

設計並分析一個在無線感測網路中基於遞轉的便車式位置管理策略¹

Design and Analysis of Free-riding-based Location Management Schemes for Wireless Sensor Networks

胡庭瑜^a、王鼎超^b、陳朝鈞^a
國立成功大學 製造資訊與系統研究所^a
南台科技大學資訊管理系^b

摘要 —我們設計在無線感測網路中基於遞轉的便車式位置管理策略並結合了多點路由以及閒談式廣播的特性來設計物體移動的資料傳輸方式，使得通訊成本降低。我們開發一個隨機 Petri 網的數學模型來分析無線感測網路下的位置管理策略行為，進而來讓方法發精進。由實驗可知，基於遞轉的便車式位置管理策略可降低無線感測網路的通訊成本。

關鍵字: 無線感測網路、位置管理策略

一、簡介

無線感測網路(Wireless sensor networks, WSN)被視為近年來新興科技的重要關鍵技術之一。在無線感測網路的監控相關應用中，位置管理技術是最重要的技術之一[1]。目前有許多相關的應用都是架構在物體位置管理技術之上。以野生動物追蹤為例，當動物學家得知暴雨來之前的遷移行為，動物學家可以送出查詢來監測現在海燕的體溫。為了服務這樣的查詢，無限感測系統首先需要找到目前海燕所在的感測器，然後查詢回傳送至海燕所在的感測器。感測器在接收到動物學家的查詢後，感測器會傳送符合動物學家要求的感測資料，並經由無線感測網路及一般網路(Internet)傳給動物學家。

根據上述例子啟發，我們感興趣的是在這樣以位置為主的應用上，客戶藉由位置管理策略來需要找到移動物體。位置查詢可藉由移動對象所停留的感測器節點獲得所需的數據。一個簡單的方法來實現位置管理是要求的感測器網絡定期發送的實際位置給位置資料庫，客戶可經由當前位置查詢操作來從移動物體的位置資料庫中來查詢所請求的移動物體。然而，這方法會浪費大量的通信成本在更新並收集移動物體的位置上，因為一個移動的物體很容易地改變位置。此外，如果位置查詢到達率是相對低，簡單的方法將成為一個低效能的服務，因為這感測器不斷消耗的能量來更新資料庫但只供應少數的位置查詢。因此，一個可行的解決方法是，收集移動物體位置資料是通過網絡數據處理技術。以上述經驗來說，無線感測網路中的位置管理可以識別作為一個新的數據管理問題：如何設計 in-network 的數據處理機制，有效率地維護不斷移動物體位置更新和位置查詢服務，其目標量最小化所產生通信成本。

我們考慮多點路由以及閒談式廣播的特性來設計新

的位置管理機制。我們發現多點路由可以和傳統的樹狀結構以及遞轉(Forwarding)結構進行非常自然的搭配；而閒談式(Gossip-based)廣播可以進行搭便車式(Free-rider-based)的資料刪除。所謂搭便車式的資料刪除是指當一個節點進行移動物體的位置更新(又稱為註冊)，另一個具有舊位置資料的節點且其位置就在註冊節點的附近可以透過閒談式廣播的特性得知使用者已經在其他節點被註冊，因此可以自己刪除舊資料。如此，無線感測系統不必花費額外的通訊成本來刪除舊資料。將上述兩個硬體特性納入位置管理機制的設計是全新的機制，而且也是針對無線感測網路量身訂做的位置管理策略，並經實驗證實可減少大量的通訊成本。

在本文中，我們提出了基於遞轉的便車式位置管理策略，藉由多點路由以及閒談式廣播的特性可使無線感測網路的位置管理策略更有效率。基於遞轉的便車式位置管理策略是最大化搭便車式資料刪除的行為，進而降低了物體移動的通信成本，通過利用感測器特有的通信特性，稱為閒談特性(Gossip property)，刪除移動物體在樹狀資料結構中過時的位置數據。主要想法是物體移動的位置非常接近，由於物體移動位置具有地理的連續性。因此，節點可以藉由閒談特性來自動刪除原來的位置數據，而無需花費額外的通信成本。

為了優化基於遞轉的便車式位置管理策略的性能，我們開發了一個以馬可夫鍊(Markov Chain)為基礎的數學分析模型，透過使用隨機 Petri 網(Stochastic Petri Net, SPN)來估算通信成本[3]。藉由這些效能評估的結果，我們可以了解我們提出的混和多點路由及閒談式廣播特性之位置管理策略的優點，並且跟模擬結果顯示基於遞轉的便車式位置管理策略優於其他位置管理策略。

