

## NEMO 環境中的群播路由協定

李龍盛 國立嘉義大學資訊工程系

謝鎧仲 國立嘉義大學資訊工程系

### 摘 要

本文提出一個應用於 NEMO(NETwork MObility)環境中的 Multicast 路由協議，由於以往運作於有線網路的 Multicast 路由協議中並沒有考慮到群播來源端(Multicast Source)移動的問題，若將發展於一般有線網路及無線網路的 Multicast 路由協議應用於 NEMO 環境中，則 Multicast Source 移動後所進行的 Multicast Tree 重新建構(Reconstruct)將耗費相當的成本。本文提出應用於 NEMO 環境中的 Multicast 路由協議，該協議以 MLD Protocol 為基礎並進行修改，使其能夠順利應用於 NEMO 環境中並支援建立及維護 Multicast Tree，在網路中群組類別較多時能夠節省相當多維護 Multicast Tree 所需要的封包，並能夠讓 Multicast Source 移動並得到新的 CoA(Care-of-Address)後以較少的成本進行 Multicast Tree 的重建，以減少不需浪費的成本。

**關鍵詞：**群播、群播路由協議、網路移動性

## **Multicast Routing Protocol in NEMO**

Long-Sheng Li, Dept. of Computer Science and Information Engineering, National Chiayi  
University

Kai-Chung Xie, Dept. of Computer Science and Information Engineering, National Chiayi  
University

### **ABSTRACT**

In this paper we propose a Multicast routing protocol used in NEMO(NETwork MObility) environment. Due to previously Multicast routing protocols operate in wired networks do not consider the movement of Multicast source. If we use the original wire or wireless Multicast routing protocol in NEMO environment, the cost of reconstruct Multicast tree after Multicast source movement will be very high. This paper propose a Multicast routing protocol which is more suitable for NEMO environment. The protocol is based on MLD protocol and modified it, so that can get more suitable applied to NEMO environment and support the construct and maintain of Multicast Tree. When the group class of networks is getting more, it can save the packets which is Multicast Tree maintenance needs. When Multicast source move around and get the new Care-of-Address, we can reduce the cost of reconstruct Multicast tree takes.

**Keyword** : Multicast 、 NEMO 、 Network Mobility

## 1. 導論

在近幾年的資訊發展下，行動計算已經幾乎是每個人在生活中所不可或缺的一部份，在無線網路的成熟發展，以及智慧型商品的普及之下，人們都可以無時無刻的連接到網路進行收發電子郵件、玩線上遊戲以及網路購物等。為了使人們的生活更為便利，許多學者紛紛提出了各種不同的網路架構並進行改良，以因應不同的需求。NEMO (NEtwork MObility) [10] 便是其中一種網路環境，這個環境可以讓人們在步行、車上甚至於在飛機上順利平穩的使用網路。例如在搭公車的時候想要連接網路觀看影片時，只要連接公車上的 MR(Mobile Router)並進行註冊等協議操作，即可連接至 Internet 並觀賞任何想看的影片。而在目前網路中，來源端傳送資料封包主要是以點對點(Unicast)的方式傳送至客戶端。假設有眾多的客戶端同時向來源端要求同一份資料，來源端必須將這份資料分別發送給眾多的客戶端，若是許多客戶端同時在同一個子網路下接收訊息封包，則可能會造成頻寬雍塞的問題，為了改善這一個問題，可以使用廣播式(Broadcast)的資料發送，來源端只要發送一個封包，即可以讓所有的客戶端都接收到這份資料。但是以這種方式進行封包發送，會造成不需要接收到這份資料的用戶也接收到相同的資料並丟棄，亦無法有效的利用網路頻寬。因此學者們發展出群播式(Multicast)的通訊協定，透過群播式的通訊協定，來源端可以只發送一次資料封包，便可以將此資料發送至眾多指定的客戶端。在現今群播式發送封包的應用已經越來越普及，舉凡視訊會議、通訊軟體、即時影音傳送等皆可以看見 Multicast 的應用。

但在 NEMO 的網路環境中，Multicast 的實施存在相當大的問題，在以往一般的環境當中，影音串流都是從遠端的伺服器以 Multicast 協定傳送資料至每個使用者，在有線網路方面 Multicast 已經有 DVMRP [3]、MOSPF [5]、PIM [1] [2] 等發展成熟的通訊協議、而在無線網路方面也有相當多的協議被學者們提出。但眾多的協議之中，幾乎沒有提供在 NEMO 環境中來源端進行移動時所需要的解決方式。在 NEMO 的環境中，Mobile Node(MN)可能會從一個 Mobile Router (MR)移動到另一個並且獲得一個新的 Care-of-Address(CoA)以繼續進行與網路的通訊。而假設今天 Multicast 的來源端為一個 MN，在發送 Multicast Packet 給群組成員時移動並連接至另一個 MR 因而獲得新的 CoA，此時要將封包發送給群組成員時就會產生很大的問題。一個主要問題是，假設今天 MN 以來源群播樹(Source Based Tree)發送 Multicast Packet 給群組成員時，移動後該如何重建 Multicast Tree 以順利將 Multicast Packet 傳送給群組成員。在 NEMO 網路環境中能夠順利將 Multicast Packet 傳送群組成員，這一點是最主要必須克服的。

在第二章節主要提出 NEMO 以及一些無線網路群播協議的介紹以及運作過程，第三章節則是提出本篇文章於上述問題的解決方法，第四章節則對本篇文章的方法做初步的評估，第五章則是本篇文章的討論與總結。

