

VG-Based Mobicast : A Mobicast Routing Protocol Based on Virtual Grid for Wireless Sensor Networks

國科會計劃編號：NSC 94-2213-E-156-001

Ren-Yuh Lue

Depart. Computer and Information Science
Aletheia University, Taiwan
am911795@email.au.edu.tw

Gwo-Jong Yu

Depart. Computer and Information Science
Aletheia University, Taiwan
yugj@email.au.edu.tw

Abstract

本篇論文融合 Mobicast 架構及 Virtual Grid 的概念提出一個新的方法，稱之為 VG-Based Mobicast Routing Protocol，使用 Virtual Grid 的概念架設 Mobicast 的架構，另外搭配協助感測模式，不讓每一個 Grid 都有感測器進行感測，期望此方法可以達到下述四個成果。(1)利用 Grid Size 和感測半徑之間的比例關係，搭配 Help Sensing Mode，達到節省電量消耗並延長網路生命週期的效益；(2)利用 Virtual Grid 的概念，藉由 Grid Head 來幫助無線感測網路的管理，並搭配 Mobicast Framework 以減少訊息傳輸的時間(Low Latency)；(3)提高感測和傳輸的成功率；(4)針對因 Coverage 所造成的 Sensing Hole 問題提出相對應的處理。

1. Introduction

無線感測網路（亦稱為無線感知網路）的應用範圍非常廣泛，依其目的不同，本文將它分成兩個部分。

一、**環境偵測**：在一個地區內，佈下許多個相同的感測器，它們可以感測環境中某些指定的物體或事件的變化。例如當溫度感測器發現溫度急速上升，便可以偵測出感測器所在的位置可能發生火災，並且盡快回報給基地台做出該有的應對措施。

二、**危險探測**：若有個人要往某一個方向移動，可以預先在下一個時間點他可能到達的地方佈下一定數量的感測器，利用這些感測器，可以探測出該處是否有危險，並對這些危險做出應對。例如感測器感測到有障礙物並通知他，他就可以改道避開。

無線感測網路最大的兩個優點就是佈點容易及成本低廉。某些文章提過可以利用飛機灑點的方式分佈感測器，這樣的佈點方式，成本較為低廉，但缺點是分佈不均，管理上也較為不便。也曾經有文章因為考慮 Coverage 的問題，精密地計算過位置後再行佈點，計算和比較佈點時的精確性都需要消耗電量或時間，致使這種佈點方式會造成較高的成本，但是可以使管理負擔大幅降低，大部份的文章採用隨機佈點的方式，隨機佈

點固然方便，卻會造成某些地區可能沒有感測器進行感測，若事件發生在該範圍內，將無法得知有關於該事件的任何資訊，本篇文章的內容也考慮了所佈下的眾多感測器是否能涵蓋全域。

無線感測網路成本低廉的其中一個原因是這些被佈下的感測器都不能充電，當它們的電量一旦耗盡，只能選擇重新佈點，或是使用可移動的感測器去取代電量耗盡的感測器，前者會不斷消耗佈點成本，也會與無線感測網路的方便性相抵觸，後者則會牽扯出不少管理方面的問題，也會增加電量的消耗，並縮短網路的生命週期，而本論文研究目的之一即為延長無線感測網路的生命週期。

Mobicast 是一種有條件的 Multicast 繞徑協定，通常被使用於感測會移動的物體或事件，例如會蔓延的火災，或是進入感測區域中的人或車，一般稱這樣的感測為目標物追蹤 (Target Tracking)。首先在一個空間裡佈下一定數量的感測器，當某一個移動中的物體或事件隨時間的變動而移動時，其周圍的感測器會因為事件的發生或物體的移動而打開它們的感測裝置，這麼做會讓大家感覺網路中進行感測的感測器正隨著事件的移動而移動，藉此偵測想要得知的資訊，並將偵測到的資訊回傳給鄰近的 Base Station 或 Sink，讓該處的人員對其做出反應，為了提高傳輸的即時性，網路中所有的感測器的通訊裝置都是開著的。

本文會使用一些在之前的論文中被提出的技術，主要有以下兩種。

一、Mobicast Framework：

Qing-Feng Huang [1]使用的 Mobicast 繞徑協定架構，現在研究 Mobicast 的學者大多參考他的論文，他的架構是以感測到的移動物體為中心，以適當半徑畫一個圓，把這個圓稱為 Delivery Zone，所有在這個圓裡的感測器在感測到物體的移動情形時，都會把搜集到的資料或訊息向外廣播出去。

以下一個時間點物體的所在位置為圓心，以更大的半徑畫圓，此時所成的圓稱為 Forwarding Zone，所有在此範圍裡的感測器在接收到從 Delivery Zone 廣播出來的訊息時，會再將這些訊

息廣播出去，這個動作被作者稱之為 Forward。

藉由不斷地 Forward 訊息，除了可以通知鄰近的感測器打開感測裝置追蹤目標物，也可隨時將最新的資訊回傳給基地台或距離較近的 Sink，如果這個移動的物體代表著一件災害的發生，傳送訊息的時間越短，就越能防止災情擴大，而這也是本文想要做到的。

二、Virtual Grid

一般的作法是將感測器所在的二元平面劃分成一個個的方格，主要優點是簡單、有規則且方便管理，也由於有規則地劃分感測器的所在區域，利用這樣的規則，可以去定義所有感測器之間的相關位置，藉以協助繞徑。

Qing-Feng Huang 提出了三篇和 Mobicast 有關的論文[1][2][3]，但是這些文章都存在著一個共同的問題，即均未考慮物體移動的方向性，換句話說，這個 Mobicast 架構僅能被使用於感測持續在同一方向上前進的物體，如果物體轉彎，就會失去追蹤的效果。

Yuh-Shyan Chen 和 Shin-Yi Ann 的文章[4]裡提到他們使用蛋型的 Forwarding Zone，並且將物體的轉向也列入考慮，但模擬出來的預測成功率大約只達到 80%，表示他們無法完全掌握正在移動中的物體，所以無法達到 100%的預測效果。

這篇論文的目的是希望研究出一個更好且實用性更高的 Mobicast 協定，不僅要能完全捕捉到移動物體的繞徑動向和資訊，也要盡量地達到即時傳輸的效果，最重要的就是要能延長整個 Mobicast 繞徑協定的生命週期。

本文在第二章裡會描述如何使用前面所提到的相關技術。然後在第三章提出我們的 Mobicast 繞徑協定，並詳細說明細節的部分。在第四章，為了驗證這篇論文的實用性，我們會對提出的方法進行模擬，並且分析模擬的實用性。最後會在第五章提出結論。

