

無線感測網路中具巡邏時間一致性之掃描覆蓋機制

王偉凱

朝陽科技大學
資訊與通訊系

s9530619@cyut.edu.tw

紀孟宏

朝陽科技大學
資訊與通訊系

s9730613@cyut.edu.tw

潘彥廷

朝陽科技大學
資訊與通訊系

s9730609@cyut.edu.tw

朱鴻棋

朝陽科技大學
資訊與通訊系

hcchu@cyut.edu.tw

摘要—在無線感測網路的眾多研究主題中覆蓋問題是一個基本且重要的問題。在本論文中將針對感測節點的掃描覆蓋(Sweep coverage)問題,藉由分析各巡邏節點所需巡邏時間之差異,提出巡邏點演算法(Patrol Point Algorithm),期使各巡邏節點之巡邏時間能趨於一致。本研究針對各種不同的節點移動速度及停留時間進行實驗模擬,來驗證本論文所提方法之可行性。

關鍵詞—掃描覆蓋、無線感測網路

一、前言

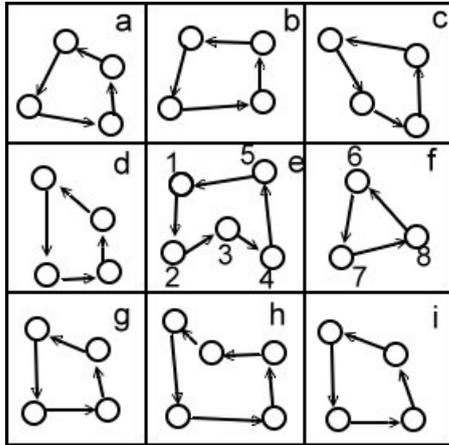
因無線感測節點的研發使得過去需要大量人力、資源才能完成的監測任務,如今只需將大量的感測節點佈署在想監測的環境中即可收集到所需的監測資訊。而感測節點所具備的無線通訊功能,讓使用者可以在安全的環境中接收感測節點所收集的資訊,而降低在監測危險環境如火山口、戰場等高危險性地區的風險。由於感測節點上述的特點,使得無線感測網路研究[6]的相關議題越來越受到重視,在眾多的重要議題中覆蓋問題是基礎且重要的議題之一。而在相關的無線感測網路覆蓋問題研究中會針對全體覆蓋(Blanket Coverage)[5, 7, 9]、柵欄覆蓋(Barrier Coverage)[2]以及掃描覆蓋(Sweep Coverage)[8]等三個方向。其中本論文著重於掃描覆蓋的研究,這是因為掃描覆蓋可應用於火場之閃燃偵測上。所謂閃燃(Flash over)[1]是於一空間內燃燒所產生之可燃性氣體蓄積於天花板附近,此可燃性氣體與空氣混合引發大範圍性火苗,造成此空間瞬間陷入火海中,因此閃燃發生後於此空間內之

人員將無法生存,故火災發生時,區域內之人員須在閃燃發生前逃出。而閃燃發生的溫度根據[3]可知為攝氏 500 度,並根據[1]可知木造建築物到達攝氏 500 度之所需時間為 330 秒,故若以掃描覆蓋應用於火災偵測上,應至少於 330 秒內於監測區域內完成數次的掃描。

而本論文將以全體覆蓋研究中[5]所提之兩階段不規則佈署演算法(Two-phase Irregular Deployment Algorithm, TIDA)為基礎,將每個子區域內所佈署之數個感測節點改由一個移動節點沿著原本子區域內感測節點所在位置依序移動之巡邏路徑,來完成子區域內之掃描覆蓋,如圖一所示。在九個子區域 a, b, \dots, i 中,以 TIDA 完成每個子區域中佈署數個感測節點之全整覆蓋,藉由子區域內佈署一個**移動節點**(mobile node),依箭頭所示之方向移動至原本欲佈署感測節點之位置(稱為巡邏點),形成**巡邏路徑**。以子區域 e 為例,利用 TIDA 需要佈署節點 1, 2, 3, 4, 5 才能完成全體覆蓋,而我們以解決旅行推銷員問題(Traveling Salesman Problem, TSP)之方法經修正後形成一個循環式巡邏路徑: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$, 因此,藉由一個移動節點沿此巡邏路徑循環式巡邏來達成掃描覆蓋。如此之作法可將原本全整覆蓋所需之 37 個固定式感測節點,改由 9 個移動節點,經過 37 個巡邏點來達成掃描覆蓋,便可有效地減少感測節點使用的數量。

