

The Online Tree Adaption Policy for Object Tracking Tree in Wireless Sensor Network

胡哲禎

中華大學資訊工程學系
m09602027@chu.edu.tw

陳旻秀

國立東華大學資訊工程學
系
mxchen@mail.ndhu.edu.tw

翁文彥

中華大學資訊工程學系
wenyen.weng@chu.edu.tw

摘要

目前在無線感測網路下既有的物件追蹤樹的建立策略都必須使用一個假設的物件移動模型。物件追蹤樹建立策略在這個模型與假設的物件移動頻率下，建立符合要求的物件追蹤樹。然而，倘若真實的物件移動環境與假設的模型有所差異時，則建立的物件追蹤樹其效能不會有預期的效果。本篇論文希望提出一個調整策略能夠修改既有物件追蹤樹架構以提升物件追蹤樹的效能。

關鍵詞：無線感測網路、物件追蹤、hierarchy tree

Abstract

The current object tracking trees establish strategy that must suppose object movement model by making assumptive object movement frequency in advance. However, if actual object movement behaviors are different from assumptive movement profile, the performance of the object tracking trees establish may be reduced. In this paper, we propose a dynamically tree adaptive strategy to modify object tracking tree architecture in order to improve performance efficiency. The simulation results also show that

it is evident that the dynamically tree adaptive strategy is sufficient to achieve a good performance.

Keywords : wireless sensor network、object tracking、hierarchy tree

1. 序論

物件追蹤是感測網路的一種應用，主要可以利用在監控動物棲息地或是軍事上的監控應用等等，我們將感測點佈署到我們想要追蹤的物件區域，利用感測點來偵測物件的位置，藉此達到物件追蹤的目的。而車輛追蹤正是其中最重要的一項應用。

假設在城市重要交通樞紐上或是危險道路上設置感測器，每個感測器都會有感測物件的能力，而且每個感測器的通訊範圍足以跟鄰近感測器做通訊。在這些感測器中，有一個特定的感測器當作行控中心負責管理整個網路，稱為資訊收集點 sink。當有一個車輛在感測器的感測範圍被偵測到時，感測器就會傳送偵測的訊息回報給 sink，sink 就能夠知道車輛資訊來做到物件追蹤。然而如果每個感測器都像資訊收集點回報感測資訊，將產生大量的回報訊息。因此，提出一個有效的資訊收集架構成為一個重要研究課題。

在本篇論文，我們使用 hierarchy tree 架構來將感測訊息有效率的傳到資訊收集點，且提出動態調整 tree 結構，讓資訊收集效率保持一定水準，並且更能夠符合真實環境需求。

本論文的第二章節會提到目前相關研究物件追蹤所提出的不同方法或架構，第三章節主要是探討問題存在和定義問題，並且提出動態調整策略，在第四章節說明模擬環境和實驗數據，最後則是結論。

2. 相關研究

我們先探討在感測網路下，使用者如何查詢物件出現的區域?最簡單方式就是直接對每一個感測點發出詢問訊息，詢問是否有感測到物件出現，但是這種方式會當感測點的數量越多時，發送的訊息量隨之增加，導致所要花費成本就越大，使得網路負載增加。

後來在[2]提出新的查詢方式，提出資料收集點(sink)概念。此方法先找取一個點當作資料收集點，資料收集點主要工作就是收集所有感測點的物件出現的資料，使用者只需要對資料收集點提出查詢，就能知道目前物件出現的區域。雖然這個方法能夠有效降低查詢成本，但是卻引發另外一個問題，就是感測元件如何傳送感測資料給資料收集點?而最簡單方式就是感測點使用泛濫方式傳送資料給資料收集點，但是這個方法會造成廣播風暴，進而增加網路負擔。後來，又陸續提出了 cluster tree 和 hierarchy tree 兩種架構來做資料收集。

有關使用 cluster tree 來解決資料傳遞問題的相關研究有[5]。而使用 hierarchy tree 來解決的相關研究則有 DAB[2]DAT[3]MT-HW[4]等。本論文將針對 hierarchy tree 的相關研究進行探討。

