

國立交通大學

資訊科學與工程研究所

碩 士 論 文

支援 IEEE 802.11s 無線網狀網路無縫交遞
之交叉點探索機制

Crossover Node Discovery for Seamless Handovers in
IEEE 802.11s Wireless Mesh Networks

研 究 生：廖冠銘

指 導 教 授：曾建超 教授

中 華 民 國 九 十 八 年 八 月

支援 IEEE 802.11s 無線網狀網路無縫交遞之交叉點探索機制
Crossover Node Discovery for Seamless Handovers in
IEEE 802.11s Wireless Mesh Networks

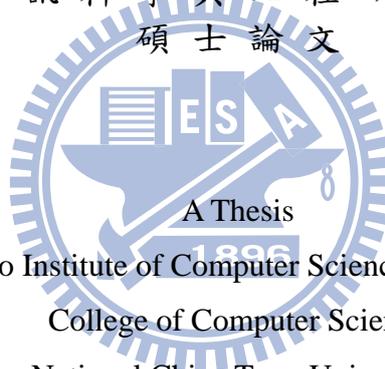
研究生：廖冠銘

Student : Kuan-Ming Liao

指導教授：曾建超

Advisor : Chien-Chao Tseng

國立交通大學
資訊科學與工程研究所
碩士論文



Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年八月

支援 IEEE 802.11s 無線網狀網路無縫交遞之 交叉點探索機制

研究生： 廖冠銘

指導教授： 曾建超 教授

國立交通大學資訊學院資訊科學與工程研究所

摘 要

本論文針對 IEEE 802.11s 無線網狀網路 (Wireless Mesh Networks)，提出一套有效率、支援行動節點 (Mobile Nodes) 換手到新網狀擷取點 (Mesh Access Points) 時欲利用交叉點 (Crossover Node) 的交叉點探索機制。所謂的交叉點即為由對應節點 (Corresponding Node，即與行動節點連線的節點) 到行動節點連接上的舊擷取點與新擷取點這兩條路徑的交點 (Intersection)。行動節點換手時即時找到並利用交叉點對於一些無縫交遞的機制是很重要的。舉凡如 Micro-mobility 便利用交叉點做為位置更新 (Location update) 的節點，利用此節點將封包導向 (Re-direct) 至新擷取點，如此便可以減少位置更新所需花費的時間以及封包繞路的行為；另外，利用交叉點做封包雙向傳送 (Bi-casting) 即是將封包複製一份分別往新、舊擷取點傳送，如此得以減少封包的遺失率 (Loss rates)。

雖然先前已有一些找尋交叉點的方法，但在 IEEE 802.11s 無線網狀網路環境中，這些方法卻無法正確地運作。在 IEEE 802.11s 無線網狀網路，網狀擷取點是做為封包路由至行動節點的代理者 (Proxy)，且是以 Layer-2 的方式做路由。此路由機制與其他網路環境不同。因為這特色，使得先前找尋交叉點的方法會有下列這些問題：第一，大多數先前研究是針對 MN 與 CN 在不同網域下的情形所設計的，因此這些方法當 MN 與 CN 在相同網域下時會找到錯誤的交叉點；第二，大多數這些方法是基於以行動節點為導向的路由環境所設計的，即封包的路由是以行動節點為目的進行傳送，並非基於 IEEE 802.11s 無線網狀網路中以代理為基礎 (Proxy-based) 的路由環境所設計，因此，即使是 MN 與 CN 在同一網域的情形，這些機制也會找到錯誤的交叉點；第三，先前研究皆是針對每一行動節點 (Per-station based) 的找尋交叉點方式，即每一行動節點連接上擷取點或換手至新擷取點便要重新做找尋交叉點的動作，這樣是很沒有效率的；最後，大多數這些機制皆僅在行動節點連接新擷取點時才做找尋交叉點的動作，如此只適用於利用交叉點做封包導向的情景，並無法運作於欲利用交叉點做封包雙向傳送的動作。因此，本篇論文致力於設計一套有效率的交叉點探索機制，此機制可運作於 IEEE

802.11s 無線網狀網路找到正確的交叉點且支援不論是封包導向或是封包雙向傳送的換手情形。

本論文提出一適用於 IEEE 802.11s 無線網狀網路之交叉點探索機制: Per-proxy-Oriented Partition-based (POP) CrN Discovery。IEEE 802.11s 無線網狀網路的路由機制使得其交叉點找尋不像其它網路環境那樣容易。如前所述，在 IEEE 802.11s 無線網狀網路，網狀擷取點是做為封包傳送至行動節點的代理者 (Proxy)。對應節點所連接上的網狀擷取點，或稱為來源網狀擷取點 (Source MAP) 與舊網狀擷取點以及新網狀擷取點便可決定一交叉點，並且，相同的舊網狀擷取點以及新網狀擷取點其交叉點會隨來源網狀擷取點的不同而不同。此外，真實 IEEE 802.11s 無線網狀網路佈建上，其路由路徑是幾乎不變的。

POP 機制有以下三個特點：針對每個網狀擷取點 (Per-proxy-oriented) 之交叉點找尋，每個網狀擷取點一次性 (One-time) 分群 (Partition-based) 之交叉點找尋以及交叉點資訊分散式化。所謂的每個網狀擷取點之交叉點找尋即 POP 是針對每個來源網狀擷取點做交叉點找尋程序，如此對於 MN 與 CN 不論在不同網域或是同一網域的連線情形都可以找到正確的交叉點；此外，POP 僅針對每個網狀擷取點利用分群的概念做一次性的交叉點找尋動作，並且其交叉點資訊是分散儲存在交叉點上，此項特點除了可以縮短交叉點找尋的程序外，還可以減少找尋交叉點程序所需花費的訊息量；最後，交叉點資訊分散式化使得此機制可以很彈性的運作於任何無縫換手時欲利用交叉點的情景，亦即封包導向或是封包雙向傳送都可以利用 POP 機制正確且快速地找到交叉點。

為了驗證 POP 機制其效能並與其它交叉點找尋機制做比較，我們使用了學術界廣為人知的網路模擬器 NS2 進行模擬實驗並與 Cross Router Pre-Discovery (CRPD) 做比較。這是由於 CRPD 也是由來源節點 (此為 GW) 去做找尋交叉點的動作，雖然並不是由來源代理者。實驗結果顯示 POP 在交叉點找尋所需花費的訊息量會較 CRPD 來的少，這是因為 POP 所花費的訊息量並不受行動節點的數量與其換手次數的影響。

關鍵詞： IEEE 802.11s 無線網狀網路、交叉點、無縫交遞、封包轉送、封包雙向傳送

Crossover Node Discovery for Seamless Handovers in IEEE 802.11s Wireless Mesh Networks

Student : Kuan-Ming Liao

Advisor : Dr. Chien-Chao Tseng

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

Abstract

In this thesis, we proposed an effective *Crossover Node* (CrN) discovery mechanism for Mobile Nodes (MN), which may change their attached *Mesh Access Points* (MAPs), in IEEE 802.11s Wireless Mesh Networks (WMNs). A CrN for an MN is the intersection of the path from a corresponding node (CN), which is communicating with the MN, to the MN's old Access Point (oAP) and the one from the CN to the MN's new AP (nAP). Finding CrNs for an imminent handover is crucial to some seamless handover schemes. Micro-mobility schemes employ CrNs to localize location updates and re-direct data packets coming from CNs to nAP, and thus can reduce both the latencies of location updates and packet deliveries during handovers. Bi-casting utilizes for generating two duplicated data stream to both oAP and nAP, to reduce packet loss rates during handovers.

Although some CrN discovery methods have been proposed previously in the literature, none of them can work correctly, not to say efficiently, for 802.11 WMNs. In 802.11s WMNs, MAPs serve as proxies that perform layer-2 routing to deliver packets on behalf of stations. The proxy-based layer-2 routing mechanism make 802.11s WMNs different than other wireless networks. As a consequence, all previous CrN discovery methods possess some of the following four problems. First, most of the previous Crossover Node discovery mechanisms are designed for an MN's inter-domain sessions with external CNs that situated at domains differ than the one visited by the MN, thus may find incorrect CrNs for the MN's intra-domain sessions with internal CNs.

Second, most of them are station-oriented and intended for the station-to-station routing, not for the proxy based one as in 802.11s WMNs. Therefore, some of the previous mechanism may find incorrect Crossover Node even for the MN' s inter-domain sessions. Third, all previous mechanisms are per-station based approaches and need to perform a CrN discovery procedure for an MN' s each communication session with a CN, each time when the MN moves. Therefore, they all cannot work efficiently for 802.11s WMNs. Last but not least, most of previous schemes perform CrN discovery procedure only when MNs attach to nAPs and thus can only use CrN for data re-direction not for bi-casting. Therefore, this thesis aims for designing an effective CrN discovery mechanism that can find the correct CrN for 802.11s WMNs and support both data re-direction and bi-casting.

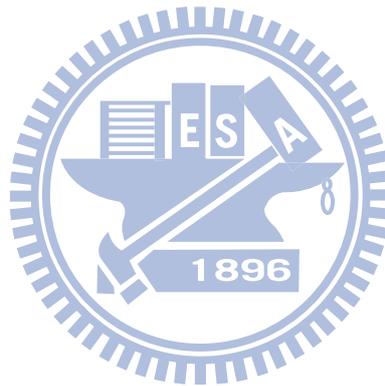
In this thesis, we propose a Per-proxy-Oriented Partition-based (POP) CrN Discovery Scheme for 802.11s WMNs. The characteristics of 802.11s WMNs make CrN discovery in 802.11s WMNs different from that in other wireless networks. In 802.11s WMNs, MAPs serve as proxies for routing packets on behalf of stations. Therefore, the serving MAP of a CN, henceforth called source MAP (sMAP), and the old MAP (oMAP) and new MAP (nMAP) of an MN determine a unique CrN for the communication path from the CN to the MN, and different sMAP may result in different CrN for a pair of oMAP and nMAP. Furthermore, in a real deployment of 802.11s WMNs, the route between two MAPs are likely to be static.

Therefore, POP adopts three design principles, Per-proxy-oriented CrN discovery, One-time Partition-based approach, and Distributed CrN tables. By adopting Per-proxy-oriented CrN discovery, POP performs a CrN discovery process for each sMAP and thus can find the correct CrN for all sessions, not matter inter- or intra-domain, of an MN. Furthermore, POP performs the CrN discovery procedure just once for each sMAP by using the concept of set partitioning and storing the CrN information at each CrN itself. The One-time Partition-based approach and Distributed CrN Tables, together, not only speedup the CrN discover processes but also reduce significantly the number of messages required for the processes. Finally, Distributed CrN Tables also make it very flexible in using CrN for seamless handover techniques; POP can

be used for both data re-direction and bi-casting.

In addition of comparing POP qualitatively with other CrN discovery schemes, we also conduct a simulation using a well known network simulator NS2 and compare POP qualitatively with Cross Router Discovery (CRPD) scheme because it also performs CrN discovery from source node, although not the source proxy. Experiment results show that POP outperforms CRPD in terms of the number of discovery messages because the number of discovery messages introduced by POP is independent of the number of MNs in a 802.11s WMN and the number of handovers each MN encountered.

Keywords: IEEE 802.11s Wireless Mesh Networks, Crossover Node, Seamless handover, Re-direct, Bi-casting



致 謝

首先要感謝曾建超老師這兩年多來的指導，提供我這麼好的研究環境，讓我可以沒有後顧之憂地完成這篇論文。在我的論文的想法一出來時，提醒我需專注的論點。謝謝老師能夠不厭其煩的提醒我常犯的錯誤及方法修正方法的缺點，讓這篇論文的完整性更加提升。

再來要感謝各位口試委員，包括了王讚彬老師、嚴力行老師以及曹孝櫟老師，在百忙之中抽空來參加我的碩士論文口試，並提供了許多寶貴的意見，讓我能更清楚瞭解自己論文的貢獻與價值。

感謝實驗室的學長群們，Gunter、Gary、大樑哥、承運。尤其是史老大，謝謝學長在繁忙的博士生涯中抽空聽我的論文且給了我許多建設性的建議，並且指正我報告的方式。謝謝同屆的夥伴們：良哥、俊延、媛莉、興遷和又仁，無論是在課業上或是撰寫論文的歷程中，都給予了我許多幫助，讓我從沒感覺到是一個人奮鬥；謝謝我最可愛的學弟妹們，讓我在撰寫論文的碩二期間從來不會感到枯燥乏味，小毛、阿盧、坤穎、蛋捲、竣晨和銘祥，很開心認識了你們。

還有瑞昱聯合研發中心的大家，謝謝幹頭讓我瞭解做任何事情應有的態度，這是很很有幫助的。

謝謝我的女朋友牛奶寶寶，一直在背後默默地給我鼓勵，讓我在求學階段最失意的時候重新燃起動力，這些年一直不能陪在你身邊真是辛苦你了，謝謝妳一直體諒任性的我。最後，還要感謝我的家人，感謝爸爸媽媽一路上的支持，姊姊以及妹妹，讓我可以順順利利的走到這裡。

目 錄

摘 要.....	iii
Abstract.....	v
致 謝.....	viii
目 錄.....	ix
圖目錄.....	xi
表目錄.....	xiii
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 研究目的.....	2
1.3 章節簡介.....	3
第二章 背景知識介紹.....	4
2.1 IEEE 802.11s 無線網狀網路.....	4
2.1.1 IEEE 802.11s 無線網狀網路架構.....	4
2.1.2 IEEE 802.11s 無線網狀網路路由機制 - Proxy-based Routing.....	5
2.1.3 IEEE 802.11s 無線網狀網路換手情形及問題.....	9
2.2 無接縫漫遊機制之交叉點使用情景.....	9
2.2.1 交叉點之定義.....	10
2.2.2 換手前使用交叉點之情景 (Bi-casting)	11
2.2.3 換手後使用交叉點之情景 (Data-Redirection)	12
2.3 單元總結.....	13
第三章 相關研究.....	14
3.1 找尋交叉點所遭遇之困難.....	14
3.2 找尋交叉點之機制.....	16
3.2.1 Use the information of routing table.....	17
3.2.2 Use the information of Binding Cache.....	20
3.2.3 Use the information of Hierarchical model.....	23
3.2.4 Use the information of Geopaging routing table.....	25
3.2.5 找尋交叉點機制之比較.....	26

3.3 單元總結.....	27
第四章 Per-proxy-oriented Partition-based (POP) Crossover Node Discovery Mechanism.....	28
4.1 簡介	28
4.2 交叉點探索機制.....	29
4.2.1 交叉點探索機制 POP 之概念.....	29
4.2.2 交叉點探索機制 POP 之方法.....	31
4.2.3 交叉點探索機制 POP 之 Model.....	36
4.3 交叉點使用之情節	42
4.4 POP 機制之延伸	45
4.5 POP 機制之探討.....	47
4.6 單元總結.....	48
第五章 模擬實驗結果與討論.....	49
5.1 簡介	49
5.2 模擬環境.....	50
5.3 實驗結果與分析.....	51
5.4 貢獻	56
5.5 單元總結.....	56
第六章 結論與未來工作.....	58
6.1 結論	58
6.2 未來工作.....	59
Reference.....	60



圖目錄

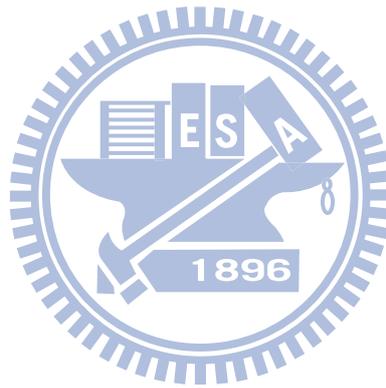
Figure 2-1 無線網狀網路與一般無線網路基本架構示意圖.....	5
Figure 2-2 無線網狀網路封包傳送示意圖(一).....	7
Figure 2-3 無線網狀網路封包傳送示意圖(二).....	7
Figure 2-4 交叉點示意圖.....	10
Figure 2-5 利用交叉點 Bi-casting 示意圖.....	11
Figure 2-6 利用交叉點做封包導向示意圖.....	13
Figure 3-1 以 GW 為出發點所找尋之交叉點示意圖.....	15
Figure 3-2 CN 與 MN 在同一個網路底下其交叉點之示意圖.....	16
Figure 3-3 CRPD 訊息示意圖.....	17
Figure 3-4 CRPD 重複性找尋交叉點示意圖.....	19
Figure 3-5 Route update 更新流程示意圖.....	20
Figure 3-6 行動裝置換手後示意圖.....	21
Figure 3-7 無線網狀網路利用 Binding Cache 找尋交叉點機制示意圖.....	22
Figure 3-8 Hierarchical Mobile IP 架構圖.....	23
Figure 3-9 Hierarchical Mobile IP Regional Registration 示意圖.....	24
Figure 3-10 OCRD 找尋交叉點示意圖.....	25
Figure 4-1 路徑生成的 Tree 示意圖.....	29
Figure 4-2 Tree rooted from A 之分群示意圖.....	30
Figure 4-3 Sub-tree rooted from C 之分群示意圖.....	31
Figure 4-4 節點 A 交叉點探索機制之流程(一).....	32
Figure 4-5 節點 A 交叉點探索機制之流程(二).....	33
Figure 4-6 節點 A 交叉點探索機制之流程(三).....	34
Figure 4-7 節點 A 交叉點探索機制之交叉點表格資訊.....	35
Figure 4-8 交叉點探索機制流程圖示意圖.....	37
Figure 4-9 POP 示例示意圖.....	38
Figure 4-10 節點 A 分群及傳遞方向示意圖(一).....	39
Figure 4-11 節點 A 分群及傳遞方向示意圖(二).....	40
Figure 4-12 節點 A 分群及傳遞方向示意圖(三).....	41
Figure 4-13 利用交叉點表格之示意圖 - 換手前.....	42
Figure 4-14 利用交叉點表格之示意圖 - 換手後.....	43
Figure 4-15 行動裝置延原路徑換手之示意圖.....	44

Figure 4-16 交叉點運用流程圖示意圖	45
Figure 4-17 POP 機制延伸之示意圖.....	46
Figure 4-18 無線網狀網路 Multiple Tree 環境之示意圖	47
Figure 5-1 模擬環境示意圖	51
Figure 5-2 POP 與 CRPD 之訊息計算方式示意圖	51
Figure 5-3 Grid 3×3 之找尋交叉點訊息量.....	52
Figure 5-4 Grid 4×4 之找尋交叉點訊息量.....	53
Figure 5-5 Grid 5×5 之找尋交叉點訊息量.....	53
Figure 5-6 Grid 6×6 之找尋交叉點訊息量.....	53
Figure 5-7 1 MN 之找尋交叉點訊息量	54
Figure 5-8 5 MN 之找尋交叉點訊息量	55
Figure 5-9 10 MN 之找尋交叉點訊息量.....	55



表目錄

Table 3-1 先前研究於非 802.11s 與 802.11s 其 inter- and intra-domain session 時比較表	26
Table 4-1 節點 A 之路由表格	32
Table 4-2 節點 C 之路由表格	33
Table 4-3 節點 F 之路由表格	34
Table 5-1 模擬參數設定值.....	50



第一章 緒論

1.1 研究動機

近年來，無線寬頻技術快速蓬勃發展，舉凡如 IEEE 802.11 [1]、3G、3.5G、WiMax 以及 3GPP 目前正在推動的長期演進技術 (Long Term Evolution - LTE [2])，目的都希望使用者能利用無線的方式來享受寬頻的網路服務。IEEE 802.11s [3]，無線網狀網路，為 IEEE 802.11 所制定的網狀網路標準，主要制定無線節點彼此間的連接與溝通方式 (Interconnection)，無線網狀網路內的網狀節點 (Mesh Point - MP) 是透過無線的方式連接，其僅需要少數幾個 (通常為一至二個) 網狀節點擔任網狀入口 (Mesh Portal Point - MPP) 的角色，即接有線連接後端的骨幹網路，負責網狀網路對外的連線。無線網狀網路目的為減少佈建網路時所需花費的時間與成本，因其不受地理因素的影響，無線網狀節點佈建好開機後，便可以主動連接起來，而不像佈建擷取點 (Access Point - AP) 其每個擷取點都必須連接後端的骨幹網路，造成佈建的不易。在擴大或縮小網路覆蓋範圍時，也只要直接增加或減少網狀節點即可，可說是相當的便利。

無線網路的核心精神就是移動性，讓使用者能夠在使用過程中到處漫遊。但是，使用者常常會遇到一個問題，就是無線網路在使用者移動的情況下會發生斷線、訊號不穩定的問題。這種情形在 802.11s 覆蓋範圍廣的無線網路環境尤其可見其發生的頻率，當使用者換手 (Handover) 時，若沒有適當的無接縫漫遊機制 (Mobility Management)，使用者使用網路的情形便會被迫中斷，尤其是使用 Voice over Internet Protocol (VoIP) 等即時性應用服務，影響更為顯著，能夠快速且順暢地在擷取點間換手是一項重要的需求。因此，解決換手所造成的上述問題，一直是許多相關研究的目標。

從以往的研究得知，交叉點 (Crossover Node) 經常在無接縫漫遊機制中扮演著重要的角色，這裡先解釋一下何謂交叉點：其即為兩條路徑最後一個相同的節點。交叉點的應用範圍很多，舉凡如換手前或換手後都可以利用交叉點來提高換手的效益，交叉點常使用的情景如下：在行動裝置即將要換手的時候，利用交叉點將原本要送往行動裝置的封包複製 (Duplicate) 一份並傳往行動裝置即將連接上的擷取點，而當行動裝置一連接上新擷取點時便可以立即取得封包，如此便得以縮短換手延遲 (Handover Latency)，此行為通常稱為 Bi-casting。而 Bi-casting 的行為並非一直持續下去，當行動裝置連接上新擷取點時還需要做位置更新 (Location Update) 的動作，此動作為更新需要被通知的節點 (通常為來源節點) 該行動裝置已經換手到新擷取點底下，不要

再將封包往舊擷取點傳送，而在更新的過程中同樣可以在交叉點收到更新訊息的時候提早將 Bi-casting 往舊擷取點傳送的封包停止傳送，僅傳送給新擷取點即可，因為行動裝置已經連接上新擷取點了，送往舊擷取點的封包實屬不必要，如此亦可以減少頻寬的浪費。

另外一個使用交叉點的情景為在行動裝置連接上新擷取點時，做位置更新的時候使用，在更新的過程中，若交叉點收到此更新訊息便可以提前將封包直接傳往新擷取點，此動作稱為封包導向 (Data-Redirection)，此交叉點的使用可以減少送往舊擷取點的封包數；若舊擷取點沒有暫存機制 (Buffer)，便可以減少封包遺失 (Packets Loss)；而若舊擷取點有暫存機制，即可以減少封包的繞路。因為若沒有利用交叉點，在來源節點收到位置更新前，封包都會一直往舊擷取點傳送，但行動裝置卻已經不連接在舊擷取點底下。此時舊擷取點收到封包後不是將之丟棄 (造成封包遺失)，就是將之轉往新擷取點 (造成封包的繞路)，視舊擷取點的機制而定。因此，若能利用交叉點提前將封包導向傳送至新擷取點，便可以減少上述這些問題。

由以上過去研究使用交叉點的情景，我們可以得知，交叉點在漫遊機制中扮演著很重要的角色的，善用交叉點確實可以有效增加換手的效益。而無線網狀網路由於網狀節點間彼此透過無線的方式連接，隨著節點傳送的 Hop 數增多，上述問題就更顯得嚴重，因此，若能在無線網狀網路環境使用交叉點，勢必也可以增加換手的效益，減少上述情況所會發生的問題。

目前尋找交叉點的研究並不適用於無線網狀網路的環境，因為這些研究通常利用該網路環境的特色來找尋交叉點，並且除了找尋訊息量過多以及找尋次數過多外，這些研究對於來源行動裝置與目的行動裝置在同一個無線網狀網路內時，也會找到錯誤的交叉點。因此本論文即是從此處出發，希望能夠改善解決此問題。

1.2 研究目的

在上一節裡，我們已經瞭解所要研究的問題以及需求所在，因此本節中將簡單敘述我們想要達成的目標。

根據先前的研究，適時的利用交叉點確實可以改善換手的延遲與減少頻寬的浪費，又由於無線網狀網路內無線節點無線連結的關係，因此，我們認為在無線網狀網路的環境使用交叉點亦可以改善換手所造成的問題。

以往的找尋交叉點研究，其找尋交叉點通常直接利用該網路的特性去尋找，而這樣的特性與無線網狀網路的環境中是不相同的，且這些研究找尋交叉點所會花費的訊息量數太高，我們稍後會對這些研究做詳細的介紹。

另外，這些研究在找尋交叉點時，是針對目的行動裝置 (Mobile Node - MN) 與來源行動裝置 (Corresponding Node - CN) 在不同網域下溝通的情形做找尋，因此沒有考慮當目的行動裝置與來源行動裝置在同一網域下溝通時，目的行動裝置換手的情形。當這些情況發生時，所要利用的交叉點便不再是以閘道器 (Gateway) 為 Root 所長出的樹狀架構的交叉點了，其真正的交叉點必須為以來源行動裝置所連接上的節點為 Root 所長出的樹狀架構，依據此樹狀架構所找到的交叉點才是正確的。其樹狀結構代表以一節點出發至其它所有節點的路徑。而先前的研究其找尋交叉點機制僅找尋了以閘道器為 Root 所找出的樹狀架構所得到的交叉點，對於在目的行動裝置與來源行動裝置在同一網狀網路下溝通時，其所找到的交叉點會是錯誤的。而當在這些情況下換手時，若使用了錯誤的交叉點，對於增進換手的效益是沒有幫助的。因此，找尋到正確的交叉點以適用於任何的換手環境是很重要的。