然而,本論文所提出的掃描覆蓋是架構在不規則之節點佈署的環境下,各子區域內之巡邏路徑長短不一,將造成移動節點巡邏時間不一致的問題,進而影響各子區域內掃描覆蓋的品質及穩定性。因此,本論文的目標是針對此一問題,提

出巡邏點演算法(Patrol Point Algorithm, PPA)，藉由微調相鄰子區域內之巡邏路徑，期使各移動節點之巡邏時間能趨於一致且符合偵測火災閃燃發生之要求。



圖一：子區域內移動節點之巡邏路徑

二、相關工作

覆蓋問題依節點佈署的方式可分為規則佈署與不規則佈署。其中規則佈署是依特定幾何形狀來佈署感測節點，來達成覆蓋整個監測區域的目標，在[4]中探討了各種規則佈署的方法，如六角形、正方形、星狀等規則佈署方法。規則佈署的優點是能有系統的規劃節點之佈署，而缺點則因監測環境中的障礙物阻擋或節點所在地點之佈署困難，導致規則佈署可能無法涵蓋整個監測區域。另外，不規則佈署是將節點隨意放置於監測區域內，藉由移動監測環境內的節點或放置新節點等方法，進行細部調整以達成完全覆蓋。不規則佈署的優點是能根據環境變化，進而調整節點分佈，因此較能符合現實環境之需求，而缺點則是在佈署節點上需考量環境變化，造成在決定節點佈署的位置需要較複雜的計算，且所使用的節點數量亦較規則佈署為多。這是因為不規則佈署的方式容易造成部份節點的感測區域重疊，使得在相同大小之區域中，不規則佈署的方式比規

則佈署的方式需要使用較多的節點。以下將針對不規則之節點佈署在全體覆蓋(Blanket Coverage)、柵欄覆蓋(Barrier Coverage)及掃描覆蓋(Sweep Coverage)等之相關研究概述如下。

(一) 全體覆蓋(Blanket Coverage)

全體覆蓋是在監測區域內放置大量的感測節點覆蓋整個監測區域。在[7]中提出將監測區域劃分成數個網格點(Grid Point)，每一個網格點為放置節點的位置，並提出 MAX_AVG_COV 和 MAX_MIN_COV 這兩個貪婪演算法來計算節點放置於此位置能偵測到和未能偵測到目標物的機率，以此作為是否要將節點放置於該位置上之依據，而此機率值則是由節點和目標的距離來決定偵測機率。當有節點已被放置在監測區域內，先計算在監測區域內之網格點被已存在的節點偵測到的機率，再決定感測節點放置在那一個網格點上以補足未被偵測到的區域。

在[9]中則是將節點分為不具移動能力之靜態節點和具移動能力的移動節點。移動節點是額外加入的節點用於補足未覆蓋區域。此方法將監測區域劃分成數個細胞(Cell)，根據細胞數量和節點數計算節點密度平均值 N_c ，若有細胞的節點密度低於 βN_c ，則此細胞區域被認定為未覆蓋區域，其中 β 為一個系統參數且 $0 < \beta < 1$ 。被視為未覆蓋區的細胞則會呼叫附近的移動節點進入該細胞補足此未覆蓋區域。另外，感測節點會因電池耗盡而成造未覆蓋區域時，則細胞內的節點會發出 Help Message 給移動節點來請求協助，藉以達成整個監測區域之覆蓋。

而在[5]中則提出兩階段不規則佈署演算法，第一階段負責佈署隨機感測節點，第二階段則包含三個子工作，分別為(1)初步節點佈署：分割子區域並在覆蓋率最低之子區域中心點放置控制節點；(2)補足未覆蓋區域：移動完全覆蓋節點及計算每個完全覆蓋節點之移動效益，移動效益最大之完全覆蓋節將優先移動，若已完成全體覆但仍有完全覆蓋節點，則將其關閉以減少節點使用量；(3)提升整體覆蓋率：負責有限制移動節點來減少節點覆蓋區域之重疊，提升整體之覆蓋率及減少節點使用數量。

(二) 柵欄覆蓋(Barrier Coverage)

