFA。BCFA 是從只有一個 node (sink node)的初始 tree 開始建造骨幹，BCFA 透過增加一個不在 tree 之中的 node 來擴大目前的 tree，因此被選擇加入 tree 的 node 必須是自身通訊範圍內覆蓋最多 non-tree nodes 的 node。

BCFA 的演算法形式描述如下：

BCFA 演算法：

- Input: 包含 sink node s 在內的 n 個 sensor nodes N 。
- Output: 一個包含 s 與 backbone nodes 的 set T 。
- Step 1. 首先，將 s 加入 T 。
- Step 2. 選擇一個自身通訊範圍內覆蓋最多 non-tree nodes 的 node。也就是說，在 $N-T$ 之中選擇一個 node p ，使得 $|C(v) \cap (N-T)|$ 最大，此處 $C(v)$ 代表在 v 的通訊範圍內覆蓋的 nodes。如果通訊範圍內覆蓋最多 non-tree nodes 的 node 不只一個，則在以上這些 nodes 之中選擇一個 node p ，使得 $\sum_{u \in V} d_T(u, s)$ 的值最小。再來，如果滿足上面這些條件的 nodes 仍然超過一個，則從其中挑出一個 node p 使得 node p 到 T 之間的 Euclidean distance 最大。否則，從滿足上述所有條件的 nodes 之中任意選擇一個 node p 。
- Step 3. 將 node p 加入 set T 。
- Step 4. 如果 T 的通訊範圍已經覆蓋住所有在 N 之中的 sensor nodes，則到 Step 5。否則，回到 Step 2。
- Step 5. 結束。

BCFA 的時間複雜度(time complexity)將被分析並且呈現在下面的定理中。

Theorem 1: BCFA 的時間複雜度為 $O(n^3)$ ， n 是網路中 sensor nodes 的數量。

Proof: 明顯地，Step 1 與 Step 3 花費常數時間(即 $O(1)$ time)。於 Step 2，每個 sensor node 必須被搜尋一次，但是不一定每次都需要搜尋到每個 sensor node，所以最大的搜尋次數是 $O(n^2)$ 。Step 4 可以很簡單地用 $O(n)$ 的時間實行。當每次增加 backbone node 的時候，會將上面提到的時間複雜度相加，至多增加 n 次，所以最後再乘以 n 。則 BCFA 演算法的時間複雜度必小於 $(1+n^2+1+n) \times n$ ，也就是 $O(n^3)$ 。

4.3 最少延遲優先演算法(The least delay first algorithm)

這一小節將呈現第二個演算法，最少延遲優先演算法(The least delay first algorithm)，簡稱 LDFA。相較於 BCFA，LDFA 使用了截然不同的方法。

BCFA 透過反覆地加入一個新的 node 來建造一棵我們想要的 tree。相反地，LDFA 每次消除一個 node。最初，所有的 nodes 被保留在一個集合

裡，LDFA 在集合裡選擇其中一個 node 使得 $\sum_{u \in V} d_T(u, s)$ 的增加量最少。

LDFA 的演算法形式描述如下：

LDFA 演算法：

- Input: 包含 sink node s 在內的 n 個 sensor nodes N 。
- Output: 一個包含 s 與 backbone nodes 的 set T 。
- Step 1. 令 T 等於 N 。
- Step 2. 依照 sensor node 本身 ID 的大小順序，從 ID 最大的開始，嘗試一次移除掉一個在 T 之中除了 s 以外的 node，並計算若移除此 node 後所產生的新的值 $q = \sum_{u \in V} d_T(u, s)$ ，然後再把此 node 加入 T 。一直進行到 T 之中除了 s 以外的每一個 node 都有計算過 q 值。如果移除某個 node 後，會使網路變成 *disconnected*，則將此時 q 的值設為 n^2 。選出一個 node p 可以使得 q 的值是最小的。如果 q 的值是最小的 nodes 不只一個，則在滿足上述條件的 nodes 之中任意選擇一個 node p 。
- Step 3. 如果 node p 使得 q 小於 n^2 ，則到 Step 4。否則，到 Step 5。
- Step 4. 將 node p 從 set T 之中移除，然後回到 Step 2。
- Step 5. 結束。