本論文所提出的「交叉點探索之機制」，目的就是要找出所有並且正確的交叉點。其主要為透過路由表的資訊以及分群的概念，利用最少的訊息傳遞量，找到無線網狀網路內以任一節點為來源節點的任兩網狀節點間的交叉點。而先前的研究為根據行動裝置的連接去找尋交叉點，因此花費的訊息量在行動裝置數量多時會很可觀，而本方法可以一次性的找尋任兩網狀節點間的交叉點且不受行動裝置連接的影響。因此，本方法可以很有效率且快速的找出所有正確的交叉點，並且適用於所有的換手環境。我們會將所提出的機制利用 Network Simulator-2 (NS-2)，嘗試與分析此機制運作的結果與效能，驗證此作法的可行性。

1.3 章節簡介

本篇論文的章節編排與內容簡介如下：

第一章：緒論，簡介本篇論文的研究動機即預期達到的目標。

第二章：背景知識概述，介紹本篇論文所要探討的主題及其所應用到的相關知識，包括 802.11s 的基本架構及其路由機制，802.11s 行動裝置換手後所會遭遇到的問題及基本漫遊機制中利用交叉點情景的簡介。

第三章：相關研究，簡介本篇論文主題之前發表過的相關研究文獻資料。

第四章：交叉點探索機制，說明我們所提出之交叉點探索機制的原理與運作流程。

第五章：效能分析與結果，針對我們所提出的交叉點探索機制，進行相關的模擬環境測試，並提供效能分析報告。

第六章：結論與未來工作，總結本篇論文的結果，以及未來可繼續研究的方向。

第二章 背景知識介紹

2.1 IEEE 802.11s 無線網狀網路

IEEE 802.11s [3]為無線網狀網路 (Wireless Mesh Network - WMN) 標準規範，為 IEEE 802.11 為了實現網狀網路所制定的標準，目前仍是草案 (Draft)，其主要定義了無線節點彼此間的連接與溝通方式。想像一個情境，在一個由許多網狀節點所形成的網路，這些網狀節點僅需要其中一、二個節點連結後端的有線骨幹網路，所有的網狀節點皆透過無線的方式相互連結。透過這些網狀節點的無線連接，可以很輕易的擴大整個網路的覆蓋範圍，且佈建的過程也不需要花費龐大的工程拉線。因此，無線網狀網路的便利性是顯而易見的。IEEE 802.11s 規格主要的目標如下 [4]：

- 增加網路的涵蓋範圍與使用上的彈性
- 可靠的運作效率
- 無縫隙的安全性
- 支援多媒體的傳輸
- 運作省電
- 與 802.11 相容
- 能經由網路的不同階層



底下就無線網狀網路架構、無線網狀網路的路由 (Routing) 機制以及無線網狀網路的換手所會遭遇的困難依序介紹。

2.1.1 IEEE 802.11s 無線網狀網路架構

802.11s 網路將節點依其功能性分為三類，網狀路口 (Mesh Portal Point - MPP)、網狀擷取點 (Mesh Access Point - MAP) 以及網狀節點 (Mesh Point - MP)，說明如下：

- MPP：為連接後端骨幹網路的節點，一個網狀網路內至少要有一個網狀路口。
- MAP：可供行動裝置連接上的節點。
- MP：不含上述兩項功能的無線網狀網路內的節點，負責封包的路由，無線網狀網路內的所有節點都可以稱作是 MP。

簡單來說，構成無線網狀網路的節點都可以稱其為 MP，負責封包的路由；若此

MP 有對外連接後端骨幹網路的能力，便可稱其為 MPP；而若此 MP 具有讓行動裝置連接的能力，便可稱其為 MAP，底下提到的網狀節點亦可以代表網狀擷取點或是網狀路口。一個網狀網路的基本架構如 Figure 2-1 所示：

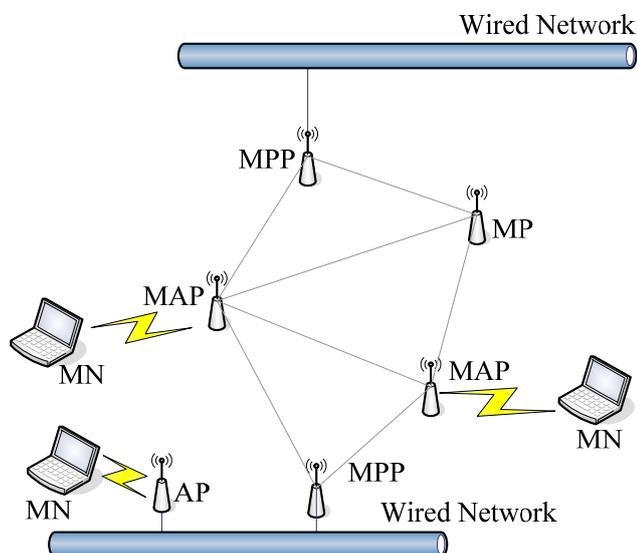


Figure 2-1 無線網狀網路與一般無線網路基本架構示意圖

2.1.2 IEEE 802.11s 無線網狀網路路由機制—Proxy-based Routing

網狀節點形成無線網狀網路後，其最重要的莫過於是資料封包處理與轉送：當無線擷取點收到來自行動裝置的封包後，要如何將封包路由到正確的目的地呢？此行為在無線網狀網路裡扮演著很重要的角色。混合型無線網狀協定 (Hybrid Wireless Mesh Protocol - HWMP) 為 802.11s 所制定的網狀路徑選擇協定 (Mesh Path Selection Protocol)，其結合了需要式路徑選擇 (On-demand Path Selection) 與主動性樹狀建立 (Proactive Tree Building)。需要式路徑選擇提供了在變動環境時路徑選擇的彈性，而主動性樹狀建立則是提供在固定環境時有較好的效率。IEEE 802.11s 建議這兩種路徑選擇協定在無線網狀網路形成時是同時存在的 [3]。

其主要概念便是，當無線網狀網路形成時，網狀路口會主動地建立以其為 Root 的樹狀網路 (Tree-based) 拓樸，此即主動性樹狀建立。建好後，網狀路口便會有到所有網狀節點的路徑 (以 <目的地, 下一個節點> 方式儲存)；而所有網狀節點同樣也會有到網狀路口的路徑。這對封包需要經由網狀路口進出其效果是很好的，不過，若有封包要送給同一個網狀網路內的節點或是行動裝置，若只靠此路徑協定，都必須先將封包送往網狀路口，再由網狀路口送至目的地，就會有繞遠路的情形。需要式路徑選擇協定就是為了解決上述問題，當有封包要送給同一個網狀網路內的節點時，來源網狀節點啟動需要式路徑選擇協定可以找到到目的網狀節點的最好路徑，此法僅在網狀節點要傳

送封包時且沒有到目的節點的路徑才會啟動。需要式路徑選擇協定是使用 Radio Aware Ad Hoc On Demand Distance Vector (RA-AODV) 此路徑協定，本論文不會詳細說明這兩種路徑選擇的細節，僅會介紹藉由上述這兩種路徑選擇找到路徑後，網狀節點如何根據這些資訊將封包傳送到目的地，若讀者有興趣可自行參考 IEEE 802.11s Draft [3]，裡頭有很詳細的說明。

無線網狀網路的封包路由是以代理為基礎的路由方式 (Proxy-based Routing)，亦即，行動裝置連接上的網狀擷取點做為代理行動裝置的角色，封包在網狀網路內的路由都是針對代理的角色 (即網狀節點) 去查找而非行動裝置本身，不像一般的網路路由都是針對行動裝置去查找。

其主要依照兩個表格的查詢在進行：路由表 (Routing Table) 以及代理表 (Proxy Table)。路由表是封包在無線網狀網路內路由時會查詢的表格，紀錄格式為 < Destination, Next Hop >，代表若要將封包路由至目的地網狀節點時，需將封包傳往哪個網狀節點；而代理表則是用來紀錄網狀擷取點與行動裝置的對應關係，紀錄格式為 < Owner, Station >，提供行動裝置連結在哪一個網狀擷取點的資訊。當一個來源網狀擷取點要幫連接上的來源行動裝置傳送封包時，會有下列幾個動作：

- 一、先查詢代理表，得知目的行動裝置連結在哪一個網狀擷取點底下。跳到 Step 2
- 二、接著查詢路由表，得知封包送給目的網狀節點要傳送給哪一個網狀節點，接著便將封包傳送給查詢到的網狀節點。跳到 Step 3
- 三、網狀節點收到封包後，檢查自己是否為目的地網狀節點，是便跳到 Step 4，否則重複 Step 2。
- 四、目的網狀擷取點將封包收下，接著再將封包傳送給連接上的目的行動裝置。

Figure 2-2 為在代理表與路由表有資訊的情況下，封包傳送的示意圖，CN 欲傳送封包給 MN：

1. MAP-S 收到 CN 的封包，查詢代理表得知 MN 於 MAP-D 底下。
2. MAP-S 接著查詢路由表，得知欲到 MAP-D 的下一個節點為 MAP-A。
3. MAP-S 將封包傳往 MAP-A。
4. 所有中間的節點皆查詢路由表，查找到 MAP-D 的下一個節點。

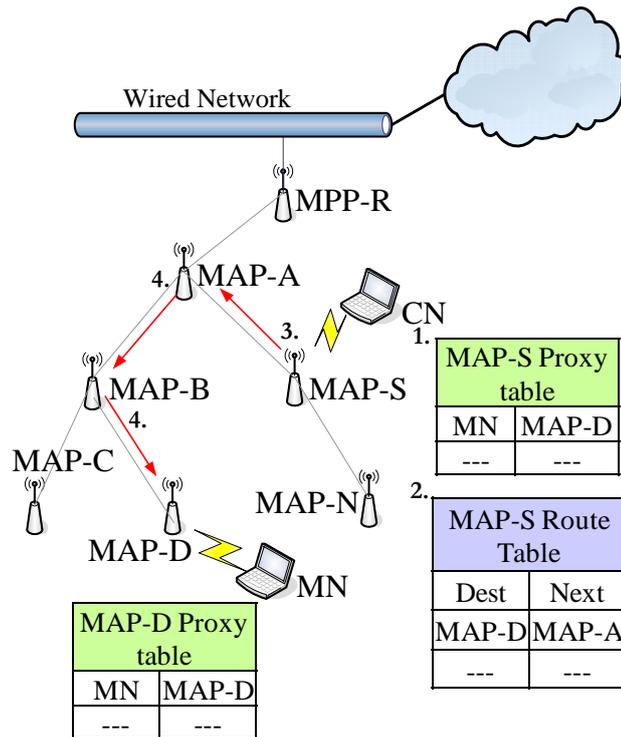


Figure 2-2 無線網狀網路封包傳送示意圖(一)

上述之行為是建立在已知目的行動裝置連結在哪一個網狀擷取點以及已知如何到此網狀擷取點的情形下的步驟，即由代理表和路由表可查詢到資訊。若是一開始的網狀網路情形，來源行動裝置可能會不知道目的行動裝置的實體位址 (MAC Address，802.11s 是根據實體位址來路由的，也就是所謂的 Layer2 Routing)，而僅會知道目的行動裝置的 IP 位址，並且路由表和代理表也尚未有相對應的資訊，此時無線網狀網路其行為又會是怎樣呢？底下便介紹在這樣情況下時行動裝置欲傳送封包其網狀節點的行為。

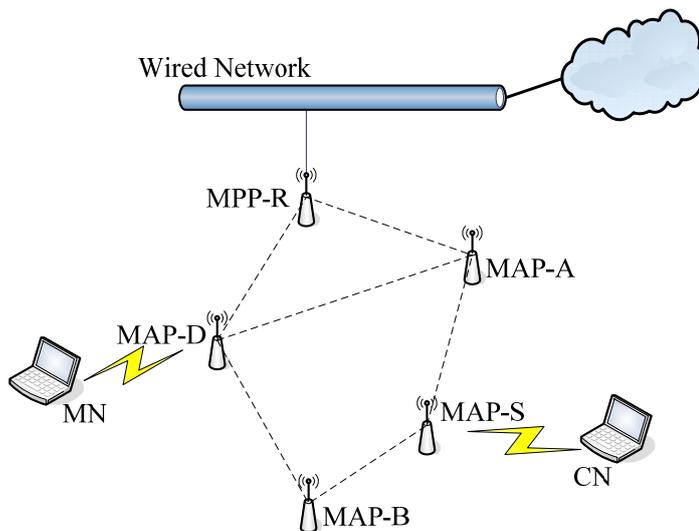


Figure 2-3 無線網狀網路封包傳送示意圖(二)

根據上述，當無線網狀網路形成時，網狀路口會啟動主動性樹狀建立，之後網狀路口便會有到所有網狀節點的路徑，反之亦然。Figure 2-3 顯示，一開始行動裝置 CN 連上網狀擷取點 MAP-S；行動裝置 MN 連上網狀擷取點 MAP-D。CN 欲傳送封包給 MN，但 CN 僅知道 MN 的 IP 位址，此時傳送封包的步驟如下：

1. CN 發送 ARP request 廣播以詢問 MN 的實體位址。
2. MN 收到 ARP request 後，回應 ARP reply 單點傳播，目的位址為 CN 實體位址。 [5]
3. MAP-D 收到此 ARP reply 後，查詢代理表以及路由表發現無此資訊（因為不知道目的位址為網狀節點或是行動裝置，所以兩個表格都會查詢），便先將此 ARP reply 傳送至網狀入口（若在沒有主動性樹狀建立的情形該節點會直接啟動需要式路徑選擇）。先傳送至網狀路口是因為不知道此目的位址是否在同一網狀網路裡，若只因不知道目的位址資訊就隨意啟動需要式路徑選擇會造成多餘的封包，因為若目的位址是在網狀路口外，此時啟動需要式路徑選擇便會找不到路徑，因此在查表無資訊時都會先傳送給網狀路口。
4. 網狀路口收到 ARP reply 後，查詢代理表以及路由表也發現沒有目的位址 CN 資訊，於是會啟動需要式路徑選擇，找尋到 CN 的路徑。MAP-S 會代替 CN 回應網狀路口，CN 在其底下。
5. 網狀路口這時便可將封包送往 MAP-S，MAP-S 收到後再將封包傳送給 CN，CN 才真正的收到 ARP reply 封包。此時，MAP-S 因為收到來自 MAP-D 幫 MN 傳送的封包，會將 <MAP-D, MN> 的代理關係記錄在代理表，緊接著查詢路由表後發現沒有到達 MAP-D 的資訊，因此會啟動需要式路徑選擇，將到達 MAP-D 的路徑建立好，存在路由表裡，以供接下來使用，需要式路徑選擇所建立的也是雙向路徑。
6. CN 收到 ARP Reply 知道了 MN 的實體位址，便可以開始傳送封包傳給 CN。這時 MAP-S 收到 CN 要傳送的封包，查詢代理表，便可知道 MN 在 MAP-D 底下，同時查詢路由表，得知到 MAP-D 的路徑，就可以將封包給傳送出去了，途中的網狀節點在剛剛的建立路徑過程中，也都知道到 MAP-D 的路徑，便可依序的將封包傳送到 MAP-D，MAP-D 收到後再傳送給 MN，之後 MN 回傳的封包便可以從反向路徑回去，不需要再送往網狀路口，此即封包的傳送過程。

總結來說，無線網狀網路形成後，網狀路口會啟動主動性樹狀建立，當來源網狀節點要傳送封包時，若路由表或代理表沒有目的地節點資訊時，會先將封包傳送給網狀路口，網狀路口收到封包後視情形決定是否啟動需要式路徑選擇（視路由表有無目的地網狀節點的資訊而定），接著網狀路口便可將封包送往目的網狀節點，收到單點傳送封包

的目的網狀節點會紀錄來源網狀節點與來源行動裝置的對應關係，並查看是否有到來源網狀節點的路徑，若無的話便啟動需要式路徑選擇找尋到來源網狀節點的路徑，之後的封包便可以依循著建好的最佳路徑傳送，如此僅有一開始的少數幾個封包需要經由網狀路口，之後都可以透過最佳路徑傳送到目的地。

2.1.3 IEEE 802.11s 無線網狀網路換手情形及問題

根據上節介紹，無線網狀網路的封包路由是倚賴路由表與代理表的查詢所完成的，即要傳送封包給某一行動裝置時，首先查詢代理表，得知此行動裝置在哪一個網狀擷取點下，接著查詢路由表，將封包路由到此網狀擷取點，目的網狀擷取點收到封包後，再將封包傳送給連接上的目的行動裝置。在這樣的網狀網路環境下，行動裝置的換手行為便會產生一些問題，若要達到無接縫漫遊，則勢必要有一些機制來解決這些問題。

行動裝置換手後，第一個會遇到的問題便是：舊網狀擷取點仍會有一段時間持續收到傳送給行動裝置的封包，但行動裝置卻已經不與舊網狀擷取點連接了，此時節點若沒有任何漫遊機制，封包便會被舊網狀擷取點丟棄 (Drop)。隨著第一點衍生出來的問題便是：來源網狀擷取點代理表的更新，因為若沒有更新的機制，來源網狀擷取點會一直紀錄著行動裝置在舊網狀擷取點底下的資訊，因此會持續的將封包往舊網狀擷取點送，行動裝置在這段時間都無法收到封包，802.11s 並沒有規範行動裝置在無線網狀網路內換手後，要如何去處理這些問題。在沒有任何的機制下，舊網狀擷取點發現其底下並沒有此行動裝置時，便會通知來源網狀擷取點，請它重新詢問行動裝置現在在哪一個網狀擷取點底下，而來源網狀擷取點便會啟動需要式路徑選擇去找尋路徑，接著再將封包送往新網狀擷取點。在這樣的機制下，此行為便會有所謂的封包遺失 (Packet Losses) 以及換手延遲 (Handover Latencies)，802.11s 並沒有規範這段時間的處理機制，這便讓研究者可以有更多的想像空間，提出一些漫遊機制以其達到無接縫漫遊。

2.2 無接縫漫遊機制之交叉點使用情景

無接縫漫遊機制其主要目的是希望讓使用行動裝置的使用者在其移動的過程中，即使行動裝置更換擷取點，使用者也不會感覺有任何斷線的情形，即使用者使用網路的情形不會因為使用者的移動而有任何停頓。要達到無接縫漫遊，一般來說要符合以下兩個條件：一為無封包遺失 (No Packet Loss)，即行動裝置在換到新擷取點時不會有任何的封包遺失掉；另一為延遲時間要短 (Low Latency)，即行動裝置從舊擷取點斷線後到連接上新擷取點重新取得封包的時間，要越短越好。無接縫漫遊機制對使用者來說是很重要的，試想一情形，當您在使用網路電話時移動，網路電話卻突然不通了，這甚至

可能會影響一筆生意的成功與否。也因無接縫漫遊的重要性，便有了許多研究，希望透過一些機制來達到無接縫漫遊，前章有提到，許多研究會利用交叉點對封包做一些處理，以改善換手的效益，常見的不外乎是在換手前利用交叉點做封包預先傳送到新擷取點的動作亦或是換手後利用交叉點提前將封包導向到新擷取點，並利用這些基本概念再去延伸一些優化的機制，底下便就這些交叉點定義、使用情景及其好處做介紹與討論。

2.2.1 交叉點之定義

既然本論文為提出一交叉點機制，我們就來介紹何謂交叉點，首先先來定義一些名詞：

- 路由路徑 (Routing Path)：即由來源節點到目的節點依序所需經過節點的集合。
- 相同路徑 (Common Sequence of Paths)：即有相同來源節點的兩條或多條路徑其路由路徑中相同的節點。
- 交叉點 (Crossover Node - CrN)：即相同路徑的最後一個節點。

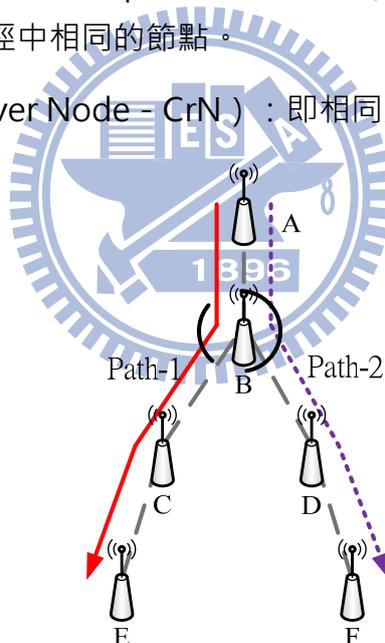


Figure 2-4 交叉點示意圖

Figure 2-4 為交叉點示意圖，由圖中可以看到路由路徑 $Path-1=\{A, B, C, E\}$ 、 $Path-2=\{A, B, D, F\}$ ，而 $Path-1$ 與 $Path-2$ 的相同路徑 $Path-1 \cap Path-2$ 為 $\{A, B\}$ ，因此，根據上述交叉點的定義，交叉點 $CrN=B$ 。

2.2.2 換手前使用交叉點之情景 (Bi-casting)

在一些可以知道新擷取點的情形下，有許多研究會在行動裝置即將換手的時候利用交叉點預先建立新擷取點的路徑 [6]~[9]、預先暫存封包 [10]、或是預先將封包複製一份並傳往新擷取點 [11]，其實這些利用交叉點的目的都是希望能縮短行動裝置連接上新擷取點後還要向舊擷取點拿取封包的這段程序。我們就交叉點將封包複製一份並傳往新擷取點的情形做探討，這種節點複製一份封包並傳往另一擷取點的行為稱為 Bi-casting，其主要精神為，當行動裝置要離開舊擷取點時，會通知交叉點，使交叉點再傳送一份封包到新擷取點，新擷取點收到後先將之暫存，等行動裝置連接上新擷取點後便可以立刻從新擷取點取得暫存資料，這樣的目的是希望減少換手延遲，當行動裝置連接上新擷取點時，便可以直接由新擷取點取得封包，而不需要再去向舊擷取點拿取封包。其延遲僅受換手過程的影響，視行動裝置多快連接上新擷取點而定，而換手過程的延遲改善並非本論文討論的重點。欲使用 Bi-casting 需要預測 (Prediction) 新擷取點，即必須知道行動裝置接下來要連接的擷取點才有辦法使用 Bi-casting 機制，因為是要先將封包送往新的擷取點，而預測新擷取點的機制亦並非本論文所討論。Figure 2-5 為利用交叉點做 Bi-casting 示意圖：

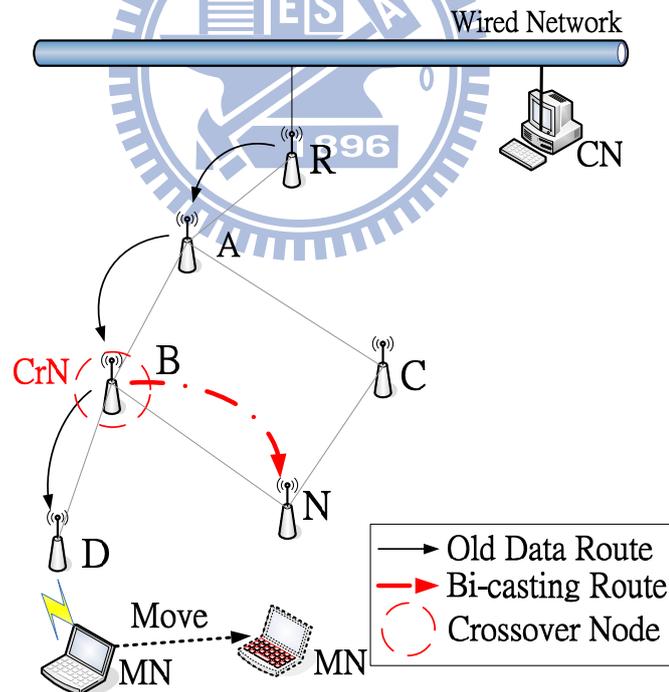


Figure 2-5 利用交叉點 Bi-casting 示意圖

如 Figure 2-5 所示，行動裝置 MN 現在連接上擷取點 D，接下來要連接的擷取點為擷取點 N，CN 與 MN 正在傳送資料，原本的傳輸路徑為 R→A→B→D，當 MN 換手連接上新擷取點 N 後，其傳輸路徑會變為 R→A→B→N，根據上述交叉點定義，此例 B 即為交叉點，因此可以在 MN 要換手的時候通知交叉點 B，請交叉點 B 將封包複製一

份並送往新擷取點 N，擷取點 N 先將封包暫存，如此當 MN 連接上新擷取點 N 後便可以立即拿取封包，而不需要再去向舊擷取點。

而在 MN 連接上新擷取點後，在有 Bi-casting 的情形下，要做位置更新的時候，同樣可以利用交叉點提前將往舊擷取點的封包不再傳送，如此便可以避免頻寬的浪費，因為此時 MN 已經連接上新擷取點，並不需要再將封包送往舊擷取點了。

讀者可能會有疑問，為什麼一定要利用交叉點去做 Bi-casting 的動作，難道其他節點不行嗎？首先要知道的是，這兩條路徑分別為從來源端到舊擷取點以及從來源端到新擷取點的最好路徑，亦即使用路徑演算法所選出的路徑。而若選擇 Bi-casting 的節點為交叉點前任一節點，在交叉點前就會有兩份相同的封包在同一條路徑上傳送，在交叉點才會開始走不同的路徑，這對交叉點前的頻寬是浪費的，因為會有兩份相同的封包走相同的路徑，這其實是不必要的；而若選擇 Bi-casting 的節點為舊路徑交叉點後任一節點，雖然此點仍可以根據路徑演算法選擇最好的路徑送往新擷取點，但封包從來源擷取點→此點→新擷取點的路徑並非是最好的（因為最好的路徑為來源擷取點→交叉點→新擷取點），因此，在來源端收到位置更新前的這段時間，封包會一直有繞路的行為，而這對頻寬亦是一種浪費，若利用交叉點做 Bi-casting 便可以減少上述的問題，因此，大部分研究皆透過換手前利用交叉點做事先與新擷取點的溝通以減少換手延遲。