本文其他章節如下:第二節描述的系統模型。第三節基於遞轉的便車式位置管理策略。第四節效能評估模型。第五節實驗結果。第六節結論。

¹ 本研究由國科會贊助，計畫編號 NSC 100-2221-E-006-267-MY2 及 NSC 102-2221-E-006-249。

- 新感測器與前一個感測器的階層位置關係有 3 種可能，下降、上升或平移，被 SPN 模擬成 ODown、OUp 及 OEqual，其中階層數不變的機率為 P_{down} 、 P_{up} 及 P_{same} 。
 - 若新舊感測器的階層位置關係為下降，ODown 放置一個標號到 OLevel 中，其中 OLevel 的標號數代表物體所在的新感測器階層數。並且開始會遞轉鍊運作，被模擬成 ODown 放置一個標號到 UpdateChain 中。
 - 若新舊感測器的階層位置關係為上升，OLevel 中一個標號會流出到 DecreaseLevel，代表階層數少一層。並且開始會遞轉鍊運作，被模擬成 DecreaseLevel 放置一個標號到 UpdateChain 中。
 - 若新舊感測器的階層位置關係為平移，代表階層數不變。並且開始會遞轉鍊運作，被模擬成 OEqual 放置一個標號到 UpdateChain 中。

- 若遞轉長度小於最大遞轉長度(K)，Forward 會流出一個標號且 AddPointer 會產生一個標號放置到 KValue 一個，其中 KValue 的標號數代表遞轉長度。
- 遞轉長度等於最大遞轉長度(K)，代表更新遞轉鍊及啟動便車式註冊程序，Forward 會流出一個標號到 ClearChain，且 KValue 的標記會全部流走。
- 當更新遞轉鍊及啟動便車式註冊程序時，錨點的位置會被更新，此被模擬成 AnchorLevel 清空自己狀態內的標號並放置 OLevel 數目的標號。
- 若物體向後移動，UpdateChain 會流出一個標號且 Back 放置一個標號。遞轉鍊會被刪除及遞轉長度會減一，此被模擬成 Back 和 KValue 各流出一個標號。

位置查詢(Location query part)

- 位置查詢產生，被模擬成一個標號從 QueryIn 移動到 Query，QueryIn 產生標號的速率為 μ 。
- 位置查詢找到物體且回傳給 sink，被模擬成一個標號從 Query 移動到 QueryProcessing，其中處理速率為 r_{pro} 。

4.3 參數化

P_{down} 、 P_{up} 及 P_{same} 可由樹狀網路的階層節點數得知，可利用目前所在上到下階層的節點數估算，以 P_{up} 來說，目前階層為 Lo 其數值為所有可能經過的節點中出現上一層的機率，同理， P_{down} 為所有可能經過節點中出現下一層的機率， P_{same} 為所有可能經過節點中出現目前層的機率，意同於下方方程式。

$$P_{down} = \frac{level(Lo + 1)}{level(Lo + 1) + level(Lo + 1) + level(Lo + 1)}$$

$$P_{up} = \frac{level(Lo - 1)}{level(Lo + 1) + level(Lo + 1) + level(Lo + 1)}$$

$$P_{same} = \frac{level(Lo)}{level(Lo + 1) + level(Lo + 1) + level(Lo + 1)}$$

其中 $level(Lo)$ 表示第 Lo 階層中的總節點數，可藉由感測場範圍中第 Lo 層所佔的面積，再加上感測場範圍及感測器個數得知，假設sink到第 Lo 層的距離(R_{Lo})及到第 $Lo-1$ 層的距離(R_{Lo-1})，如式

$$level(Lo) = \frac{R_{Lo}^2 - R_{Lo-1}^2}{R^2} \times N$$

P_{fwd} 是物體移動到非前一個位置節點的機率，用數學表示成所有鄰居節點數出現不是物體回到歷史走過的節點的機率。可用下列式子表示。其中 N_{sur} 所有鄰居節點數(感測器周圍1hop內節點數)在下列實驗中數值約為9。

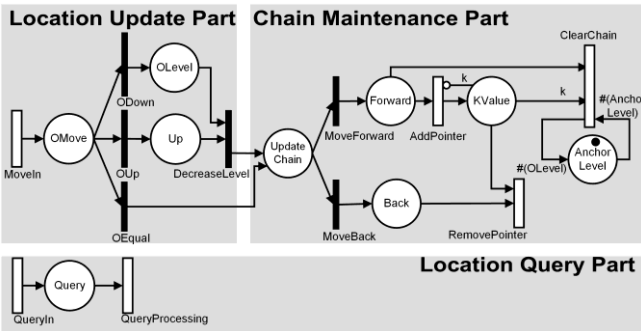
$$P_{fwd} = \begin{cases} 1, & \text{if Mark (KValue) = 0} \\ \frac{N_{sur} - 1}{N_{sur}}, & \text{otherwise} \end{cases}$$