Wen-Chih Peng, 等人在 [5] 提出 HTM (heterogeneous tracking model) 針對於物件追

蹤所提出不同的模型，此模型是一種基於預測物件移動和訓練學習的方式。利用 cluster tree 的架構做資料收集，所有感測點做分群，然後在每一個群中選出一個分群點(cluster head)來負責收集同一群內的物件資料，最後再由分群點將資料傳送給資料收集點。然後透過 Variable memory Markov (VMM)預測出物件移動的樣本，並且訓練這些樣本使得準確度提高，同時使用 Emission tree 來做維持預測架構，當預測失敗時，會進入重新尋找物件的程序。缺點就是架構複雜而且計算也很費功夫，當預測失敗時，所要付出的成本會更大，只能用於小型感測網路。

H. T. Kung 等人在 [2] 提出了 DAB (Drain-and-Balance) 利用 hierarchy tree 架構來解決感測資料收集的問題。由資料收集點當作樹根，建立樹狀結構，其他的感測點只要沿著邊線往樹根方向傳送資料即可，並且在中間的感測點會負責保留物件資訊，避免重複收到相同訊息。DAB 是根據物件出現的頻率定義不同的門檻值並且分不同的層次，同一個層次的感測點相連合併成平衡樹。DAB 改善了原本以氾濫的方式作傳送資料，而且能用在大型感測網路上。但是 DAB 的缺點在於二元樹，如果感測點很多時，樹愈高所花費的成本就會增加，而且需要額外多配置感測點當做中間點幫忙轉送。

為了改善DAB缺點，Chih-Yu Lin等人在[3]提出以多元樹架構來為基礎的 DAT(Deviation-Avoidance Tree)改善。DAT利用存在於感測網路的感測點當做中間點，不需要額外配置，並且提出新的建構階層樹的方式。然而DAT的複雜度高，當感測點太多，建立樹所花費之計算成本太高。為了加快計算速度，提出Z-DAT的概念將感測網路分割後分別建立子樹，然後再合併。此外，尚提出了QCR來平衡query cost與資訊傳遞維護成本的花費。缺點就是由於是多元樹，傳送資料所花費成本會增

加，而且所建立出來的樹不會是最佳解，而會因為資料收集點的位置所影響。

改善DAT缺點，由於單一sink作為資料收集，會因為sink所在位置而建立出效能不同的資料收集樹。因此Chih-Yu Lin等人在[4]提出MT-HW(multi-tree construction with the high-weight-first property)的概念，能夠加快查詢回答時間，同時也減輕感測網路傳遞訊息的負擔，並且提出新的更新和查詢機制，建立multi-sink tree結構，已達到有效訊息傳輸，降低更新和查詢成本。但是sink點愈多，代表建立hierarchy tree也愈多，要維持hierarchy tree所付出的更新成本就會增加，造成網路負擔。

我們在[1]提出TAP (tree adaptation procedure)，針對已經建立好的邏輯樹來做調整，從最底層的感測點開始，假設和鄰近的感測點作連結並計算其update cost，挑選update cost 最小值來確定最後要連結的感測點，最後以bottom up方式往上層的感測點作相同計算一直到sink結束，表示調整完整個hierarchy tree。此調整方式能通用任何以建立好的邏輯樹，能改善update cost值。但是只做一次TAP調整的邏輯樹，並不能得最小update cost值，需要做三到四次的TAP才能得到最小update cost值。

3. 問題討論

3.1 問題定義

除了HTM是另外提出預測模型，其他不論是DAB、DAT或是MT-HW都需要利用物件移動的頻率來建立hierarchy tree，而在模擬環境時，都必須使用假設的物件移動頻率的模型，來產生資料收集樹。假設的物件移動模型與真實常有所差距。因此，使用假設物件移動模型所建構的資料收集樹，應用在真實環境時

其效能將會大打折扣。因此，為了提昇資料收集樹的效能，一套能動態調整樹狀結構的機制顯得相當重要。而據我們所收集資料可知，目前並無相關研究根據此一問題提出解決方案，所以我們將在本論文提出一套解決方案。

3.2 調整策略

為了能夠動態調整樹狀結構，每個感測器要知道與鄰居之間的傳送訊息頻率(event rate)，藉此來判斷是否要觸發調整策略。當hierarchy tree在收集物件移動更新資訊時，傳送訊息的內容會紀錄從自己本身感測點到下一個感測點的物件資訊。根據此資訊我們可以在感測器上多建立一個table負責紀錄鄰近感測點與自身的傳送訊息頻率。