2.2.3 換手後使用交叉點之情景 (Data-Redirection)

交叉點使用的情景並不僅侷限於換手前的運用，畢竟預測新擷取點這件事並不會是百分之百正確的，因此另有研究便致力於換手後如何增進換手的效益 [8][12]。在行動裝置連接上新擷取點的時候，如前所述，通常需要做位置更新的動作，讓來源端不要再將封包往舊擷取點傳送，不過在來源端收到更新前的這段時間，封包還是會持續往舊擷取點傳送，而若能在交叉點收到便提前將封包往新擷取點傳送 (Redirection)，便可以減少封包遺失，當然，即使是舊擷取點有暫存再轉送 (Buffer and Forward) 的機制，利用交叉點做封包導向的動作同樣可以減少封包的繞路。Figure 2-6 為利用交叉點做封包導向示意圖，在交叉點收到更新封包後便直接將封包導向新擷取點，此時交叉點接著繼續將位置更新封包往來源端傳送，來源端收到後才真的將封包傳向新擷取點。

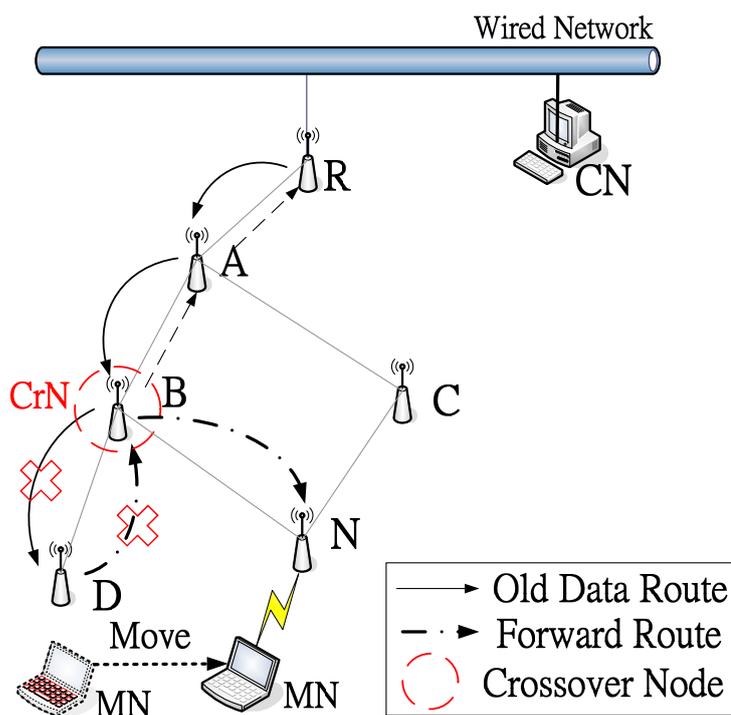


Figure 2-6 利用交叉點做封包導向示意圖

所謂的暫存再轉送便是當行動裝置從舊擷取點離開後，舊擷取點會暫時將要送給行動裝置的封包暫存住 (Buffer)，而當行動裝置連接上新擷取點後，由新擷取點去向舊擷取點要求暫存的資料，舊擷取點便可將暫存的資料轉送給 (Forward) 新擷取點，這時行動裝置便可以拿到這些資料。

2.3 單元總結

本章介紹了無線網狀網路的基本架構及其傳送封包的特性，說明了在換手時所會遭遇到的一些問題，並介紹交叉點在無接縫漫遊機制其扮演的角色為何。由過去研究可知，交叉點確實可以縮短換手延遲、減少頻寬的浪費與減少封包遺失，總結來說，善用交叉點確實可以有效的增加換手的效益，前述我們也說明了，相較於交叉點前、後的節點，交叉點是對頻寬影響最小的節點，所以使用交叉點做上述之行為，對於增進換手的效益，尤其是無線網狀網路的環境來說，是很好的方式，因此我們希望在無線網狀網路也可以使用交叉點來增進換手的效益，不過在無線網狀網路環境下，要如何將交叉點找尋出來呢？先前的研究並不適用於無線網狀網路環境，同時亦有其缺點，而這便是本篇論文要討論的重點，詳細內容會在下一章介紹與討論。

第三章 相關研究

3.1 找尋交叉點所遭遇之困難

前章我們介紹了交叉點在無接縫漫遊機制中通常扮演著很重要的角色，有許多研究皆透過交叉點的利用以增進換手的效益，在無線網狀網路中，網狀節點間是透過無線連結的方式連接在一起，換手所受到的延遲影響會更加顯著，尤其當節點間傳輸的 Hop 數增長時，因此，若能在無線網狀網路的環境善用交叉點，勢必也可以有效的減少換手時所會遇到的問題。也因為如此，如何在無線網狀網路找到正確交叉點便成了一件很重要的議題。

依據交叉點的定義：自來源端到新、舊擷取點路徑的最後一個相同的節點。因此，若可以明確的知道兩條路徑所需經過的節點，便可以輕易找出交叉點，但是，網路的路由並不一定得知道完整的路徑才可以將封包傳送到目的。在路由協定 (Routing Protocol) 的分類中，可以依運作特性而區分出所謂的連結狀態路由 (Link State) 以及距離向量路由 (Distance Vector)。連結狀態路由協定是這樣運作的，首先，每個節點會蒐集自己周遭鄰居的資訊 (Link State Information)，其次，節點會與同區域的所有其它節點交換彼此之前蒐集到的連結狀態資訊，這個階段完成後，每個節點都會得到所有其它節點以及本身的連結狀態資訊，最後，各個節點會根據前述資訊，以最短路徑優先 (Shortest Path First) 演算法算出以自己為基準點至所有其他節點的 Tree (Topological Tree)，然後，再以此 Tree 為基礎，建構它自己的路由表；而距離向量路由呢？它的建構路由表之依據，完全依賴來自鄰居所提供的資訊 (包括自己知道的資訊)，而鄰居提供的資訊，又則是鄰居的鄰居提供... [13]。簡單來說，這兩種分類其主要區別為連結狀態路由會擁有來源節點要傳送封包給目的節點時所要經過的所有節點資訊，也就是完整路徑；而距離向量路由卻只會有到目的節點要傳送封包給下一個節點的資訊。根據前章介紹的網狀網路路由封包機制，其路由是以後者的方式運行，由於沒有完整的路徑資訊，無法直接比對兩條路徑以找尋分點，交叉點是不容易找尋到的。

在過去的研究裡，雖然也有研究是在以距離向量路由的資訊下去尋找交叉點，但這些研究其找尋交叉點的方式並不適用於無線網狀網路，因為這些研究係是透過其網路的特性去將交叉點找尋出來，但無線網狀網路的環境裡並無這樣的特性，因此便無法以這樣的方式去尋找交叉點，另外，某些研究其找尋交叉點所需花費的訊息量過高亦是這些機制的缺點，因為這些機制都是針對每個行動裝置去作找尋的動作，這些詳細內容會在

下一節做介紹。

而這些研究所存在的另一主要問題為，根據定義，交叉點是自一節點出發到任兩節點路徑的最後一個共同點，而任一節點都會有以它自己到所有節點的路徑，任一節點都會以它自己生成一個樹狀拓樸，而這樣的架構就會有一個現象，任兩節點間的交叉點會隨著出發節點的不同而有所改變。以往研究其交叉點的找尋都是根據閘道器 (Gateway - GW) 出發所長出的樹狀拓樸去找尋交叉點 [6]~[12]，因此，其結果為任兩節點間的交叉點就只會有一個點，如 Figure 3-1 所示：擷取點 D 與擷取點 N 的交叉點僅會有 A 這一點，因為先前研究通常僅是考慮 Correspondent Node (CN) 在 GW 外，MN 進行換手的情形，因此，不論 CN 在網路外哪個地方，封包都會由 GW 進入，因此交叉點便僅需由 GW 出發去找尋。

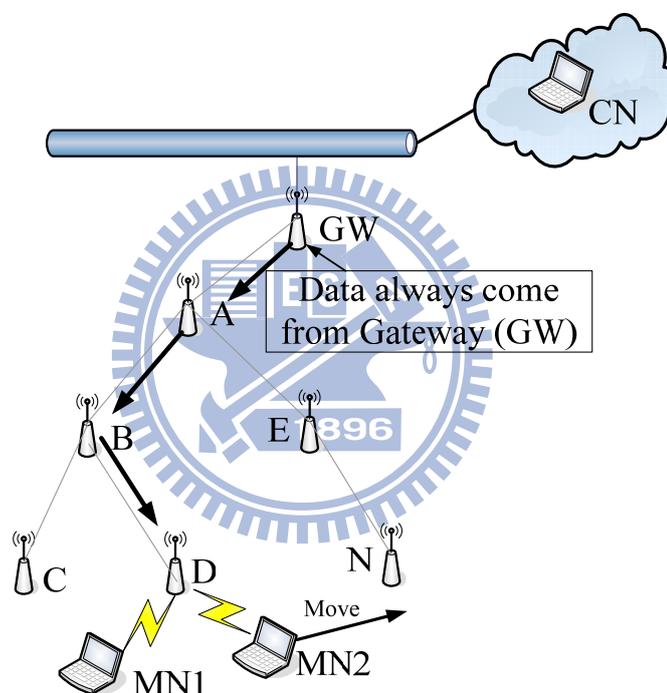


Figure 3-1 以 GW 為出發點所找尋之交叉點示意圖

由 GW 找尋交叉點的機制或許適用於部份情況，倘若現在考慮到，CN 與 MN 在同一個網域底下，MN 進行換手，那又是會有什麼現象呢？其結果是任兩節點間的交叉點就會受 CN 位置不同而有所改變。舉 Figure 3-2 例子來看：

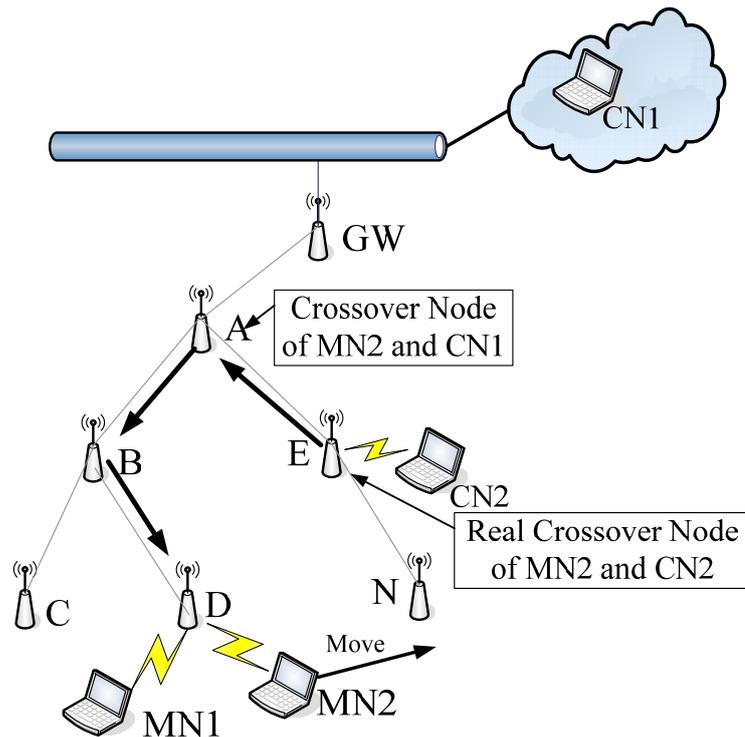


Figure 3-2 CN 與 MN 在同一個網路底下其交叉點之示意圖

MN2 行動裝置與 CN2 行動裝置正在溝通，當 MN2 由舊擷取點 D 換到新擷取點 N 時，交叉點便不再是以 GW 出發所找尋之結果，圖中可以看到 CN2 到舊擷取點 D 其封包所走的路徑為實線，而 CN2 到新擷取點 N 其封包所走的路徑為 E→N，此時交叉點的找尋必須是以 CN2 所連接的擷取點出發去找尋其結果才會是正確的交叉點，在這裡舊擷取點 D 與新擷取點 N 的交叉點即為 CN2 連上的擷取點。因此，兩兩節點間的交叉點會受到 CN 的位置不同而不同。若 CN 與 MN 不在同網域內，那交叉點當然都是一樣的，因為都是以 GW 出發所找尋之結果；但若 CN 與 MN 在同一網域內，就必須是以 CN 所連上的擷取點出發所找尋之結果，而這是先前研究中沒有考慮到的。

下一節會跟讀者介紹先前研究中找尋交叉點之機制及其優缺點，以其利用交叉點在換手前、後做一些機制增進換手的效益。如前所述，交叉點的用處是可以很多的，其主要都是為了行動裝置在從舊的擷取點移動到新的擷取點後，利用交叉點加速與優化行動裝置接收封包情形，以達到無接縫漫遊的目標。

3.2 找尋交叉點之機制

本節將會介紹幾種以往研究中找尋交叉點的方式，我們將這些研究分為幾類，分別是：1.利用路由表的資訊找尋，此法將會是本論文比較的對象，因其符合無線網狀網路的路由環境。2.利用 Routing cache 的資訊找尋，此法較適用於 Cellular IP 這種會有 Binding cache 機制的網路環境。3.利用知道所有路徑的資訊找尋，此法較適用於

Hierarchical Mobile IP 這類會知道整條路徑資訊的網路環境。4.利用 Geopaging routing table 的資訊找尋，此法必須節點有支援 Geopaging 才適用。上述這些方法都有其優缺點，我們會詳細介紹。雖然這些研究都可以找到以 GW 出發所建立的樹狀拓樸之任兩節點的交叉點，但對於 CN 與 MN 在同一網域底下，MN 換手的情形均沒有探討到，在這種情況下，上述研究所找尋之交叉點便是錯誤的，底下將介紹這些方法的運作方式。

3.2.1 Use the information of routing table

D. H. Cuong [6][7]提出了 Crossover Router Pre-Discovery (CRPD) 找尋交叉點的機制，其主要精神為：利用 Fast Mobile IPv6 (FMIPv6) [14]的一些訊息機制，在這些訊息中加入作者找尋交叉點的方法，使其找尋交叉點的時機與節點間的交叉點資訊都可以透過這些訊息交換的過程中得以完成，Figure 3-3 為 CRPD 找尋交叉點的訊息示意圖：Fast Neighbor Advertisement (FNA)、Binding Update (BU) 以及 Binding Acknowledgement (BA) 為 Mobile IPv6 中原本就有的訊息交換機制：FNA 是用來通知連接上的擷取點已進入該網域；BU 即為通知 CN 來源行動裝置 MN 行動裝置現在的位置為何；BA 即 CN 所回應的更新確認訊息。而 CRPD_Act 訊息為本研究作者所新增之訊息，BA+CRD 的 CRD 則是本研究作者加在 BA 訊息裡的新欄位 (Option)。

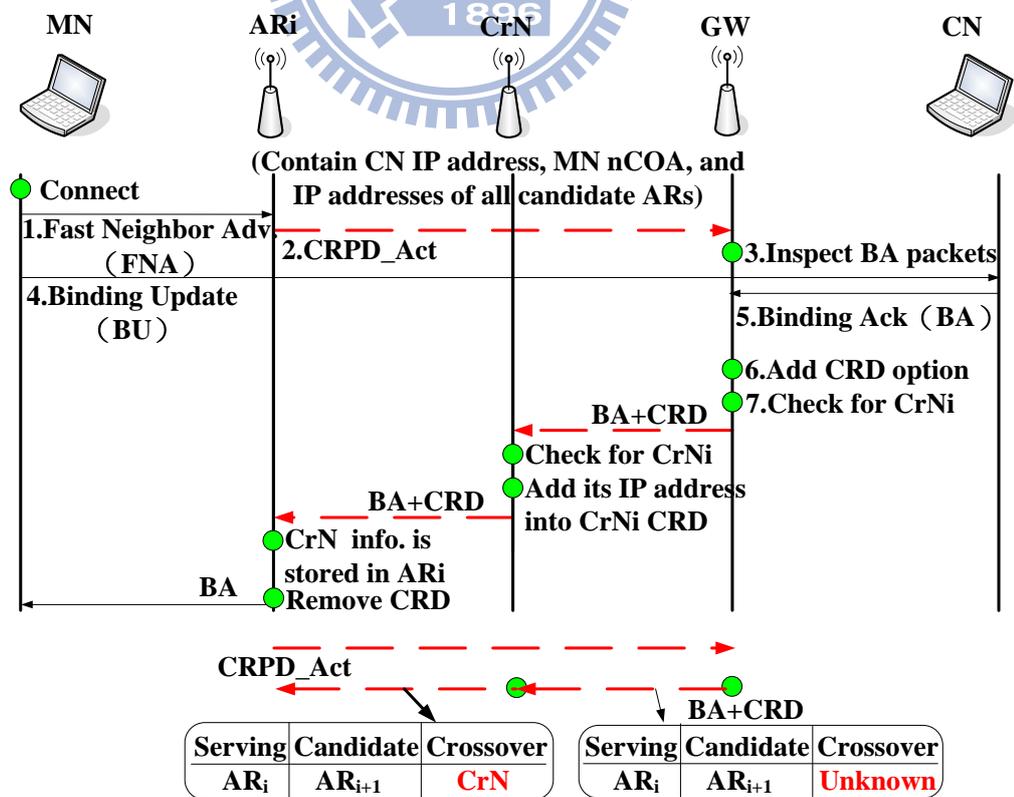


Figure 3-3 CRPD 訊息示意圖

接下來介紹此找尋交叉點之機制，當行動裝置 MN 連接上擷取點 AR_i ：

在 FMIPv6 架構下 MN 會發出 FNA 告知 AR_i 已進入該網域。此時 AR_i 便會啟動 CRPD 機制， AR_i 會發出 CRPD_Act，此訊息會挾帶著來源端 CN 的 IP 位址、MN 的 nCOA (此即為行動裝置在此網路所使用的 IP 位址) 以及 MN 換手後有可能連接的擷取點 - 候選擷取點 (Candidate ARs) 的 IP 位址。至於要如何得知候選擷取點呢？作者並沒有探討此問題，僅說明可以用手動的方式設定或者其它預測的方式得知。以下為 CRPD 訊息傳遞流程：

1. MN 會發出 FNA 告知 AR_i 已進入該網域
2. 此 CRPD_Act 會被沿著往 GW 的路徑傳送。
3. 當 GW 收到此訊息後，便開始監控由 CN 傳送過來的 BA 訊息。
4. 接著 MN 會再送出 BU 封包給 CN 更新其位置。
5. CN 回應 BA。
6. GW 收到 CN 傳送的 BA 時，由於已經啟動監控 CN 的 BA 訊息，因此會在 BA 訊息的欄位加入額外的資訊 - 候選擷取點 (BA+CRD)。
7. GW 與中間收到此訊息的節點代表著要找尋 AR_i 與這些候選擷取點 AR_{i+1} 的交叉點，其找尋的方式為：節點會查詢路由表，分別比對到 AR_i 與到所有 AR_{i+1} (所有候選擷取點) 的下一個節點是否相同。若相同的話，表示還不知道交叉點為何；而若是不同的話，則表示該節點即為交叉點，此時便會紀錄 AR_i 與比對到相同下一個節點的 AR_{i+1} 的交叉點為該節點，並標示成已找尋，其餘與 AR_i 有相同下一個節點的 AR_{i+1} 仍標示成未找尋。接著，便將此訊息繼續往 AR_i 的下一個節點傳送，下一個節點收到後，重複上述動作，比對到 AR_i 與到其餘仍標示未找尋的 AR_{i+1} 的下一個節點是否相同，有相同的下一個節點仍然是未找尋，不同的則該節點本身就是交叉點，將資訊填完後再將訊息往到 AR_i 的下一個節點傳送。

以此類推，當 AR_i 收到後，便將附在 BA 後的 CRD (即 AR_i 與所有 AR_{i+1} 的交叉點為何的資訊) 儲存在 AR_i 上，接著再將 BA 傳給 MN。如此， AR_i 便有了與候選擷取點的交叉點為何的這筆資訊了。

而當要利用交叉點做事時， AR_i 直接查詢交叉點表格得知交叉點為何後便可以將欲傳送的訊息傳送給交叉點了，其交叉點表格傳遞的內容如 Figure 3-3 的下部份所示。

此方法固然可以找到交叉點，不過卻有些缺點，若 MN 與 CN 在同一個網域底下，

如同前述，這方法僅找尋了由 GW 所建立之樹狀拓樸的節點間交叉點，因此所找到的交叉點會是錯誤的；二為找尋訊息觸發的太過頻繁，此機制每當行動裝置連接上擷取點與換手都必須觸發找尋交叉點的動作，若另一行動裝置的候選擷取點是相同的話，所找到的交叉點也會是相同的，如此便會多浪費網路頻寬去取得相同的資訊，而當行動裝置一多，整個網路便會充斥著這些找尋交叉點的訊息；三為無法適用於所有欲利用交叉點的情景，此方法必須把連接上的擷取點與所有候選擷取點的交叉點資訊一併往 AR_i 傳送，由 AR_i 紀錄著這些資訊，因此此法僅適用於換手前利用交叉點之情景，若換手後想要利用交叉點，還必須請 GW 重新做找尋動作，因為僅有舊擷取點有此交叉點表格資訊。

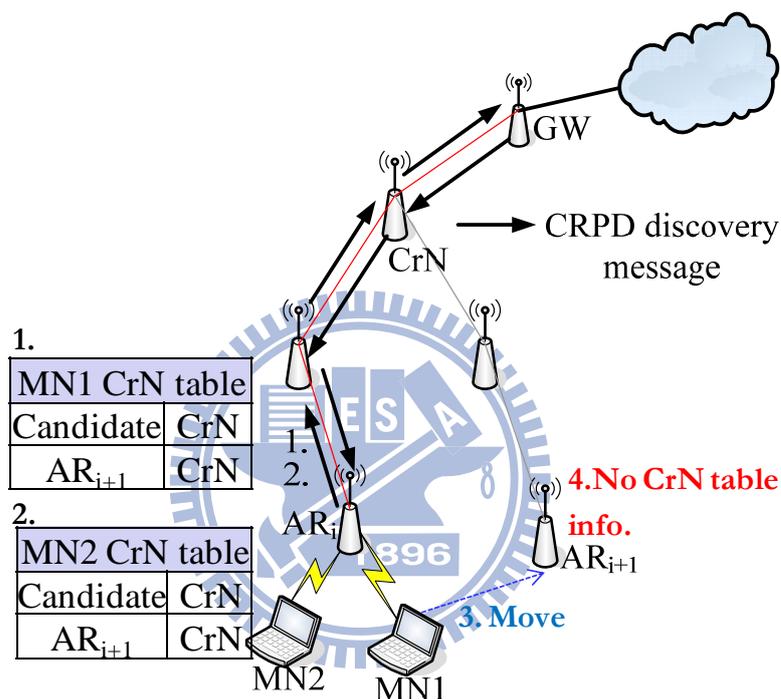


Figure 3-4 CRPD 重複性找尋交叉點示意圖

Figure 3-4 為 CRPD 重複性找尋交叉點示意圖，圖中分別為不同的行動裝置 MN1 與 MN2 連上相同的擷取點 AR_i ，其候選擷取點皆為 AR_{i+1} ，在這樣的情形下，其交叉點以 GW 出發找尋會是相同的，但在 CRPR 的機制卻還需要做一次交叉點找尋的動作，如此便造成多餘不必要的找尋，當行動裝置多與換手次數頻繁時，勢必造成網路內部充斥著這些找尋訊息。

而我們亦可以由 Figure 3-4 看到交叉點表格儲存在舊擷取點上的，本機制是利用交叉點預先建立連線到新擷取點，因此並無考慮到換手後利用交叉點的情景，當行動裝置 MN1 換手到新擷取點 AR_{i+1} 後，若欲利用交叉點，此時 AR_{i+1} 是沒有交叉點的資訊的，因為必須再重新往 GW 做找尋的動作才能得知交叉點為何，因此機制對於換手後欲利用交叉點的情景是不適用的。

3.2.2 Use the information of Binding Cache

在 A. T. Campbell [8]與 J. Y. Hu 等 [12]裡利用 Cellular IP [15]裡的 Binding Cache 特性得以找到新、舊擷取點間的交叉點，並利用交叉點對封包做重新導向的動作，我們稱此找尋機制為 Binding Cache-based (BCB) Mechanism for Crossover Node Discovery。在 Cellular IP 網路內其路由是利用這項資訊得以將封包正確的傳送到目的地的。當 MN 進入 Cellular IP 網路內時，便會向其所連上的基地台(Base Station - BS) 發出一個 Binding update 的訊息，此 Binding Update 會被從 BS 沿著往 GW 的路徑更新，沿路上的節點收到此 Binding Update 都會更新其 Cache 紀錄，當 GW 收到更新後，之後有封包從 GW 來便可依據 Cache 的資訊將封包正確的傳送到 AR，如 Figure 3-5 所示：