2. Related Work

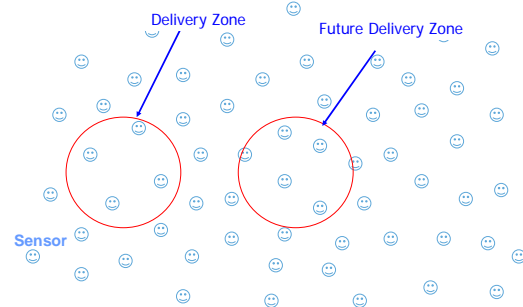
2.1 Mobicast Framework

在 Qing-Feng Huang 的論文[1]裡提到了一個 Mobicast Framework 的技術，使用這個技術可以讓在移動中的物體可能前進方向上的感測器及時打開感測裝置進行感測。假設在時間為 T_0 時發生了一個事件，以這個事件為中心，設立一個半徑畫圓，這個圓應該盡可能地涵蓋所有可能感測到此事件的感測器，作者稱這個圓為 Delivery Zone，處於這個圓裡面的所有感測器，會把感測到的事件訊息 (Delivery Message) 向 Delivery Zone 的外面廣播。

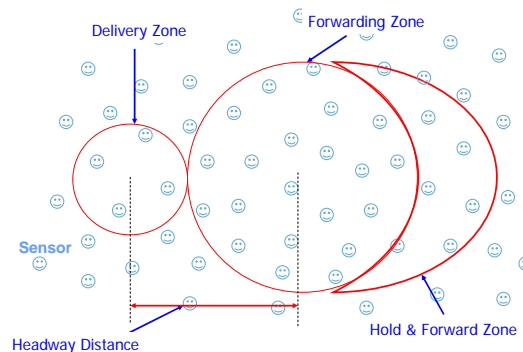
假設此事件發生的下一個時間點為 T_1 ，估算出在 T_1 時事件發生的所在位置，以另一更大的半徑畫圓，作者稱這個圓為 Forwarding Zone，處於這個圓裡面的所有感測器，在接收到 Delivery Message 後，會再將此訊息朝這個圓的外面廣

播，這時被廣播出去的訊息稱為 Forwarding Message。

Forwarding Zone 的周圍有一部份感測器在接收到 Forwarding Message 後，會將此訊息留住，不再向外廣播，這個部份的感測器涵蓋的範圍作者稱之為 Hold and Forward Zone，不屬這些區域內的所有感測器，會忽略掉它們接收到的所有訊息，這麼做的目的是不讓距離事件極遠的感測器也打開感測裝置準備感測事件，如果物體或事件的移動速度不是很快，又提前打開位於移動方向上感測器的感測裝置，勢必會使這些感測器持續處於等待感測的狀態下，感測裝置也持續地為感測器消耗不必要的電量，在等待事件或物體到來的時間裡造成無謂的電量浪費。



(a) Delivery Zone & Future Delivery Zone



(b) Delivery Zone, Forwarding Zone, and Hold and Forwarding Zone

Figure 1 Mobicast Framework

Figure 1(a)和 Figure 1(b)是原作者所提出的 Mobicast 架構，但是這個架構僅能處理等速且等向移動的事件或物體，本篇論文會將這個觀念應用在 Virtual Grid 之中，並且試著改進這個缺點。

2.2 Virtual Grid

在 Ya Xu, John Heidemann 和 Deborah Estrin 的文章[5]中，Virtual Grid 的作法是將感測器所在的二元平面劃切割成一個個相同大小的方格，它的優點是簡單、有規則且方便管理，也由於有規則地劃分整個網路，利用這樣的規則，可以去定義所有感測器之間的相關位置，藉以協助繞徑。

在 VG-Based Mobicast Routing Protocol 中，

是以基地台建立一組虛擬的 x 軸和 y 軸，藉以產生虛擬的座標計算感測器之間的相對距離關係，再以長度為 d 的邊長由內而外地建立一組 Virtual Grid，這個長度為 d 的邊長，本文稱之為 Grid Size。本文會利用 Virtual Grid 的規則去定義感測器之間的相關位置，由於在 Virtual Grid 建立完成之後，所有方格的大小均相同，只要對不同的 Grid 編號，就可以知道某一個感測器是位於哪一個 Grid 裡，即當 Virtual Grid 建立完成以後，差不多已經知道所有感測器和基地台之間的相對位置，知道了這樣的資訊以後，將會使得繞徑更為方便。

3. Real-Time Mobicast Routing Protocol

3.1 Protocol Overview

假設感測器的佈點相當地均勻，這樣的假設有個好處，將不會有所謂的 Coverage Hole 出現，Figure 2(a)中紫色虛線所涵蓋的區域中沒有任何的感測器存在，一般稱這個區域為 Coverage Hole。

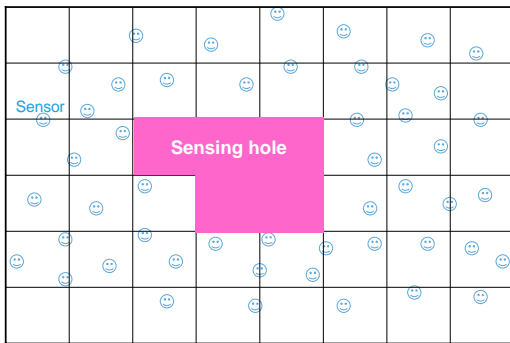
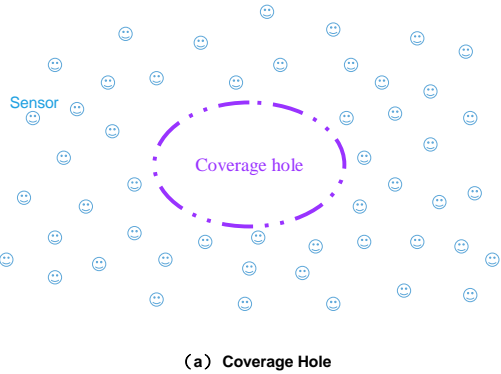


Figure 2 Coverage Hole & Sensing Hole

如果佈點情形不如本文假設般地完美，代表 Coverage Hole 存在於網路之中，那麼待 Virtual Grid 建立完成之後，可能會發生某些 Grid 中沒有感測器的情形，因而導致 Sensing Hole 的出現。Figure 2(b)即是因為 Figure 2(a)的 Coverage Hole 所形成的 Sensing Hole，由於整個網路已經被切割成大小相同的方格，可以藉此計算出 Sensing Hole 的大小，本文視這個情形為一個特例，而至