柵欄覆蓋是以大量感測節點將一特定區域或目標圍住，形成一個圍繞著特定區域或目標物的柵欄。當有任何物體通過由感測節點所構成的柵欄，則會被感測節點偵測到，並通知監測方。在[2]中提出以相吸力和互斥力讓原本散佈於各地的節點，自行排列成一直線的柵欄。相斥力應用在 X 軸，隨著節點距離拉長而跟著減弱。當節點間之距離等於感測半徑時，則相斥力為零，目的是希望節點距離能適度拉開，而不致因節點間距過於近而需使用較多的節點來形成柵欄。相吸力則應用在 Y 軸，隨著節點間之距離拉近而減弱，當節點在 Y 軸的間距為零，相吸力方為零，否則相吸力會持續作用到兩節點在 Y 軸上呈現近似水平狀態為止，其目的在於希望各節點能排列成近似水平之直線以形成柵欄。

(三) 掃描覆蓋(Sweep Coverage)

掃描覆蓋則是由一個或數個具有移動能力的感測節點組成群組，該群組會在監測區域內來回移動，並在移動的過程中不斷偵測監測區域內是否有目標物。在[8]的論文中提出 POI (Points of Interest) 的概念，每一個 POI 是移動節點必須經過的點，且不同的 POI 因重要性的不同可容許較長的等待時間，以等待移動節點的到訪。藉由解決旅行推銷員問題(Traveling Salesman Problem) 找出最短路徑，得出每一個 POI 最少的循環時間，以此發展 CSWEEP (Centralized Sweep Algorithm) 演算法。該演算法先找出移動至全部 POI 所需之循環路徑，將該循環路徑分割成數個區段，每一個區段皆由一個移動節點負責巡邏，由該移動節點在負責區域中找出符合該區域之各 POI 所需巡邏間隔時間。由於 CSWEEP 是集中式演算法需知道所有節點位置才能決定節點移動的起點和終點位置。因此，提出 DSWEEP (Distributed Sweep Algorithm) 演算法，讓每個移動中的感測節點互相交換位置資訊和來回時間，並由節點決定是否改變現有移動路徑。

綜合上述之相關研究，可知在不同的覆蓋問題上有各種不同的解決方式及其優缺點。而本篇論文將針對掃描覆蓋的問題，提出一個巡邏點演算法(Patrol Point Algorithm, PPA)，藉由微調相鄰

子區域內之巡邏路徑，期使各移動節點之巡邏時間能趨於一致以符合偵測火災閃燃發生之要求。

三、巡邏點演算法

掃描覆蓋概念下之不規則的狀況有兩個問題需要解決：(1)各巡邏點是不規則分佈在監測區域內，將面對移動節點走訪那些巡邏點及其到訪順序的問題；(2)移動節點所須經過的巡邏點數量和巡邏點之間的距離不相等，導致各移動節點在巡邏時間上的差異而造成覆蓋品質不一的狀況。其中第一個問題可由已廣泛被討論的旅行推銷員問題(Traveling Salesman Problem, TSP)的解法經修改後可解決此一問題。但第二個問題則要縮短各移動節點巡邏時間之差異使其能趨於一致。因此，在本論文中我們提出巡邏點演算法(Patrol Point Algorithm)來解決此問題。以下先針對整個問題及適用環境做以下幾項假設：

- (1) 整個監測區域為二維的平面。
- (2) 移動節點的感測範圍為理想狀態，即其感測範圍為圓形。
- (3) 移動節點不受移動及訊號干擾、衰減、漫射及散射等因素影響。
- (4) 移動節點不因長期的移動而造成能源耗盡而失效，即本研究不考慮各種能源消耗問題。
- (5) 移動節點可依需求移動至監測區域內之任一位置。

(一) 巡邏時間分析

在不規則節點佈署(TIDA)的環境下因每一個節點之間的距離不相同且每個子區域內所需經過之巡邏點個數不同。因此，移動節點的巡邏時間歸納如下：

$$T = \sum_{i=1}^k \frac{d_i}{v} + kw \quad (1)$$

其中 k 為巡邏區內之所有巡邏點之個數； d_i 為第 $i-1$ 個巡邏點和第 i 個巡邏點之間的距離； v 是感測節點移動速度； w 是每到一個巡邏點所需停留的時間。