圖1表示在車輛移動情境下，已經建立好資料收集樹做物件追蹤和訊息更新。當B偵測到有Car A此車輛出現在感測範圍內，就會記錄此車輛資訊，並且回報更新訊息給上層的感測器，上層的感測器會查看自己是否有此車輛的資訊，如果有就不再回報，如果沒有此資訊，就會繼續傳送訊息給更上層的感測器，重複步驟直到sink表示車輛目前位置訊息正確更新。當車輛穿越感測區域時，像是Car B從E移動到D，此時E就要傳送更新訊息給上層的感測器，通知此車輛要離開E的測範圍到D的感測範圍，並且將此訊息次數做紀錄。D跟開始偵測到車輛的運作一樣做回報更新。假如sink要做車輛追蹤時，只要跟下層的感測器詢問是否有此查詢車輛的訊息，一直追蹤到最底層的感測器就能得到目前車輛的位置。

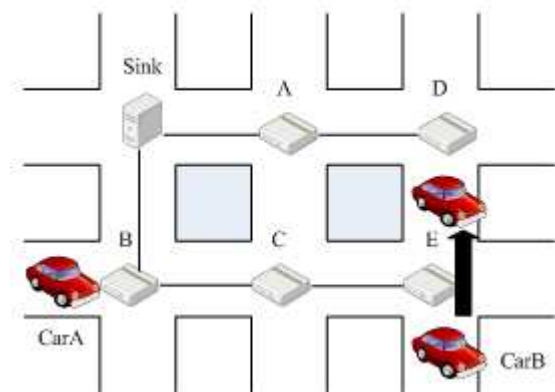


圖 1 物件追蹤和訊息更新

經由上述資料更新方式，每個感測器都會知道與鄰居的傳送訊息頻率。然後，感測器可根據目前的傳送訊息頻率與之前紀錄傳送訊息頻率計算變動率，並比較變動率是否超越門檻值 θ 。如果沒有，則表示目前此感測網路以下的子樹上不需要做調整。若真實比例值超過 θ ，則必須進行結構調整。變動率的公式定義如下。

ratio of change =

$$\frac{\text{new event rate} - \text{original event rate}}{\text{original event rate}}$$

接下來介紹調整策略。調整策略主要概念是由 sink 來做管理和做調整決策。如圖 2 的 step 1 所示，每個感測點都會定期比較其傳送訊息頻率，並且計算其變動率。假如變動率沒有超過門檻值就不會回報給 sink，如圖 2 的 step 2 所示。若有感測點發現其變動率超越門檻值則會將這訊息通知給 sink，如圖 2 的 step 3 所示。經過上述過程我們會得到感測器因為變動率超越門檻值，而發送的需要調整訊息通知給 sink，但是因為傳送到 sink 會花費一些時間，而且如果門檻值 θ 太小，會有很多感測器發送調整訊息，此時如果每接收到一個調整訊息 sink 就做調整，很容易造成網路負擔，所以我們假設在一段時間內收到的調整訊息才會做處理。

sink 收到感測器傳送需要做調整訊息

後，接著 sink 再跟 branch node 要求傳送相關訊息，這裡相關訊息包括了車輛資訊和記錄傳送訊息頻率，以方便資料收集點當做調整樹的依據，如圖 2 的 step 4 所示。由於感測器自己本身會暫存車輛資訊，這裡的車輛資訊包含了最底層的車輛資訊，也就是說蒐集每個感測器資訊不必要求到最底層的感測器，只要到每個 branch node 要求資訊即可。當感測器收到來自 sink 傳送過來的要求訊息時，則回報給 sink。

sink 收到相關訊息作為調整 tree 的依據後，最後再由 sink 透過調整演算法計算出需要調整的感測點，如圖 2 的 step 5。最後，sink 送出調整樹的訊息給需要做調整的感測器，感測器收到訊息後，作邏輯樹的調整，如圖 2 的 step 6。經過此方法後，就能夠動態調整資料收集樹，並且也能保持做車輛追蹤。