於遇到如同 Sensing Hole 這樣的特例時該如何處理，在稍後的章節裡，我們會做簡單的介紹。

另外，假設所有 Sensor 已經透過某種方法(例如：GPS[6]、MDS[7]或三角定位法[8]等定位協定或設備。)得知自己的位置資訊。當一個感測器想要對外連繫時，可以藉由封包的廣播，告知所有位於自己通訊範圍裡的感測器自己的位置資訊，使得 Grid 的內部管理和 Grid 之間的連繫得以更方便地進行。

3.1.1 Distribution Method

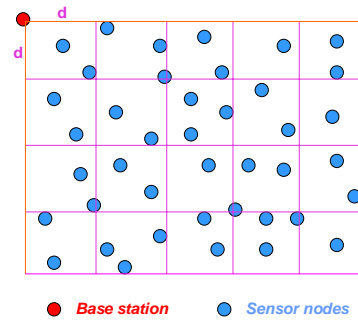
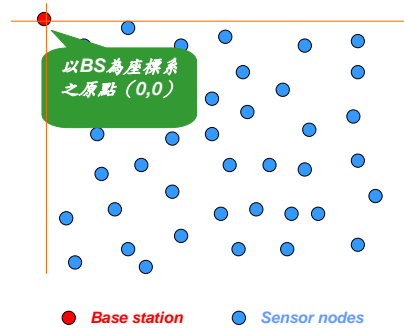


Figure 3 Settings of Virtual Grid

我們建立 Virtual Grid 的方法，如 Figure 3(a)和 Figure 3(b)所示，是以 Base Station 為中心，畫虛擬的 x 軸及 y 軸各一條，從 x 軸向上下，y 軸向左右延伸，以長度為 d 的距離畫垂直和水平線，使其交錯形成大小相同的方格，利用這些大小相同的方格將感測器所在的空間切割成許多相同大小的區塊，每一個區塊，都是一個 Grid。

如同 Figure 3(b)所示，本文定義 Grid Size 的意義為 Grid 的邊長，即為 d ，我們令 d 和感測器的 Sensing Range 的比值 (d/R_s) 要小於或等於 $1/\sqrt{2}$ ，這樣的假設可讓一個 Grid 裡只需要任何一個感測器打開感測裝置進行感測，其感測範圍就足以涵蓋整個 Grid，值得注意的是，每一個感測器的 Sensing Range 是固定的，所以 VG-Based Mobicast Routing Protocol 能夠調整的是 Grid Size，Grid Size 的大小會對 VG-Based

Mobicast Routing Protocol 的效能有所影響。

本文定義 Grid Cells 的意義為整個網路中的 Grid 總數，令 n 為整個網路中的 Grid 總數， n 會和 d 呈現反向的比例關係，換句話說，Grid 總數越多，則表示 Grid Size 越小。

當整個感測器所在的空間已經被切割成許多大小相同的 Grid 時，每一個感測器的所在位置會變得容易計算，可以對每一個 Grid 編號，藉以決定感測器和 Base Station 的相對位置，使得繞徑的建立更為容易，每一個 Grid 都是一個集中式的管理模式。本文在一個 Grid 裡使用一個 Head 管理 Grid 的運作，而這個 Head 則是由該 Grid 中的感測器輪流擔任，這個情境稱之為 Grid 的成員管理模式，可以藉由 Grid 的成員管理模式，達到節省電量消耗和接近 100% 涵蓋全域的感測效果。

3.1.2 Management Method

本文採用分散式管理模式，視每一個 Grid 為一個獨立的群組，如此一來不但可以加強省電的功能，也是為了在發生事件時，Base Station 能夠盡可能快地接收到警告訊息。想像每一個 Grid 即為一個 Cluster，裡面有一個 Grid Head 負責管理其他同在 Grid 內部的感測器，即每一個 Grid 的運作會由該 Grid 中的 Head 負責指揮，為了使 Grid Head 不論位於 Grid 內部何處均可和本身的上、下、左、右四個 Grid 內部的 Head 連繫，本文假設 d 和感測器的通訊半徑 (R_c) 之比值為 $1/\sqrt{5}$ ，但是這會導致不同 Grid 之間的通訊會互相干擾，為了避免這個問題，在 Real-Time Mobicast 繞徑協定裡讓不同的 Grid 使用不同的頻道進行通訊，Grid 之間便不會互相干擾。

在 VG-Based Mobicast Routing Protocol 裡，我們對一個 Grid 中的感測器定義了四種狀態。第一種是 Head，僅開啟通訊裝置而不開啟感測裝置，因為 Head 最主要的工作即為對 Grid 裡的成員發號施令。它的工作相當地多，電量的消耗也相對地較大，假如持續由某一個感測器擔任這個職務，電量很有可能在短時間內消耗殆盡，這也是 Head 不開啟感測裝置以及必須由 Grid 中的感測器輪流擔任的原因。第二種是 Idle Node，它的感測裝置也是關著的，在 Virtual Grid 尚未建立完成之前，已經打開通訊裝置但未競爭到 Head 職務的感測器均處於這種狀態。第三種是 Sensing Node，這種感測器處於正在感測的狀態中，由於本文的設定，一個 Grid 裡只會有一個這種狀態下的感測器，為了感測移動的物體，它的感測裝置是開著的。最後一種是 Sleeping Node，通訊裝置和感測裝置均尚未開啟。處於第二和第三種狀態的感測器，它們的通訊裝置都必須持續地打開，這是因為他們必須隨時準備好接收 Head 所給的指令並予以執行，假設某個 Idle Node 接收到 Head 傳送過來的指令，指令內容是將感測裝置打開並進行感測工作，此 Idle Node 便會遵從指令打開感

測裝置並開始感測，此時 Idle Node 的狀態也從 Idle 轉變成為 Sensing。

Head 的工作摘要如下：

- 一、控制 Grid 中感測器工作的輪替。
- 二、通知感測器打開感測裝置（電量越大者優先權越高）。
- 三、作為 Grid 和 Grid 之間通訊的橋樑。
- 四、傳送 GMP 封包給鄰近的 Grid。
- 五、將感測到的資訊回傳給 Base Station。
- 六、儲存並更新 Head Table。

Head Table 裡的資訊被用來協助繞徑，它的欄位包含有 Grid 成員感測器的 ID、狀態，和所屬的頻道，以及鄰近 Grid 的 Head 位置及所屬頻道，當然也包含了基地台的位置資訊。