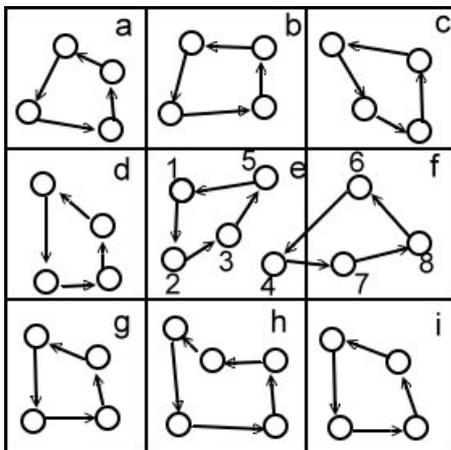
在不規則節點佈署環境下，若採用相同移動

速度的移動節點來完成掃描覆蓋，則由式(1)可看出各移動節點之巡邏時間將不一致。

(二) 巡邏時間平衡

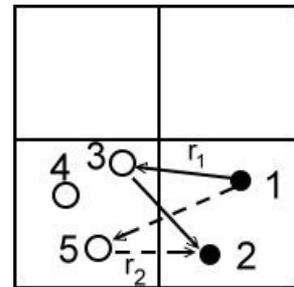
在沿用 TIDA 演算法中的子區域概念做為選擇巡邏點的依據，移動節點需走訪所屬子區域內的所有巡邏點，但因 TIDA 可能會發生單一子區域內只有一個巡邏點的情形，導致移動節點只能固守在單一巡邏點上，為避免此問題且不影響覆蓋品質下能有效減少巡邏節點使用數量，我們將四個子區域合併成一個群組的方式，並讓移動節點走訪該群組內所有巡邏點，以此架構來調整移動節點之巡邏節點來達成巡邏時間平衡之目標。

在縮短各移動節點巡邏時間差異的問題上，因各個子區域內的節點數和節點間距的不同，造成各子區域內的移動節點在走訪巡邏點的時間有所差異。因此在巡邏點承接上將考量各移動節點的巡邏時間和平均巡邏時間，若移動節點的巡邏時間大於平均巡邏時間則釋放一個巡邏點由鄰近子區域的移動節點進行巡邏點承接。在圖一中，子區域 *e* 之巡邏點路徑為 1→2→3→4→5→1，因其巡邏時間超過平均巡邏時間。因此，將原本所需經過之巡邏點 4 改交由相鄰子區域 *f* 的移動節點負責，其巡邏路徑由巡邏點 6→7→8→6 改為巡邏點 6→4→7→8→6，如圖二所示。



圖二：相鄰的子區域之巡邏點承接

在移動節點所欲承接之巡邏點是由承接方所決定，其依據往返時間 *m* 來決定。所謂“往返時間”為移動節點承接其他子區域之巡邏點後所增加之間時。以圖三為例，移動節點 A 負責之巡邏路徑為巡邏點 1→2→1；移動節點 B 負責之巡邏路徑為巡邏點 3→4→5→3。但因移動節點 B 之巡邏時間超過平均巡邏時間，故將原本所經過之巡邏點擇一交由移動節點 A 承接。因此，可能之巡邏路徑為 1→3→2→1 或 1→5→2→1，其中實線路徑 *r1* (1→3→2) 和虛線路徑 *r2* (1→5→2) 則為往返時間。若實線路徑 *r1* 之往返時間小於虛線路徑 *r2*，則移動節點 A 將承接巡邏點 3。



圖三：移動節點承接巡邏點為最短來回時間

為避免兩相鄰子區域之移動節點因巡邏點的承接與釋放，導致巡邏點承接與釋放之動作頻繁發生而造成乒乓效應。因此，我們以門檻值 *L* 作為相鄰移動節點是否有資格承接巡邏點之依據。即當移動節點之巡邏時間小於 *L* 才允許承接巡邏點。而 *f* 則是相鄰移動節點預測承接巡邏點後之巡邏總時間，用來判斷相鄰移動節點是否有能力承接新的巡邏點。即當 *f* 大於門檻值 *L*，則不承接新巡邏點。*L* 和 *f* 的關係式分別如下所示：