## 2. 相關工作

### 2.1. NEMO Basic Support Protocol(NEMO BS)

NEMO Basic Support Protocol 是由互聯網工程任務組(IETF)提出用於NEMO環境的基本運作協議，該協議運作如圖 1 所示，若 Correspondent node(CN)要傳送封包給位於巢狀網路中行動路由器(MR3)所管理的行動節點(MN3)時，首先會將封包送往 MR3 的 Home Agent(MR3\_HA)，待封包傳送至 MR3\_HA 後，HA 會查詢 Binding Cache 中的資料，得知 MR3 目前被 MR2 所管理，所以又需要將封包送往 MR2\_HA。等到封包送至 MR2\_HA 且查詢 Binding Cache 後，又得知 MR2 目前為 MR1 所管理，所以需要將封包送至 MR1\_HA。封包送至 MR1\_HA 後，MR1\_HA 得知目的端為自己所管理的節點，所以將封包透過 Tunnel 將封包傳送至目前管理 MR1 的 Access Router(AR)，AR 收到封包後知道目的節點為自己所管理的，所以將封包送至 MR1，MR1 收到後會將封包轉送至 MR2，MR2 會在將封包轉送至 MR3，MR3 再將封包轉送至 MN3。

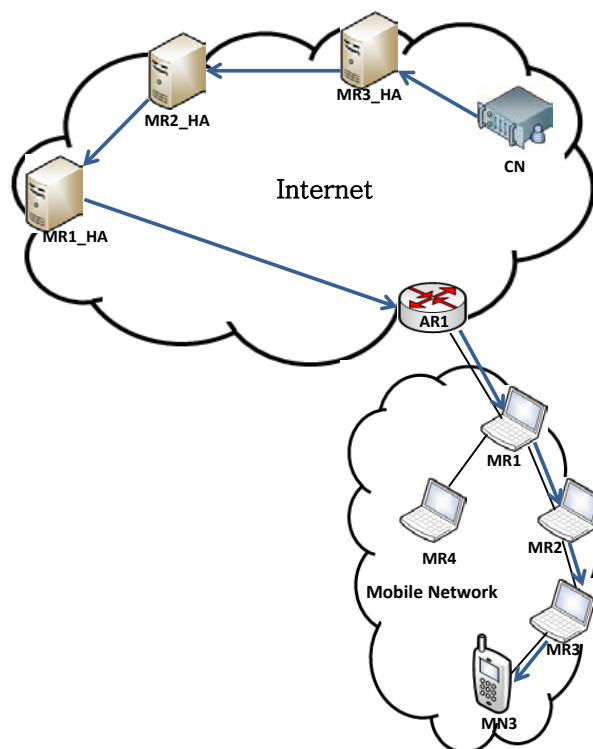


圖 1 NEMO Basic Support Protocol

在 NEMO BS 中有個很重要的問題，名為 Pinball Routing Problem，如圖 1 所示，若 CN 要將傳送封包至 MN3 則先必須經過 MR1\_HA、MR2\_HA 與 MR3\_HA，若 MN 位於層數多的巢狀式網路時，則經過 HA 所產生的花費是非常可觀的。

## 2.2. Remote Subscription

Remote Subscription [4] 是 mobile IP 中一種容易實施 Multicast 的架構，在這個架構中 MN 直接利用 CoA 加入 Multicast Tree。當 MN 加入群組並接收 Multicast 封包時移動並得到新的 CoA，若欲繼續接收該群組資料時，則只需用新的 CoA 加入原來的 Multicast Tree 即可。這個架構的優點在於方便實行，且不需要像 Home Subscription 使用 tunnel 傳送資料，且能夠較快接收到 Multicast 封包。而缺點是造成較多封包遺失。

### 2.3. Multicast Listener Discovery Protocol(MLD)

MLD [7] 是一種應用在 IPv6 Multicast Router 與 Host 之間的協議，最早版本紀錄於 RFC2710，主要用於讓 Host 與 Multicast Router 交換群組的成員狀態資訊。MLD 所使用的訊息是屬於 ICMPv6 的封包，假設一個網段內同時有好幾個 IPv6 Router，則這些 Router 會自動選出一個負責調查群組的 Querier Router。Querier Router 會定時發送 MLD Query Message 詢問網段下的 Host 欲加入的群組。當 Host 收到 Query Message 後會依照欲加入的群組回送 Report Message。或一個網段中有多台主機欲加入同一個群組，則可以利用成員關係報告抑制機制減少相同群組的 Report Message 發送。若有主機欲離開群組，則發送 Done Message 告知 Querier Router，當 Querier Router 收到 Done Message 後則會發送 Multicast-Address-Specific Query Message 進行特定群組的詢問，以得知此網段中是否有其他主機想要繼續接收該群組的 Multicast Packet。若一段時間後沒有任何主機回應則表示已經沒有主機需要此 Multicast Group 的資料。

### 3. 路由演算法

由於 Multicast 有很大的機會被使用於傳送即時影音、即時通訊以及視訊會議等等。因此本文所提出在 NEMO 環境的 Multicast 演算法，是以氾濫的概念進行來源分散樹的建構，並且將 MLD 中 Query、Report 與 Done 等交換成員狀態的方法應用在 Multicast Tree 的維護，並修改與新增功能以能夠順利在 NEMO 環境中執行，在來源端移動後進行 Multicast 的 reconstruct 機制時能夠減少網路的負擔。