所有感測器並不會持續地維持在同一種狀態下，會根據本文中訂定的規則不斷輪替，在一個 Grid 中，假設還沒有任何一個 Head 的出現，稱這個 Grid 尚未建立完成，這個 Grid 裡也不會有感測器進行感測。當某個感測器在此時打開通訊裝置，若沒有任何裝置發送封包，就更改本身的狀態，成為這個 Grid 中的首個 Head。在一段時間後，若有第二個感測器打開感測裝置並聽到 Head 的運作時，會要求和 Head 比較電量，電量較大者成為 Head，較低者則回復 Grid 成員的身份，依此類推，比較的動作會一直持續到每一個 Grid 的成員都和暫時擔任 Head 的感測器比較過為止，這個動作在本文中稱之為競爭 Head，在比較的過程裡，比較雙方會消耗一定的電量，為避免電量的浪費，我們規定週期時間內每一個感測器只能競爭一次，當所有在這個 Grid 裡的感測器都競爭過了，此時的 Head 就是這個 Grid 裡真正的第一個 Head，其他感測器都是 Grid 成員，必須聽其號令。若有某一個感測器位於兩個 Grid 的邊界上，這個感測器會和這兩個 Grid 之中率先建立完成的 Grid 裡的 Head 註冊，成為該 Grid 的成員。

當真正的第一個 Head 已經出現，本文稱這個 Grid 已經建立，此時哪一個感測器應該打開感測裝置或是哪一個感測器應該休息，都由 Head 負責管理，Head 為 Grid 中電量最大者，通常第一個打開感測裝置的感測器，電量僅次於 Head。

這個管理機制會遭遇到下列四個問題，(1) Grid 成員電量即將耗盡；(2) Grid 成員突然毀損；(3) Head 電量即將耗盡；(4) Head 突然毀損。本文把 (1) 和 (2) 歸於同一項。

一、Grid 成員電量即將耗盡或突然毀損：當某個 Grid 成員感測到事件或物體時，會把發現的資訊 (Discovery Message) 封裝在管理封包 (Management Packet, MP) 裡傳送給 Head，此時 Head 會回傳回覆管理封包 (Replication Management Packet, RMP) 並且順便調查其他成員的剩餘電量來更新 Head Table，可以對每一個

擔任 Head 的感測器的電量設立一個 Threshold 值，如果發現有成員的電量高出這個 Threshold 值，Head 和這個成員的狀態會互相交換，一般稱這個動作為換手 (Handover 或 Handoff)。

假設 Head 對某個成員發出調查電量的封包，這個成員久久不曾回覆，Head 會等候一段時間之後再度重傳，重覆數次之後若該成員仍未給予回覆，可以得知這個成員已經毀損，此時 Head 只要直接將此成員在 Head Table 上除名即可。

假設某個成員知道自己的電量即將耗盡，會傳送 MP 封包通知 Head 並轉換自己的狀態，若 Head 收到此 MP 封包不需回覆，直接將該成員於 Head Table 上除名即可，若這個 MP 封包在傳送過程中毀損，且該成員已無多餘電量再度傳送 MP 封包，即此成員電量直接耗盡，本文將這種情形和上述成員已毀損的情形視為相同。

二、Head 的電量即將耗盡：同樣地，本篇論文對每一個感測器的電量設立一個低的 Threshold 值，當 Head 的電量低於這個 Threshold 值的時候，Head 會根據 Head Table 上的電量資訊選擇一個適當的成員與其溝通，然後進行換手的動作，在換手的過程中，除了狀態的轉變和工作的交接之外，同時也要將 Head Table 一併轉移給新的 Grid Head。如果這個合適的成員一直沒有回應，則 Head 會在等待一段時間後，根據 Head Table 裡的電量資訊，找尋下一個合適的成員。在 Real-Time Mobicast Routing Protocol 裡，所有的工作排班都是根據電量來進行的。

三、Head 突然毀損：此時這個 Grid 又變回尚未建立的階段，所有的感測器從這個時候開始，重新再進行一次競爭 Head 的動作。

3.2 Packet Format

目前使用到的封包分為八種，分別是 Grid 訊息封包 (GMP)、回覆 Grid 訊息封包 (RGMP)、傳遞訊息封包 (DMP)、回覆傳遞訊息封包 (RDMP)、管理封包 (MP)、回覆管理封包 (RMP)，和要求換手封包 (HRP) 及回覆要求換手封包 (RHRP)。

Grid 訊息封包 (Grid Message Packet, GMP) 用以調查周圍是否有 Grid 為空，當一個 Grid 建立完成後，Head 會發出 GMP 告知上下左右四個 Grid 自己的位置、所在 Grid 的 ID 和使用的 Channel，以便連結鄰近 Grid 之間的通訊，它有四個欄位，包含標頭、Grid ID、Grid Channel，和該 Grid Head 所在的位置，利用 Figure 5 來說明，假設 Grid (5)，即編號為 5 的 Grid，欲傳送 GMP 給 Grid (6)，Grid (6) 的 Head 在收到 GMP 後會回覆一個 RGMP (Replication Grid Message Packet, RGMP) 的封包給 Grid (5) 的 Head，對 Grid (6) 來說，RGMP 不只用來確認收到 GMP，也同時將自己 Head 的所在位置回傳給 Grid (5)，RGMP 和 GMP 的欄位資訊几乎是

完全相同的。

GMP	ID	Location	Channel
-----	----	----------	---------

RGMP	ID	Location	Channel
------	----	----------	---------

(a) GMP & RGMP

DMP	SR	DR	RR	DeM
-----	----	----	----	-----

RDMP	SR	DR	RR	RM
------	----	----	----	----

(b) DMP & RDMP

HRP	HR1	HR2	HT
-----	-----	-----	----

RHRP	HR1	HR2
------	-----	-----

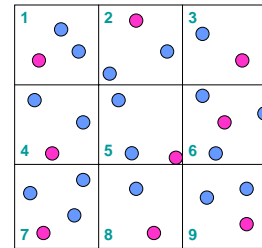
(c) HRP & RHRP

MP	HR	SenR	DiM	MM
----	----	------	-----	----

RMP	HR	SenR	DiM	MM
-----	----	------	-----	----

(d) MP & RMP

Figure 4 Packet Format



● Grid head ● Idle nodes

Figure 5 Example of Transmission on Virtual Grid

傳遞訊息封包 (Delivery Message Packet, DMP) 主要的用途是不同 Grid 之間的訊息傳送。在 DMP 中有五個欄位，分別是標頭、來源端記錄 (Source Record, SR)、目的端記錄 (Destination Record, DR)、中繼點記錄 (Relay Record, RR)，以及欲傳送的訊息 (Delivery Message, DeM)，SR 裡包含了第一次發出此訊息 Grid 的 Grid ID、Grid Channel，和此 Grid 裡的 Head 所在位置，DR 和 RR 裡包含的資訊和 SR 大致相同，但是 DR 裡的資訊是屬於目的端的，RR 則是在記錄此封包經過的所有中繼點的資訊，當目的端收到來源端所傳送的 DMP 封包，會將 RR 裡的記錄存成一個 Table，並從裡面選擇一條較佳的路徑回傳一個