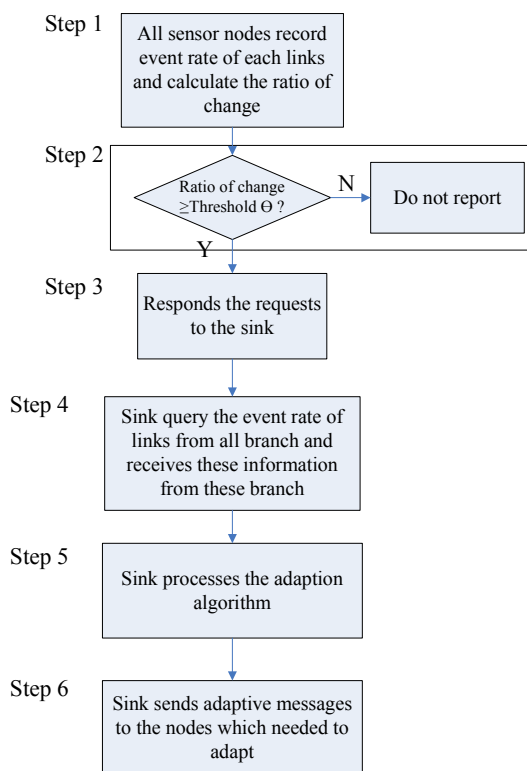


圖 2 調整策略流程圖

圖 3 是說明整個調整的過程。圖 3(a)表示，黑色點代表感測點發現其變動率超越門檻

值，發送需要調整訊息通知 sink，也就是圖 2step 1 到 step 3。圖 3(b)表示，sink 收到這些通知需要調整訊息後，向 branch node 要求傳送相關訊息，來做調整樹的依據。圖 3(c)則表示，branch node 收到從 sink 傳來的要求訊息，並回報相關訊息給 sink，圖 3(b) (c)也就是圖 2 的 step 4。圖 3(d)表示，當 sink 收到 branch node 所提供相關訊息後，透過調整演算法，計算出需要調整的感測器，並且傳送調整訊息給需要調整的感測器，也就是圖 2 的 step 5 和 step 6。圖 3(e)表示，最後根據調整訊息做調整後的結果。

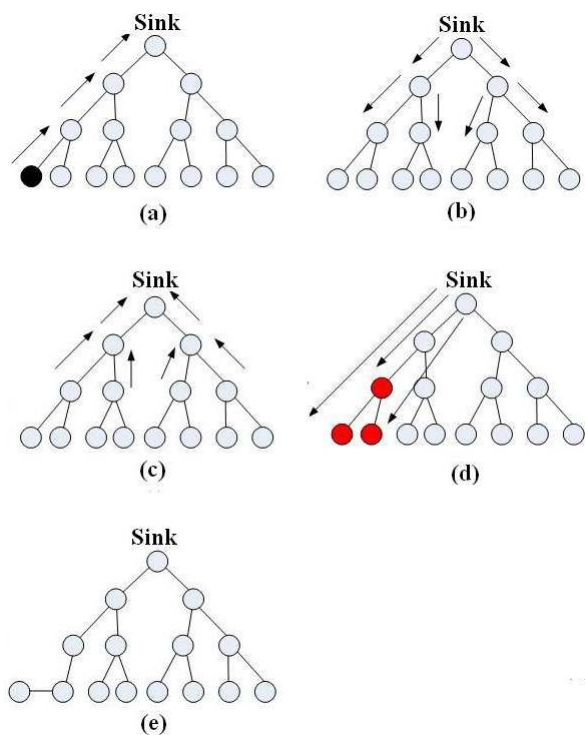


圖 3 調整策略過程

器所負責感測範圍。當有一個車輛在此感測範圍被偵測到時，負責此區域的感測器會傳送偵測的訊息回報給 sink。

當車輛跨越感測範圍移動到其他感測器的感測範圍時，舊的感測器會回報一個更新訊息通知 sink 車輛離開，並紀錄發生頻率。如果兩個感測器共用一個感測範圍邊界，則稱為鄰居。我們假設每個感測器的通訊範圍都能夠跟其鄰居彼此溝通和交換訊息。因此，我們可以將此感測器網路定義成圖 4 物件移動模型 $G(V_G, E_G)$ ， V_G 表示感測器分佈(圖 4 中圓圈)，而圓圈內的字表示感測器的 ID， E_G 表示每個感測器的鄰居之間相連的連線(圖 4 中相連的虛線)，每個感測器的連線上會有 event rate 值， $event\ rate(A,B) \in E_G$ ，表示車輛跨越感測範圍的發生頻率總和，因為不論車輛從哪個方向跨越感測範圍，新的和舊的感測器都會傳送訊息回報給 sink。