LDFA 的時間複雜度(time complexity)將被分析並且呈現在下面的定理中。

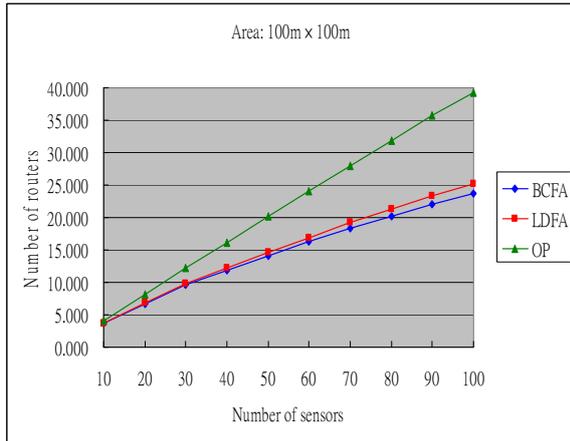
Theorem 2: LDFA 的時間複雜度為 $O(n^4)$ ， n 是網路中 sensor nodes 的數量。

Proof: 於 Step 1，令 T 等於 N ，需要 $O(n)$ 的時間。Step 2 要計算 $q = \sum_{u \in V} d_T(u, s)$ 的值的時，使用 Dijkstra's shortest path algorithm，花費的時間複雜度為 $O(n^2)$ ，而為了要計算 T 之中除了 s 以外的每個 node 的 q 值，至多需要的搜尋次數是 $O(n)$ ，所以 Step 2 的時間複雜度為 $O(n^2 \times n)$ ，也就是 $O(n^3)$ 。很明顯地，Step 3 與 Step 4 花費常數時間(即 $O(1)$ time)。當每次將 node p 從 set T 之中移除的時候，會將上面提到的時間複雜度相加，至多移除 n 次，所以最後再乘以 n 。則 LDFA 演算法的時間複雜度必小於 $(n + n^3 + 1 + 1) \times n$ ，也就是 $O(n^4)$ 。

五、模擬實驗

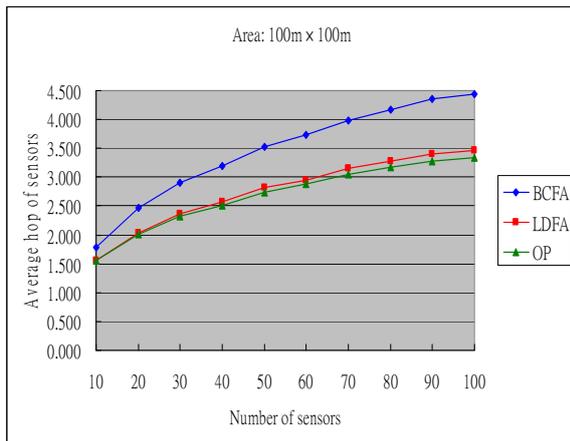
本實驗使用 C 語言撰寫程式，在配備 Intel Core 2 Duo E6300 CPU 和 2GB DDR2-800 RAM 的個人

電腦上進行模擬，每次模擬均視需求的 Sensor 數量，以隨機亂數產生不重複位置的 Sensor，散佈於有限制大小的正方形或長方形的場地(Area)之中。其中每個 Sensor 傳輸半徑固定為 10m，並且保證所有 Sensor 均能完整連接之，然後分別使用 BCFA 與 LDFA 建造骨幹後，與 Ma 的 Ordinal Pruning[8] 做效能上的比較。實驗結果為統計在不同的 Sensor 個數的情況下，各做 1000 次模擬的平均值。此處所謂的 average hop of sensors 是把所有 Sensor 到 Sink 的 hop 相加總後，除以 Sensor 的個數。



圖二：不同網路密度下所需要的 Router 個數 (Area: 100m x 100m)

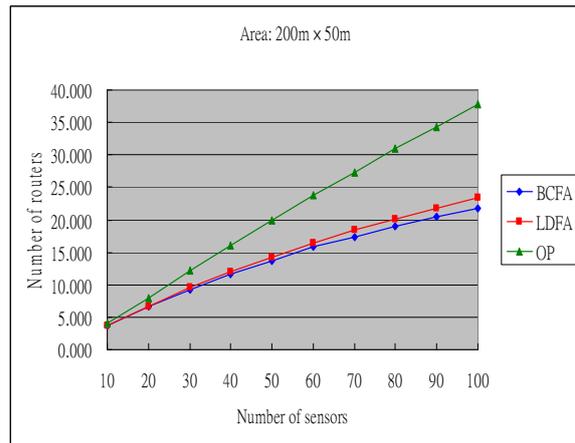
由圖二可以發現到 BCFA 與 LDFA 均能有效地大幅降低需要成為骨架上的 Router 的 Sensor 數量，而且與 OP 相比，Router 的數量非常少。且隨著部署的 Sensor 個數的增加，也就是較密集部署時可節省電量的 Sensor 數量也會隨之增多。



圖三：不同網路密度下的 Average hop (Area: 100m x 100m)