RDMP (Replication Delivery Message Packet, RDMP) 封包, 和 DMP 相比之下, RDMP 的欄位少了 DeM, 增加了一個新的欄位, 稱之為反應訊息(Response Message, RM), RM 是 Base Station 或 Sink 在接收到 DeM 之後所做出的相對回應, 例如 DM 的內容是“發現一個入侵者”, RM 的內容可能是“持續地追蹤它的動向, 並隨時向基地台回報。”

當某個 Grid 的 Head 發現自己的電量即將消耗殆盡, 或是低於一個門檻值的時候, 它會根據 Head Table 裡的電量資訊, 對最適當的感測器發出要求換手封包 (Handover Request Packet, HRP), HRP 裡包含有標頭、原 Head 的記錄 (HR₁)、接手 Head 的記錄 (HR₂), 還有 Head Table (HT), HR₁ 裡描述原 Head 的位置和剩餘電量, HR₂ 則記錄欲接手的感測器的位置。收到 HRP 的感測器, 會立即轉換成 Head State, 並回覆一個 RHRP (Replication Handover Request Packet, RHRP) 封包給原本的 Head, RHRP 裡沒有 HT 這個欄位, 其餘內容則和 HRP 相同。此時原 Head 應設立一個終止時間 T_r, 若原 Head 在 T_r 之前接收到 RHRP, 則將自己的狀態轉換成一般感測器, 卸下 Head 的職務, 若超過了終止時間 T_r, 原 Head 仍然沒有接受到 RHRP 封包, 表示欲接手的感測器可能損壞, 那麼原本的 Grid Head 就必須從 Head Table 裡尋找下一個適合的感測器傳送 HRP, 假使原 Head 傳送 HRP 給數個感測器皆失敗, 且自己的電量已然耗盡, 那麼 Grid 則會恢復到沒有 Head 前的情形, 所有的 Sensor 必須再次進行搶當 Head 的動作, 重新建立此 Grid 的管理機制。

管理封包 (Management Packet, MP), 是 Grid 裡功能最多的封包, 在 Grid 之中, 除了 Head 的換手之外, 其他的管理都要靠它來協處理, 例如電量管理、感測管理, 和工作排班的管理。MP 裡包含有標頭、Head 記錄 (Head Record, HR)、感測器記錄 (Sensor Record, SenR)、發現訊息 (Discovery Message, DiM), 以及管理訊息 (Management Message, MM)。當某一個感測器感測到事件的發生, 會將感測到的資訊記錄在 DiM 裡, 並傳送 MP 給 Head 請示該如何反應, 此時 MM 欄位為空, Head 收到後會回覆 RMP 封包 (Replication Management Packet, RMP) 給此感測器, RMP 的欄位則和 MP 相同, Head 會將指令存放在 MM 裡, 當感測器接收到 RMP 時, 會比對 DM 裡的資訊是否無誤, 若確定無誤則立即執行 MM 裡的指令。MP 不只可由感測器主動發出, 也可由 Head 發出, 當 Head 有指令要交付給某一感測器執行時, Head 也同樣可以發出 MP 封包, 此時 MM 欄位裡即為欲交付的指令, 而 DM 欄位為空。換言之, 在任一個 Grid 裡, 不論這個感測器的型態為何, 都可以主動發出 MP

封包, 利用 DiM 及 MM 欄位來達到不同的管理作用。

3.3 Protocol Details

前述的文章提到, 假設 Grid Size 和 Sensor 的 Sensing Range 的比值 (d/R_s) 小於等於 $1/\sqrt{2}$, 則每個在 Grid 中的 sensor, 感測範圍都可以涵蓋住整個 Grid, 可以利用 GMP 封包調查鄰近的 Grid 是否為空, 若相鄰的 Grid 為空, 而且感測器在 Grid 中所在的位置靠近 Grid 中央, 由於 Grid Size 和感測半徑的比值關係, 這個感測器甚至可以協助鄰近 Grid 進行感測, 這麼一來在同一個 Grid 裡, 只需要一個感測器打開感測裝置進行感測, 可以藉以省下部份的電量。

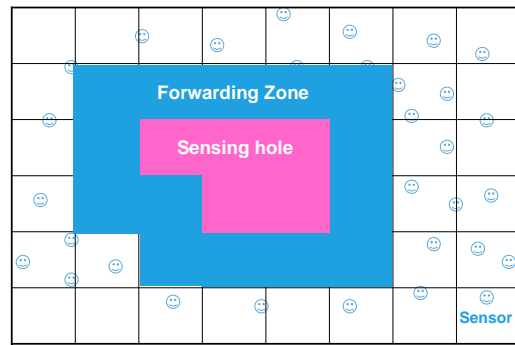


Figure 6 Sensing Hole & Forwarding Zone

在 3.1 裡提到, 如果以隨機的方式進行佈點, 可能會遇到 Sensing Hole 的問題, 本文試著把 Mobicast 的架構放大, 在移動物體即將進入 Sensing Hole 時, 讓 Sensing Hole 周遭的所有 Grid 有感測器打開感測裝置進行感測, 移動物體進入 Sensing Hole 之後, Mobicast 架構轉變為沒有 Delivery Zone, 而 Forwarding Zone 恰巧即為 Sensing Hole 周圍的所有 Grid 圍繞起來的形狀, 這個形狀並非都是矩形, 會依據 Sensing Hole 的形狀而決定, 如 Figure 6 所示。

假設 Sensing Hole 已然存在, 欲建立 Forwarding Zone, 首先必須知道哪些 Grid 位於 Sensing Hole 的周遭, 在前述的文章裡提過, Grid 建立完成後, Grid Head 會利用 GMP 封包調查鄰近的 Grid 是否為空, 對於被調查的 Grid 而言, 必須收到來自於上、下、左、右四個方向的 GMP 封包, 如果遲遲未收到某一方向 Grid 傳送過來的 GMP 封包, 可以得知該方向的 Grid 為空, 假設此 Grid 遲遲未收到由左方傳送來的 GMP 封包, 表示在這個 Grid 的左方出現了一個 Sensing Hole, 若 Sensing Hole 的範圍不大, 其周圍的 Grid Head 可以叫醒該 Grid 中距離 Sensing Hole 較近的感測器協助感測, 試圖利用自己的感測範圍將 Sensing Hole 覆蓋。若 Sensing Hole 無法被完全覆蓋, 使用上述辦法只會加速距離 Sensing Hole 較