$$L = t_{avg} + w \tag{2}$$

$$f = t_i + w + m_i \tag{3}$$

在式(2)中 t_{avg} 為平均巡邏時間；*w* 為移動節點每到一巡邏點必須停留的時間。若相鄰的移動節點之巡邏時間低於 *L* 則表示該移動節點在不

計“往返時間”的前提下，仍有能力可應付新增的巡邏點。在符合門檻值 L 條件下，將由巡邏時間最短之移動節點優先承接巡邏點。另外，在式 (3) 中， t_i 為第 i 個移動節點的巡邏時間； w 為移動節點每到一巡邏點必須停留的時間； m_i 為移動節點 i 從原先所經過之任一巡邏點，移動到欲承接新巡邏點的最小“往返時間”。當 f 大於門檻值 L ，表示移動節點若承接此巡邏點則巡邏時間將超過平均巡邏時間，則該移動節點將不承接此巡邏點。若 f 小於門檻值，則預測時間 f 最短者將優先承接巡邏點。以下為本論文所提之巡邏點演算法(Patrol Point Algorithm, PPA)的虛擬碼：

PPA(P, w)

```

1: //巡邏節點集合  $A=\{a_1, a_2, \dots, a_j\}$ ;
2: //巡邏點集合  $P=\{p_1, p_2, \dots, p_j\}$ ;
3: //每個巡邏節點之巡邏總時間  $T=\{t_1, t_2, \dots, t_j\}$ ;
4: //要承接巡邏點之巡邏節點  $S=\{s_1, s_2, \dots, s_k\}$ ;
5: //預估承接巡邏節點後之巡邏時間集合
    $F=\{f_1, f_2, \dots, f_k\}$ ;
6: do
7:   計算巡邏點  $a_i$  之巡邏時間,  $i=1..j \forall a_i \in A$ 
8:    $t_{avg} = \frac{1}{j} \sum_{x=1}^j t_x$ ;
9:    $L=t_{avg}+w$ ; //  $w$  為巡邏點之停留時間
10:  for ( $x=1; x \leq j; x++$ )
11:    if ( $t_x > t_{avg}$ )
12:      找出巡邏節點  $s_i$ , 其為巡邏節點  $a_x$ 
        的鄰近節點且  $t_i = \min\{T\}$ ;
13:      if ( $t_i < L$ )
14:        //  $m_i$  為由  $s_i$  至  $p_i$  最短巡邏時間;
15:         $f_i = t_{avg} + w + m_i$ ;
16:        if ( $f_i < L$ )
17:           $S = S \cup \{s_i\}$ ;
18:           $F = F \cup \{f_i\}$ ;
19:        end if
20:      end if

```

```

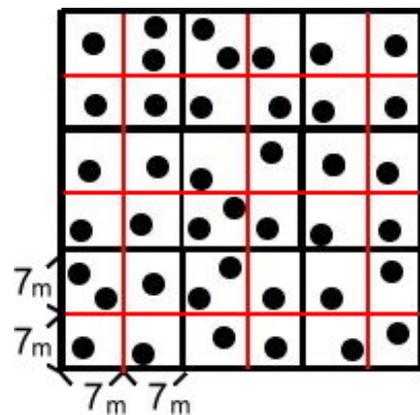
21:      end if
22:    end for
23:  if ( $S \neq \emptyset$ )
24:    找出巡邏節點  $s_i$ , 其  $f_i = \min\{F\}$  且  $f_i \in F$ 
25:     $S = S - \{s_i\}$ 
26:  end if
27: while ( $S \neq \emptyset$ )

```

在上述之 PPA 演算法中，Step 7 到 Step 9 為計算各子區域的巡邏時間、平均巡邏時間和門檻值 L ；Step 10 到 Step 22 為檢驗超過平均巡邏時間之移動節點，其相鄰巡邏點是否可承接巡邏點；Step 23 到 Step 26 則是找出所有欲承接巡邏節點之移動節點且預測巡邏時間最短者將優先承接巡邏點。

四、模擬結果

在本節中我們使用 MATLAB 軟體進行實驗模擬，監測區域之大小為 42 公尺×42 公尺，如圖四所示。將四個由 TIDA 所產之 36 個 7 公尺×7 公尺的子區域合併成 9 個群組（即 14 公尺×14 公尺）且每個群組內放置一個移動節點，共計放入 9 個移動節點。在此模擬環境下，我們將針對移動節點在巡邏點的停留時間分別為 0, 1 及 2 秒且節點移動速度分別為時速 10、20、30、40 及 50 公里，進行十次模擬統計移動節點的平均巡邏時間之變化。