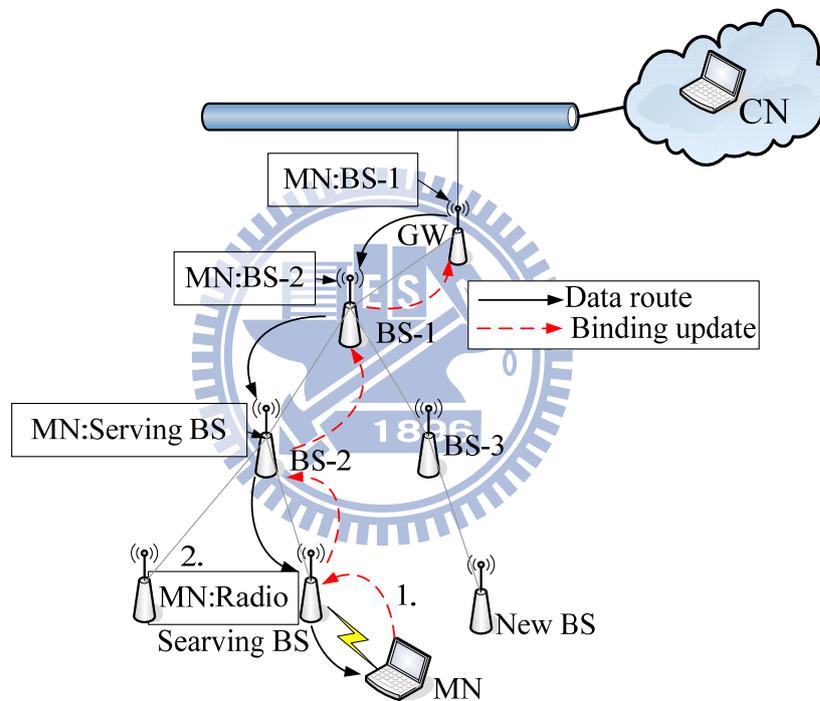


Figure 3-5 Route update 更新流程示意圖

利用此 Cache 機制的特性便可以很輕易的找到交叉點，其方法是，當行動裝置連接上新擷取點時後，同樣會送一個 BU 給新擷取點，而候選擷取點收到此訊息後，同樣的沿著往 GW 的路徑發送此 BU，當沿路上的 BS 收到此訊息時，若無此 MN 的 Cache 記錄，則繼續將訊息往 GW 的路徑傳送；若先前有此 MN 的 Cache 紀錄，且是從不同的 BS 傳送過來的，則此 BS 本身便是交叉點了，如 Figure 3-6 所示：當 BS-1 收到此訊息時，它原本有 MN 的 Cache 資訊（往 BS-2），在此時必須修改成 BS-3，如此，BS-1 就會知道自己是 Serving BS 與 New BS 間的交叉點了，而當知道自己為交叉點後，BS-1 便直接將封包往新 BS 傳送，做導向的動作，接著再延著往舊 BS 的路徑將 Cache 清除。

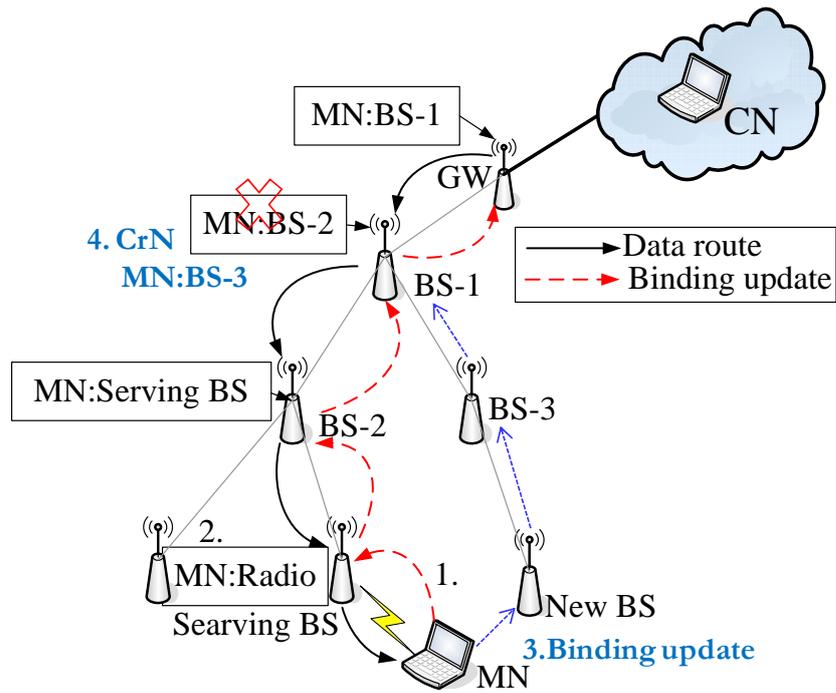


Figure 3-6 行動裝置換手後示意圖

此方法同樣僅能找到以 GW 出發的交叉點，對於 MN 與 CN 在同一個網域底下還是不適用，而其主要缺點是，必須從新 BS 出發才能找尋到交叉點，若從舊 BS 出發是沒有辦法找到交叉點的，因為舊 BS 到 GW 沿路都有 Cache 行動裝置的資訊，無法利用此方法找到交叉點。這對換手後欲利用交叉點的一些機制可以使用，不過若要在換手前利用交叉點做 Bi-casting，此法是不適用的，因為換手前行動裝置連接上的是舊 BS。不過，在本篇研究有提到可以在換手前先發一個 Semisoft 訊息給新 BS，接著再馬上連回舊 BS，新 BS 依據上述依然可以找到交叉點，並讓交叉點再傳送一份封包給新 BS，此法同樣也可以達到預期的結果，不過，MN 必須預先連上新 BS 就有點不實用。

然而，撇開此缺點不談，此機制依然是完全不適用於無線網狀網路的環境，其主要原因為，無線網狀網路並不會只有『GW 到行動裝置連接上的網狀擷取點這條路徑上的網狀節點知道行動裝置的資訊』，無線網狀網路內的任何網狀節點都有可能要與行動裝置溝通因而知道如何到達行動裝置，亦即，任何網狀節點皆有可能會有行動裝置的資訊，我們舉個本機制在網狀網路環境下找到錯誤分叉點的例子，如 Figure 3-7 所示：

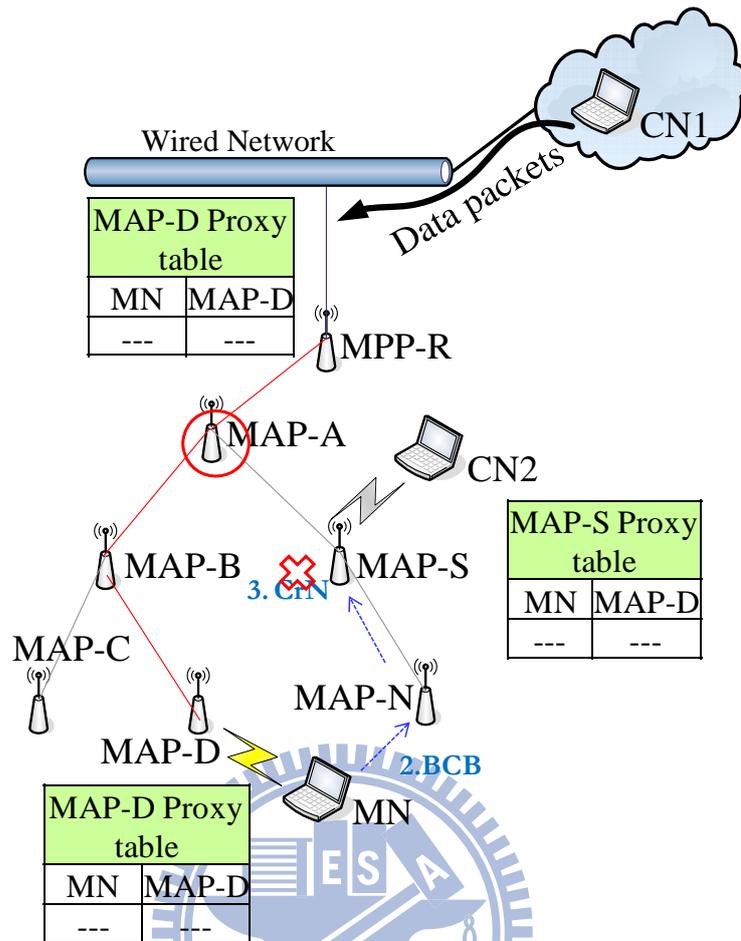


Figure 3-7 無線網狀網路利用 Binding Cache 找尋交叉點機制示意圖

MN 行動裝置現在連接上 MAP-D 網狀擷取點，CN2 行動裝置連接上 MAP-S 並與 MN 溝通，過一段時間後，CN2 便離開了，但此時 MAP-S 仍有 MN 在 MAP-D 底下的資訊，而此時 CN1 與 MN 溝通。如前章所述，路由的方式是先查詢代理表得知行動裝置在哪一個網狀擷取點下，再將封包依據路由表的查詢路由到此網狀擷取點，中間網狀節點只要查詢路由表即可。不同於 Cellular IP，在無線網狀網路環境下並非是 MPP-R、MAP-A、MAP-B 會有如何到達 MAP-D 的資訊，其餘的網狀擷取點如 MAP-S 都有可能知道 MN 連接上 MAP-D，並且有如何 MAP-D 的資訊，當這些網狀節點有和行動裝置溝通過，這種情況就有可能發生，而這樣便無法以 Binding Cache Mechanism 資訊找到交叉點，我們可以由圖中看到，若利用 Binding Cache Mechanism 資訊的找法，MAP-S 會認為自己是交叉點，而這是錯的，MAP-A 才是真正 MAP-D 與 MAP-N 的交叉點。因此，利用 Binding Cache Mechanism 資訊其找尋交叉點的機制是不適用於無線網狀網路的，即使在 MN 與 CN 在不同網域的情況，在無線網狀網路的環境亦無法找到正確的交叉點。

3.2.3 Use the information of Hierarchical model

在 D. Tandjaoui [10]等是利用 Hierarchical Mobile IP [16]的特性去找尋交叉點，本篇研究主要精神是希望在行動裝置要換手前，利用交叉點暫存封包，等到行動裝置連接上新擷取點時便可直接向交叉點拿取封包，而不需要到再向舊擷取點拿取，目的是希望減少封包到達新擷取點的順序錯誤(Mis-ordered)與換手延遲，所謂的 Mis-ordered 即換手後新擷取點通知交叉點將封包傳往新擷取點的封包會比舊擷取點暫存再轉送的封包快到達新擷取點，因為在樹狀結構下，更新的封包勢必會先經過交叉點接著才到達舊擷取點，如此會導致來源節點較晚傳送的封包反而比較快到達新擷取點，因此換手前利用交叉點暫存封包便可以避免此問題，在行動裝置換手後便不需要再向舊節取點取得封包，僅需要跟交叉點要求暫存的封包即可。

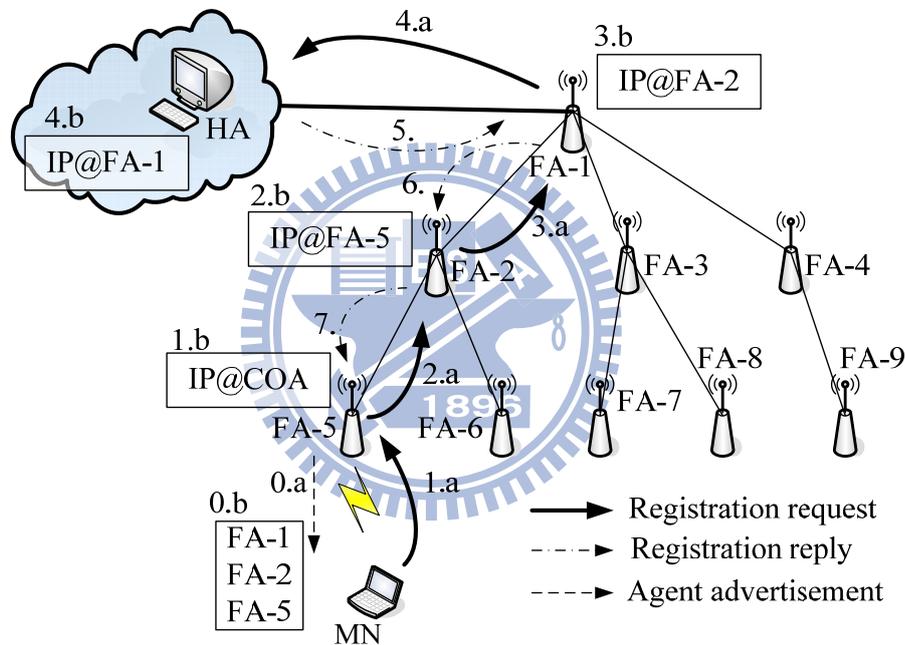


Figure 3-8 Hierarchical Mobile IP 架構圖

Hierarchical Mobile IP 的架構如 Figure 3-8 所示：當行動裝置 MN 進入 Hierarchical 網路時，會收到 Foreign Agent (FA，這裡為 FA-5) 所發出的 Agent advertisement，此訊息會帶著從 GW (這裡為 FA-1) 到此 FA-5 所會經過的所有路徑 (這裡為 FA-1→FA-2→FA-5)。而當 MN 連上後，會發出註冊要求 (Registration request)，通知其 Home Agent (HA) 更新行動裝置的位置，要將封包傳送往行動裝置現在所在的網路，HA 更新後亦會回註冊回應 (Registration Reply)，之後封包便可以依圖中的紀錄依序丟給行動裝置了。

在 Hierarchical Mobile IP 中是利用交叉點做 Regional Registration，如此當行動

裝置換到新擷取點後便不用每次都要向 HA 註冊，僅需註冊到交叉點更新其行動裝置的記錄資訊，因為交叉點前的資訊都是一樣的。而此架構下找尋交叉點方式很簡單，因其擁有完整的路徑資訊，如 Figure 3-9 所示：當 MN 連接上新擷取點時(此為 FA-6)，同樣的，也會收到 FA-6 所發出的 Agent Advertisement，此訊息同樣會帶著 GW 到此 FA-6 會經過的所有路徑(這裡為 FA-1→FA-2→FA-6)，MN 在與前一筆資料比對後(FA-1→FA-2→FA-5 與 FA-1→FA-2→FA-6)便可以很清楚得知交叉點為 FA-2，此時便緊要傳送 Regional Registration 訊息至 FA-2 即可，之後交叉點 FA-2 便會將封包往新擷取點傳送，而舊擷取點暫存的資料接著再去向其拿取即可，而若僅利用交叉點做註冊這樣的方式便會有上述的封包順序錯誤(Mis-ordered)與換手延遲的問題。

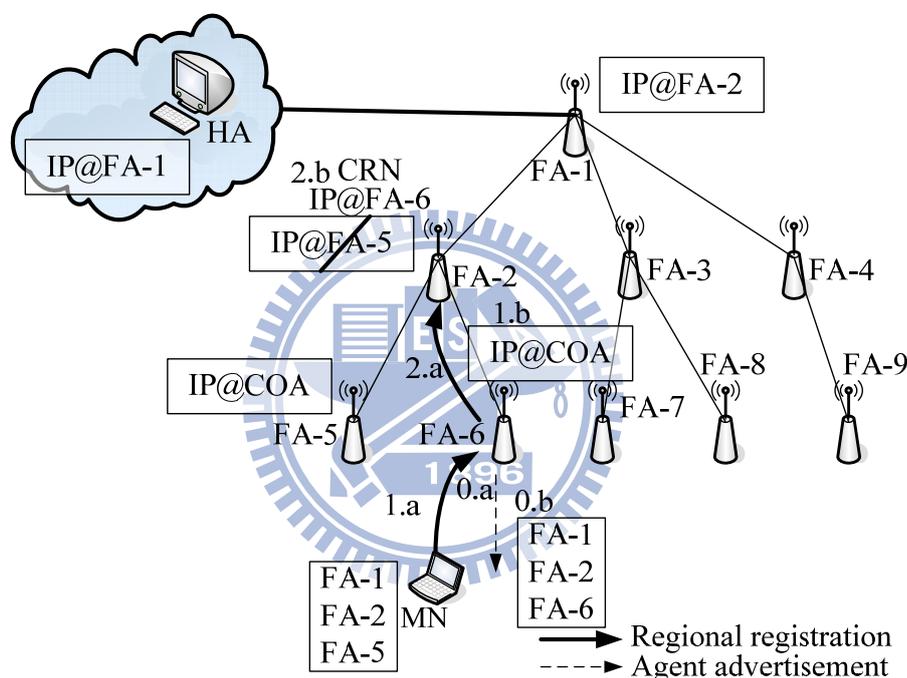


Figure 3-9 Hierarchical Mobile IP Regional Registration 示意圖

而此篇研究是希望利用交叉點做暫存封包的動作，找尋交叉點的方式同上述，皆是透過完整的路徑資訊找尋出交叉點的，其主要精神如下：當 MN 移動到新擷取點的範圍，聽到新擷取點發出 Agent Advertisement 並決定要換手過去時，會請舊擷取點發一個通知訊息給新擷取點並且帶著舊擷取點其 GW 到舊擷取點的所有路徑資訊 (FA-1→FA-2→FA-5)，新擷取點收到後同樣會有 GW 到新擷取點的所有路徑資訊 (FA-1→FA-2→FA-6)，新擷取點再依據上述之方式比對後便可以很輕易的知道交叉點為何，接著新擷取點便可以通知交叉點請它暫存要送給 MN 的封包，而當行動裝置連接上新擷取點時，依據前述會找出交叉點並做 Regional Registration 的動作，當交叉點收到此 Regional Registration 後，將暫存的封包送往新擷取點，新擷取點收到後再傳送給 MN 即可。此找尋交叉點的方法同樣是完全不適用於無線網狀網路的，因為網狀

節點不會有完整的路徑資訊，不過由此篇研究我們也可以知道交叉點確實在漫遊機制中扮演著很重要的角色。

3.2.4 Use the information of Geopaging routing table

在 R. V. Ferré [9]等是利用 Geopaging routing table [17]的資訊去尋找交叉點，利用交叉點在換手前與新擷取點做一些訊息交換的動作，取代了原本需由舊擷取點與新擷取點的訊息交換，以達到無接縫漫遊。此研究主要是基於 FMIPv6 [14]的架構下做探討，其方法名稱為 Optimal Crossover Router Discovery (OCRD)。Geopaging 為一種 IPv6 的群播協定 (Multicast Protocol)，其目的是希望有效率地在 IPv6 架構 (IPv6-based) 的 Cellular network 裡傳送 paging 資訊。Geopaging 其各個 Cell 的位址為根據 Cell 實際的地理位置去配置出來的，而 Geopaging Routing Table 則是根據所配置出來的位址來建構，Geopaging Routing Table 其建制過程本論文不會介紹，僅會探討此篇研究是如何利用 Geopaging Routing Table 特性找出交叉點。OCRD 的機制如 Figure 3-10 所示：

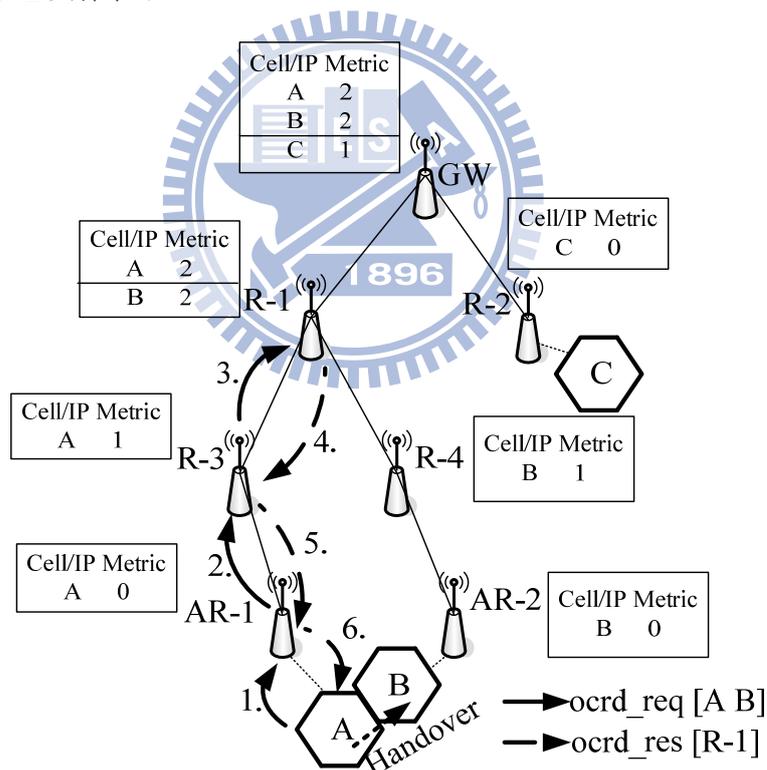


Figure 3-10 OCRD 找尋交叉點示意圖

圖中表格為根據 Geopaging 之位址配置所建構的路由表，Cell/IP 代表此節點底下有的 Cell。以 GW 舉例，其底下有 A、B、C 三個 Cell，A 與 B 在同一個分區內，表示要到 A 與 B 需往同一個 GW 的介面傳送，要到 C 則往另一個 GW 的介面傳送，Metric 則表示還需要經過幾個 Router。當 AR-1 收到行動裝置發出的 Router Solicitation for

Proxy (Rtsolpr) 後 (此訊息會帶有行動裝置所掃描到的擷取點，目的是向 AR-1 取得這些擷取點的資訊，為 FMIPv6 機制所制定的訊息)，AR-1 便會觸發找尋交叉點機制，即 orcd_req[A B]，A 為現在的 Cell，B 為即將要換手過去的 Cell，此訊息會被沿著往 GW 方向的路徑傳送，沿途上的 Router 都會依據 Geopaging routing table 檢查自己是否為 A 與 B 的交叉點，當 R-1 收到後，檢查 A 與 B 在不同的分區即可知道自己為交叉點，此時便回應 orcd_res[R-1]，AR-1 收到後便會將交叉點 R-1 挾帶在 Proxy Router Advertisement (PrRtAdv) 訊息裡傳送給行動裝置 (此訊息為回應 Rtsolpr，即所找到的候選擷取點的資訊)，行動裝置接著會發出 FBU (此訊息告知 AR-1 其即將連上的擷取點為何) 並挾帶交叉點的 IP，AR-1 收到後便會將此訊息轉送給交叉點，交叉點此時便可代替原本 FMIPv6 中舊擷取點的角色，與新擷取點做接下來的訊息交換 (交換 HI 與 Hack 訊息)，之後交叉點便可以直接將封包傳給新擷取點而不需要再透過舊擷取點了。上述之 Rtsolpr、PrRtAdv、FBU、HI 與 Hack 皆為 FMIPv6 所制定的訊息，與交叉點無直接關係，本論文不多介紹。

此方法雖然可以透過舊擷取點找到交叉點，但缺點為整個網路都需要支援 Geopaging 此協定，透過此協定建立的路由表才可以找到交叉點，但在一般的網路環境並不會有節點支援這些協定的，尤其是無線網狀網路，再者，要多實做此一機制對網路來說亦是額外的負擔，因此，此法亦是不適用於無線網狀網路的。

3.2.5 找尋交叉點機制之比較

Table 3-1 先前研究於非 802.11s 與 802.11s 其 inter- and intra-domain session 時比較表

Env. Mech.	Non-802.11s Inter-domain session	Non-802.11s Intra-domain session	802.11s Inter-domain session	802.11s Intra-domain session
CRPD	Yes	No	Yes	No
Binding Cache	Yes	No	No	No
Hierarchical Mobile IP	Yes	No	No	No
Geopaging	Yes	No	No	No

Table 3-1 將先前研究的找尋交叉點機制做一個比較，主要看這些機制於 MN 與 CN 在同一網域與不同網域底下是否適用，以及這些找尋機制可否運作於 802.11s 網路環境。上列 (Environment) 為在非 802.11s 網路環境與 802.11s 網路環境其 MN 與 CN 在同一網域與不同網域的情形，左列 (Mechanism) 為先前研究介紹的四種找尋交叉點機制。由表格中可以看到，這些機制皆是針對 MN 與 CN 在不同網域下設計的，因此對於 MN 與 CN 在同一網域下的情形皆會找錯交叉點，至於在 802.11s 環境下時，由於除了 CRPD 外，其它方法皆是利用該網路的特性去找尋交叉點，因此這些方法即使在 MN 與 CN 在不同網域下皆無法找到交叉點。

3.3 單元總結

本章介紹了多種找尋交叉點的機制，並探討其優缺點，其大多都不適用於無線網狀網路的環境，尤其是在 CN 與 MN 在同一網域，MN 換手的情況，這些方法所找到的交叉點會是錯誤的。3.2.1 節所介紹的 CRPD 方法對於無線網狀網路 MN 與 CN 在不同網域的情況或許可以適用，但仍有其缺點，在 3.2.1 節都已經有提到。由這些研究也可以得知交叉點在漫遊機制中的確有許多利用的方式，不論是換手前或換手後都可以利用交叉點達到減少延遲、不浪費頻寬的效果、增加換手的效益，因此，若能在無線網狀網路的環境也能利用交叉點，勢必能增進換手的效益，本論文係針對這些研究的缺點提出一個適用於無線網狀網路環境的找尋交叉點機制，以其可以利用最少量的訊息量找到最多的資訊，並且在任何的換手情況底下都可以找到正確交叉點，利用正確的交叉點去增進換手的效益。

第四章 Per-proxy-oriented Partition-based (POP) Crossover Node Discovery Mechanism

4.1 簡介

本論文提出一探索交叉點之機制，我們稱為 Per-proxy-oriented Partition-based discovery (POP)，透過此機制可以針對每一網狀擷取點在一次性的探索交叉點過程中將以此出發點網狀擷取點所建立的樹狀拓樸其任意兩網狀擷取點的交叉點資訊建立出來，此機制有以下幾個特點：

- 針對每個網狀擷取點 (Per-proxy-oriented) 之交叉點找尋
- 每個網狀結點一次性 (One-time) 分群 (Partition-based) 之交叉點找尋
- 交叉點資訊分散式化