近的感測器電量消耗的速度，為了避免這種情形發生，本文預先對每一個位於 Sensing Hole 周圍的感測器下達一個指令：“當自己位於 Sensing Hole 的周圍並且接收到 Forwarding Message 的時候，立刻將此訊息 Forward 出去。”

此時可能會遇到一個問題，在 Figure 7 中的綠色區域並非直接和 Sensing Hole 有所連接，它們都會收到來自於自己四個不同方向傳遞來的 GMP 封包，所以並不曉得它們也必須轉換成 Forwarding Zone，進行 Forward 訊息的動作。

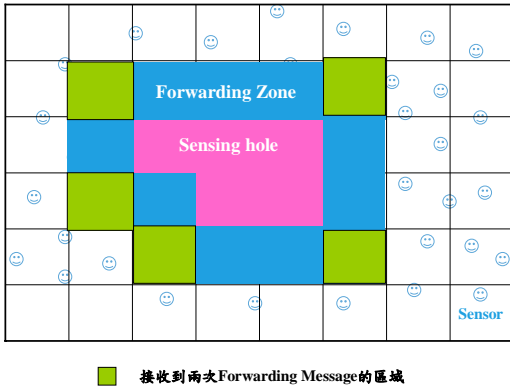


Figure 7 Settings of Forwarding Zone

本文針對這個問題想出了一個解決的方法，這些綠色的區域雖然會接收到四個不同方向傳送來的 GMP 封包，認為自己不需改變成 Forwarding Zone，但是隨後也會接收到兩個以上的方向傳送而來的 Forwarding Message，所以我們要對所有感測器下一個指令：“所有接收到鄰近兩個以上（含兩個）Grid 傳送 Forwarding Message 的 Grid，即屬於 Forwarding Zone。”

根據本篇文章提出的規則，任何一個符合下列兩個條件其中之一的 Grid，就會在移動物體靠近的時候轉變成為 Forwarding Zone。

- (1) 沒有接收到某一方向傳送的 GMP 封包，即在任何一個 Sensing Hole 旁邊。
- (2) 接收到兩個以上方向傳送過來的 Forwarding Message。

相同地，可以將類似的方法使用在 Forwarding Zone 的周圍，藉以建立 Hold and Forward Zone，當 Forwarding Zone 以及 Hold and Forward Zone 放大到足以將 Sensing Hole 完全覆蓋，無論移動物體從哪一個方向離開 Sensing Hole，都可以立即發現並且馬上進行追蹤。

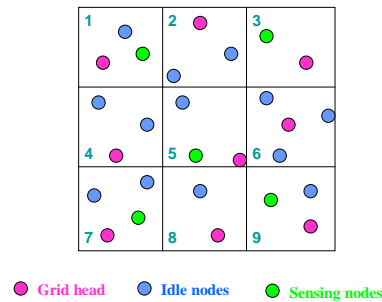
在一個 Grid 中，所有成員都必須把通訊裝置打開，這是為了方便 Grid Head 對成員發出指令，以達到即時通訊、即時反應的效果，為了讓每一個 Grid 都能和自己的上、下、左、右四個 Grid 都能進行通訊，假設 Grid Size 和通訊半徑的比值 (d / R_c) 小於或等於 $1/\sqrt{5}$ ，即每一個感測器的通訊半徑必須大於或等於 Grid Size 的 $\sqrt{5}$ 倍。

不能讓網路中的每一個 Grid 裡的所有感測器都把通訊裝置關掉，也不能只打開邊界的通訊裝置，假設移動物體的出現或事件的發生是在網路的內部，就容易發生追蹤不及的情形，若發生的事件可能造成重大災害，例如：火災，那麼將導致無法挽救的局面。

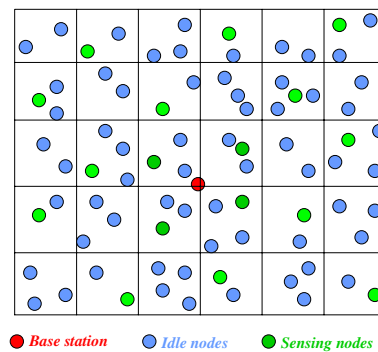
同樣地，也不能讓所有的 Grid 中都有感測器打開通訊裝置，如果一直沒有事件發生，也沒有移動物體出現，感測變成了多餘的動作，加上因為 Grid Size 和感測半徑的比值關係，網路中會有一些區域至少有兩個以上的感測器在進行感測，這些情形都勢必會造成電量的浪費，所以在下一個段落將會介紹另一個可以協助省電的方法，本文希望藉由這個方法，除了盡量維持即時傳輸訊息的功能外，也能夠延長整個網路中的感測器的生命週期。

除了前述所提到的省電方法之外，本文對於 Grid 內部和整個網路的電量控制也做了安排。

對整個網路而言，剛建立完 Grid 時，如果一直沒有事件發生，而所有 Grid 又持續地處於感測狀態下，為了盡可能地減少電量的消耗，本文不希望所有的 Grid 都進行感測的動作，在此使用一個“守望相助”的感測方法 (Help Sensing Mode) 來決定哪一個 Grid 應該進行感測。



(a) Help Sensing Mode 的九宮格範例



(b) Example of Help Sensing Mode

Figure 8 Help Sensing Mode

守望相助感測方法的原理是讓感測的 Grid 成方格棋盤狀的分佈，使正在進行感測的 Grid 協

助沒有感測動作的 Grid 感測事件，這麼做會犧牲一小部份的即時性，在一個九宮格中，如 Figure 8(a)所示，若 Grid (1) 進行感測，可以得知 3、5、7、9 四個 Grid 也同時在進行感測的動作，反之若 Grid1 為空，則可以知道 2、4、6、8 四個 Grid 正在進行感測。注意！這個模式只會關閉感測的功能，並不會關閉通訊裝置。

它的作法是讓基地台周圍的四個 Grid 持續處於感測的狀態下，利用這四個 Grid 其中之一作為守望相助的原點，架設守望相助 x 軸和 y 軸各一條，由此 Grid 向上下左右架設，因為此 Grid 鄰近於基地台，所以此 Grid 持續地進行感測，根據上段文章敘述，此 Grid 上方的 Grid 不需進行感測，在上面的那一個則必須參與感測，下左右三個方向的架設方法亦相同，如 Figure 8(b)。