P_{back} 是物體移動到前個位置節點的機率，用下列公式表示成所有鄰居節點數出現物體回到上次走過節點的機率。

$$P_{back} = \begin{cases} 0, & \text{if Mark (KValue) = 0} \\ \frac{1}{N_{sur}}, & \text{otherwise} \end{cases}$$

QueryProcessing 的速率為 r_{pro} ，代表查詢封包會經過 n_q 個 hop，換句話說每秒有 n_q 個查詢封包傳出，其中 n_q 為查詢封包從 sink 到 Anchor 沿著遞轉鍊到物體所在的節點，可用下列式子表示。

$$r_{pro} = \frac{1}{n_q} = \frac{1}{\text{Mark(Anchor Level)} + \text{Mark(KValue)}}$$



圖二：基於遞轉的便車式位置管理策略的 SPN 模型

表 I SPN 模型中 Place 和 Transition 的定義

名稱	定義
OMove	mark(OMove)=1 代表物體移動到另一個感測器
OLevel	mark(OLevel) 代表物體所在的階層數
Up	新舊感測器的階層位置關係為下降
UpdateChain	物體移動產生新的鏈結構
Forward	物體向前移動 (即移動到一個新的節點)
Back	物體向後移動 (即，移動到的前一節點)
KValue	遞轉鍊長度
AnchorLevel	mark(AnchorLevel) 代表物體所在的階層數
Query	mark(Query)=1 代表有一個位置查詢
MoveIn	物體移動，其速率為 λ
OUp	新舊感測器的階層位置關係為上升，其中機率為 P_{up}
ODown	新舊感測器的階層位置關係為下降，其中機率為 P_{down}
OEqual	新舊感測器的階層位置關係為不變，其中機率為 P_{same}
DecreaseLevel	階層數少一層
MoveForward	物體向前移動，其中機率為 P_{fwd}
MoveBack	物體向後移動，其中機率為 P_{back}
Addpointer	遞轉鍊長度加一，其速率為 r_{add}
RemovePointer	遞轉鍊長度減一，其速率為 r_{rem}
ClearChain	更新遞轉鍊，其速率為 r_{cc}
QueryIn	位置查詢產生，其速率為 μ
QueryProcessing	位置查詢被服務完，其速率為 r_{pro}

遞轉鍊的運作(Chain maintenance part)

- 物體移動後會運作遞轉鍊，被模擬成一個標號在 UpdateChain。一個標號可以移動著一個新節點或向後移到前一個節點，換句話說，遞轉長度會加一或減一。SPN 模型區分這種情況下使用 MoveForward 和 MoveBack，其機率為 P_{fwd} 及 P_{back} 。
 - 若物體向前移動，UpdateChain 會流出一個標號且 Forward 放置一個標號。

Addpointer 及 RemovePointer 的速率為 r_{add} 及 r_{rem} ，因為建製遞轉鍊只需要傳送一個註冊封包及一個ack給前個節點，所以只需要2個hop，可用下列式子表示。

$$r_{add} = r_{rem} = \frac{1}{2}$$

位置更新服務的平均成本(C_{update})為加/刪遞轉鍊的成本(C_{ptr})及更新遞轉鍊的成本，其中更新遞轉鍊的成本可分成向sink進行位置更新的成本(C_{res})及刪除舊資料的成本(C_{del})，可用下列式子表示。

$$C_{update} = C_{ptr} + (C_{res} + C_{del}) = P_{ptr} \times 2\tau + P_{chain} \times (C_{res} + C_{del})$$

其中 P_{ptr} 為發生加/刪遞轉鍊的機率，可由SPN中Forward及Back得知； P_{chain} 為發生更新遞轉鍊的機率可由SPN中KValue得知； C_{res} 代表平均物體更新到sink的註冊成本； C_{del} 代表平均刪除過時資料的解註冊成本。

$$C_{res} = \text{Mark}(\text{OLevel}) \times \tau$$

C_{del} 代表刪除錨點到 sink 及遞轉鍊過時資料的解註冊成本，特別不同的是，在我們策略中有便車式註冊程序可減少傳送解註冊封包的次數(N_{FR})，其中 $FR(i)$ 代表第 i 階欲解註冊的節點是否為 Free Rider 解註冊的節點， n 為目前物體所在節點的階層數， $d(i)$ 表示錨點和物體所在節點的第 i 層父節點之間的距離，若其距離小於閒談的最大距離表示可用閒談特性聽到註冊封包，反之則否。

$$C_{res} = (\text{Mark}(\text{Anchor Level}) + \text{Mark}(\text{KValue}) - N_{FR}) \times \tau$$

$$N_{FR} = \sum_{i=0}^n FR(i), \text{ where } FR(i) = \begin{cases} 1, & \text{if } (d(i) < R_{gossip}) \\ 0, & \text{otherwise} \end{cases}$$