前章提到當 MN 與 CN 在同一網域內時，交叉點會隨著 CN 的位置不同而有所改變，必須是以 CN 所連接上的擷取點去找尋交叉點而非 GW。有鑑於此，POP 針對每個網狀擷取點(Per-proxy-Oriented)去找尋交叉點，建立各個網狀擷取點的交叉點資訊，如此，在任何行動裝置換手的情形時，所欲利用的便會是最正確的交叉點；所謂的一次性 (One-time) 分群 (Partition-based) 之交叉點找尋則是本機制透過分群的概念，在一次的探索程序中便將以此出發點至所有其它節點建立之樹狀拓樸其任兩節點間的交叉點資訊找尋完成，利用最少的訊息建立最多的交叉點資訊為本機制之特色。無線網狀網路的環境下其網狀節點通常不具有行動能力，我們可以假定封包的路徑不太改變，而透過這種一次性探索的動作便可以減少不必要的重複性找尋，上一章有提過，先前研究通常是利用行動裝置連接上擷取點時才去找尋交叉點，當不同的行動裝置連上相同的擷取點，若與先前行動裝置有相同的來源節點與候選擷取點時，其交叉點便會相同的，此時若還要再找尋一次，則會造成不必要重複性找尋，對頻寬亦是種浪費；此外，本機制的交叉點資訊是分散在找尋交叉點路徑的網狀節點上，這樣的方式其好處是當行動裝置連接上新擷取點後，便可以立即利用交叉點，亦即，不管換手前、後手後欲利用交叉點都可以立即使用，而不需要再找尋一次。不同於相關研究章節中介紹的 CRPD 機制，其交叉點表格資訊必須沿路傳遞到行動裝置連接上的擷取點，所傳遞的資料量以及 Hop 數都較本機制多，且行動裝置換手後亦不能立即利用交叉點，這是因為新擷取點沒有交叉點表格資訊，當然，CRPD 的機制原本便不是拿來做換手後利用交叉點的情景。

4.2 交叉點探索機制

本論文之交叉點探索機制 - POP 為一利用分群 (Partition) 的概念，依據已建立好的路由表建立交叉點資訊，透過此交叉點資訊表格便可以很清楚的呈現兩兩節點間以任一來源網狀擷取點的交叉點為何。而此機制的特點在於交叉點資訊分散在交叉點上，亦即只有交叉點本身會知道自己為哪兩個節點的交叉點，除了將儲存空間分散在交叉點外，對於換手前、後欲利用交叉點的情景都可以立即得知交叉點為何。底下就交叉點探索機制之概念以及交叉點探索機制之 Model 做介紹。

4.2.1 交叉點探索機制 POP 之概念

首先，本機制有個假設 (Assumption)，即所有網狀節點其路徑表格皆已經建立完成，也就是說所有網狀節點都具有到其它網狀節點的路徑資訊。依據前述，網狀網路路由演算法在當有封包要傳送，且不知道如何到目的節點時便會觸發，因此在無線網狀網路運行一段時間後，網狀節點的路由表便會建立完成，因此假設所有路徑建立好的情形是合理的。在無線網狀網路環境，每個網狀節點在路由演算法完成後，都會有以此節點到其它所有節點的路徑資訊，將此節點到其它所有節點的路徑連接起來即為一個樹 (Tree)，此節點為樹的 Root。如 Figure 4-1 所示：節點 A 有其自己生成的 Tree，同樣的，節點 C 亦有其自己生成的 Tree。

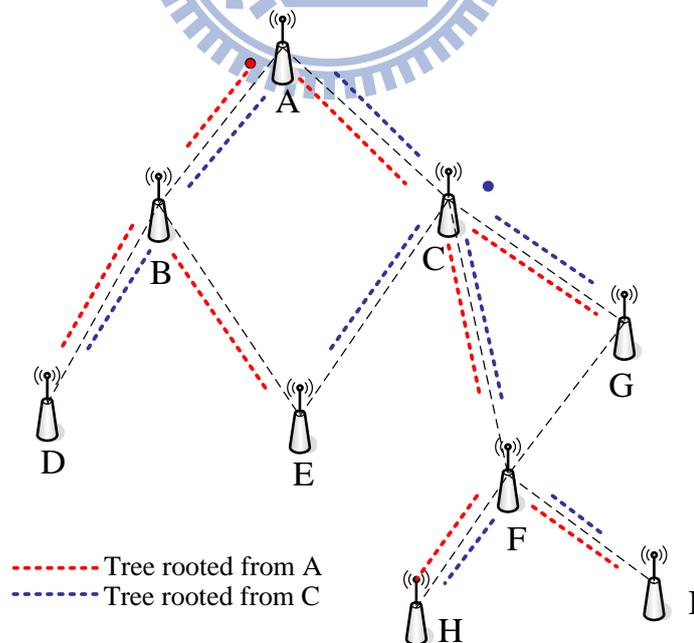


Figure 4-1 路徑生成的 Tree 示意圖

而對於一個 Tree 來說，其可以被分群為數個不同的 Sub-trees，分別如下所示：

1. Root 節點自己本身
2. 以其 Child 為 Root 之 Sub- trees

若我們視每一 Sub-tree 裡的所有節點為一集合 (Set) · 即一個 Sub-tree 為一個 Set · 這些 Sets 便為這個 Tree 的分群 (Partition) 。我們可以由 Figure 4-2 看到 · 由 Root 為 A 的 Tree 所分群而來的便是 : {A} 本身以及 {B, D, E} 以及 {C, F, G, H, I} 這三個 Sets 。

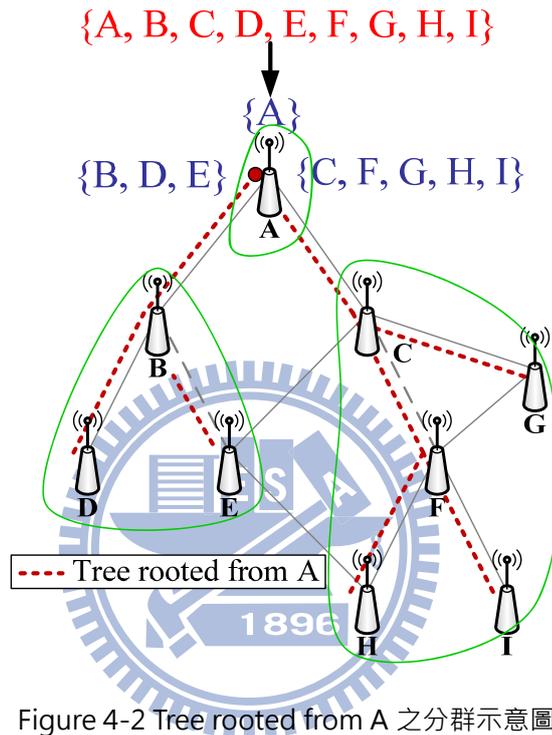


Figure 4-2 Tree rooted from A 之分群示意圖

根據分群 · 我們可以得到一個資訊 · 分群節點本身 (節點 A) · 即為這些不同 Set 裡頭任選二個節點的交叉點 · 亦即 · {A, B} {B, C} {D, F} 等以此類推的交叉點即為節點 A 本身 · 當來源擷取點為 A 時 ·

這樣的觀念同樣可以套用到每一個 Sub-tree 上 · 每一個 Sub-tree 同樣可以再被分群為數個不同的 Sub sub-trees · 分別如下所示 :

1. Root 節點自己本身
2. 以其 Child 為 Root 之 Sub sub-trees

因此 · 我們可以由 Figure 4-3 看出 · Sub sub-tree {C, F, G, H, I} 同樣可以被分群為 {C} Root 節點本身以及 {F, H, I} 與 {G} 這三群 · 同樣地此分群所呈現的資訊亦同上 : {C, F} {F, G} 等以此類推的交叉點即為節點 C 本身 · 當來源擷取點為 A 時 ·

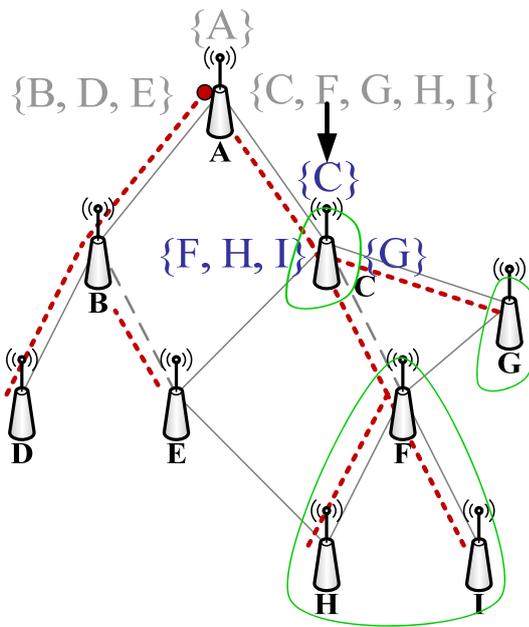


Figure 4-3 Sub-tree rooted from C 之分群示意圖

4.2.2 交叉點探索機制 POP 之方法

有了上述的分群概念後，本節將介紹如何將這觀念套用到真實的網狀網路節點上。上節我們可以由一樹狀架構拓撲很清楚的知道如何分群，但站在節點的角度上要如何知道怎麼分群呢。前面有提到此 Tree 其實為根據路由演算法所形成的，亦即由路由表的資訊將此 Tree 建立起來，因此，我們便可以知道，依據路由表裡的 Next hop，便可以做分群的動作。接著我們描述每個節點進行分群的步驟，我們假定路由表有到所有節點的資訊，起始節點：

1. 將路由表中的每一節點依據 Next hop 做分群，相同的 Next hop 為同一群。
2. 將這些分群儲存在自身的交叉點表格上，供之後使用。這邊有一點要注意的是，節點自己本身即自成一群，但路由表內部並不會有到『本身節點的下一個節點為何』這一個資訊，因此若有節點本身要分群，我們便自行在交叉點表格創立節點本身這一群，而其 Child 的分群則由路由表查詢並將之分群。
3. 針對每分群裡有兩個以上的節點，分別將之傳遞給其 Next hop，讓 Next hop 節點再針對這些未分群的節點依 1.之概念查詢路由表做分群動作。

上述為交叉點探索機制之基本流程，我們接著直接舉一個以一節點出發之交叉點探索流程：

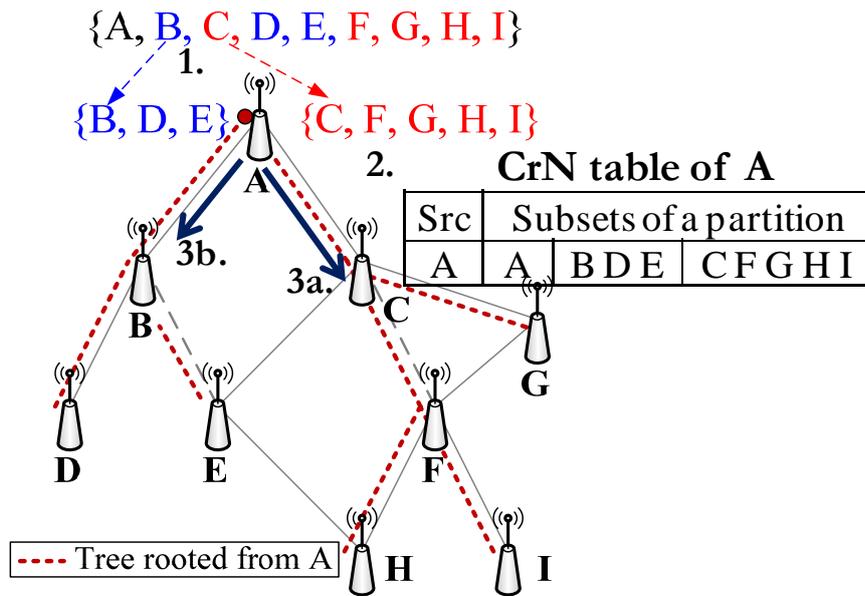


Figure 4-4 節點 A 交叉點探索機制之流程(一)

如 Figure 4-4 所示，節點 A 開始此程序，其 Table 4-1 路由表中會有到 B, C, D, E, F, G, H, I 的下一個節點為何之資訊，由路由表便可知，到 B, D, E 的下一個節點為 B，到 C, F, G, H, I 的下一個節點為 C。因此可以將節點分為 {B, D, E} 與 {C, F, G, H, I} 兩群。接著將此儲存在交叉點表格上，上述說明了節點本身為一群，因此交叉點表格會有節點自己，另外再將分群出來的儲存至交叉點表格上，即為圖中的 CrN table of A。而分群出來且有兩個以上節點的為 {B, D, E} 與 {C, F, G, H, I}，所以還需要向下一個節點去做分群：{B, D, E} 往下一個節點 B，{C, F, G, H, I} 往下一個節點 C。

Table 4-1 節點 A 之路由表格

A Routing Table	
Destination	Next Hop
B	B
C	C
D	B
E	B
F	C
G	C

H	C
I	C

接著我們舉節點 C 收到{C, F, G, H, I}為例：

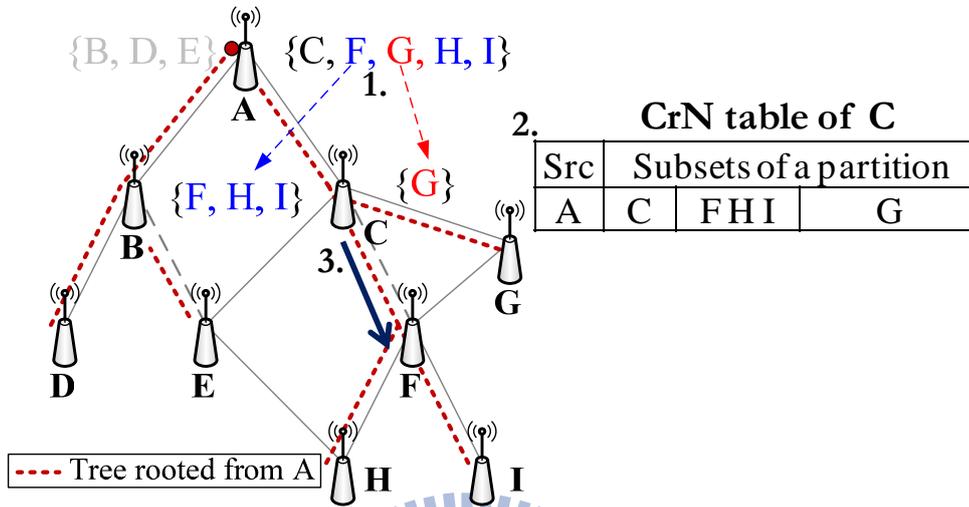


Figure 4-5 節點 A 交叉點探索機制之流程(二)

如 Figure 4-5 所示，C 欲分群節點為{C, F, G, H, I}，其查詢 Table 4-2 路由表裡的 F, G, H, I 的下一個節點為何之資訊，由路由表便可知，到 F, H, I 的下一個節點為 F，到 G 的下一個節點為 G。因此可以將節點分為{B, D, E}與{C, F, G, H, I}兩群。接著將此儲存在交叉點表格上，同樣的，交叉點表格會有節點 C 自己，另外再將分群出來的儲存至交叉點表格上，即為圖中的 CrN table of C。而分群出來且有兩個以上節點的為{F, H, I}，所以還需要向下一個節點去做分群：{F, H, I}往下一個節點 F。

Table 4-2 節點 C 之路由表格

C Routing Table	
Destination	Next Hop
A	A
B	A
D	A
E	A
F	F

G	G
H	F
I	F

接著舉節點 F 收到{F, H, I}為例：

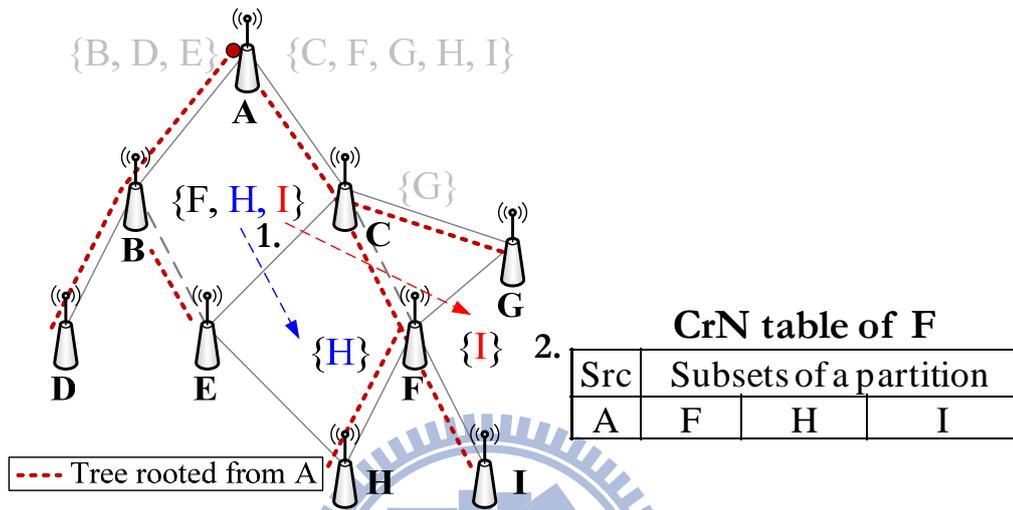


Figure 4-6 節點 A 交叉點探索機制之流程(三)

如 Figure 4-6 所示，F 欲分群節點為{F, H, I}，其查詢 Table 4-3 路由表裡的 H, I 的下一個節點為何之資訊，由路由表便可知，到 H 的下一個節點為 H，到 I 的下一個節點為 I。因此可以將節點分為{H}與{I}兩群。接著將此儲存在交叉點表格上，同樣的，交叉點表格會有節點 F 自己，另外再將分群出來的儲存至交叉點表格上，即為圖中的 CrN table of F。而分群出來的皆沒有兩個節點以上，因此不需要再往下找尋，此節點便不用再往下做交叉點探索的程序。

Table 4-3 節點 F 之路由表格

F Routing Table	
Destination	Next Hop
A	C
B	C
C	C
D	C

E	C
G	G
H	H
I	I

同樣的，對於節點 A 分群出的{B, D, E}也是做相同的分群動作，因此，最後的交叉點資訊如 Figure 4-7 所示：

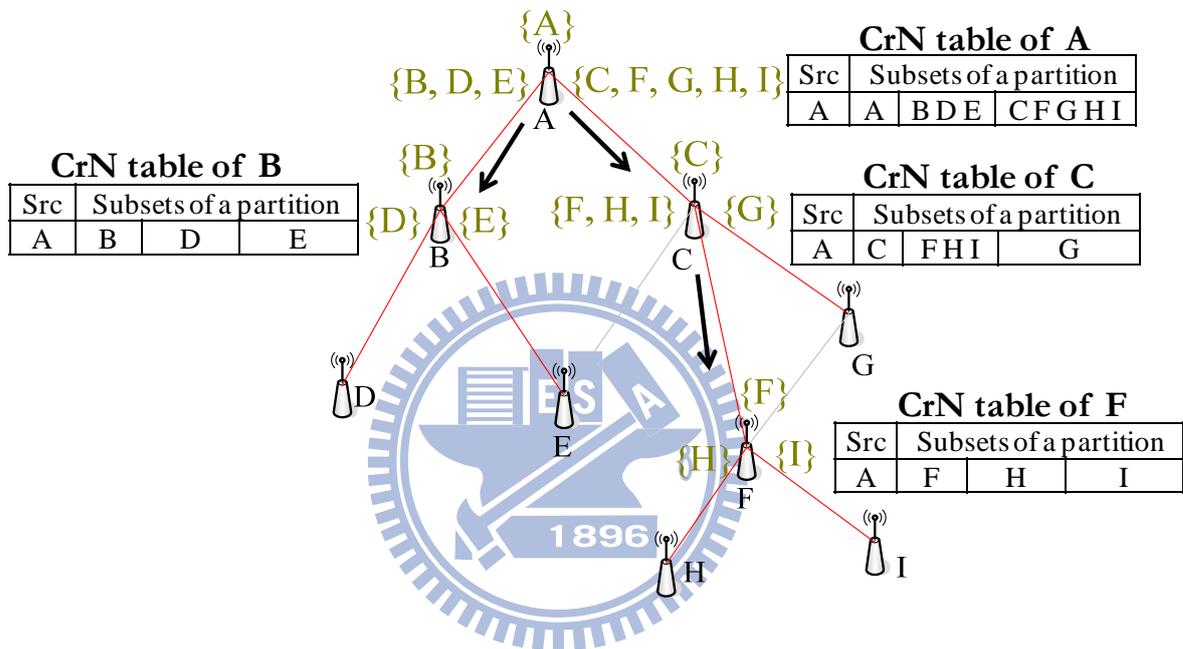


Figure 4-7 節點 A 交叉點探索機制之交叉點表格資訊

我們可以看到，交叉點表格的資訊為分散在交叉點本身上，如此可適用於由新舊擷取點直接延來源節點方向找尋便可知道交叉點為何。而其它節點同樣依據這樣的機制便可將所有節點生成的樹其交叉點資訊建立完畢，如此便可適用於任何 MN 與 CN 溝通的情形。以上介紹的便是 POP 交叉點探索機制。

如此便可以把以節點 A 為來源節點其底下任兩節點間的交叉點資訊建立完成，透過這樣一次性探索，可以用最少的訊息量建立最多的交叉點資訊，我們可以看到，僅需三個訊息，便可以建立好來源節點 A 的交叉點表格資訊，而這是不會浪費的，因為我們僅是將資訊建立好，要查找的時候再針對特定的節點去做找尋即可。而對任何一個節點所長出的 Tree 都可以透過這樣的方式將以此節點為出發點其底下任兩節點間的交叉點資訊建立完畢，接下來我們透過一個比較公式的概念來將本交叉點找尋機制做 Model。

4.2.3 交叉點探索機制 POP 之 Model

介紹完了交叉點探索機制的程序後，接著便來介紹我們 Model 出的 POP 交叉點探索機制，只要依循此流程，便可以很輕易地利用上述觀念對任一網狀擷取點做一次性分群動作，其是將上節介紹的交叉點探索機制做較正式的定義。首先定義此 Model 的一些名詞定義：

V : A set of nodes in WMNs (即無線網狀網路內的網狀節點代號)

N_u : A set of branches of node u (即網狀節點 u 其所有的分支，包括節點本身)

RT_u : The routing table of node u (即網狀節點 u 的路由表資訊)

Define:

$$\forall u \in V, RT_u = \{ \langle dst, next \rangle \mid (\forall dst \in V \wedge dst \neq u), next \in N_u \} \quad (1)$$

RT_u 的表示方式為 $\langle dst, next \rangle$ ，代表欲到此目的節點 dst 需將封包往下一個節點 $next$ 傳送。 dst 為網狀節點內的所有點但不包括 u 節點本身， $next$ 為節點 u 的鄰居。

UP_u^{proxy} : The element of nodes used to be partitioned in u of $proxy$ (即以 $proxy$ 為出發點，在節點 u 上需要分群的節點)

Define:

$$\begin{aligned} \forall proxy \in V, UP_{proxy}^{proxy} &= V \\ \forall proxy, u \in V, UP_u^{proxy} &= P_{(v,u)}^{proxy} \end{aligned} \quad (2)$$

UP_{proxy}^{proxy} 即表示出發節點欲要分群的節點，通常代表所有網路內的節點，而 UP_u^{proxy} 即為節點 u 收到欲要分群的節點。

$P_{(u,v)}^{proxy}$: The partition on the (u,v) of $proxy$ (即以 $proxy$ 為出發點，節點 u 對分支 v 的分群)

Define:

$$\forall proxy, u \in V, v \in N_u, P_{(u,v)}^{proxy} = \{ x \mid \forall x \in UP_u^{proxy}, \text{if } \langle x, v \rangle \in RT_u \} \quad (3)$$

$P_{(u,v)}^{proxy}$ 即代表依據 UP_u^{proxy} 資訊，查詢 RT_u 裡頭 $next$ 為節點 v 的節點，代表著對 UP_u^{proxy} 欲要分群節點所做的分群。