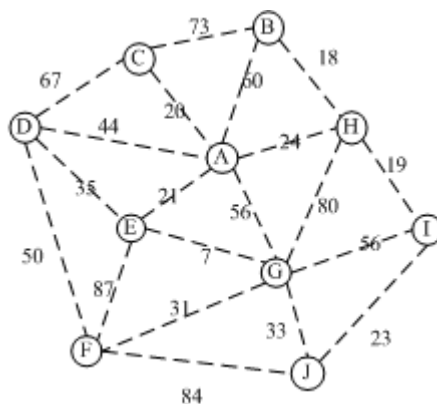


圖 4 物件移動頻率模型

4. 實驗結果

4.1 網路模型

根據我們在序論所假設的物件追蹤應用情境，對於此感測器網路所形成的感測範圍，我們使用一個 Voronoi graph 來表示每個感測

4.2 模擬環境

在模擬環境中，我們設定分佈範圍為 256x256 的單位平方區域，在此封閉的空間內以隨機的方式散佈 100 到 1000 個不會移動的感測器。sink 位置固定在角落，門檻值 θ 從 5% 到 95%，event rate 則是利用物件移動頻率模型隨機產生一個值來模擬。在模擬環境中，我們

重視在環境變化後對於資料收集樹的效能影響，因此物件的移動模型採用隨機方式產生。

在模擬測試時所比較的效能有兩個，其一為 update cost，定義在[3]。另外一個則為調整樹狀結構時，所需傳送訊息的花費。update cost 主要意義為計算感測器傳送更新訊息時所花費的成本，update cost 值愈小表示 sink 收集資料所花費成本小，執行效率高，update cost 值愈大時則反之。

而在計算結構更新的訊息花費，可用公式表示，其中感測網路上的感測點 $\in i$ ，發現有問題的點通知 sink 花費的 hop count $\in problem_node$ ，如圖 2 的 step 3。sink 要求感測點回傳訊息所花費的 hop count $\in sink_gather$ ，如圖 2 的 step 4。sink 通知需要調整的感測點所花費的 hop count $\in adapt$ ，如圖 2 的 step 6。公式：

$$overhead = \sum problem_node_i + \sum sink_gather + \sum adapt_j$$

在以下各圖的圖示中，original_DAT 則表示在移動模型環境改變之後，原本建立的 DAT 所產生的效能。updated_DAT 代表是在移動模型環境改變之後，重新建立 DAT。

圖 5 圖 6 所要探討的是在不同的感測器數量下，感測器位置不變，門檻值 θ 設為 10%，當模擬的移動模型環境發生變化時，update cost 的改變。然後加入我們所提出的調整機制後，update cost 的改變以及結構更新的訊息花費。從圖 5 可以看到環境發生變化時，如果能重新做調整，在此調整演算法還是採用 DAT，從圖中能看到 update cost 值能夠下降，同樣會隨著感測器數量愈多，下降差距愈大。這說明了如果環境改變，若沒有做調整還是使用原本的資訊收集樹，效能會隨著數量越多而下降越大，必須要經過調整後才會得到較佳的效能。圖 6 根據我們所提出調整策略來計算結構更新的訊息花費，當數量愈多，overhead 值愈大。