查詢找出指定對象的平均值成本(C_{query})每一次查詢從sink到錨點再到物體所在感測器的通訊成本。

$$C_{query} = P_q \times (\text{Mark}(\text{Anchor Level}) + \text{Mark}(\text{KValue})) \times \tau$$

五、 實驗結果

在本節中，我們取得實驗數據來進行分析，藉由SPN分析模型計算得知。表 II 中顯示了實驗系統的參數和設置。此章節中我們重製遞轉策略(pointer forwarding scheme, PFS)並符合無線感測網路來做為比較對象[4]。與基於遞轉的便車式位置管理策略最大的不同點為沒有便車式註冊程序，換句話說，遞轉策略需要對每一個有過時資料的節點發送解註冊封包。

表 II. SPN 模型中 Place 和 Transition 的定義

名稱	定義
感測範圍(R)	500×500 平方公尺
感測器個數(N)	1000
通訊範圍	150 公尺
最大階層數	30
物體移動速率(λ)	[0.1~1.0] 個/秒
查詢的到達速率(μ)	0.1 個/秒
物體移動和查詢的比值($QMR=\lambda/\mu$)	0.5
資料傳遞一個 hop 的成本(τ)	1

5.1 效能的比較

圖 2 顯示在基於遞轉的便車式位置管理策略和遞轉策略在不同 QMR 對總通訊成本的影響。我們發現當移動次數越高(QMR 越大)時，基於遞轉的便車式位置管理策略的成本遠低於遞轉策略。這是因為基於遞轉的便車式位置管理策略利用閒談特性來進行過時資料刪除，與遞轉策略相比降低了傳送過時資料的刪除封包，所以在物體

移動比例越高時，可大幅降低通訊成本。

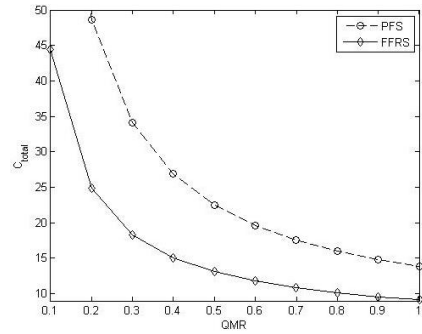


圖 2. QMR 對基於遞轉的便車式位置管理策略及 PFS 總通訊成本

5.2 敏感度分析

圖 3 顯示在基於遞轉的便車式位置管理策略和遞轉策略在不同通訊距離(又為閒談半徑)對兩策略總通訊成本差的影響。我們發現基於遞轉的便車式位置管理策略會隨通訊距離越大而讓改善成本高，這是因為基於遞轉的便車式位置管理策略利用閒談特性來進行過時資料刪除，當通訊距離越大代表可利用閒談特性刪除過時資料的節點變多，換句話說，發送解註冊封包次數減少；遞轉策略成本不受通訊距離影響，換句話說，不會因通訊距離減少封包發送次數，所以當感測器通訊距離越大，通訊成本變小。

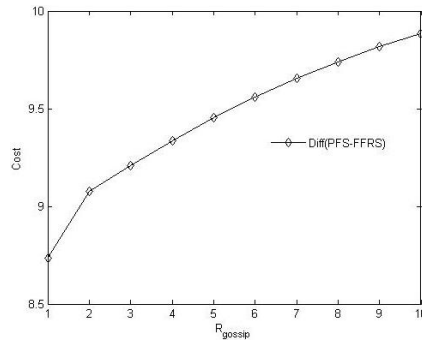


圖 3. 通訊距離對基於遞轉的便車式位置管理策略及 PFS 總通訊成本

結論

此篇論文的貢獻為(1)設計基於遞轉的便車式位置管理策略，並結合了多點路由以及閒談式廣播的特性，使得通訊成本降低。(2) 開發一個 SPN 數學模型來分析無線感測網路下基於遞轉的便車式位置管理策略的行為，進而來分析基於遞轉的便車式位置管理策略的方法效益。

參考文獻

- [1] K. R.V., F. A. and V.G.K., "Computational Intelligence in Wireless Sensor Networks: A Survey," IEEE Communications Surveys & Tutorials, vol.13, no.1, pp.68-96, First Quarter 2011.
- [2] C-C C and C-H L, "Model-based Object Tracking in Wireless Sensor Networks", Wireless Networks, Vol.17, No.2, February 2011, pp. 549-565.
- [3] C. H. B. T, and K. T., "SPNP: Stochastic petri nets.version 6.0.", volume 1786, pages 354-357. Springer Berlin, Heidelberg, 2000.
- [4] Y. L and I-R. C. "Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks." IEEE Transactions on Mobile Computing, 10(3):349-361, 2011.