Figure 4-8 為 POP 交叉點探索機制之流程圖示意圖，下述例子皆可依循此流程圖運作：

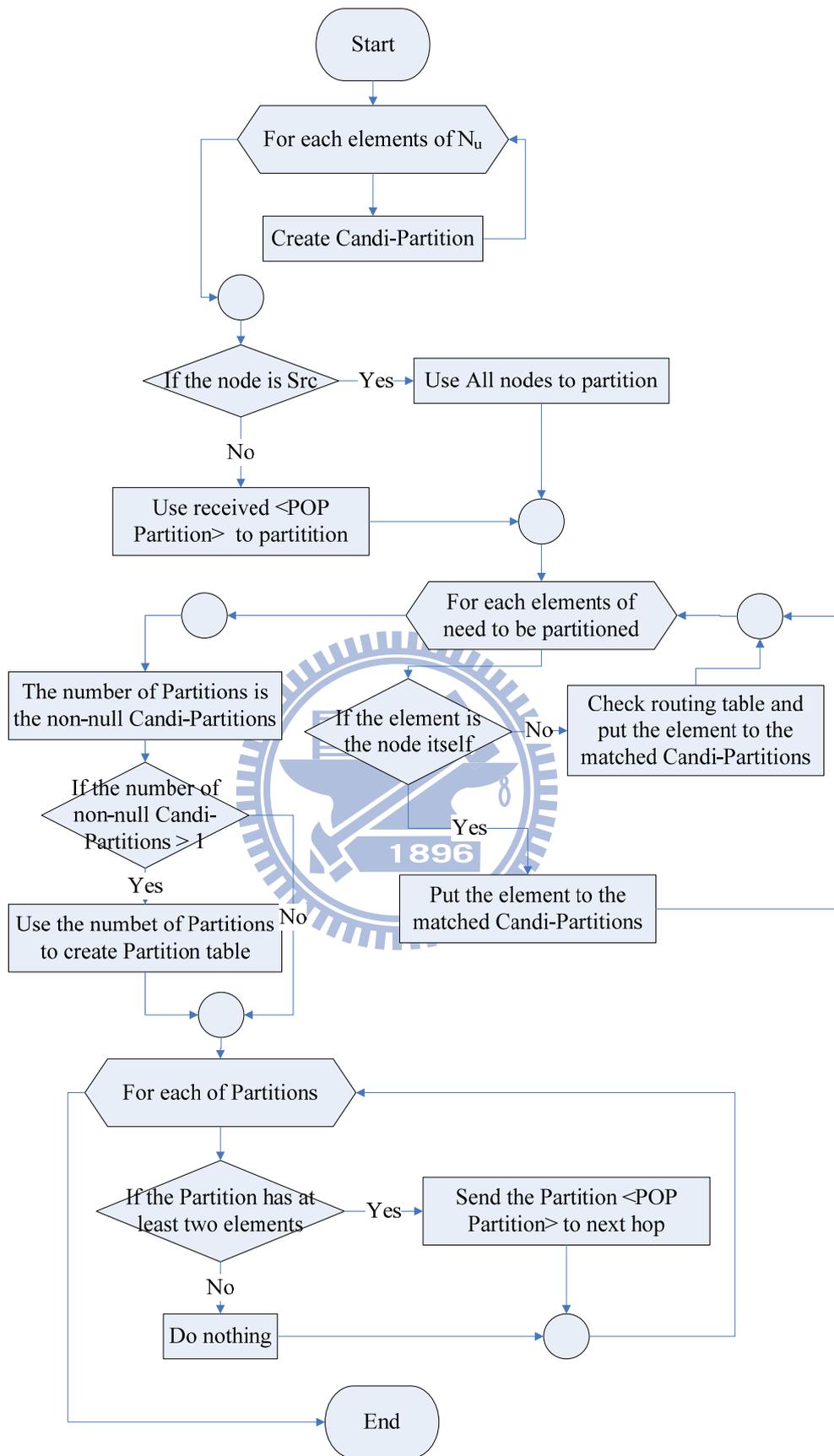


Figure 4-8 交叉點探索機制流程圖示意圖

接下來會舉一個示例來驗證本論文所提的式子，如 Figure 4-9 所示：以節點 A

出發來說明，圖中紅線代表節點 A 到其它所有節點的路徑。對每一節點均可以透過此 Model 找出以其出發所建立之樹狀拓樸之任兩節點間之交叉點：

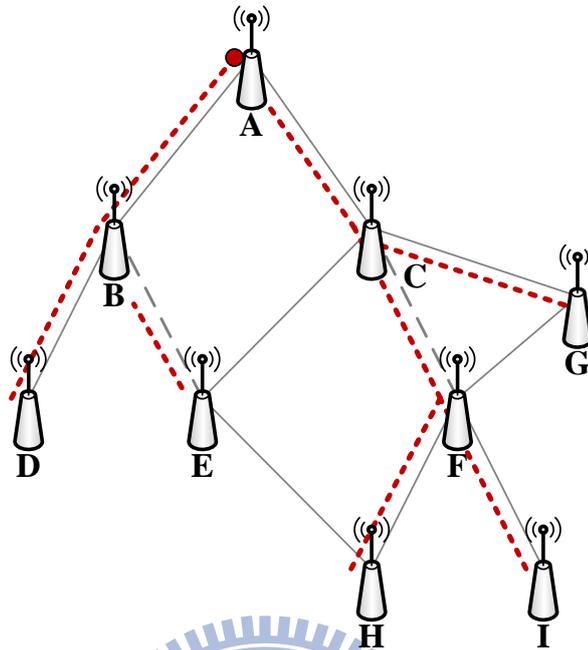


Figure 4-9 POP 示例示意圖

先將上述的式子定義出來：

$V = \{A, B, C, D, E, F, G, H, I\}$ ，為網狀網路內的所有網狀節點。

$N_A = \{A, B, C\}$ ，為節點 A 其分支。

$RT_A = \{\langle B, B \rangle, \langle C, C \rangle, \langle D, B \rangle, \langle E, B \rangle, \langle F, C \rangle, \langle G, C \rangle, \langle H, C \rangle, \langle I, C \rangle\}$ ，為節點 A 的路由表資訊： $\langle F, C \rangle$ 代表欲到節點 F，需將封包傳送往下一節點 C。此由上圖虛線便可以看出這資訊。這些資訊可由路由表得知的，

接著，由節點 A 出發執行 POP 程序，欲分群的節點為網狀網路內所有的節點，因此：

$$UP_A^A = \{A, B, C, D, E, F, G, H, I\}$$

接著依據上式開始做分群的動作，依據 $N_A = \{A, B, C\}$ ，節點 A 最多可以分為三群，因此最多可以分為 $P_{(A,A)}^A$ 、 $P_{(A,B)}^A$ 與 $P_{(A,C)}^A$ ，將欲分群的節點 $UP_{(A,A)}^A : \{A, B, C, D, E, F, G, H, I\}$ 依定義並查詢路由表後填入相對應的分群中。節點 A 為節點自己本身，因此可以直接將之分群至 $P_{(A,A)}^A$ 。而欲到節點 B 的下一個節點為 B，因此節點 B 屬於 $P_{(A,B)}^A$ ；欲到節點 C 的下一個節點為 C，因此節點 C 屬於 $P_{(A,C)}^A$ ；欲到節點 D 的下一個節點為 B，因此節點 D 屬於 $P_{(A,B)}^A$... 依序將剩餘節點 E, F, G, H, I 分群後填入 $P_{(A,B)}^A$ 或 $P_{(A,C)}^A$ ，其結果如下：

$$P_{(A,A)}^A = \{A\}$$

$$P_{(A,B)}^A = \{B, D, E\}$$

$$P_{(A,C)}^A = \{C, F, G, H, I\}$$

依據上節定義，不同的分支可以分為這三群，此三群代表的意義為：節點 A 本身為 $P_{(A,A)}^A$ 、 $P_{(A,B)}^A$ 與 $P_{(A,C)}^A$ 這三群間的交叉點，即任兩節點 $\{A, B\}$ 、 $\{A, D\}$ 、 $\{A, E\}$ 、 $\{A, C\}$ 、 $\{A, F\}$ 、 $\{A, G\}$ 、 $\{A, H\}$ 、 $\{A, I\}$ 、 $\{B, C\}$ 、 $\{B, F\}$ 、 $\{B, G\}$ 、 $\{B, H\}$ 、 $\{B, I\}$ 、 $\{D, C\}$ 、 $\{D, F\}$ 、 $\{D, G\}$ 、 $\{D, H\}$ 、 $\{D, I\}$ 、 $\{E, C\}$ 、 $\{E, F\}$ 、 $\{E, G\}$ 、 $\{E, H\}$ 、 $\{E, I\}$ 間的交叉點為節點 A。而同一分群裡的節點若有兩個以上即代表此群裡頭裡的節點其交叉點尚未得知，因此必須繼續往下分群， $P_{(A,B)}^A$ 往節點 B 做分群； $P_{(A,C)}^A$ 往節點 C 做分群，如 Figure 4-10 所示：

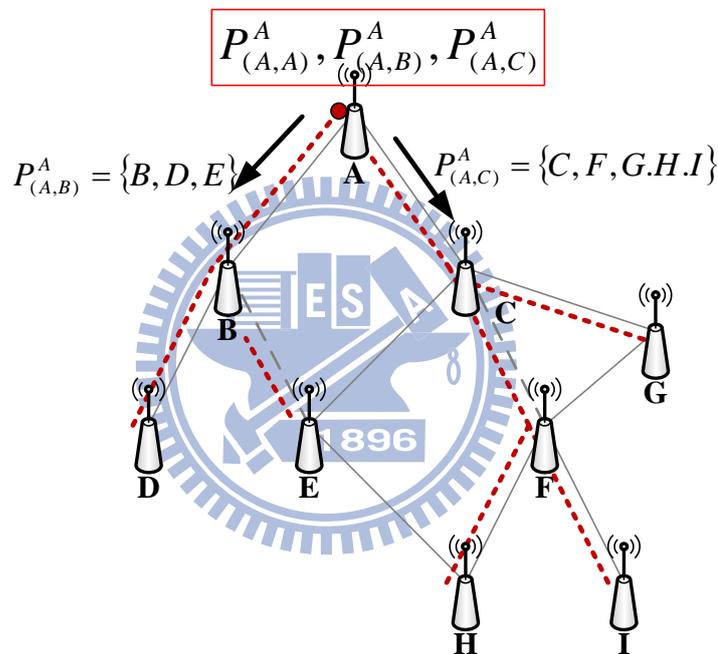


Figure 4-10 節點 A 分群及傳遞方向示意圖(一)

節點 C 收到 $P_{(A,C)}^A$ 後，即表示節點 C 欲要分群的節點為即為 $\{C, F, G, H, I\}$ ，因此： $UP_C^A = \{C, F, G, H, I\}$ ，這裡就是所收到的節點，我們將之轉成易懂的符號表示。而節點 C 有以下資訊：

$$N_C = \{A, C, E, F, G\}$$

$$RT_C = \{\langle A, A \rangle, \langle B, A \rangle, \langle D, A \rangle, \langle E, E \rangle, \langle F, F \rangle, \langle G, G \rangle, \langle H, F \rangle, \langle I, F \rangle\}$$

同樣依據上式開始做分群的動作，節點 C 依據 N_C 表示最多可以有 $P_{(C,A)}^A$ 、 $P_{(C,C)}^A$ 、 $P_{(C,E)}^A$ 、 $P_{(C,F)}^A$ 、 $P_{(C,G)}^A$ ，將欲分群的節點 $UP_C^A : \{C, F, G, H, I\}$ 依定義並查詢路由表後填入相對應的分群中。節點 C 為節點本身，因此其屬於 $P_{(C,C)}^A$ 。欲到節點 F 的下一個節點為 F。

因此節點 F 屬於 $P_{(C,F)}^A$; 欲到節點 G 的下一個節點為 G · 因此節點 G 屬於 $P_{(C,G)}^A$; 欲到節點 H 的下一個節點為 F · 因此節點 H 屬於 $P_{(C,F)}^A$; 欲到節點 I 的下一個節點為 F · 因此節點 I 屬於 $P_{(C,F)}^A$ · 其結果如下 :

$$P_{(C,A)}^A = \{\phi\}$$

$$P_{(C,C)}^A = \{C\}$$

$$P_{(C,E)}^A = \{\phi\}$$

$$P_{(C,F)}^A = \{F, H, I\}$$

$$P_{(C,G)}^A = \{G\}$$

由於欲分群節點 $\{C, F, G, H, I\}$ 其下一個節點沒有節點 A 與節點 E · 因此 $P_{(C,A)}^A$ 與 $P_{(C,E)}^A$ 均為空集合 · 依據上式 · 節點 C 便將欲分群節點分為三群 $P_{(C,C)}^A$ · $P_{(C,F)}^A$ 與 $P_{(C,G)}^A$ · 同樣的 $P_{(C,C)}^A$ · $P_{(C,F)}^A$ 與 $P_{(C,G)}^A$ 的交叉點即為節點 C 本身 · $P_{(C,G)}^A$ 由於僅剩下一個節點 G · 表示不用再做分群的動作 · 因此不用再往節點 G 做分群 ; $P_{(C,F)}^A$ 則同樣必須往節點 F 做分群 · 節點 C 欲分群節點及傳遞方向如 Figure 4-11 所示 :

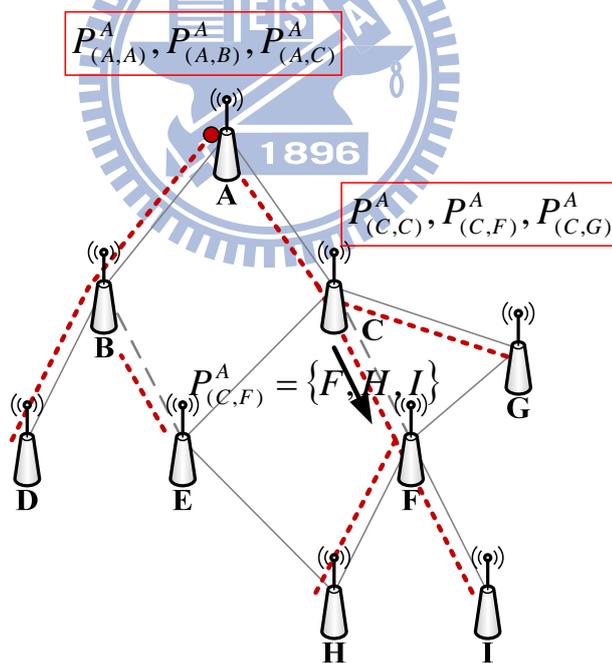


Figure 4-11 節點 A 分群及傳遞方向示意圖(二)

節點 F 收到 $P_{(C,F)}^A$ · 同樣將之轉換為欲分群的節點表示符號 $UP_{F\setminus}^A$ · 因此 :

$$UP_F^A = \{F, H, I\}$$

而節點 F 有以下資訊 :

$$N_F = \{C, F, G, H, I\}$$

$$RT_F = \{\langle A, C \rangle, \langle B, C \rangle, \langle C, C \rangle, \langle D, C \rangle, \langle E, C \rangle, \langle G, G \rangle, \langle H, H \rangle, \langle I, I \rangle\}$$

同樣將欲分群的節點 $UP_F^A : \{F, H, I\}$ 依定義並查詢路由表後填入相對應的分群中，節點 F 最多可分群為 $P_{(F,C)}^A$ 、 $P_{(F,F)}^A$ 、 $P_{(F,G)}^A$ 、 $P_{(F,H)}^A$ 、 $P_{(F,I)}^A$ 。節點 F 為節點本身，因此其屬於 $P_{(F,F)}^A$ ；欲到節點 H 的下一個節點為 H，因此節點 H 屬於 $P_{(F,H)}^A$ ；欲到節點 I 的下一個節點為 I，因此節點 I 屬於 $P_{(F,I)}^A$ ，其結果如下：

$$P_{(F,C)}^A = \{\phi\}$$

$$P_{(F,F)}^A = \{F\}$$

$$P_{(F,G)}^A = \{\phi\}$$

$$P_{(F,H)}^A = \{H\}$$

$$P_{(F,I)}^A = \{I\}$$

欲分群節點 $\{F, H, I\}$ 其下一個節點沒有節點 C 與節點 G，因此 $P_{(F,C)}^A$ 與 $P_{(F,G)}^A$ 為空集合，節點 F 便將欲分群節點分為三群 $P_{(F,F)}^A$ 、 $P_{(F,H)}^A$ 與 $P_{(F,I)}^A$ ，同樣代表 $P_{(F,F)}^A$ 、 $P_{(F,H)}^A$ 與 $P_{(F,I)}^A$ 間的交叉點即為節點 F 本身，而這些分群因其欲分群的節點皆僅剩一個，因此不需要再做分群的動作。同樣的，原本由節點 A 左群的 $P_{(A,B)}^A$ 欲做分群的其分群動作亦同，最後的分群結果及訊息傳遞流程如 Figure 4-12 所示：

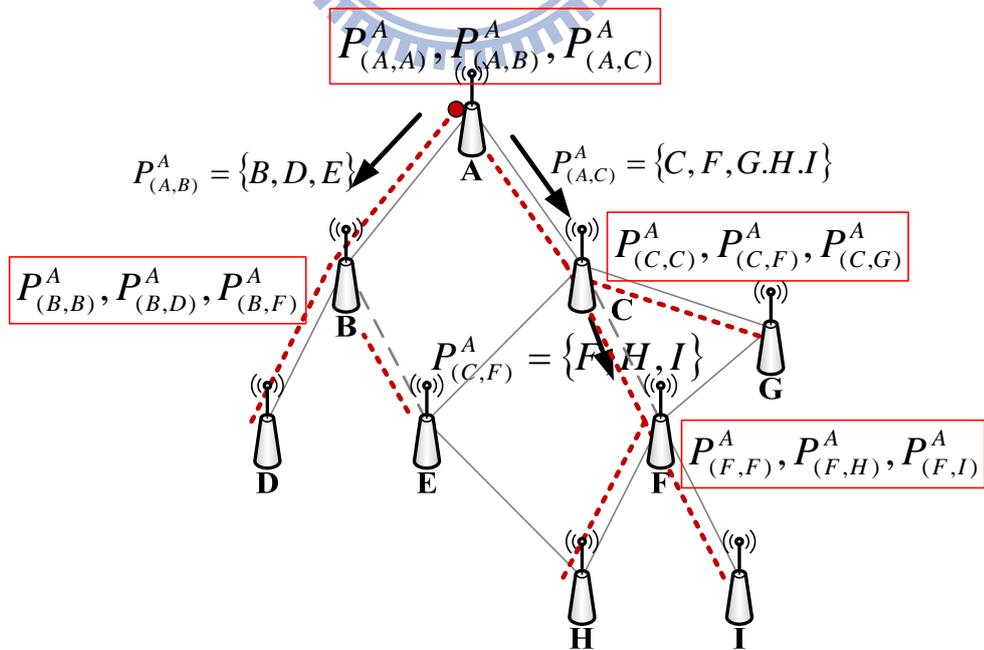


Figure 4-12 節點 A 分群及傳遞方向示意圖(三)

4.3 交叉點使用之情節

經由 POP 的程序，便可以將任兩節點間的交叉點資訊表格建立出來，此表格所代表的資訊也已經說明了，本節就來介紹當行動裝置即將換手時，如何利用交叉點來做 Bi-casting 以及一些可以利用交叉點的情境。

當 MN 要換手時，其可以傳送分叉點查詢 (Lookup) 訊息，此訊息沿著來源網狀擷取點的方向傳遞，當然，欲從舊網狀擷取點傳送或新網狀擷取點傳送皆是可行的方式，端看設計者如何使用。而沿路上的節點在還未找到交叉點前都會查詢交叉點表格以檢查自己是否為新、舊網狀擷取點的交叉點，其每個節點收到會做以下動作：

1. 先查到符合的 source MAP 的交叉點表格
2. 再檢查訊息裡頭欲查詢的網狀擷取點是否存在於此交叉點表格內。
 - i. 如果有的話，則自己為交叉點，此時便可以利用交叉點做事情 (Bi-casting 或是 Data-Redirection)，視機制而定。
 - ii. 如果沒有的話，則繼續往 source MAP 的 next hop 傳送去查找交叉點。

我們接下來舉兩個例子說明，如 Figure 4-13 所示：

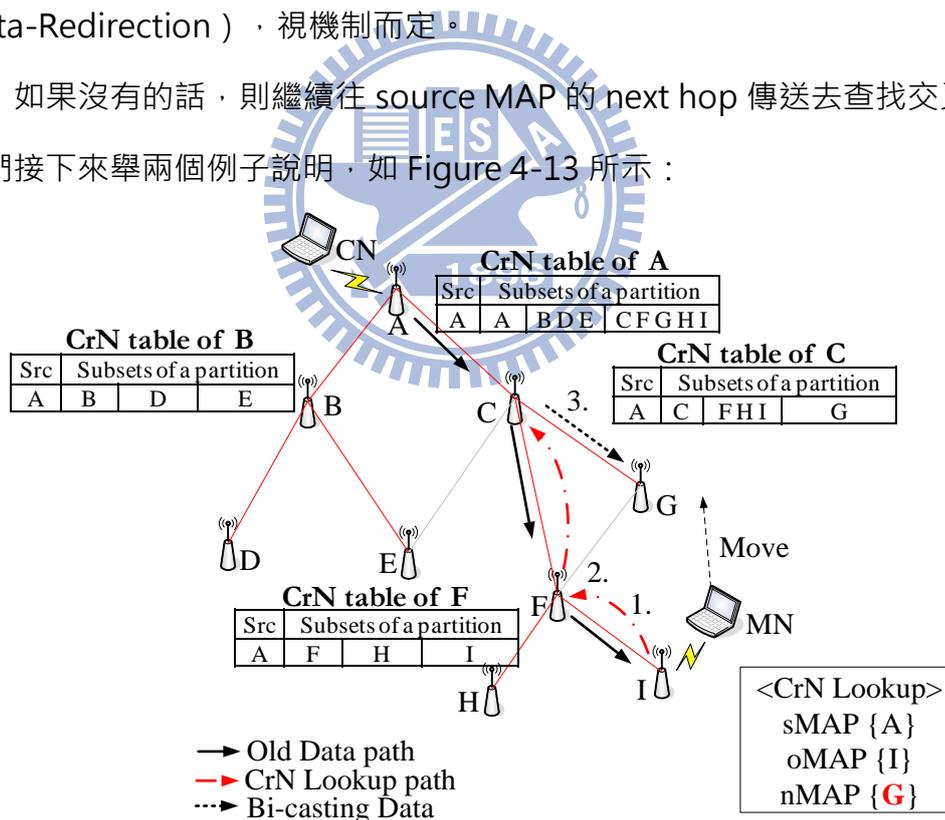


Figure 4-13 利用交叉點表格之示意圖 - 換手前

行動裝置 CN 連接上 A，行動裝置 MN 連接上 I，CN 傳送封包給 MN 的路徑為 A→C→F→I，當 MN 要換手時，它預測接下來會連接 G，此時會發出 <CrN Lookup> 訊息，請交叉點做 Bi-casting 的動作。此訊息會挾帶著 CN 連接上的網狀擷取點，即來源網狀擷取點 (此為 A)，以及欲要找尋交叉點的節點 (此為 G)，可多個，接著 < CrN

Lookup > 訊息會被沿著原封包傳送路徑往來源網狀節點的方向傳送，節點本身與沿路上的網狀節點收到此訊息後會依據所挾著的訊息內容查詢交叉點表格，檢查自己是否為交叉點。

舉例來說：當 F 收到 < CrN Lookup > 訊息後，會先查詢符合 Source MAP 的交叉點表格，接著再查詢訊息裡的候選網狀擷取點（此為 G）是否存在此交叉點表格中。不需查詢 MN 現在連上的網狀擷取點（Current_MAP，此為 I）是因為此 < CrN Lookup > 訊息僅會沿著原路徑往來源網狀節點傳送，而 MN 連上的網狀擷取點一定都會在沿路網狀節點的交叉點表格中，由 Figure 4-13 便可以清楚看到：I - 即 MN 連上的網狀擷取點存在於 F→C→A 路徑上網狀節點的交叉點表格中，因此只需要查詢候選網狀擷取點（Candidate_MAP，此為 G）是否存在交叉點表格中即可。F 收到後，發現交叉點表格沒有 G 的資訊，表示尚未找到交叉點，於是繼續往原路徑傳送；當 C 收到後，同樣查詢交叉點表格，發現 G 的資訊，此時便可知道自己為交叉點，C 便會將封包做 Bi-casting 給 G，而 G 收到後會將封包暫存起來，當 MN 連接上便將暫存的資料傳送給它。

接著來看另一使用交叉點之情景，當 MN 連上 G 後，依據前述會做位置更新的動作，此時也可以利用交叉點來減少頻寬的浪費，如 Figure 4-14 所示：

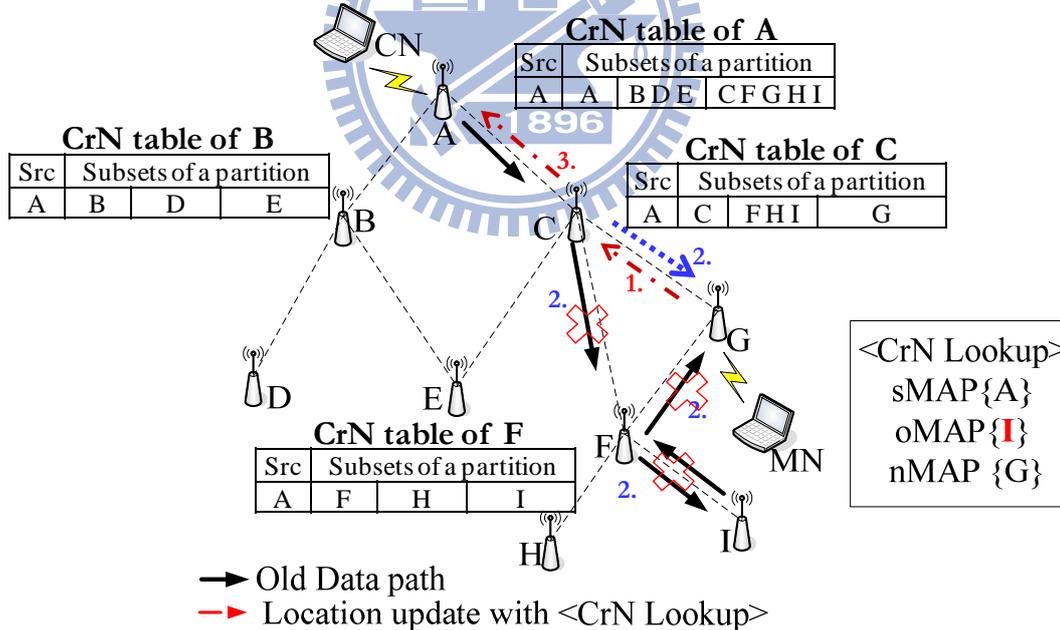


Figure 4-14 利用交叉點表格之示意圖 - 換手後

此 < CrN Lookup > 訊息可夾帶在位置更新訊息中，訊息接著便會被沿著往來源擷取點的路徑傳送，當 C 收到此訊息後，依據訊息內容，同樣先找尋 Source Map 為 A 的交叉點表格，接著再查詢得知 Candidate_MAP（此為 I）存在交叉點表格中，便知道自己為交叉點，此時便可立刻不再將封包往舊網狀擷取點 I 傳送了，而位置更新訊息