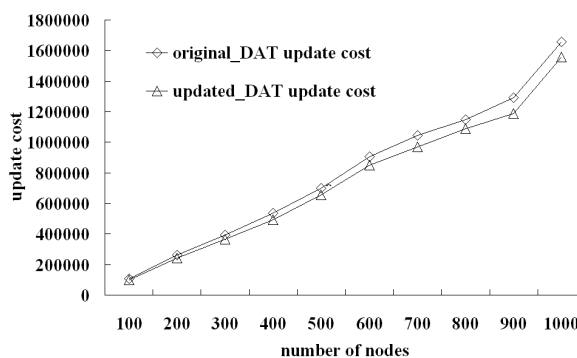


圖 5 在不同感測器數量下，環境改變後的 DAT 結構不變，和根據新環境重新建立的 DAT 的 update cost 比較

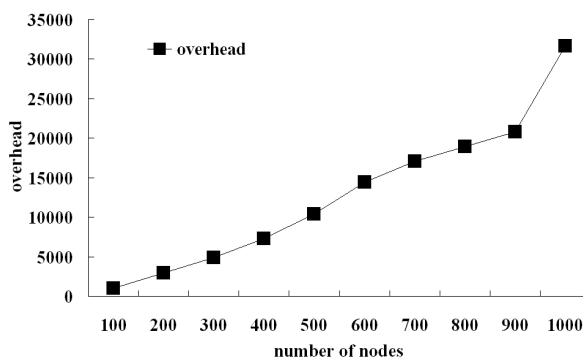


圖 6 overhead

圖 7 圖 8 所要探討的是當感測器數量不變固定 1000 點，感測器位置不變，門檻值 θ 設為 10%，我們隨機挑選 1 到 100 的點改變其環境，update cost 的改變。然後加入我們所提出的調整機制後，update cost 的改變以及結構更新的訊息花費。從圖 7 可以看到隨著改變數量愈多，移動模型環境發生變化時，同樣採用 DAT 做重新調整的演算法，隨著改變數量愈多，update cost 差距愈大，update cost 逐漸下降效能提高。在圖 7 的 update cost 會有雜亂情況是因為我們改變其環境時，是以隨機產生 100 個點的方式，所以 update cost 並不會呈現線性的方式，但還是可以看到選取改變數量愈多，update cost 差異越大。圖 8 計算結構更新的訊息花費，當改變數量愈多，overhead 值愈大。

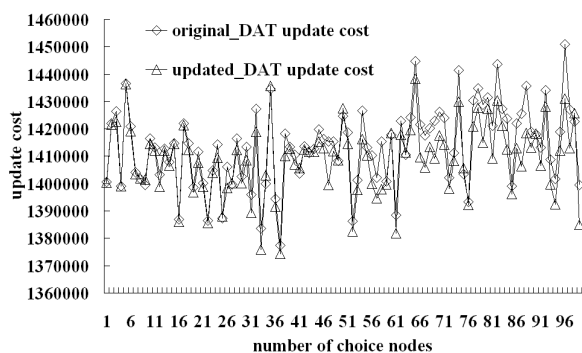


圖 7 隨機挑選 1 到 100 的點改變其環境下，環境改變後的 DAT 結構不變，和根據新環境重新建立的 DAT 的 update cost 比較

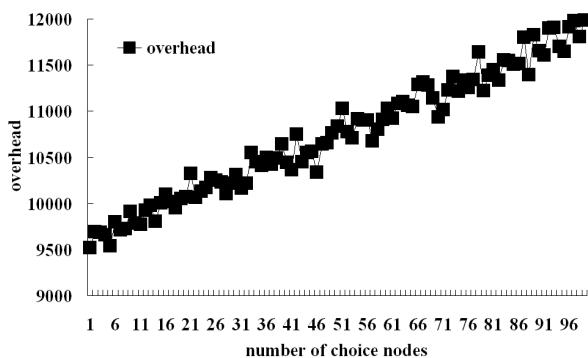


圖 8 overhead

圖 9 圖 10 圖 11 所要探討的是當感測器數量不變固定 1000 點，感測器位置不變，我們固定挑選感測器之間連線數量和位置，取全部連線數量的一半當作固定挑選的值，在不同的門檻值 θ 下，update cost 的改變。然後加入我們所提出的調整機制後，update cost 的改變以及結構更新的訊息花費。從圖 9 可以得知門檻值 θ 的改變，在固定環境變化下，同樣也是採用 DAT 做重新調整的演算法，並不會影響 update cost 減少或是增加。在圖 10 可以看到當門檻值 θ 越大，整個 overhead 會下降。

在圖 11 中我們分析結構更新訊息中，感測器回報訊息花費與資訊集中點通知更改訊息花費所佔的比例。overhead_report 表示在 overhead 計算中，有問題的點回報給 sink 的更

新訊息花費計算，如圖 2 的 step 3。overhead_adjant 則是表示在 overhead 計算中，由 sink 決定需要調整的感測器，進而傳送調整訊息的更新訊息花費計算，如圖 2 的 step 6。從圖 11 可以看到門檻值 θ 越大，因為已發調整的機會就越低，則發現需要進行結構調整的點也就越少，因此回報的訊息量就減少了。而調整訊息則不會隨著 θ 而改變。由此可知 θ 將會影響結構更新訊息中回報的訊息量。