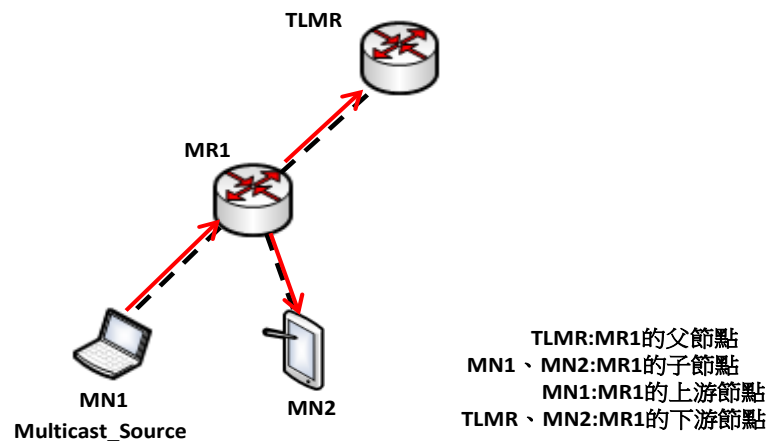


圖 2 易混淆詞彙標示說明

在介紹路由演算法之前，必須先詳細定義在演算法的介紹中會出現的一些易混淆詞彙。如圖 2 所示，因為在 NEMO 環境中 MR1 是經由 TLMR 得到 CoA，所以 TLMR 稱為 MR1 的父節點；而 MN1 以及 MN2 是從 MR1 獲得 CoA 以能夠與網路進行通訊，所以 MN1 及 MN2 為 MR1 的子節點。而圖 3 中 MN1 為 Multicast Source 並向 MR1 進行 Multicast Flow 的發送，MR1 再將 Multicast Flow 往 TLMR 和 MN2 進行發送，所以 MN1 稱為 MR1 的上游節點，而 TLMR 及 MN2 稱為 MR1 的下游節點。

在演算法中規定幾個具有特定功能的封包，用來進行 Multicast Tree 的構建、維護以及當來源端移動時進行 Multicast Tree 的重建所需。以下介紹封包的名稱與用途：

1. Construct：用於進行 Multicast Tree 的建立。
2. Query：用於查詢下游節點所加入或進行轉發的群組種類，或是查詢特定群組的成員。
3. Report：節點回報所加入的群組或需進行轉發的群組。
4. Done：成員欲停止接收群組的 Multicast Packet。
5. Init：當 MR 帶著 Multicast Source 移動後告知 Source 進行 Multicast Tree 的重建。
6. Reconstruct：Source 移動後進行 Multicast Tree 的重建。

在有線網路的 MLD 協議中有成員關係報告抑制的機制，而在本文提出的協議中則繼承了這個功能，因為以原本的封包格式在 NEMO 環境中進行此機制會造成問題，所以 Report 訊息無法使用原本的 ICMP 封包格式，因此新的 Report 訊息中目的位址一樣為 Group ID，而封包中必須新增一個欄位以紀錄 Upstream node 的位址，以避免發生聽到不同上游的鄰居所發送的 Report 訊息而抑制訊息的情況。

表 1 Multicast Table 紀錄欄位

GroupID	Upstream node_address	S bit
group1	MR1_CoA	1
group2	MR2_CoA	1

在以下所提出的演算法中，我們規定 MR 與 TLMR 中額外新增的 Multicast Table 紀錄欄位為(Group ID, Upstream node address,S bit)，Group ID 為 IPv6 所規定的群組位址，Upstream node address 表示收到的 Multicast packet 的上游節點位址，S bit 表示是否需要將 Multicast Packet 進行轉發，如表一所示。當 MR 與 AR 接收到 Multicast Packet，會查詢 Routing Table 中的 S bit 以判斷是否需要轉發至下游節點，當 S bit 為 1 表示需要進行封包轉發；S bit 為 0 表示不需進行轉發。由於 NEMO 環境中節點會動態的變化，因此 Multicast Table 也需要定期進行更新。在演算法中 MR 與 TLMR 會定期對鄰近節點發送 Query 訊息，當下游節點接收到此訊息後，會回送 Report 訊息表示仍須進行轉發或接收群組的 Multicast Packet，若一段時間後 MR 與 TLMR 未接收到某一群組的 Report 訊息，會將該群組在 Multicast Table 中所記錄的 S bit 更改為 0，並且在接收到該群組的 Multicast Packet 時不進行轉發。

### 3.1. 群播註冊

在一開始，在 NEMO 環境中的 Multicast Source 會以廣播的方式發送 Construct Packet 進行 Multicast Tree 的構建，當 MR 或 TLMR 接收到 Construct Packet 後會將該群組的群組

位址與收到 Construct Packet 的上游節點記錄在 Multicast Table 中，並將 S bit 設為 1，因為 NEMO 環境為樹狀拓樸，因此路由路徑並不存在迴圈，Construct Packet 的上游節點亦為 Multicast Packet 的上游節點。將 S bit 設為 1 為表示預設下游存在成員。紀錄後會將 Construct Packet 發送至除了上游節點外的其他鄰近節點。鄰近的 MR 收到 Construct Packet 後會進行相同步驟直到封包廣播至整個子網。當 AR 接收到 Construct Packet 後，AR 則利用有線網路的 Multicast 路由協議(EX:DVMRP、PIM-DM)等進行有線網路的 Multicast Tree 構建。在無線網路中的 Multicast Source 發送 Multicast Packet 到 AR 時，AR 則能夠將 Multicast Packet 轉發至有線網路以提供資料給群組成員。在本文章所提出的協議中，規定當成 Multicast Gateway 的 AR 擁有與在無線網路中的 Multicast Source 發送相同 Multicast Message 的能力，在有線網路的 Multicast 路由協議將 Multicast Packet 轉發至 AR 後，AR 則當成 Multicast Source 進行在 NEMO 環境中進行群播註冊。

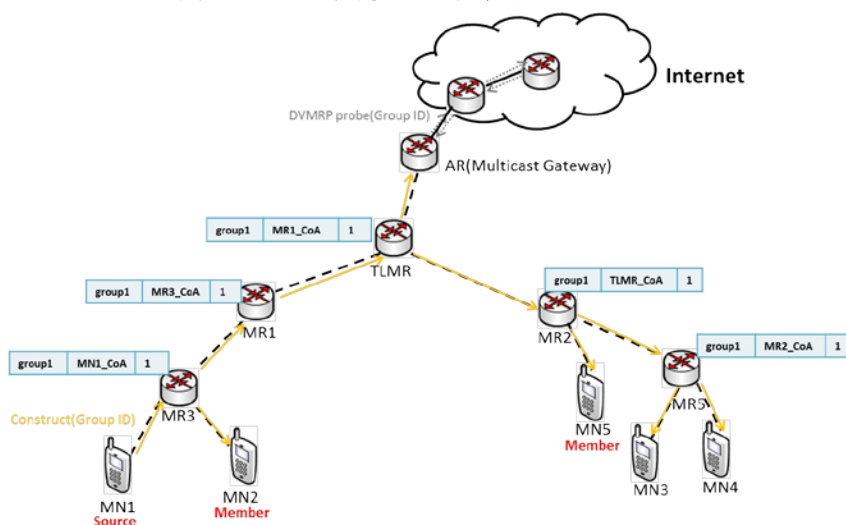


圖 3 方法的群播註冊

以圖 3 為例，Construct Packet 自 Multicast Source 的 MN1 開始進行發送，當 MR3 收到 Construct Packet，則在 Multicast Table 中紀錄 MN1 的 CoA 的上游節點，並將 S bit 設為 1，接著將 Construct Packet 轉發至除了上游節點外的其他鄰近節點。當 MR3 接收到從 MR1 發送的 Construct Packet，亦將 MR1 的 CoA 記錄在 Multicast Table，S bit 設為 1，並將 Multicast Packet 轉發至除了上游節點外的其他鄰近節點。Construct Packet 會持續被轉發至整個子網以尋找群組成員。在 AR 接收到 Construct Packet，則以自己當成 Multicast Source 進行有線網路的 Multicast Tree 構建，當 Multicast Packet 轉發至 AR，則 AR 再將 Multicast Packet 轉發至有線網路，並透過有線網路的 Multicast 路由協議將資料轉發到群組成員

每個 MR 每隔一段時間會發送 Quer 訊息詢問鄰近節點所加入的群組或是需要轉發的群組封包類別，當 MN 接收到上游 MR 所發送的 Quer 訊息，則依照所加入的群組分別發送 Report 訊息告知上游 MR；若是 MR 接收到上游 MR 所發送的 Quer 訊息，則進行 Multicast Table 的查詢，若是 Upstream node address 的欄位所記錄的位址與上游 MR 的位址相同且 S bit 為 1，則發送 Report 訊息告知上游 MR。而此處也應用了 MLD 的成員關係

報告抑制的機制，因為 Report 訊息的目的位址為 Group ID，所以在發送 Report 訊息前聆聽到周圍的其他 MN 或 MR 已發送相同 Group ID 與 Upstream node address 的 Report 訊息，則停止發送相同的 Report 訊息以減少路由負擔。因為路由器只需知道下游是否有該群組成員，所以一個群組只需要從下游節點接收到一個 Report 訊息即可。當 MR 接收到 Report 訊息，則令 Multicast Table 中該群組的 S bit 為 1，若無該群組資料則新增在群播表中，父節點設為上游節點，S bit 設為 1。一段時間後 MR 會將 Multicast Table 中未接收到 Report 訊息的群組的 S bit 設為 0，表示下游目前無該群組的成員。經過 Query 和 Report 的交換成員資訊 MR 可以判斷是否需要進行 Multicast Packet 的轉發，而有修剪 Multicast Tree 的效果。在 Multicast Source 發送 Construct Packet 後的一段時間即開始發送 Multicast Packet，當 MR 接收到 Multicast Packet，會依照群播表中該群組的 S bit 而判斷是否該進行 Multicast Packet 的轉發。

### 3.2. 成員加入

若新成員欲加入群組，則不需要等待上游 MR 發送 Query 訊息就先發送 Report 訊息告知上游的 MR，當 MR 收到 Query 訊息後，即對 Multicast Table 進行查詢，若 Multicast Table 中未有此群組的資料，則在 Multicast Table 新增群組資料，Upstream node address 欄位設為父節點，將 S bit 設為 1，並繼續向上游 MR 發送該群組的 Report 訊息。若 Multicast Table 有該群組資料且 S bit 為 1，則表示已加入 Multicast Tree 而不再發送 Report 訊息並進行 Multicast Packet 的轉發；若 S bit 為 0，則將 S bit 變更為 1，並繼續向上游發送 Report 訊息直到加入 Multicast Tree。

### 3.3. 成員離開

若成員欲離開群組，則發送 Done 訊息告知上游 MR，當 MR 收到 Done 訊息後則會發送目的位址為 Group ID 的 Query 訊息向下游節點詢問是否還存在該群組的成員或需要進行該群組的資料轉發。若有收到該群組的 Report 訊息則 S bit 不變，若 S 一段時間未收到 Report 訊息則將 S bit 變更為 0，並停止進行該群組的 Multicast Packet 的轉發，在上游 MR 發送 Query 訊息詢問時，也不再發送該群組的 Report 訊息。

### 3.4. MR 的移動

在 NEMO 環境中 MR 是會進行移動的，當 MR 從一個 MR 移動到另一個 MR 底下，因為 Multicast Table 還記錄原來的上游節點，所以需要視情況進行更新以避免資料錯誤。

當 MR 移動到新的 MR 底下並獲得新的 CoA 後，則必須對 Multicast Table 的內容進行更改，若 MR 發現群組的上游節點資料為自己的子節點，表示自己是帶著 Multicast Source 進行移動的，此時則必須進行 Multicast Tree 的重建；若 MR 發現群組的上游節點位址不是自己的子節點，表示上游節點位址是前一個父節點的 CoA。此時則將新的父節點位址覆蓋舊的上游節點位址。若 S bit 為 1 則以 Query 與 Report 的方式重新加入 Multicast Tree。MR 在移動並獲得新的 CoA 後會對 Multicast Table 中每個群組都進行判斷及更新。

### 3.5. 來源端的移動

Multicast Source 在 NEMO 環境中是會進行移動的，所以當 Multicast Source 移動後必須進行 Multicast Tree 的重建。當 Multicast Source 移動並獲得新的 CoA，則會發送



Reconstruct Packet 進行群播樹的重建。當 MR 接收到 Reconstruct Packet，則進行 Multicast Table 的查詢，並依以下情況進行 Multicast Table 的更新：

1. 若 Multicast Table 無此群組資料，則新增該群組的資料，S bit 預設為 1。
2. 若 Multicast Table 中紀錄的上游節點資料與收到的 Reconstruct Packet 的上游節點位址不同，則將該群組的上游節點資料覆蓋成 Reconstruct Packet 的上游節點位址，並將 S bit 變更為 1。
3. 若 Multicast Table 中紀錄的上游節點資料與收到的 Reconstruct Packet 的上游節點位址相同，表示封包流向不變而不須做任何更改。

在 Multicast Table 更新後 MR 依照 S bit 判斷是否需進行轉發，若 S bit 為 1 則將 Reconstruct Packet 轉發至除了上游節點外的其他鄰近節點。依照這個規則可以減少重建 Multicast Tree 的花費，因為在某些原本已經進行確定下游節點沒有成員的 MR，在 Multicast Tree 重建時就不須再進行建立的步驟。

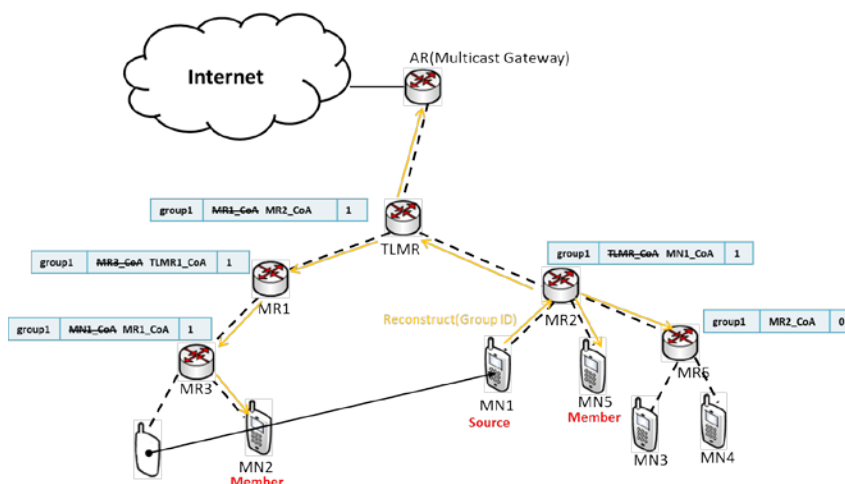


圖 4 來源端移動

以圖 4 為例，作為 Multicast Source 的 MN1 從 MR3 底下移動到 MR2 底下並獲得 CoA 後，需要進行 Multicast Tree 的重建。則 MN1 發送 Reconstruct Packet 給父節點 MR2 以重建 Multicast Tree，當 MR2 接收到 Reconstruct Packet，則檢查 Multicast Table 的資料，搜尋後得知 Multicast Table 中存在該筆資料且紀錄的上游節點資料(TLMR\_CoA)與 Reconstruct Packet 的上游節點(MN1\_CoA)不同，則 MR2 將 TLMR\_CoA 覆蓋成 MN1\_CoA，將 S bit 設為 1，並將 Reconstruct Packet 轉發至除了上游節點外的其他鄰近節點。當 MR6 接收到 Reconstruct Packet，檢查 Multicast Table 後得知上游節點資料與 Reconstruct Packet 的上游節點相同，且 S bit 為 0，則不需將 Reconstruct Packet 轉發。其他 MR 則以相同的規則進行 Multicast Tree 的重建。

### 3.6. MR 攜帶來源端的移動

在 NEMO 環境中 Multicast Source 可以利用更換 CoA 來得知已經移動並需要重建 Multicast Tree。但若是 MR 帶著 Multicast Source 進行移動，Multicast Source 並不會更

換 CoA，而不知道自己需要進行 Multicast Tree 的重建。所以更換 CoA 的 MR 必須發出訊息告知 Multicast Source 以進行 Multicast Tree 的重建。

在章節 3.4.提到當 MR 發現群組的上游節點資料為自己的子節點，則必須進行 Multicast Tree 的重建。此時 MR 會發出紀錄 Group ID 的 Init 訊號給 Multicast Table 所記錄的上游節點。當節點收到 Init 訊號，若自己是該群組的 Multicast Source，則開始發送 Reconstruct Packet 進行 Multicast Tree 的重建；若不是群組的 Multicast Source，則依照 Multicast Table 所記錄的上游節點位址進行發送，Init 最終會被轉發至 Multicast Source 並進行 Multicast Tree 的重建。

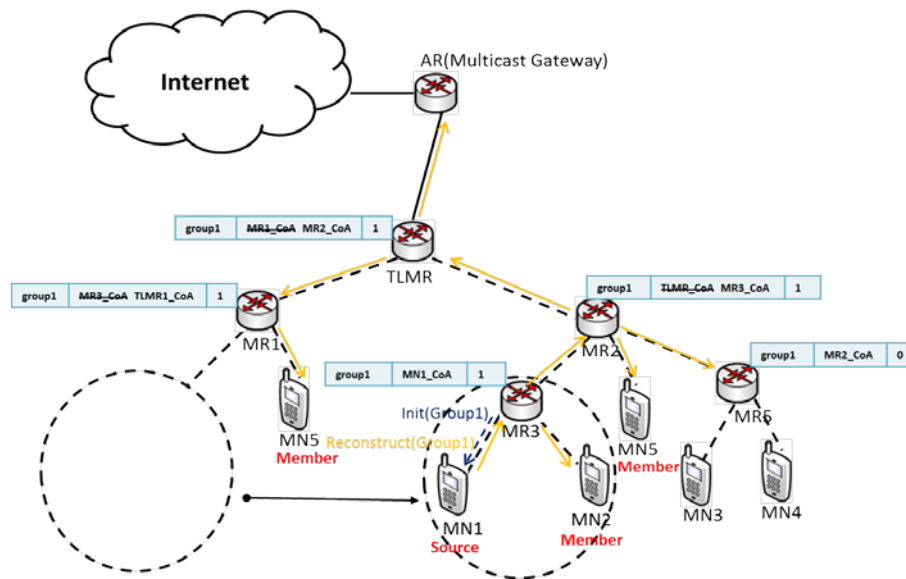


圖 5 MR 攜帶來源端移動

圖 5 為攜帶來源端移動的範例，MR3 帶著身為 Multicast Source 的 MN1 移動至 MR2 底下並獲得新的 CoA，這時 MR 在更新 Multicast Table 時發現 group1 的上游節點為自己的子節點，而須進行 Multicast Tree 的重建。此時 MR3 發送 Init 訊號給 Multicast Table 所記錄的 MN1。身為 Multicast Source 的 MN1 收到 Init 訊息而開始進行群播樹的重建。Multicast Tree 的重建同於章節 3.5.所敘述的演算法。

#### 4. 效能評估

此章節主要評估的方面在於無線網路中進行 Multicast Tree 的構建、維護、成員加入及離開等所需要的花費，評估的對象選擇與 NEMO 環境較相似的 Ad-Hoc 網路中被提出來的 Multicast 路由協議，分別是 ODMRP [6] [8] 與 NSMP [9]。首先簡要介紹這兩種 Multicast 路由協議。

ODMRP 是 IETF 所提出來支援 Multicast 功能的協議，作法是定期向整個 Ad-Hoc 網路廣播 JoinReq 封包，若是成員接收到 JoinReq 封包，則沿著路徑往回發送 JoinReply 封包進行 Multicast Tree 的構建。由於 ODMRP 只定期重複做 JoinReq 與 JoinReply 的發送，雖然簡單且能夠在變動的拓樸中順利進行群播樹的維護，但是會產生比較高的花費，因為

JoinReq 需定期廣播至整個 Ad-Hoc 網路。

NSMP 作法為 Multicast Source 會先對整個網路發送 FLOOD-REQ 封包，當成員接收到 FLOOD-REQ 封包後會沿著路徑回送 REP 封包給 Multicast Source 以進行群播樹的構建，而 Multicast Source 只會發送一次 FLOOD-REQ 封包，接下來則定期發送 LOCAL-REQ，與 FLOOD-REQ 不同的是 LOCAL-REQ 只會沿著 Multicast Tree 與 Multicast Tree 節點的鄰居發送，也就是距離 Multicast Tree 2 個 hop 的成員若欲加入 Multicast Tree，則在接收到 LOCAL-REQ 封包後回送 REP 封包即可。若距離 2 個 hop 外的成員欲加入 Multicast Tree，則廣播 MEM-REQ 封包尋找距離最近的 Multicast Tree 中的節點。節點回覆 Reply discovery 的封包告知欲加入的成員，最後成員再發送 REP 封包給 Multicast Source 以完成 Multicast Tree 的加入。

在效能評估的拓樸方面，主要是在 NEMO 環境中的協議運作進行評估，拓樸為 1 個層數為  $L$  的完全樹，每個節點下都擁有  $K$  個子節點，第  $L$  層的節點為方便計算則全部設為群組成員，也就是說拓樸的第  $L$  層共有  $K^{L-1}$  個成員，來源端則位於完全樹的樹根。以此拓樸進行效能評估。

#### 4.1. 群播註冊效能評估

3 種方法在上述拓樸，且  $K=4$  時進行 Multicast Tree 的構建所需封包量如圖 6，由圖 6 可知我們提出的方法稍微節省一些封包需求，主要的原因是因為我們的方法繼承了 MLD 協議的成員關係報告抑制機制，可以有效的減少 Report 訊息的發送，在  $K$  越大的情況下抑制機制的效果更好而節省更多的訊息發送。由於 ODMRP 與 NSMP 在構建 Multicast Tree 的步驟是相同的，所以耗費相同數量的封包。

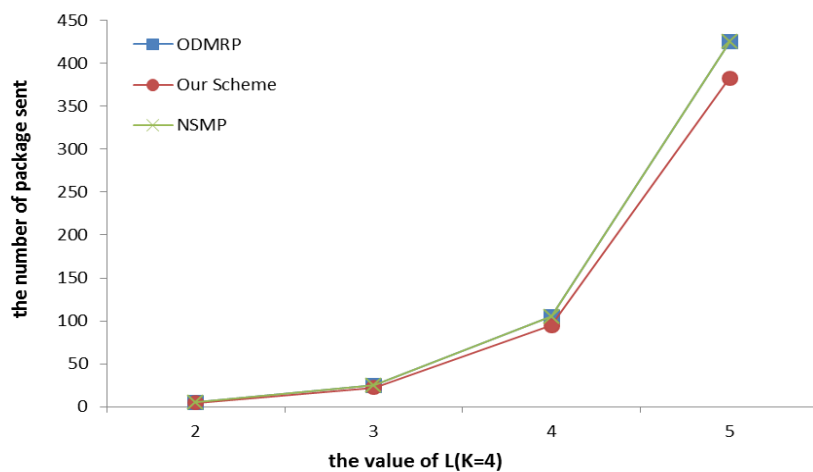


圖 6 當  $K=4$  時群播註冊的評估

#### 4.2. Multicast Tree 維護效能評估

圖 7 與圖 8 分別為  $K=3$  時與  $L=3$  時在拓樸中 3 種方法在維護群播樹時所需要的封包數量，由圖可知當  $L$  越大時我們的方法可以節省更多需要的封包，這也是因為成員關係報告抑制機制可以減少網路中 Report 封包的發送，而 ODMRP 與 NSMP 發送的封包數量相

同所造成的原因是，由於我們所設定的網路拓樸中，第 L 層所有節點均為成員，也就是說整個網路拓樸相當於已構建好的 Multicast Tree，所以 NSMP 無法看出改良的效果。

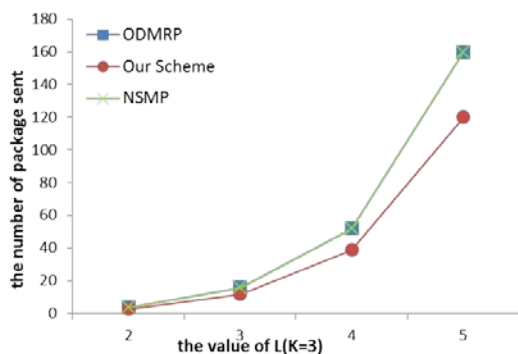


圖 7 當 K=3 時 Multicast Tree 維護所需封包

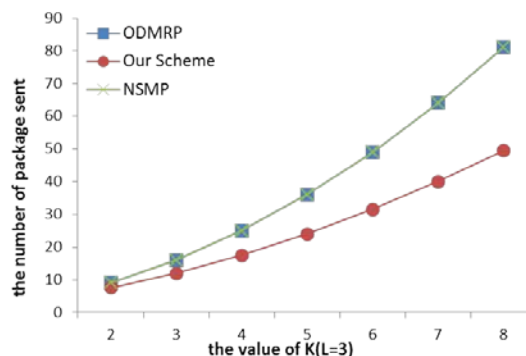


圖 8 當 L=3 時 Multicast Tree 維護所需封包

圖 9 則是 K=3 且 L=3 時拓樸中群組數量(N)變多而需要的封包量評估。由圖 9 可知當拓樸中群組數量越多時，我們的方法可以節省越多的封包。這是因為，在 ODMRP 與 NSMP 的方法中每個群組進行 Multicast Tree 維護的封包數量是分開計算的，但是在我們的方法中，每個 MR 只需向周圍節點發送一次 Query 訊息，周圍節點便會依照所加入的群組或是需進行封包轉發的群組來回應對應的 Report 訊息，也就是說 Query 訊息是群組間所共用的，因此相對於其他 Multicast 路由協議，我們的方法可以減少一定程度的網路負擔。

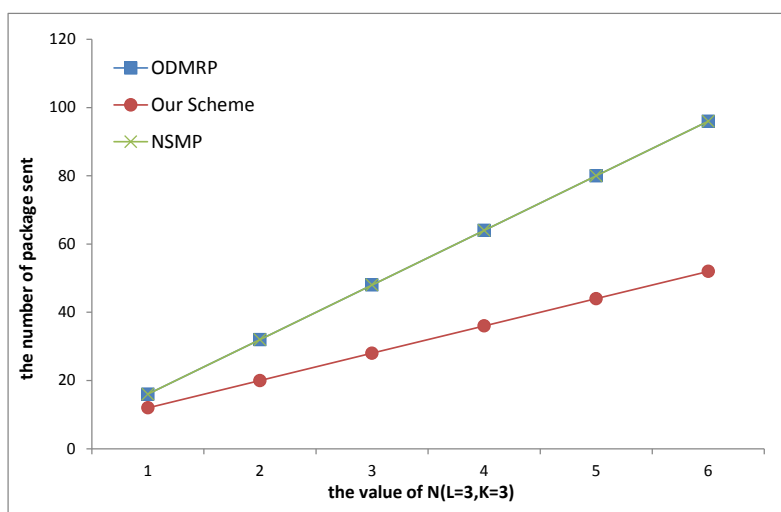


圖 9 當 L=3, K=3 時群組數量變多時所需封包

#### 4.3. 成員加入效能評估

令一個 MR 攜帶一群節點形成一個 D 層的完全樹，每層的節點皆連接 K 個子節點，並移動至拓樸中第 L 層的其中一個節點底下，形成兩個完全樹的连接。假設第 L+D 層的其中一個節點欲加入 L 層的 Multicast Tree，此處則將節點加入 Multicast Tree 所需要的額外花費進行評估。圖 10 是當 K=3 與 L=3 時隨著 D 層的多寡在進行 Multicast Tree 的加入時所造成的額外花費。因為 ODMRP 並沒有成員加入的因應機制，所以並沒有成員加入所需要的額外花費，但若是成員欲加入群組則必須等到下一次接收到 JoinReq 封包並進行

JoinReply 封包的回應才能加入 Multicast Tree，對品質而言相對地會受到影響。NSMP 在  $D$  小於 3 時，因為有將 LOCAL-REQ 發送至 Multicast Tree 相鄰 2 個 hop 的機制，所以額外花費在於將 REP 封包送至 Multicast Source。而若是  $D$  大於等於 3 時，則成員必須利用加入 Multicast Tree 的機制，成本因而大大的提高。而我們的方法則只是往上游節點發送 report 訊息，所以造成的花費較 NSMP 來的小。

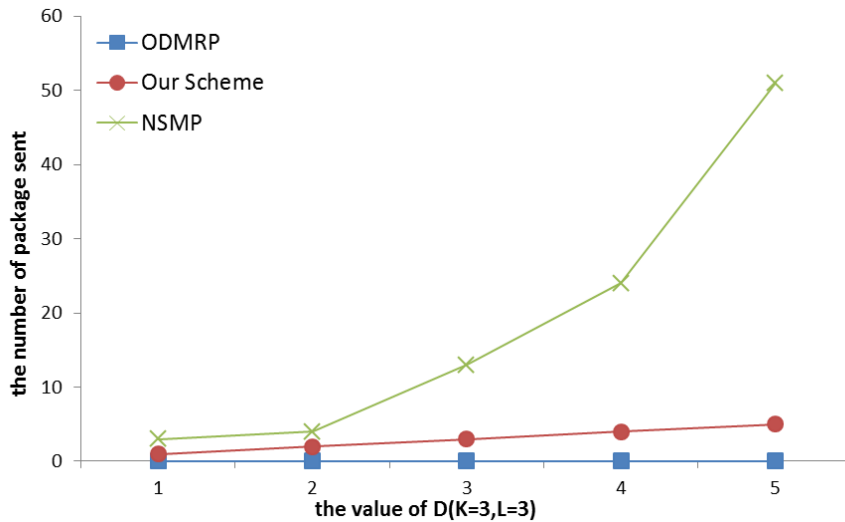


圖 10 當  $K=3, L=3$  時 Multicast Tree 的加入所需額外花費

#### 4.4. 成員離開效能評估

圖 11 為成員離開時為了維護 Multicast Tree 所造成的額外花費評估，拓撲則是一開始只有  $L$  層的完全樹。在成員離開方面我們有主動查詢的機制，也就是當成員欲離開群組時，會發送 Done 訊息告知上游節點，上游節點再發送特定群組的 Query 訊息查詢是否還有該群組的成員存在，若還存在該群組的成員，則成員必須回覆 Report 訊息。雖然主動查詢的機制造成一些額外花費，但若是經查詢後已無該群組成員，則 MR 不需再進行 Multicast Packet 的轉發，在下次常規的 Query 訊息發送前所收到的該群組的 Multicast Packet 可不需進行轉發以減少網路負擔。而若是成員因移動而斷線，則 MR 只需利用一般定期發送的 Query 訊息進行群播樹的維護即可，所以被動的查詢不需發送額外的訊息。而 ODMRP 與 NSMP 亦無成員離開的即時對應機制，所以亦無消耗額外的訊息花費。

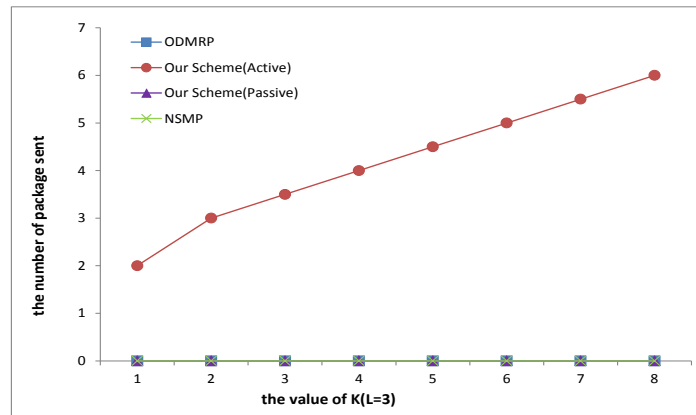


圖 11 當 L=3 時成員離開所需額外花費

## 5.結語.

在無線網路越來越普及的現今，網路頻寬的有效利用也就越顯得重要，本文提出的 Multicast 路由協議主要優點在於若 NEMO 網路中存在多種群組的 Multicast Tree，則可以減少網路中維護 Multicast Tree 的花費，且當來源端進行移動時，能夠順利的進行群播樹的重建，並減少不必要的封包支出。在評估後顯示了與本文提出的路由協議相比，較早所提出的協議需要較多額外的花費，在 NEMO 環境中我們期望該協議能夠擁有良好的效能，以及減少不必要的開銷。在現今行動裝置普及的世代，希望本文所提出來的 Multicast 路由協議能夠為社會做出相當的貢獻。

## 參考文獻

1. A. Adams, J. Nicholas, and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)" IETF RFC 3973, January 2005.
2. D. Estrin, et al., "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification" IETF RFC 2362, June 1998.
3. D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol" IETF RFC 1075, November 1988.
4. I. Romdhani, et al., "IP mobile multicast: Challenges and solutions" IEEE Communications Surveys & Tutorials, (Volume:6 , Issue: 1):18-41.
5. J. Moy, "Multicast Extensions to OSPF" IETF RFC 1584, March 1994.
6. Mario Gerla et al., "On-Demand Multicast Routing Protocol (ODMRP) for Ad-Hoc Networks" IETF, draft-ietf-manet-odmrp-00.txt, November 1998.
7. S. Deering, W. Fenner, and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6" IETF RFC 2710, October 1999.
8. Seong-Yee Phang, et al., "Multicasting Support for NEMO based on ODMRP" Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference, page:611-616, 6-8 Aug. 2008.
9. Seungjoon Lee, and Chongkwon Kim, "Neighbor supporting ad hoc multicast routing protocol" Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop, page:37-44,2000.
10. V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol" IETF RFC 3963, January 2005.