再繼續往來源網狀擷取點的路徑傳送。若按照原本，位置更新必須等到 CN 連上的網狀擷取點收到更新後才不會將封包送往舊網狀擷取點，而利用交叉點便可以提前將封包停止送往舊網狀擷取點了。

現在我們舉另外一個例子來說明，當行動裝置延著原路徑換手時，依然可以利用建立好的交叉點表格找到交叉點，我們舉個例子說明，如 Figure 4-15 所示：

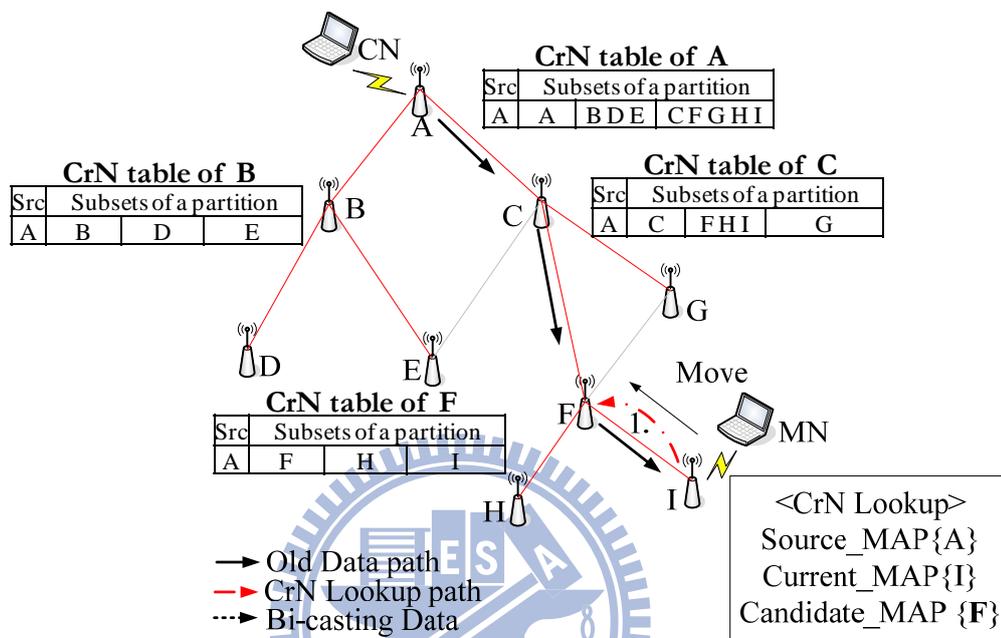


Figure 4-15 行動裝置延原路徑換手之示意圖

當 MN 欲換手到 F 時，會發出 < CrN Lookup > 訊息，I 會先檢查自己是否為交叉點，但其無 Source MAP 為 A 的交叉點表格，因此往上傳送。當 F 收到後，查詢交叉點表格，而交叉點表格有此資訊，這時會知道自己即為交叉點，同樣的會做 Bi-casting 的動作傳送封包給 F，即自己，F 便會暫存封包，待 MN 連上後便可立即取得 F 幫忙暫存的封包，位置更新也是同樣的動作，F 知道自己為交叉點後便停止將封包送往舊網狀擷取點 I。

即使有多個候選網狀擷取點，依然可以使用上述方法找尋出交叉點，此 < CrN Lookup > 訊息是可以挾帶多個候選網狀擷取點的，而只要將已經找過的候選網狀擷取點做註記或取消再依序往來源網狀擷取點的路徑傳送即可。僅利用此表格，不管來源網狀擷取點為何，都可以利用最正確的交叉點去做 Bi-casting 的動作或是做其他需要利用到交叉點的機制，以達到短延遲時間，並減少不必要頻寬的浪費。Figure 4-16 為上述收到 < CrN Lookup > 之交叉點運用示意圖，用框框起來的為交叉點知道自己為交叉點後，欲要請交叉點做何事，這裡舉 Bi-casting 為例，當然，如前所述，交叉點使用情景不僅於此：

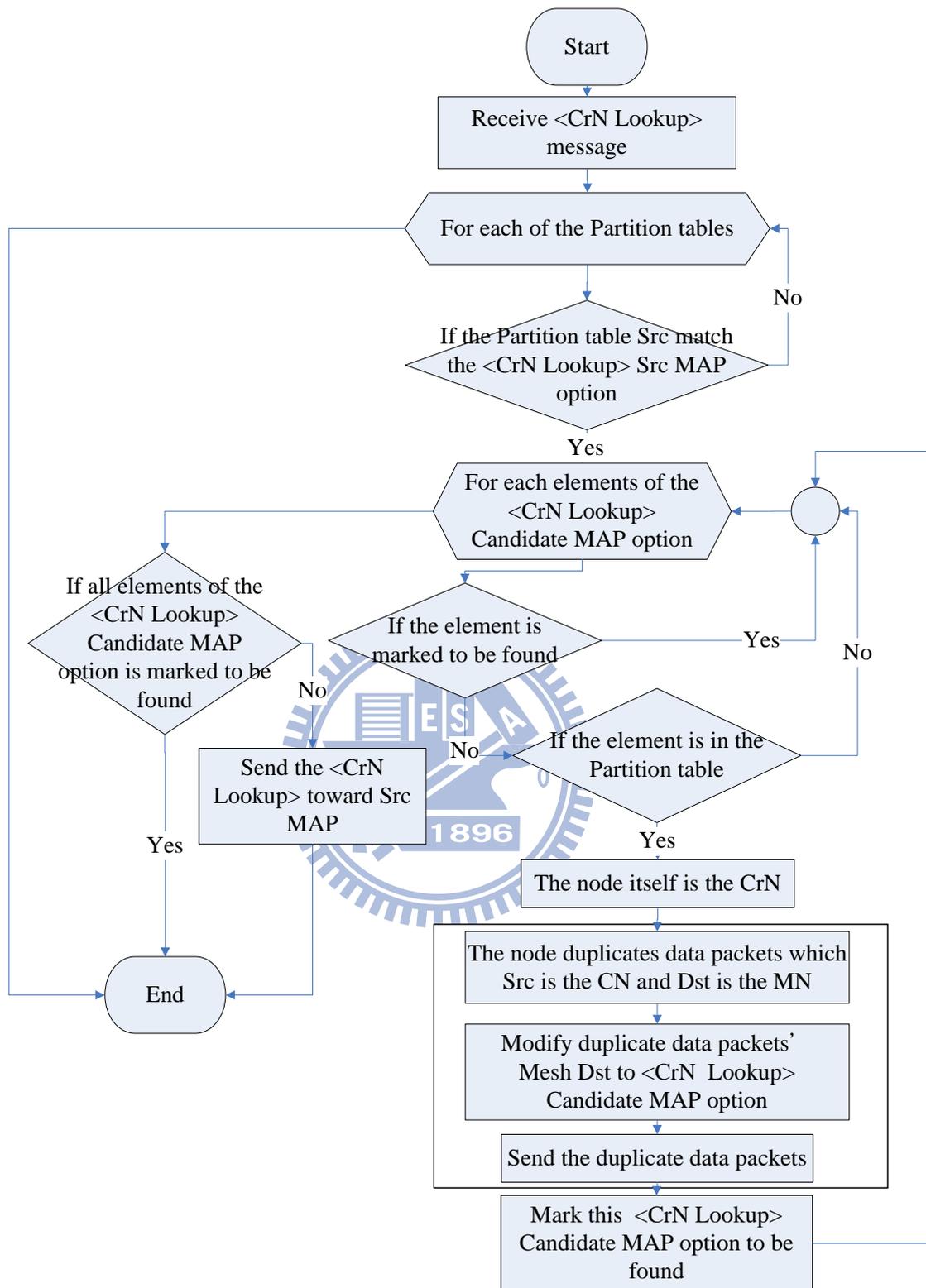


Figure 4-16 交叉點運用流程圖示意圖

4.4 POP 機制之延伸

本交叉點探索機制是提供一個觀念，將任意兩節點間的交叉點表格都建立起來，但

若是僅要建立某些節點間的交叉點表格，同樣是可以做到的，如 Figure 4-17 所示：

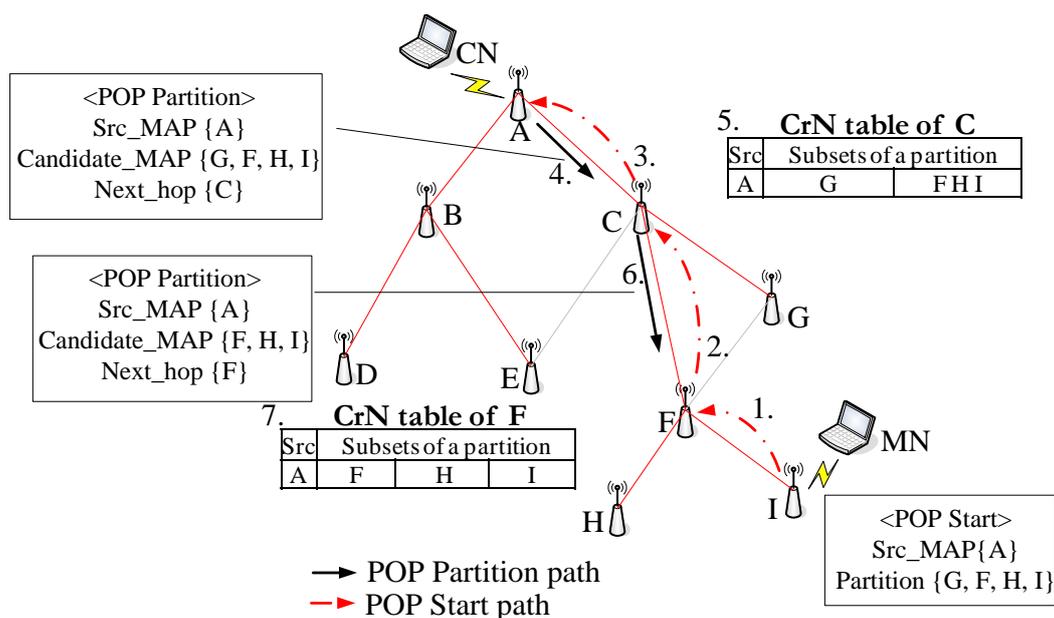


Figure 4-17 POP 機制延伸之示意圖

行動裝置 CN 與行動裝置 MN 在溝通，此時 I 可以根據某些機制（如地理位置），認為 MN 僅有可能換手到其鄰近的地方 - F、G、H，因此欲將 I 與這些候選網狀擷取點的交叉點表格建立出來，其實就是請來源網狀擷取點對這些節點做分群的動作即可。此時便會發出 <POP Start> 訊息，此訊息會挾帶著 CN 所連上的網狀擷取點、MN 連上的網狀擷取點以及候選網狀擷取點並往 CN 所連上的網狀擷取點傳送。

此訊息是用來執行 POP 探索機制，即針對某些節點建立交叉點表格資訊，當 A 收到後，便會開始執行 POP，與原本不同的是，原本的定義是所有網狀節點，表示要對所有的節點做分群，但在這裡則只要對 <POP Start> 訊息內容裡的 G、F、H、I 做分群即可，我們可以看到，G、F、H、I 的下一節點皆為 C，因此只能分一群出來，而由於僅有一群，因此不需要建立交叉點表格，接著便往 C 的方向去繼續做分群動作，如此便可以僅將欲找節點間的交叉點表格建立好。

而當要利用交叉點做 Bi-casting 動作時，同樣可依據上節的方式去執行，交叉點表格即代表了節點間交叉點的資訊，此法是較適用於路徑變動較大的時候使用，如此可以不需要一開始將所有交叉點表格建立好。此機制流程與相關研究裡的 CRPD 類似，但其解決了 CRPD 的缺點：一是傳遞過程的資料量少，CRPD 必須將整個 I 與 F、G、H 的交叉點為何沿著路徑傳上，而 POP 僅需要傳遞尚須要分群的節點即可。二是 CRPD 必須將交叉點表格傳遞到 MN 所連上的網狀擷取點（這裡為 I），而 POP 僅需傳遞尚須分群的節點到最後一個交叉點上（這裡為 F）即可，即 POP 所傳遞的 HOP 數至少會比 CRPD 少一個，對頻寬亦是一種節省。三是 CRPD 的方式僅能在換手前利用交叉點做事，

因為其交叉點表格是儲存在 MN 連接的網狀擷取點，所以若換手後欲要利用交叉點，還必須要重新做一次交叉點表格的建立，而 POP 將交叉點表格建立後，不論換手前、後也都可以直接利用交叉點做漫遊機制，因為交叉點表格是在交叉點上，因此可以直接查詢使用。

4.5 POP 機制之探討

為什麼需要本論文之交叉點探索機制呢？為什麼不能直接由舊擷取點或新擷取點往來源行動裝置所連接上的擷取點去尋找交叉點呢？如此不就可以不需要本論文的程序不是嗎？在傳統的網路其答案是“是的”，但在無線網狀網路的環境下答案卻是不行的。原因就是傳統網路的路由路徑是一個單一樹 (Single Tree)，也就是所有節點長出的樹是唯——個樹；但在無線網狀網路卻是一個多樹 (Multiple Tree) 架構，也就是每個節點所長出的樹合起來並非唯——個樹，如 Figure 4-18 所示，由 MPP 到其他所有節點的路徑所生成的樹以及由 oMAP 到其他所有節點的路徑所生成的樹並不是一樣的。因此，若直接由 oMap 往到來源擷取點的方向直接做交叉點找尋動作，可以看到，當在 oMAP 比對到 MPP 與到 nMAP 的 Next hop，發現到 MPP 的 Next hop 為 MP2，到 nMAP 的 Next hop 為 nMAP，因此 oMAP 就認為自己為交叉點，但這結果是錯的，因此才需要本論文方法先將交叉點資訊建立起來。

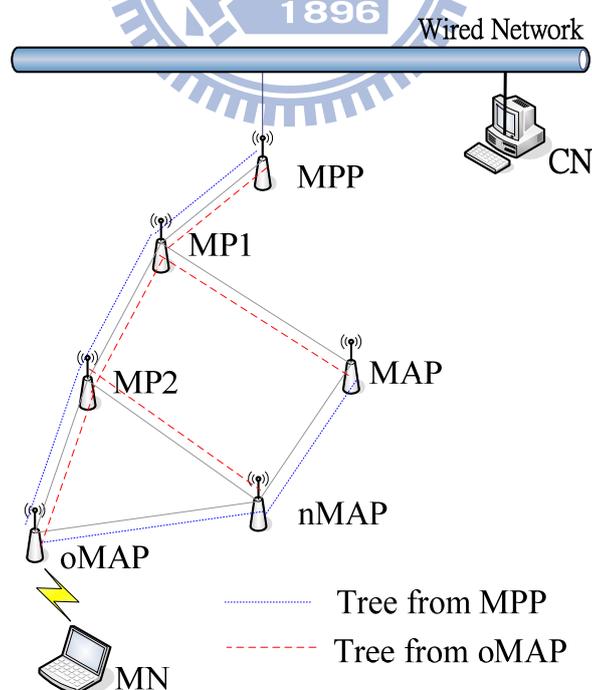


Figure 4-18 無線網狀網路 Multiple Tree 環境之示意圖

4.6 單元總結

本章介紹了此論文提出的 POP 交叉點探索機制，針對其抽象觀念、實際操作的方式都有詳細的解說，此機制可以利用最少量的訊息一次性的將任兩節點間的交叉點探索完成，建立的交叉點表格即表示了節點間的交叉點資訊，且是針對每一個可能的來源網狀擷取點做探索，使此機制可以適用於任何的換手環境，且其交叉點資訊儲存在交叉點本身上，如此除了可以將傳遞的訊息資料量減少，對於換手前、後欲利用交叉點的漫遊機制亦可適用，依據此機制流程之觀念，亦可將本機制延伸為僅探索部份節點交叉點之行為，以適用於較具路徑變動性的網路環境。因此，本論文所提出的交叉點探索機制並不局限於在一開始網路成形時便將所有交叉點資訊建立完畢，這是一個方式，但應因網路較變動的環境時，卻也同樣可以使用本論文所提出的探索機制，僅建立需要的交叉點資訊即可。



第五章 模擬實驗結果與討論

5.1 簡介

本篇論文適用於無線網狀網路環境中，因此若要測試其效能，最有效直接的方法便是佈建許多無線網狀節點，在其上實作本篇論文所提出的交叉點探索機制，觀察其在真實無線網狀環境中的實際運作情形。

然而由於目前市面上所販售的無線擷取點鮮少具有 802.11s 標準的功能，也不易取得其產品原始碼，考慮這些因素，因此我們透過網路模擬器來模擬本篇論文所提出的探索機制。網路模擬器具有將真實網路環境作模擬的動作，可以大大降低佈建真實網路環境所需花費的成本，得以提供學者驗證自己所提出的方法，亦可以讓初學者可以驗證學習到的網路知識，並可提供各式各樣的參數讓使用者設定，這比真正建構一個真實網路環境還要容易的多 [18]。

目前的網路模擬器主要分成兩大類，第一類是透過載入不同程式將硬體裝置模擬成所需的設備。第二類則是純軟體的模擬器，而這一類的軟體又可以分為兩種，一是分析欲測試的網路之效能，注重於理論的分析驗證，如 NS2、OPNet、Qualnet 等，另一種是以模擬實際的網路設備為主要目的，適用於參加驗證檢定、實務訓練，例如 Boson Netsim [19]。

本篇論文選擇以學術研究中頗具參考價值的 NS2 [20]，NS2 最早係來自於 1989 年的 Real Network Simulator，NS 一直以來都在吸收全世界各地研究者的，經過多年的發展之後，NS2 成為在學術研究上最具影響及代表性的網路模擬器。

NS2 是由 C++ 和 OTcl (Objetc Tool Command Language) 做為開發的語言。之所以使用兩種程式設計語言，是因為模擬器有兩方面的事情需要做。一方面為具體協定的模擬和實現，需要高效率的處理位元組 (Byte)、標頭 (Packet Header) 等資訊，能夠應用合適的演算法在大量的資料集合上進行操作。為了達到這個目的，程式內部模組的運行速度 (Run-time speed) 是非常重要的，這種情況下，C++ 語言是非常合適的；另一方面，許多網路的研究工作都圍繞著網路組件的設置和環境參數的改變而進行的，通常都需要在短時間內快速的開發和模擬出所需要的網路環境 (Scenario)，以方便修改以及發現、修復程式中的 Bug。這種情況下，腳本語言 (Script) 便有很大的優勢，Tcl 在 NS2 中主要是用來描述腳本的，簡單的說就是用來描述要模擬的網路環境和參數設定等[21][22]。

5.2 模擬環境

本篇論文所使用的 NS2 版本為 2.33 版，由於網路上 NS2 所提供的無線網狀網路模組目前還尚未成熟，因此我們便在 NS2 路由層 (Routing Layer) 實做無線網狀網路路由機制，以模擬封包在無線網狀網路中傳送的情形，我們實做了無線網狀網路中路由表與代理表的對應關係，以及其位址轉換的機制。而本論文所提出的方法是基於網路內節點其路由表建立好後的無線網狀網路環境去做分群的動作。

本實驗分別在 3×3、4×4、5×5、6×6 的網格網路 (Grid Network) 下進行，觀察本機制與 CRPD，在 MN 變動及其換手次數變動的環境下，所需要花費找尋交叉點的訊息量之比較。第三章有提到，CRPD 為根據每次 MN 的連接與換手都必須觸發找尋交叉點機制，對於網路拓樸與環境較不易變動的無線網狀網路來說，每一次的找尋都會造成網路的負擔，尤其當 MN 數量多與換手次數頻繁的情況下，多餘的找尋是不必要的，因此才有本論文的方法來改進此缺點。使用網格網路的原因是我們想觀察在網路節點數增多的情形下，本論文方法及 CRPD 對交叉點找尋的訊息量變化，而網格網路為一種較普遍的網路佈建法。實驗參數詳細情形如 Table 5-1：

Table 5-1 模擬參數設定值

NS-2 Environment Argument	
Item	Value
NS-2 版本	2.33
PHY 層協定	IEEE 802.11 PHY
MAC 層協定	IEEE 802.11 MAC
Layer 3 Routing 層協定	Mesh proxy-based routing
Mesh node 個數	3×3Grid、4×4Grid、5×5Grid、6×6Grid
MN 個數	1、5、10
Handover 次數	1、5、10、15、20
Handover 模式	鄰近節點亂數 Handover

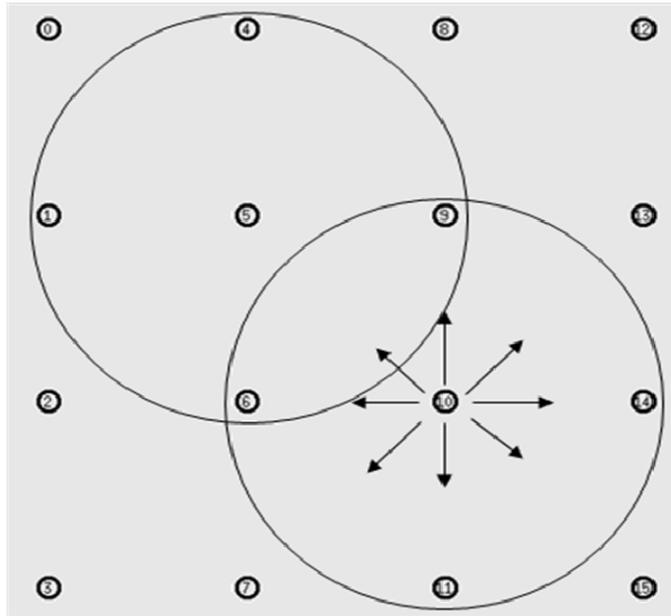


Figure 5-1 模擬環境示意圖

Figure 5-1 為本論文模擬的網路拓樸環境，可以看出這是一個 4×4 的網格網路，圓圈即代表節點訊號範圍，箭頭代表行動裝置可以換手的候選網狀擷取點，我們設計其行動裝置可以連結上四面八方的網狀擷取點，以亂數的方式去選擇一鄰近的候選擷取點，我們根據這樣的行為去計算行動裝置連接上擷取點後，其 CRPD 所需花費的找尋交叉點訊息量，節點 0 為預設之 GW，因為 CRPD 為以 GW 去找尋交叉點。至於本方法則根據每一個節點去做交叉點探索機制，計算其一次性分群所花費之訊息量。

5.3 實驗結果與分析

由 Figure 5-2 可以知道 POP 與 CRPD 的訊息量計算方式。POP 為針對每一個節點做一次性的找尋，而 CRPD 為針對行動裝置每次連接上擷取點的觸發性找尋，因此當換手後的訊息量是以累計的方式往上累加的。

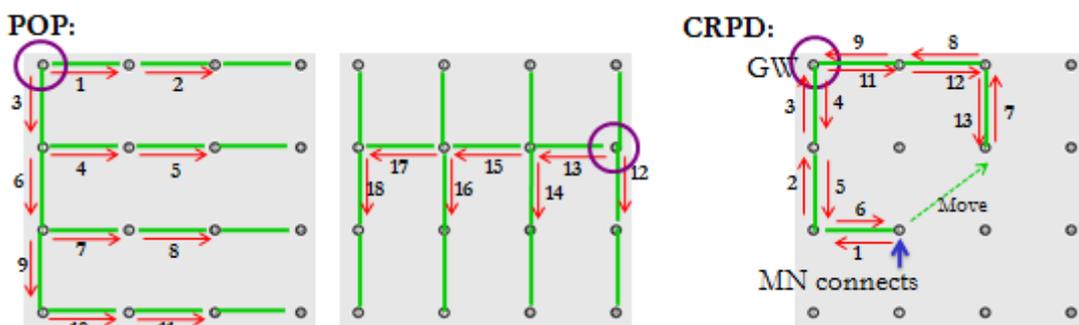


Figure 5-2 POP 與 CRPD 之訊息計算方式示意圖

雖然本論文方法在網路成型時（此為路由表格建立好後）會花費一些訊息去針對每一節點做交叉點探索機制，不過此訊息量是很小的，而且只要網路的路由不變動，就不

需要再觸發交叉點探索機制；反之來說，CRPD 的每一次的找尋交叉點，對網路便都是一種負擔。Figure 5-2 為在 Grid 3×3 的拓樸環境下，POP 與 CRPD 隨著行動裝置與其換手次數的增加，所需要花費的訊息量數。圖中的橫座標代表單一行動裝置的換手次數，縱座標為所花費的找尋交叉點訊息量數，其訊息量數的計算方法為 (訊息×所走的 Hop counts) 。