在前述的文章裡提到，這個模式不會關閉通訊裝置，這是為了能盡快將感測到的事件訊息傳送出去，藉以彌補在感測時犧牲的即時性，達到 Mobicast 即時傳輸的精神，而這個“守望相助”演算法，會在等待移動物體出現之前省下大約一半的感測電量，這是本文想要達到的成果之一，但是守望相助演算法只適合使用於被偵測的實體較少時，若於受偵測的實體太多時使用守望相助演算法，會導致事件發生時網路來不及對其做出反應。

4. Simulation and Performance Analysis

因為使用 Virtual Grid 管理，搭配 Help Sensing Mode 的省電機制，可以達到節省電量 (Power-Saving)、減少傳輸的延遲 (Low Latency) 等優點，為了突顯這些優點，所以進行以下的模擬。

首先設定模擬的環境，本文希望和 Qing-Feng Huang 的 Spatiotemporal[1]做比較，所以環境設定的方面仍有少部分維持相同，本文利用 Dijkstra 演算法[9]計算網路中的最短 Hop 數，模擬 Spatiotemporal，由於本文假設均勻佈點，所以會先在每一個 Grid 裡隨機佈入兩個感測器，再把剩餘的感測器隨機佈入整個網路，換句話說，每一個 Grid 之中至少會有兩個以上的感測器，可以避免 Coverage Hole 的問題發生，而以下是其餘的相關設定：

- Area : 300*300 (m)
- Node Number : 101 (個)
- Grid Size : 50 (m)
- Velocity of the Entity : 20 (m/s)
- Delivery Zone radius : 50 (m)
- Sensing Range (Rs) : 71 (m)
- Communication Range (Rc) : 112 (m)
- Initial Energy of Node[1] to Node[100] : 100 J
- Energy Consumption as get a Message of

each Node : 0.030 J

- Energy Consumption as sent a Message of each Node : 0.081 J
- Initial Location of the Entity : (245,125)
- Moving Direction of the Entity : west

本文中主要是模擬網路中的第一象限內的情形，設立一個 Base Station，置於網路平面的原點，Base Station 被視為第 0 個點，基地台的電量是無限大，剩餘的 100 個點全為感測器。在後序的文章裡，會利用這個網路分佈的情形進行模擬，這些模擬主要是針對執行 Forward 時的網路效能，並和 Spatiotemporal 比較，Spatiotemporal 1 秒做一次 Forward 的動作，故其頻率為 1 (次/s)，而 VG-Based 的頻率則是由 Grid Size / Velocity of Entity，所以在模擬出來的折線圖中，會發現折點不在同一時間點上，這是由於兩者做法的不同造成的結果，VG-Based 是每跨越一個 Grid 做一次 Forward 的動作，如果起始時間設為 0 秒，在本文的設定中，跨越一個 Grid 需耗時 2.5 秒，但因為起點不在 Grid 的邊界，所以第一個時間點會是 2.25 秒 ((245-200)/20)，第二個時間點會是 2.25 秒加 2.5 秒，即是 4.75 秒，第三個時間點會是 7.25 秒，以此類推。

由於 VG-Based 劃分成許多 Grid，且每一個 Grid 中只有一個點進行 Forward，本模擬不考慮網路中雜訊所造成的影響，在 VG-Based Mobicast 裡，不同的 Grid 使用 Multichannel 的方式避免互相干擾，而同一個 Grid 內的通訊則由 Head 管理，不會產生碰撞的情形，所以也不考慮碰撞的發生。

一、比較 Forwarding Ratio :

令 k_1 為 Forwarding Zone 中參與 Forward 的感測器數目， k_2 為 Forwarding Zone 中的總感測器數目。本文中將 Forwarding Ratio 定義為

$$k_1 / k_2$$

Forwarding Ratio 越低，表示在 Forwarding Zone 裡參與 Forward 的點數最少，能用最少的 Forwarding Zone Node 進行 Forward 的動作，而不影響 Mobicast 的運作，就能降低 Forward 時的成本，也能有效地延長 Mobicast 的生命週期。

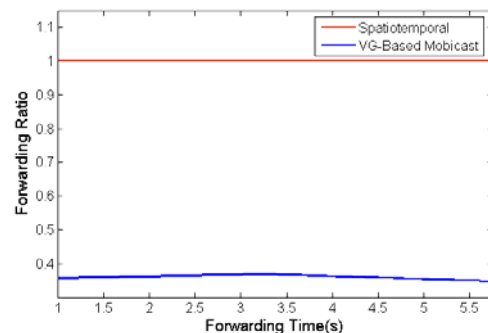


Figure 9 Forwarding Ratio V.S Forwarding Time

從 Figure 9 可以看出當 Forwarding Zone 裡的感測器越多時，VG-Based 的 Forwarding Ratio 會比 Spatiotemporal 更低。

二、比較 Forwarding Overhead :

在這次的模擬中，忽略感測器處於閒置狀態時所消耗的電量，假設每一個感測器只因為接收或傳送訊息而消耗電量，我們參考 Mote2 的電量消耗數值[10]，不考慮訊息的大小，令接收一次訊息的電量為 E_g (模擬時設為 0.030 焦耳)，傳送一次訊息的電量為 E_d (模擬時設為 0.081 焦耳)，以最靠近 Delivery Zone 的 Forwarding Zone Node 開始計算，假設 Spatiotemporal 裡有 P 個感測器參與 Forward，雖然每一個感測器只會進行一次的 Forward 動作，但是它們接收到訊息的次數卻不盡相同，所以令在 Forwarding Zone 裡的感測器接收訊息的總次數為 P_g ，本文將 Forwarding Zone 中的總耗電量 (E_f) 定義為

$$E_f = P_g * E_g + P * E_d$$

VG-Based Mobicast 因為使用了 Virtual Grid 的技術，Forwarding Zone 分佈如 Figure 10 所示。

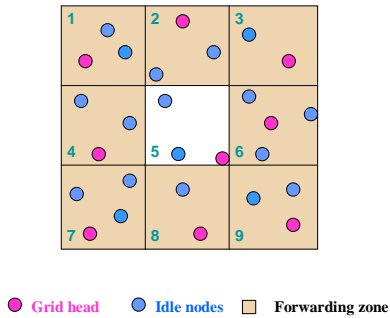


Figure 10 Virtual Grid & Forwarding Zone

因為和 Delivery Zone 相鄰，Delivery Zone 裡的感測器在發現移動物體時會立即將訊息向相鄰的 Grid 廣播出去，所以 Grid (2)、Grid (4)、Grid (6)、Grid (8) 四個 Grid 會接收到一次訊息，以 Grid (1) 來說，會收到來自 Grid (2) 和 Grid (4) 兩個方向的 Forwarding Message，所以是接收兩次，將 VG-Based 套入上述的公式裡，VG-Based 的 Forwarding Overhead 為

$$\begin{aligned} E_f &= 4 * (E_g + E_d) + 4 * (2E_g + E_d) \\ &= 12 * E_g + 8 * E_d \end{aligned}$$