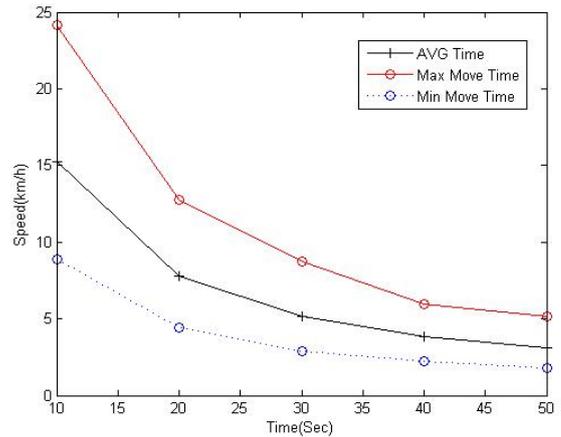


圖四：實驗模擬環境

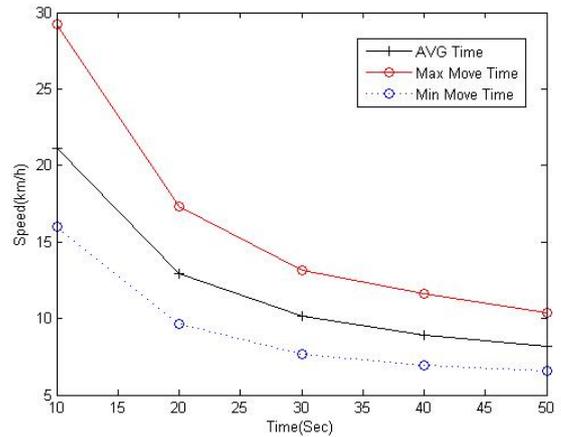
在圖五、圖六及圖七中橫軸為移動節點的移動速度，縱軸為巡邏時間，分別表示在巡邏點停留時間為0秒、1秒及2秒。在圖中可明顯看出最大巡邏時間和最小巡邏時間有相當大的差距，這是因在不規則分佈環境下每個子區域內之巡邏點個數不相同，造成各子區域之移動週期差異。隨著節點移動速度越快，最大和最小巡邏時間差距也隨之縮小，這是因節點移動速度的加快，使得移動節點在各巡邏點之間移動所花費之時間減少。而從平均巡邏時間來比較，可知隨著在巡邏點上停留時間的減少，移動節點在巡邏時間上花費也隨之減少。顯示在同樣速度下縮短巡邏點停留時間可有效地減少移動節點巡邏時間，但相較於將速度增加來減少移動節點的巡邏時間，採用縮短巡邏點停留時間之效果更為顯著。因此可瞭解移動節點隨著節點移動速度加快，其停留時間成為巡邏時間上的瓶頸。

另外，移動節點在巡邏點停留時間為0秒且節點移動速度為時速10、20、30、40和50公里時(如圖五所示)，其平均巡邏時間分別為16.1、8.1、5.3、3.9和3.1秒，其巡邏時間標準差分別為3.7、1.6、1.2、0.9和0.7秒(如表一所示)。隨著速度提升(時速大於10公里)其標準差更低於2秒。且從上述結果可看出巡邏時間之標準差僅佔平均巡邏時間約22%，且此數值不受節點速度變化的影響，因此顯示各移動節點之巡邏時間能趨於一致。移動節點在不同時速及不同巡邏點停留時間下平均巡邏時間之標準差，如表一所示。顯示所提出的巡邏點演算法在高速的移動節點及低巡邏點停留時間下，能有效地減少移動節點巡邏時間之差異。