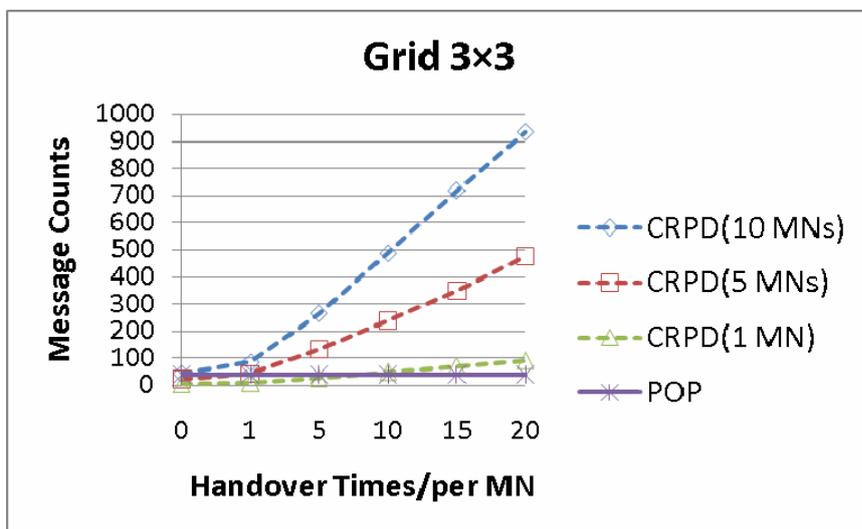


Figure 5-3 Grid 3x3 之找尋交叉點訊息量

Figure 5-2 中可以看到，POP 除了在一開始需花費少量的交叉點探索訊息外，其後不管行動裝置或是其換手次數的增加，訊息量並不會增加，因為 POP 之探索機制是建立在一開始網狀網路路由表建立完成後去執行，並把所有的交叉點資訊建立好，行動裝置及其換手次數並不會再次觸發探索交叉點的程序，因此可以看到 POP 的線會是以平行直線的方式呈現。相較於 POP，CRPD 的訊息量數便成長的很快，因為行動裝置的連接以及每次的換手都會觸發一次找尋交叉點的動作，隨著換手次數的增加，所花費的訊息量數當然也會跟著增加，在僅有 9 個網狀節點的無線網狀網路環境底下時，可以由 Figure 5-2 中觀察只要有 5 個行動裝置，各換手一次後，其所花費的找尋交叉訊息量便會超越 POP 了，即使僅有 1 個行動裝置，在換手 10 次後，其所花費的訊息量亦會超越 POP，而且隨著行動裝置的加入與換手，與 POP 的訊息量差距只會越拉越大。

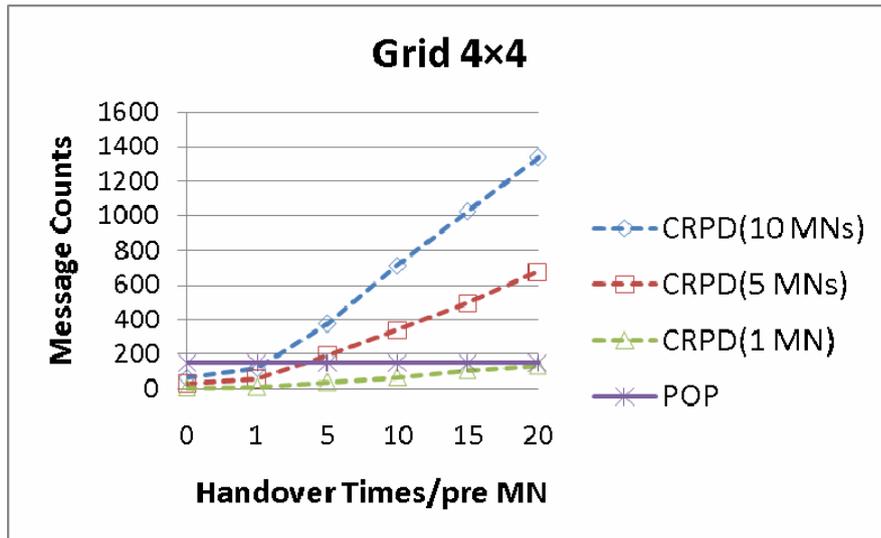


Figure 5-4 Grid 4x4 之找尋交叉點訊息量

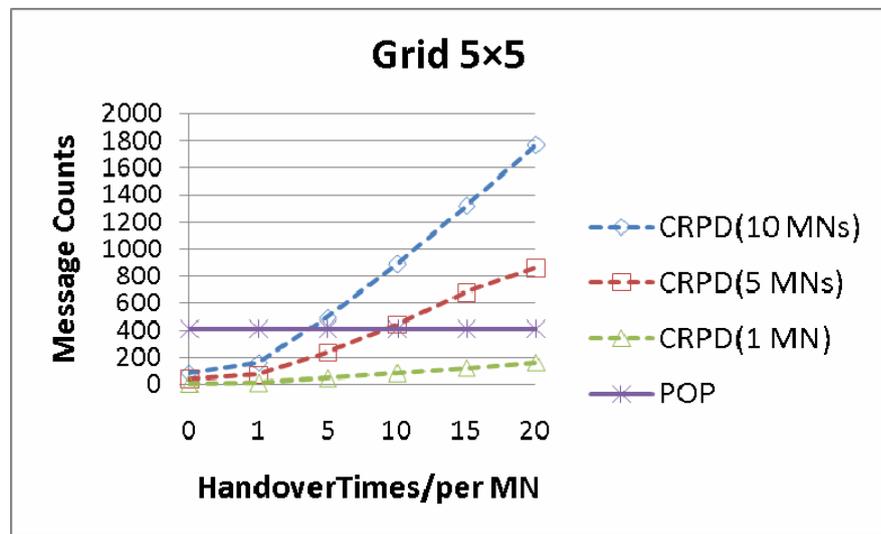


Figure 5-5 Grid 5x5 之找尋交叉點訊息量

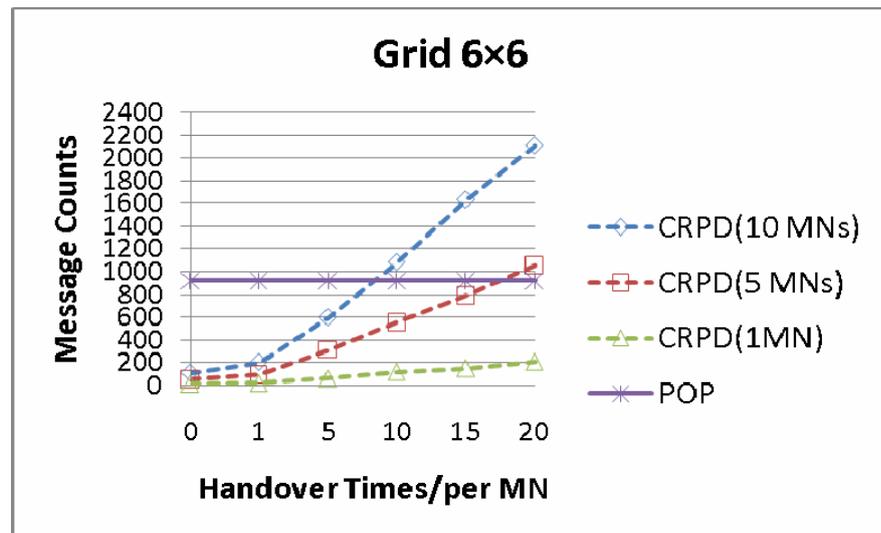


Figure 5-6 Grid 6x6 之找尋交叉點訊息量

Figure 5-3、Figure 5-4、Figure 5-5 分別針對 Grid 4×4、Grid 5×5、Grid 6×6 的網狀網路節點數去觀察 POP 與 CRPD 找尋交叉點所需花費的訊息量數。由於 POP 是以每一個節點去做探索交叉點的動作(如此才能確保任何 MN 與 CN 溝通的情景所利用的交叉點是正確的)，因此所需花費的訊息量數會隨著網狀網路節點數的增加而慢慢增加。Figure 5-3、Figure 5-4、Figure 5-5 顯示 POP 其訊息量數的不同，這是因為網路節點數量改變所至。由圖中可以看到在網狀網路節點數比較小的環境下，行動裝置及其換手次數不多時，CRPD 所花費的訊息量數便會超越 POP；而當網狀網路節點數較大時，行動裝置及其換手次數雖然要比較多時才會超越 POP，不過，只要隨著時間其行動裝置連上與其換手次數的增加，訊息量數很快便會超越 POP 所花費的訊息量數。

接著我們所要觀察的是，隨著網狀網路節點數的增加，POP 所需花費的訊息量數的變化，並且加入 CRPD 當對照組，觀察其方法所需花費的訊息量數受網路節點數的影響有多少，如 Figure 5-6 所示，橫座標為網狀網路的節點數，縱座標同樣為所花費的找尋交叉點訊息量數。

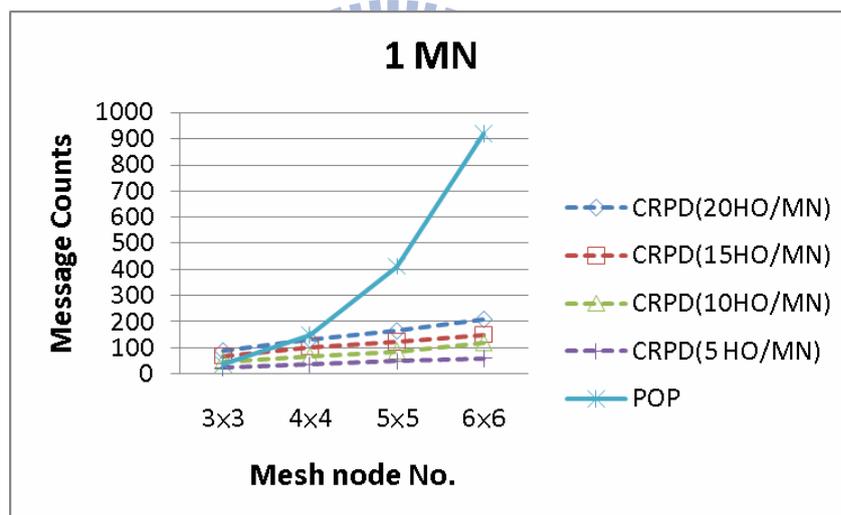


Figure 5-7 1 MN 之找尋交叉點訊息量

Figure 5-6 為在這些網狀網路中有一個行動裝置時，其換手次數對找尋交叉點的訊息量數的影響，分別觀察網狀網路節點為 Grid3×3、Grid4×4、Grid5×5、Grid6×6 訊息量數的變化。由圖中可以看到，POP 會呈現比較大幅度的成長，前面有提到，這是因為 POP 為針對每一個節點的一次性探索，因此對於節點數較多的網路環境時，勢必要花較多的訊息量數去將所有交叉點資訊建立好，因此，在僅有一個行動裝置的時候，當網狀網路節點數多時，本方法的優勢便較不顯著，可以看到節點數在 Grid4×4 以上的網路環境時，即使行動裝置換手次數達 20 次時，CRPD 所花費的找尋交叉點訊息量數仍少於 POP 所需花費的訊息量數。這是因為 CRPD 其所花費的找尋交叉點訊息量數較不受網狀網路節點數的影響，可以看到雖然隨著網狀網路節點數的增加，CRPD 所需花

費的訊息量數也會增加，但是增加的量並不顯著，其主要的影響還是在行動裝置的數量及其換手的次數。接著我們來看當行動裝置增加時 POP 與 CRPD 找尋交叉點所花費的訊息量數隨著網狀網路節點數量增加的變化為何，如 Figure 5-7、Figure 5-8 所示：

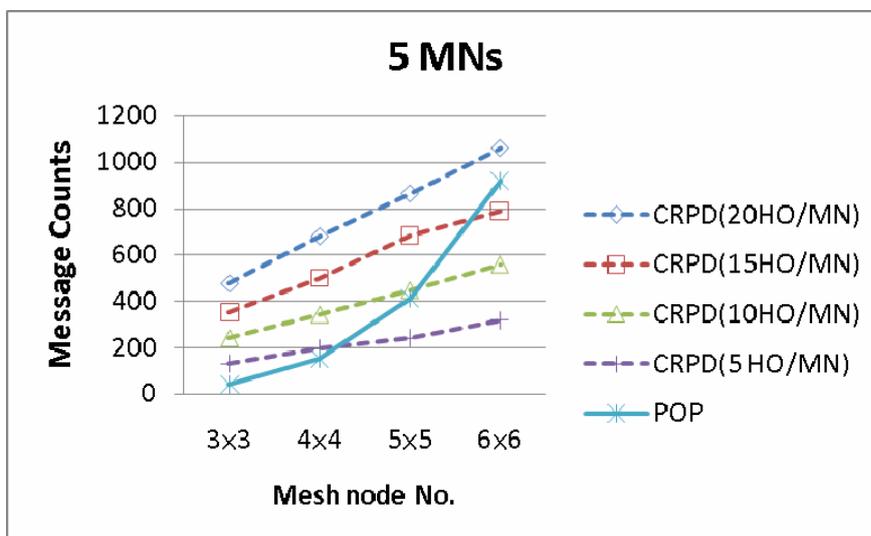


Figure 5-8 5 MN 之找尋交叉點訊息量

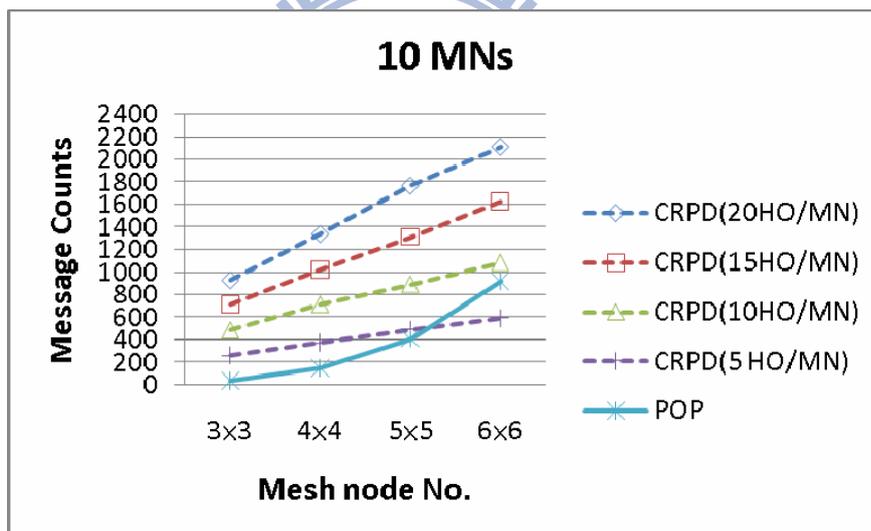


Figure 5-9 10 MN 之找尋交叉點訊息量

由圖中可以看到，POP 其訊息量數的變化不管是網狀網路環境內有 5 個或是 10 個行動裝置其和僅有一個行動裝置時的情形會是一樣的。而 CRPD 其不同於一個行動裝置的情形，所需花費的找尋交叉點訊息量數在 5 個行動裝置時候便可看到有相較於 1 個行動裝置的情形有了很顯著的成長，由 Figure 5-7，在網狀網路節點數少時如 Grid3x3 與 Grid4x4，在換手次數僅有 5 次的情形時，CRPD 所需花費的訊息量數便比 POP 多了不少，即使在 Grid5x5 的中型網路，只要換手次數超過 10 次，其訊息量同樣會超越 POP。而在網狀網路節點數較大的網路環境如 Grid6x6 時，雖然換手次數少時 CRPD 所花費的訊息量不會比 POP 多，但只要換手次數

在二十次其所花費的訊息量數便會大於 POP，而只要再經過一段時間，隨著節點數增加與換手次數的增加，CRPD 的訊息量數便會超越 POP 了。而當行動裝置數量為 10 的情形下，由 Figure 5-8 所示，CRPD 其訊息量會更快便會超越 POP，即使在 Grid6×6 的環境下，當行動裝置僅換手 10 次後，CRPD 所花費的訊息量便會超越 POP 了。

5.4 貢獻

本論文所提出之交叉點探索機制 - POP 適用於無線網狀網路的網路環境，並且對於目的行動裝置與來源行動裝置在同一個網域時所會有的『兩節點交叉點隨著來源行動裝置的位置改變』的情形都可以找到正確的交叉點，如此在要換手時或換手後都可以利用正確的交叉點去增進換手的效益。本方法針對無線網狀網路內的所有節點做交叉點探索的動作，因此可以找到以任何節點為來源點其任兩節點間的交叉點，此為先前的作法所沒有考慮到的，在第三章有提到，先前機制皆僅考慮來源行動裝置在 GW 外的地方，封包傳送到目的行動裝置均經過 GW，因此僅需要找尋以 GW 為來源點其節點間的交叉點即可，但這樣的方式並不適用於目的行動裝置與來源行動裝置在同一網域內時、目的行動裝置移動的情形，本論文所提出的交叉點探索機制改善了這樣的問題。

另外，本機制為針對每個節點一次性分群之交叉點探索機制，藉由一次性分群的概念便得以利用最少量的訊息將以此來源節點其任兩節點間的交叉點資訊建立完成，而這些交叉點資訊便呈現了『兩兩節點間的交叉點為何？』這筆資訊。利用了最少量的訊息建立最多筆的資訊為本機制之目的與優點，資訊建立好後，便不需要每次行動裝置連接上或換手後還需要觸發找尋交叉點機制，以免造成多餘的找尋及頻寬的浪費。而 POP 其交叉點資訊儲存在交叉點本身上，對於換手前或換手後欲要利用交叉點的情景都可以直接地由新舊網狀擷取點沿著來源點的方向得知正確的交叉點為何。

！

5.5 單元總結

由模擬結果可知，POP 其找尋交叉點所花費的訊息量數在一網狀網路環境內並不會受到行動裝置的增加與其換手次數的增加影響，雖然在路由表完成後需針對每個節點花費一次性探索交叉點機制，但其僅需要一次的建立，之後不管行動裝置如何增加亦不管其如何換手，都可以直接利用所建立好之交叉點資訊得知正確的交叉點為何，而不需要再做找尋交叉點的動作，因此找尋交叉點的訊息量數並不會因此

而增加；相較於 POP·CRPD 花費的找尋交叉點訊息量便會隨著行動裝置的增加與其換手次數的增加而成長，且其訊息量的成長幅度很大。

影響 POP 訊息量數為網狀網路節點數量，其針對每一節點的一次性交叉點探索會隨著網狀網路節點的數量增加而要花費較多的訊息；相較於 POP·CRPD 所需花費的訊息量數所受網狀網路節點數量的影響就叫不顯著，雖然還是會有成長，因為行動裝置離 GW 的距離有可能隨著節點數量多而加長，因此所需花費的訊息量數仍會增多，但主要影響 CRPD 還是以行動裝置的多寡及其換手次數的多寡為主。雖然在行動裝置少時，POP 的效果沒那麼顯著，但只要隨著時間的進行，當行動裝置數量多以及其換手次數多時，CRPD 所花費的訊息量數便會超越 POP，並且此訊息量數成長仍會持續增加下去。



第六章 結論與未來工作

6.1 結論

POP 之交叉點探索機制為利用以每一節點其到所有節點的路徑為一 Tree，依據其分枝分為多個 Sub-trees，而 Tree 之 Root 即為這多個 Sub-trees 的交叉點之概念，利用此概念所衍生的一次性分群之交叉點探索機制。並利用這些分群後的 Sub-trees 建立交叉點資訊，此交叉點表格儲存在交叉點本身上，此資訊即代表了自己為當以某個來源節點時，分別為哪兩個網狀節點的交叉點；亦即，當有某個來源節點送出封包時，此節點即為目的行動裝置所連接上的節點以及此目的行動裝置接下來會連接上的節點的交叉點。有了此交叉點表格，在無接縫漫遊機制中，無論是換手前、換手後都可以查找到最正確的交叉點，以利用交叉點增進換手的效益。

在 POP 的交叉點探索機制架構下，不論來源行動裝置與目的行動裝置是否在同一網域底下時，欲使用交叉點增進換手的效益時，都可以利用到最正確的交叉點，選擇正確的交叉點才能有效率的增加換手的效益。而當交叉點資訊建立好之後，只要網路拓樸不更動，便不再需要花費找尋的訊息，即不論行動裝置的連接與換手，都不需要再一次找尋，對於訊息量的花費是很少的。POP 具有以下特色：

針對所有節點的交叉點探索 (Per-proxy-oriented)：由於先前的研究僅考慮到來源行動裝置與目的行動裝置在不同網域時，目的行動裝置換手的情形，因此找到的交叉點便僅會有一個，但當來源行動裝置與目的行動裝置在同一個網域時，其交叉點是會隨著來源節點的位置而有所改變的，因此本機制針對每一節點之交叉點探索便可以建立所有可能交叉點資訊並可利用最正確的交叉點達到換手效率的增加。

每個節點的一次性分群之交叉點探索 (Partition-based)：以往的研究總是在行動裝置連接上以及換手的時候都必須做一次交叉點找尋的動作，如此當網路內行動裝置一多時，網路便會充斥著這些交叉點找尋的訊息，並且有可能要找的都是重複的交叉點，因此，在網路拓樸較不易變動的網路環境下時，透過這種一次性分群之探索，便可以將以此節點之任兩節點間的交叉點資訊建立好。而當有行動裝置連上或是換手後，均不需要再做找尋的動作，進而減少不需要的訊息量。

交叉點表格分散式化：由於本機制之分群方式，使得其交叉點資訊儲存在交叉點本身上。先前研究其交叉點無法適用於所有欲利用交叉點的環境，因為其交叉點

資訊僅行動裝置連接上的擷取點知道，因此不適用於換手後欲利用交叉點的環境，因為還需要重新找尋交叉點才有資訊。而本方法其交叉點資訊分散式化使得任何欲利用交叉點的情景都可以使用。

6.2 未來工作

由於目前市面上的網狀網路產品不多，所以本機制是以模擬的方式查看其效果，因此希望之後可以實做在實際的網狀網路產品上以查看本機制效益，並且測試利用交叉點對換手的效益改善有多少。

另外，與以往研究不同的地方，在利用交叉點的時候，先前研究 CRPD 在節點上查找表格便可直接得知交叉點為何，因此通知交叉點的訊息是以直接的方式將訊息直接傳遞給交叉點，而本論文其通知交叉點的方式為沿著到來源節點的路徑傳遞訊息，路徑上每個節點檢查自己是否為交叉點。因此這兩種通知交叉點的方式其時間上或許會有些微的差異，在模擬環境上，我們無法得知封包在一個節點上處理所需花費的時間，因此對這項數據無法測量，不過可以知道這會跟每個節點其處理的能力有關，雖然我們認為這兩種方式所花費的時間差異量不大，但在有市面產品的實際運作下，還是可以將這時間量算出。

最後針對本機制的缺點做修改，由於每次路徑的改變，本機制必需再花費一次找尋的機制，如此，在網路環境變動大的情形底下時，其所花費的訊息量還是會很可觀，雖然本論文有提供一套延伸機制，但若能僅針對變更的路徑做部份的重新找尋，便可以使本機制不受路徑變動大的情況下的影響，亦可以更增強本機制。

Reference

- [1] IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," *IEEE Standard 802.11*, 1999
- [2] Wikipedia – "LTE," Available from:
http://en.wikipedia.org/wiki/3GPP_Long_Term_Evolution
- [3] IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," *IEEE P802.11sTM/D2.03*, 2008
- [4] 802.11s 無線網狀網路介紹:
http://netlab18.cis.nctu.edu.tw/Short_course07/
- [5] Wikipedia – "Address Resolution Protocol," Available from:
http://en.wikipedia.org/wiki/Address_Resolution_Protocol
- [6] D. H. Cuong, D. Guhat, J. K. Choi, J. S. Park, and H. J. Kim, "An Enhanced Fast Handover Scheme with Crossover Router Pre-Discovery support in Mobile IPv6," *Optical Internet and Next Generation Network, 2006. COIN-NGNCON 2006. The Joint International Conference on*, vol., no., pp.6-11, 9-13 July 2006
- [7] D. H. Cuong, J. K. Choi, D. Guhat, J. S. Park, H. J. Kim and J. H. Hahm, "Proactive Crossover Router Discovery with FMIPv6," *Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, vol., no., pp. 104-107, Oct. 2006
- [8] A. T. Cappbell, J. Gomez, S. Kim, A. Valk, C. Y. Wan, and Z. R. Turnyi, "Design, implementation, and evaluation of cellular IP," *IEEE Personal Communication Magazine*, vol. 7, no., pp. 42-49, Aug. 2000
- [9] R. V. Ferré and J. P. Aspás, "Improving Fast handover for mobile IPv6: Optimal crossover discovery using Geopaging routing tables," *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 5, no., pp. 2994-2998, Jun. 2005
- [10] D. Tandjaoui, N. Badache, H. Bettahar, A. Bouabdallah, and H. Seba, "Towards a smooth handoff for TCP and real time applications in wireless network," *International Network Conference*, vol., no., pp., Jul. 2002
- [11] K. A. Kim, J. D. Kim, C. K. Kim, and J. Y. Park, "An Enhanced Handoff Mechanism for Cellular IP," *Lecture Notes in Computer Science (LNCS)*, Springer, vol. 3090, no., pp. 164-173, Aug. 2004
- [12] J. Y. Hu and C. C. Yang, "Enhancement of Cellular IP Routing by Redirection at Crossover Base Stations," *Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, vol. 4, no., pp. 2551-2555, Sept. 2005
- [13] 連結狀態路由或距離向量路由:
http://www.tkwu.net/index.php?option=com_content&task=view&id=61&Itemid=42

- [14] RFC4068 - Fast Handovers for Mobile IPv6:
<http://www.faqs.org/rfcs/rfc4068.html>
- [15] “Cellular IP,” *Internet draft, Internet Engineering Task Force, draft-ietf-mobileip-cellularip-00*, Dec. 1999
- [16] Gustafsson et al., “Mobile IPv4 Regional Registration,” *Internet draft, Internet Engineering Task Force, draft-ietf-mip4-reg-tunnel-00*, Nov. 2004
- [17] R. Vidal and J. Paradells, “Geopaging: a novel multicast approach to IP paging delivery,” *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 3, no., pp. 1896-1900 Vol. 3, Sept. 2004
- [18] NS2 簡介與安裝:
<http://blog.pixnet.net/NS2lab/post/5645626>
- [19] 利用網路模擬器提升網路管理的自我效能:
<http://library.ksvs.kh.edu.tw/article/disquisition/2005103114512927.pdf>
- [20] The Network Simulator - ns-2:
<http://www.isi.edu/nsnam/ns/>
- [21] 網路類比環境 NS-2 模擬軟體簡介:
http://net.zdnet.com.cn/network_security_zone/2007/0918/511555.shtml
- [22] TCL 基本使用:
<http://vaplab.ee.ncu.edu.tw/chinese/pcchang/course2004b/comsp/network/3.4.htm>