由 Figure 11 可以看出，由於 Forwarding Zone 的定義方式不同，VG-Based 的 Forwarding Overhead 會比 Spatiotemporal 來得更少，Forwarding Overhead 較少，則代表耗電量較少，

這對於延長網路生命週期有明顯的幫助。

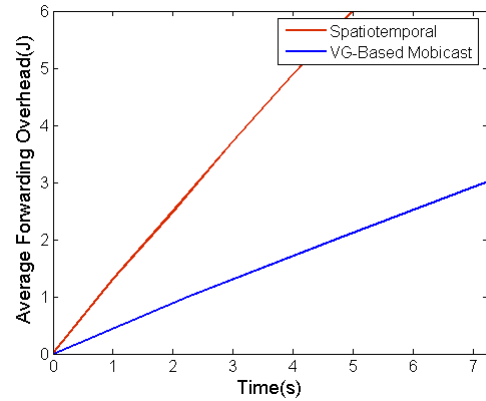


Figure 11 Average Forwarding Overhead

三、比較平均剩餘電量 :

在這個模擬裡，只考慮 Forwarding Zone 裡的電量消耗，而不考慮感測器處於閒置狀態下和 Delivery Zone 裡的電量消耗。令 ΔT 為模擬時間，本文定義平均剩餘電量 (E_l) 為剩餘的網路總電量除以總點數，計算方式如下

$$E_l = \Delta T * \text{單位時間消耗電量} / \text{網路中的感測器數}$$

由 Figure 12 可以看出，VG-Based 的平均剩餘電量較多，表示整個網路的生命週期明顯較長。

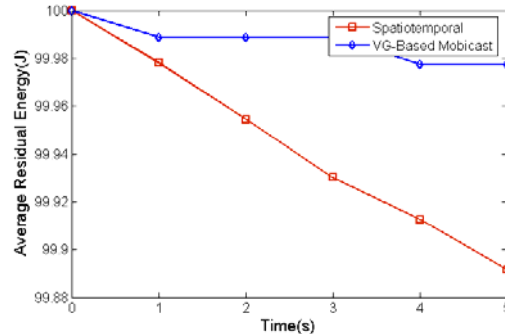


Figure 12 Average Residual Energy

四、比較 Delay Ratio :

這個模擬參考 Spatiotemporal 裡對 Delay Time 的設定，每個訊息在感測器上停留的時間為 ΔT_d ，傳送的訊息大小為 B (byte)，假設在 Forwarding Zone 裡的任兩點最短 Hop 數集合的最大值是 $Shortest_Hop$ ，Delay Ratio (R_Delay) 的值應為

$$R_Delay = (Shortest_Hop) * B * \Delta T_d / \text{模擬時間}$$

在 VG-Based 裡，因為要讓所有的 Forwarding Zone Node 都接收到 Forwarding Message 只需要 2 Hops，所以在 VG-Based 裡 Delay Ratio 的值為

$$R_Delay = 2B \Delta T_d / \text{模擬時間}$$

這是一個定值，從 Figure 13 中可以看出兩者之間的落差。

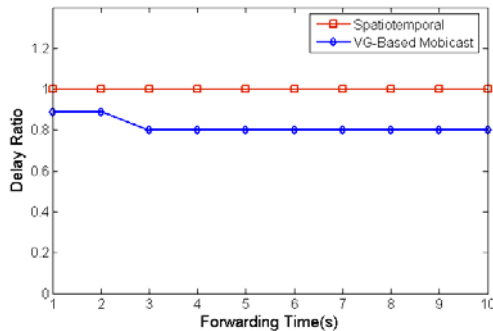


Figure 13 Delay Ratio & Forwarding Time

5. Conclusion

本篇論文提出 VG-Based Mobicast Routing Protocol，利用 Virtual Grid 搭配成員管理模式降低了管理的負擔，也藉此減少傳輸時間的延誤，更重要的是藉由 Virtual Grid 和協助感測模式的搭配，有效地延長網路的生命時間，達到了我們的目標。

本文將 Coverage Hole 視為特例而不予列入考慮，但是大多數的論文為了節省成本，幾乎都採用隨機佈點的方式在網路中佈入感測器，在隨機佈點的情形下，非常容易造成 Coverage Hole，伴隨而產生的 Sensing Hole 會使得感測工作難以執行，造就許多問題一一浮現，這是我們未來要試著解決的問題。

6. Acknowledgement

作者特別感謝國科會以計劃編號 NSC 94-2213-E-156-001- 及 NSC 94-2524-S-156-001- 提供對本論文研究部份財務上的支持。

References

- [1] Qing-Feng Huang, Chen-Yang Lu, and Gruia-Catalin Roman, "Spatiotemporal Multicast in Sensor Networks," *First ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, November 2003.
- [2] Qing-Feng Huang, Chen-Yang Lu, and Gruia-Catalin Roman, "Design and Analysis of Spatiotemporal Multicast Protocols for Wireless Sensor Networks," in *Wireless Underground Communications System (WUCS)*, USA, August 2004.
- [3] Qing-Feng Huang, Chen-Yang Lu, and Gruia-Catalin Roman, "Reliable Mobicast via Face-Aware Routing," *InfoCOM 2004*, Hong Kong, China, March 2004.
- [4] Yuh-Shyan Chen and Shin-Yi Ann, "VE-Mobicast: A Variant-Egg-Based Mobicast Routing Protocol in Wireless Sensor Networks," in

Proceedings of The 40th IEEE International Conference on Communications, (IEEE ICC 2005), Seoul, Korea, May 16-20, 2005.

[5] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[6] Dragos Niculescu, "Positioning in Ad Hoc Sensor Networks," *IEEE Network*, July/Aug. 2004.

[7] Yi Shang, Wheeler Ruml, Ying Zhang, Markus P.J. Fromherz, "Localization from Mere Connectivity," *Proc. ACM MobiHoc*, June 2003.

[8] Dragos Niculescu, and Badri Nath, "Ad Hoc Positioning System," in *GLOBECOM. IEEE*, San Antonio, November 2001.

[9] Richard E. Neapolitan, Kumarss Naimipour, "Foundations of Algorithms Using C++ Pseudocode Third Edition," page 156.

[10] Akhtar Rahman Al Eimon, and Choong Seon Hong, "DDMPC : Energy Efficient Data Dissemination Protocol in Sensor Networks," in *Proceedings of IEEE international conference, IEEE Trans. Comp.*, January 2005.