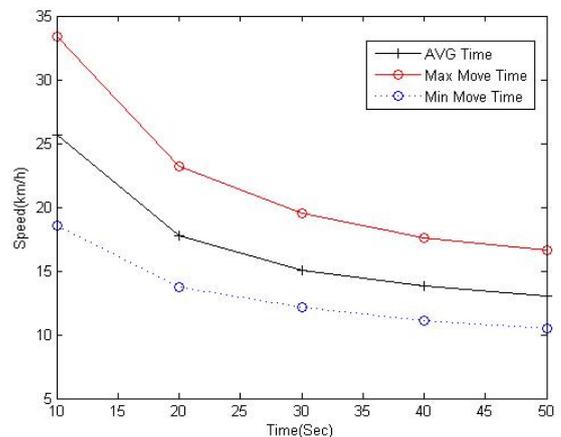
綜合上述之模擬結果，我們所提之PPA演算法在節點停留時間為2秒之模擬，其最大巡邏總時間為34秒，移動節點在火場內巡邏至少9次，符合偵測火災之閃燃發生之要求(330秒)之時間。因此，感測節點能在溫度到達閃燃之臨界點前發出警報，使消防人員或受困人員可在所處空間發生閃燃前提早逃出該區域。



圖五：在巡邏點停留時間為0秒時，移動節點的最大、最小及平均巡邏時間



圖六：在巡邏點停留時間為1秒時，移動節點的最大、最小及平均巡邏時間



圖七：在巡邏點停留時間為2秒時，移動節點的最大、最小及平均巡邏時間

表一：移動節點在不同時速及不同巡邏點停留時間下平均巡邏時間之標準差

停留時間 時速	0 sec	1 sec	2 sec
10 km	3.79 sec	4.17 sec	4.57 sec
20 km	1.65 sec	2.43 sec	2.98 sec
30 km	1.29 sec	1.75 sec	2.41 sec
40 km	0.91 sec	1.44 sec	2.01 sec
50 km	0.72 sec	1.21 sec	1.96 sec

五、結論

本論文所提之巡邏點演算法藉由模擬結果顯示在監測區域在 42 公尺x42 公尺的環境下，各移動節點之巡邏時間趨於一致且符合偵測火災閃燃發生的要求。而在掃描覆蓋的研究議題中，縮短巡邏點停留時間比增加移動速速能更有效地減少巡邏時間。因此，若以掃描覆蓋之架構下且需搭配巡邏點停留等應用上，停留時間成為巡邏時間上的瓶頸。

雖然本論文之研究是基於 TIDA 的方法，利用其全體覆蓋及畫分子區域的方式來達成具巡邏時間一致性之掃描覆蓋機制，但本研究可有彈性地更換此方法，而改採其他全體覆蓋機制後，即可應用所提之巡邏點演算法來達成掃描覆蓋。因此，在未來工作方面，可研究與設計較佳之全體覆蓋機制來取代 TIDA，以提升巡邏時間之一致性。

誌謝

本研究要感謝行政院國家科學委員會 NSC 98-2221-E-324-021 研究經費的支持。

參考文獻

[1] 陳弘毅, “火災學,” 鼎茂圖書出版有限公

司, 第七版, 中華民國九十七年十二月。

- [2] C. Shen, W. Cheng, X. Liao and S. Peng, “Barrier Coverage with mobile Sensor”, International Symposium on Parallel Architectures, Algorithms and Networks, pp. 99-104, May 2008.
- [3] Cote, “Fire Protection Handbook,” National Fire Protection Association, Vol. 1, Edition 19, 2003.
- [4] E. S. Biagioni and G. Sasaki, “Wireless Sensor Placement for Reliable and Efficient Data Collection”, IEEE International Conference on System Sciences, Jan. 2003.
- [5] H.-C. Chu, W.-K. Wang, L.-H. Chang and C.-J. Li, “The Study of Coverage Problem in Wireless Sensor Network”, The 4th Workshop on Wireless Ad Hoc and Sensor Networks (WASN 2008), Sep. 4-5, Tainan, 2008.
- [6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A Survey on Sensor Network”, IEEE Communications Magazine, vol. 40, pp. 102-114, Aug. 2002.
- [7] S. S. Dhillon and K. Chakrabarty, “Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Network”, IEEE Conference on Wireless Communications and Networking, pp. 1609-1614, Mar. 2003.
- [8] W. Cheng, M. Li, K. Liu, Y. Liu, X. Y. Li, and X. Liao, “Sweep Coverage with Mobile Sensors”, IEEE International Symposium on Parallel and Distributed Processing, pp. 1-9, Apr. 2008.
- [9] X. Du and F. Lin, “Improving Sensor Network Performance by Deploying Mobile Sensor”, IEEE International Conference on Performance, Computing, and Communications, pp. 67-71, 2005.