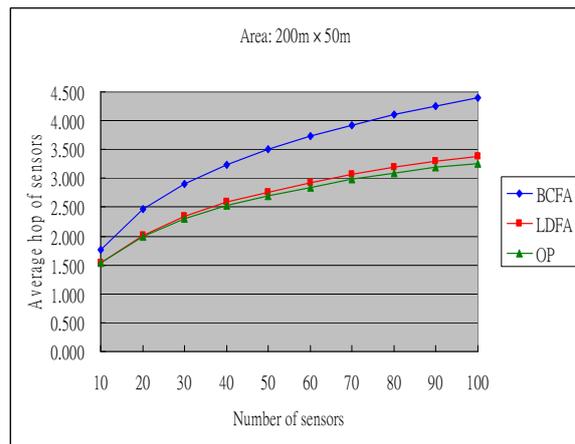
由圖三得知 BCFA 的 Average hop(即代表所需 delay)比起 OP 有少量的增加，但是值得注意的是 LDFA 的 Average hop 非常地低，與 OP 十分相近。我們也發現到，BCFA 與 LDFA 在 Sensor 的佈建更

加稠密的情況下，會有更好的表現。



圖四：不同網路密度下所需要的 Router 個數 (Area: 200m x 50m)

由圖四可知當部署場地從正方形變成長方形的時候，BCFA 與 LDFA 在減少 Router 數量的效果上仍然比 OP 好很多。而且相較於正方形的場地，效果會更好一些。



圖五：不同網路密度下的 Average hop (Area: 200m x 50m)

同樣地，由圖五我們發現到 BCFA 與 LDFA 即使在部署場地從正方形變成長方形的時候仍然有保持水準的表現，LDFA 的 Average hop 依舊非常低。

六、結論與未來展望

由模擬實驗結果可以發現到，本篇論文提出的 BCFA 與 LDFA 兩種演算法相較於 Ordinal Pruning 均可大量減少需要成為 Backbone 的 Sensor 數量，達到非常高的省電效益，但相對只增加一些或是極小的通訊延遲。由實驗也可得知隨著 Sensor 個數的增加，可節省電量的 Sensor 也會隨之增長。

LDFA 在減低通訊延遲上面有非常好的表現，但是其時間複雜度亦非常之高，如果 Sensor 數量很大的時候，在一般機器上面運行的速度有待加強。

目前我們討論的演算法是架構在每個 Sensor 的傳輸範圍都是固定的情況上面，且在真實的無線感測網路系統上，通常是各個 Sensor 各自做分散式計算，而不像此演算法掌握全部 Sensor 的位置後再做集中式的計算。如果能夠應用分散式處理也就是每個點透過周圍狀況來決定自己是否要成為 Backbone，這樣不只在時間效益上會有更佳的處理速度，且對於真實無線感測網路上也會比較符合網路狀況。

未來將在更進一步地詳細考慮各式因素後，加以改進演算法的效率，使之能在異質感測網路上面實作發生最大效用。

七、參考文獻

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, March, 2002.
- [2] A.M. Zoubir, I.J. Chant, C.L. Brown, B. Barkat, and C. Abeynayake, "Signal processing techniques for landmine detection using impulse ground penetrating radar," *IEEE Sensors J.*, vol. 2, no. 1, pp. 41-51, 2002.
- [3] Xiaojiang Du and Fengjing Lin, "Efficient multi-class routing protocol for heterogeneous mobile ad hoc networks," in *International Conference on Broadband Networks*, 2005, vol. 1, pp. 651-658.
- [4] Rituparna Ghosh and Stefano Basagni, "Napping backbones: energy efficient topology control for wireless sensor networks," *IEEE RWS*, 2006, pp. 611 - 614.
- [5] X. Wang and T. Berger, "Self-organization redundancy-cellular architecture for wireless sensor networks," *IEEE WCNC*, 2005, pp. 1945-1951.
- [6] X. Wang and T. Berger, "Topology control, resources allocation and routing in wireless sensor networks," In *Proceedings of 12th IEEE /ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS)*, 2004.
- [7] Y. C. Tseng and M. S. Pan, "Quick Convergecast in ZigBee/IEEE 802.15.4 Tree-Based Wireless Sensor Networks," *ACM International Workshop on Mobility Management and Wireless Access Protocols 2006*, pp.60-66.
- [8] Jian Ma, Min Gao, Qian Zhang, Lionel M. Ni, and Wenwu Zhu, "Localized Low-Power Topology Control Algorithms in IEEE 802.15.4-based Sensor Networks," *IEEE ICDCS*, 2005, pp. 27 - 36.