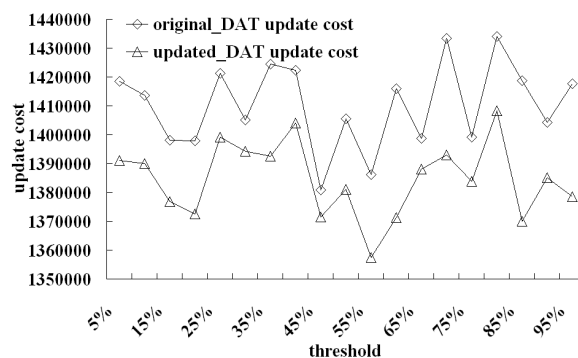


圖 9 固定挑選感測器之間連線數量和位置下，環境改變後的 DAT 結構不變，和根據新環境重新建立的 DAT 的 update cost 比較

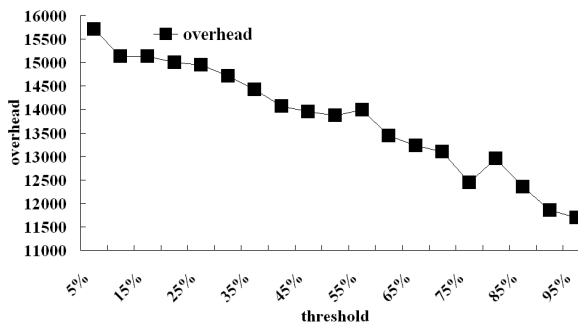


圖 10 overhead

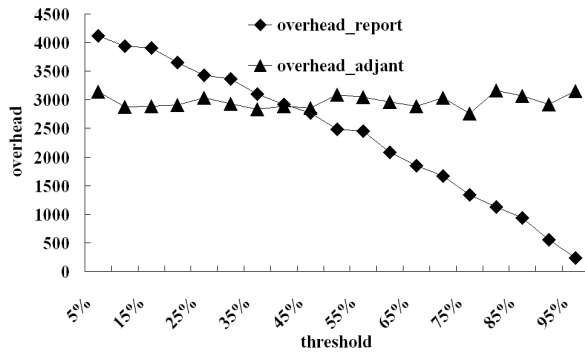


圖 11 有問題的點回報給 sink 和 sink 傳送調整訊息的比例

5. 結論

在相關論文研究中，可以發現到建立 hierarchy tree 都是基於物件移動頻率模型，但是真實環境是無法事先預測到物件移動的頻率，當假設的物件移動模型與真實常有所差距時，其效能會不如預期，所以我們提出動態調整策略。每個感測器都會查看與鄰居的傳送訊息頻率和之前紀錄的傳送訊息頻率差異是否超過門檻值，藉此觸發動態調整策略來調整 hierarchy tree 結構，使得效能和 update cost 都能保持一定水準。

參考文獻

- [1]. 王英鼎,陳旻秀 “無線感測網路於物件追蹤之應用”, WASN, 2007, 台北.
- [2]. H. T. Kung and D. Vlah “Efficient Location Tracking Using Sensor Networks”, IEEE International Conference on wireless communications and networking, 2003.
- [3]. Chih-Yu Lin, Wen-Chih Peng, and Yu-Chee Tseng, “Efficient In-Network Moving Object Tracking in Wireless Sensor Networks”, IEEE Trans. on Mobile

Computing., Vol. 5, pp. 1044-1056, Aug. 2006.

[4]. Chih-Yu Lin, Yu-Chee Tseng, Wen-Chih Peng, “Message-Efficient In-Network Location Management in a Multi-sink Wireless Sensor Network”, IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing Page(s): 496 – 505

[5]. Wen-Chih Peng, Yu-Zen Ko, Wang-Chien Lee, “On Mining Moving Patterns for Object Tracking Sensor Networks”, 2006 International Conference on Mobile Data Management Page(s): 41- 41

[6]. Sam Phu Manh Tran and T. Andrew Yang, “OCO:Optimixed Communication & Organization for Target Tracking in Wireless Sensor Networks”, Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing.

[7]. Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”, IEEE International Conference on System Sciences, 2000.

[8]. Yao-Chung Chang, Zhi-Sheng Lin and Jiann-Liang Chen “Cluster Based Self-Organization Management Protocols for Wireless Sensor Networks” IEEE Transaction on Consumer Electronics, 2006.

[9]. Yuhua Liu, Haiyan Zhu, Kaihua Xu, and Wei Teng “An Improved Route Repair Approach of Wireless Sensor Networks” Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences.