

A Multipath 6LoWPAN Ad Hoc On-Demand Distance Vector Routing

一個使用 IPv6 協議之多路徑低功率無線個人區域 網路距離向量路由策略

Hsin-Yun Yang(楊心運) Jian-Ming Chang(張建明) Ting-Yun Chi(紀廷運) Han-Chieh Chao(趙涵捷)
National Ilan University, National Dong Hwa University, National Taiwan University, National Ilan University,
Taiwan, ROC Taiwan, ROC Taiwan, ROC National Dong Hwa University,
smallchecken@gmail.com a0128866@niu.edu.tw louk.chi@gmail.com Taiwan, ROC
hcc@niu.edu.tw

Abstract

In order to move towards the current development of ubiquitous network environment. To integrate the different types of network environment is going on. The IEEE 802.15.4 standard is a low cost, low-rate, Low-power Wireless Personal Area Network technology. IP for Smart Objects seeks to extend the use of IP networking into resource-constrained devices over a low-power network. IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) are specified by the IETF Working Group, and advanced by the IPSO alliance. 6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD) is based on AODV, and simplify the routing protocol for 6LoWPAN mesh network. Although the LOAD simplify the AODV for 6LoWPAN mesh network, but the route discovery by broadcasting route request(RREQ) still huge whole network overhead. So we would like to use the concept of Ad-Hoc On-Demand Multipath Distance Vector Routing (AOMDV) for LOAD to find multipath routes during the route discovery. If the path fail we still have alternate path can be used for reducing the overhead of route discovery for the network.

Keyword : IPv6, LoWPAN, AODV, Multipath Routing

摘要

為了朝向目前無所不在的網路環境發展趨勢，將不同種類的網路環境整合為一，已經是目前積極進行中的工作。IEEE802.15.4 標準是一個低價格、低速率以及低功率的無線個人區域網路技術，IP 對智慧物件的設計，朝向在資源受限的低功率網路設備中，擴展 IP 網路的使用。IPv6 協議的低功率無線個人區域網路正在 IETF 工作團隊中制定，並且由 IPSO 聯盟推動。由於 6LoWPAN 使用的 LOAD 是以隨意隨選距離向量路由協定(AODV)為基礎，並針對 6LoWPAN 網狀網路簡化而設計。雖然 LOAD 簡單化了原本 AODV 的設計，並且適用於 6LoWPAN 網狀網路，但每次因找尋路由路徑而廣播的路由請求封包依然會造成整個網路系統龐大的負擔。因此我們希望在 LOAD 協定上使用 AOMDV 的概念，也就是在每次路由路徑尋找的同時找尋到多個路由路徑，讓每次路由路徑失效時還有替代路徑可以使用，來減少路由路徑找尋的次數，以降低整個網路系統對路由路徑找尋的負擔。

關鍵詞—IPv6、低功率無線個人區域網路、隨意隨選距離向量路由協定(AODV)、多路由路徑

一、導論

近年來網路發展趨勢的正朝向無所不在的概念發展，目前也已經擴展到感測網路環境整合中。傳統使用 IEEE802.15.4 協定的感測網路使用具轉換功能的閘道器與外界 IP 網路通訊連接，在感測網路中每個裝置均擁有一個識別代碼，但不具有直接對外溝通的能力，而 IP 針對智慧裝置(Smart Object)的設計也已經朝向擴展 IP 網路的使用到該類資源受限的低功率無線個人區域網路。

為了使傳統感測網路能與外界 IP 網路直接通訊溝通，國際標準組織 IETF 在 2007 年開始針對使用 IEEE802.15.4 的低功率網路進行 IPv6 協議相關制定工作，IPv6 over Low-power Wireless Personal Area Network 簡稱 6LoWPAN[1]相關規範已由 IETF 工作小組正在制定中，並且由 IPSO 聯盟推動[2]。6LoWPAN 使用 IP 技術透過邊界路由器與傳統 IP 網路通訊溝通，使用目前現有的有線網路、無線網路以及電信網路做為連結橋梁，將 IP 網路擴展到低功率無線個人區域網路。

在這些先天條件的限制下，IPv6 協議對於低功率無線個人區域網路上的設計中產生了許多嚴重的議題討論，包含啟動流程的簡化、相關路由機制、服務發現機制、安全性考量等相關議題都正在被討論。6LoWPAN Ad Hoc On-Demand Distance Vector Routing[3] 簡稱 LOAD 是針對 6LoWPAN mesh under 的網路模式所提出來的路由方法，其主要是簡化了傳統使用在 IEEE802.11 的 AODV (Ad Hoc On-Demand Distance Vector Routing)[4]相關路由機制，以提供適合 IPv6 協議低功率無線個人區域網路的路由方法。雖然 LOAD 簡化了 AODV 的路由機制，但在路由計算上存在著與 AODV 一樣的問題，在每一次的路由尋找(Route Discovery)上需要發送出路由(Route Request)請求封包，對整個網路將造成泛流(Flooding)情況來尋找單一路徑。

因此，本文中將 LOAD 加入多路徑的概念，在每一次的路由路徑尋找時找尋到多條路徑，以提供每次路徑失效後還有替代路徑可以使用以減少路由尋找的次數，降低對整個網路造成泛流的情況，以減少整個低功率無線個人網路的能量消耗。

二、文獻討論

(一) AODV 簡介

AODV 全名 Ad-Hoc On-Demand Distance Vector Routing[4]是專門針對 IEEE802.11 標準來設計的被動式平面距離向量路由協議，路由的建立是由整個網路的節點各自去維護自己的路由表(Routing Table)，只有在每次需要使用到路由時才會建立路由。如圖 1 所示，AODV 在每次來源端節點 S 需要路由找尋時，會先對網路廣播一個路由請求封包 (RREQ, Route Request)，當目的端節點 D 收到後，會回送路由回應(RREP, Route Reply)給來源端節點 S 來建立起雙向路由。雖然在每次傳送資料花費在路由建立上的時間較長，但相較於其他主動式路由機制來的節省網路資源。

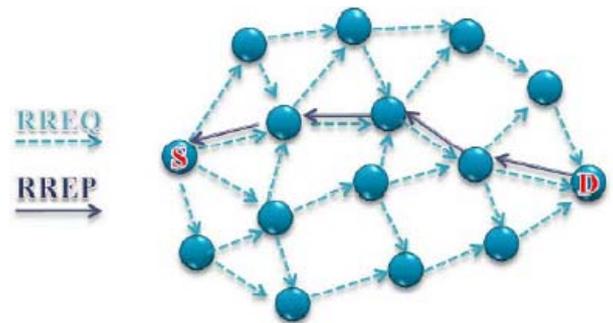


圖 1：AODV 路由建立示意圖

(二) AOMDV 簡介

AOMDV (Ad-Hoc On-Demand Multipath Distance Vector Routing) [5][6]是由美國辛辛那

提大學(University of Cincinnati) 的 Samir R. Das 與 Mahesh K. Marina 兩位所發表的多路徑 AODV 路由方法。AOMDV 路由演算法將使用單一路徑的 AODV 路由協議改進為多路徑路由方法，在每一次的路由找尋中同時找尋到多條鏈路不相交的路徑，讓在路徑失效時還有替代路徑可以使用，而減少路由找尋時對整個網路的負擔。如圖 2 所示，AOMDV 在找尋到多條路徑後，會以最小路由花費的路徑為主要路徑 (Primary Route)，而其他路徑為路徑失效後的替代路徑 (Alternate Route)，當主要路徑失效後還可以使用替代路徑繼續傳輸資料，減少路由尋找的次數。

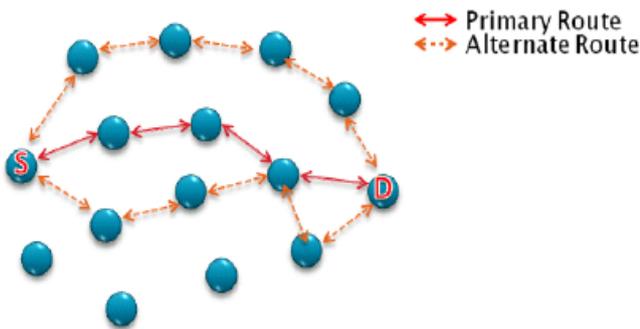


圖 2：AOMDV 的多路徑示意圖

(三) 6LoWPAN 簡介

6LoWPAN 是使用 IPv6 協議的低功率無線個人區域網路(IPv6 over Low-power Wireless Personal Area Network)的縮寫，其標準堆疊架構如圖 3 所示，主要是制定適應層 (Adaptation Layer) 來讓 IPv6 能在 IEEE802.15.4 標準使用。6LoWPAN 由 IETF 工作小組於 2004 年開始制定 [1]，已經於 2007 年八月提出定義 6LoWPAN 的相關簡介內容 [8]，以及 2007 年九月提出制定 6LoWPAN 的封包格式定義與 IPv6 標頭壓縮格式 [9]。另外還有其他的草案還在制定當中，如設計與應用範圍 [10]、路由機制的需求與問題

[11]、鄰居的發現 (Neighbor Discovery) [12] 與標頭壓縮 [13]。

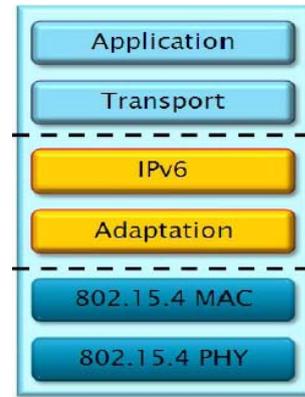


圖 3：6LoWPAN 堆疊架構圖

所以在 6LoWPAN 的定義適應層如表 4 所示主要制定了標頭壓縮 (IPv6 Header Compression)、封包切割 (Fragment Header) 與 Layer 2 層的 Forwarding 機制 (Mesh Addressing Header)，以提供低功率無線個人區域網路中做 IPv6 協議的啟動流程與封包傳輸。



圖 4：6LoWPAN 封包格式

(四) LOAD 簡介

LOAD [3] 是 6LoWPAN Ad Hoc On-Demand Distance Vector Routing 的縮寫，主要是將原有 AODV 路由協議精簡化後使用在 6LoWPAN 的 Mesh Under 網路中。因為在 6LoWPAN 裡面設計路由需要考慮到 IEEE802.15.4 標準的限制，LOAD 的設計將原有的 AODV 協議上有些較不適用在低功率無線個人區域網路的機制移除，以提供程序更精簡、消耗功率更低以及記憶體花費更小的路由方法。

LOAD 採用 IEEE802.15.4 標準的兩種 Link Layer 位址模式來使用，在設計上不使用原有 AODV 的 Destination Sequence Number，用來減

少路由封包的大小以及簡化路由找尋的處理程序。為了確保路徑不產生迴路，在 LOAD 協議中只允許目的端節點能夠對回應來源端節點的路由請求，而並非像原有 AODV 裡面可允許中間節點回應。LOAD 也將原有 AODV 的 Hello Message 功能移除，避免頻繁的 Hello Message 對網路造成功率消耗，替代判斷鏈路狀況的方式則是使用 Link Layer 的 ACK 機制，在每次傳輸封包時才去判斷路徑的狀況。並且也移除了 Precursor List，避免記憶體的使用以及減少對網路的功率消耗，當發現路徑失效後只需要傳輸 RRER 訊息通知正在傳輸的來源端節點。

傳統的 AODV 在路由計算上使用跳用次數 (Hop Count) 來評估路由的花費，而 LOAD 則使用實體層的鏈路品質指示 (Link Quality Indicator) 來計算路由的花費，以提供更強健的路由連結來降低封包的損失率。

三、本文方法

(一) MLOAD 簡介

儘管 LOAD 簡化了原有 AODV 協議的相關路由處理程序與路由資料格式，提出了較適合在低功率無線個人區域網路使用的單一路徑路由方法。但在每次路由找尋時對網路廣播的路由請求封包依然造成網路能量消耗的負擔。

因此，本文提出 MLOAD (Multipath 6LoWPAN Ad Hoc On-Demand Distance Vector Routing) 演算法使用多路徑的概念來改進 LOAD 路由，在每次的路由找尋同時找尋到多個路徑。如圖 5 所示，當主要路徑 (Primary Route) 失效時還有替代路徑 (Alternate Route) 可以使用來減少路由找尋的次數，以降低對整個網路的路由找尋負擔。MLOAD 路由演算法以 LOAD 路由演算法為基礎來設計，延用原有 LOAD 路由演算法中的相關設計，並使用 AOMDV[5][6] 路由演算法概念加以改進單一路徑的 LOAD 路由演算法。

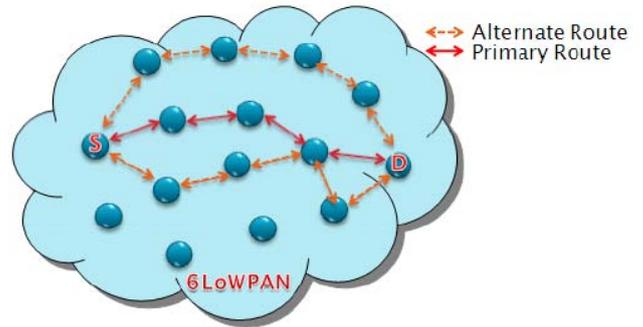


圖 5：MLOAD 路由演算法多路徑示意圖

(二) 環境說明

如圖 6 所示，本文的研究環境使用 IEEE802.15.4 標準的 6LoWPAN Mesh 網路環境，在這環境下只包含一個邊界路由器 (Edge Router) 與多個 6LoWPAN 節點 (Node)，每個節點均使用 IEEE802.15.4 標準的全功能裝置 (Fully Function Device)。而在 6LoWPAN 網狀網路環境中的節點將不固定位置，所有節點將可於 6LoWPAN 網狀網路環境中自由移動。

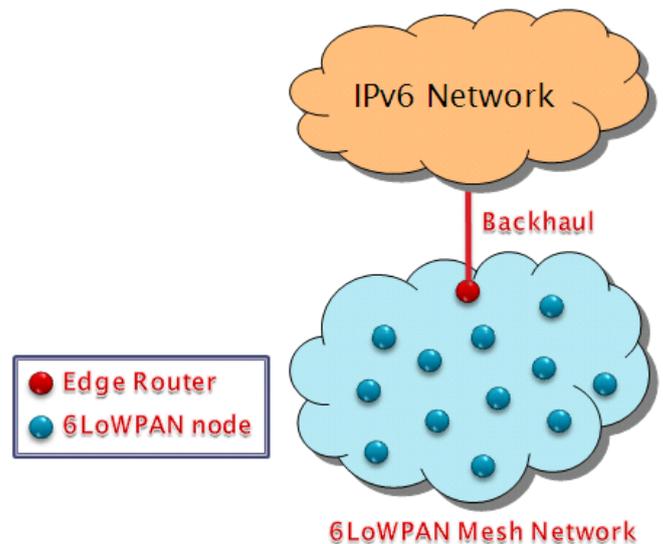


圖 6：6LoWPAN 網狀網路環境圖

(三) MLOAD 資料結構

本文的 MLOAD 路由演算法修改為了使用

多路徑計算而將 LOAD 路由演算法的兩種路由表格內容加以改進，一種是路由表(Routing Table)用於每次傳輸資料時的下一跳路由判斷。另一種是路由請求表(Routing Request Table)，只使用在路由尋找的階段，用來儲存路由尋找的相關資訊，當路由找尋完成或路由找尋時間終止時即會刪除該路由請求表項目，以節省記憶體消耗。

(1) MLOAD 路由表

MLOAD 路由表(Routing Table)如表 1 所示，主要修改了原有的 Next Hop Address 欄位資訊，將其改名為 Next Hop Address List 來存放多個 Next Hop Address，以達到多路徑演算法的計算。

表 1：MLOAD 路由表

MLOAD Routing Table			
Destination Address	Status	Next Hop Address List (Next Hop 1) (Next Hop 2) ...	Life Time

MLOAD 的路由請求表(Route Request Table)如表 2 所示，主要修改了原有的 Reverse Address 欄位資訊，將其改名為 Reverse Address List 來記錄多個往來源節點路由的 Reverse Address 資訊，以達到多路徑演算法的計算。以及修改了 Forward Cost 欄位資訊，改名為 Advertised Forward Cost，記錄 Reverse Address List 欄位中最大的路由花費(Route Cost)。另外新增了 Forward Address List 欄位，用來記錄往目的端節點的多個 Forward Address 資訊。而 Reverse Cost 欄位則改名為 Advertised Reverse Cost，用於記錄 Forward Address List 欄位中最大的路由花費。

表 3-2：MLOAD 路由請求表

MLOAD Route Request Table						
Route Request ID	Originator Address	Reverse Address List (Next Hop 1, Last Hop 1, Cost1) (Next Hop 2, Last Hop 2, Cost2) ...	Advertised Forward Cost	Forward Address List (Next Hop 1, Last Hop 1, Cost1) (Next Hop 2, Last Hop 2, Cost2) ...	Advertised Reverse Cost	Valid Time

(2) 封包格式

MLOAD 使用原有 LOAD 的封包格式，並做了些小部份的欄位新增以達到多路徑計算。MLOAD 的路由請求(RREQ, Route Request)封包格式如圖 7 所示，主要新增了 L 欄位與 Last Hop Address 欄位來記錄計算多路徑的資訊。

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Type										R	D	O	L	Reserved					CT	WL	RREQ ID										
RC										Link Layer Destination Address																					
Link Layer Originator Address																															
Last Hop Address																															

圖 7：MLOAD Route Request 封包格式圖

L 欄位占 1 位元為我們新增的欄位，使用原本保留(Reserved)欄位的 1 位元位置，用來判斷最後一跳位址(Last Hop Address)欄位的位址模式，1 為 IEEE802.15.4 標準的 16 位元 PAN ID 位址，0 為 64 位元的裝置 EUI-64 位址。Last Hop Address 欄位為存放到來源端節點的最後一跳 (Last Hop)位址資訊，也就是來源端的鄰居節點位址，為 IEEE802.15.4 標準的 16 位元 PAN ID 位或 64 位元的裝置 EUI-64 位址。

MLOAD 的路由回應(RREP, Route Reply)封包格式與 RREQ 格式相同，差別在 Type 欄位設定為 2，以及 RREQ ID 欄位填入收到的 RREQ 資訊。而路由錯誤(RRER, Route Error)封包格式則使用原本 LOAD 相同。

(四) 路由找尋流程

本文提出的 MLOAD 路由演算法使用 AOMDV[5][6]的概念改進 LOAD 路由演算法，在每次路徑找尋發送路由請求時，同時找尋多

條鏈路不相交的路徑(Link-Disjoint Path)，並且以其中一條路由花費最低的路徑為主要傳輸路徑(Primary Route)作資料傳輸，而其他路徑則為替代路徑(Alternate Route)，當主要路徑失效後則使用替代路徑繼續傳輸資料。

路由找尋流程主要分為兩個階段，分別為一開始來源節點需要找尋到目的端節點的路徑時而對 6LoWPAN 網路廣播的路由請求(RREQ, Route Request)封包階段，另一個為當目的端節點收到路由請求封包後對來源端節點使用單播(Unicast)回應的路由回應(RREP, Route Reply)封包階段，而此兩階段將由以下作詳細說明。

(1) 路由請求階段

當來源節點需要傳輸資料到目的節點時，如果來源節點的路由表(Routing Table)中沒有往目的節點的路由時，來源節點將發送出路由請求封包開始找尋路由，此時發送的路由請求封包中的 Last Hop 欄位將為空值。

當中間節點 i 收到節點 j 廣播的路由請求後處理流程如圖 8 虛擬碼所示，節點 i 會先更新路由請求封包目前的花費(1)，接著判斷路由請求封包中的 Last Hop 欄位是否為空值(2)，如果是則代表節點 i 為來源端的鄰居節點，然後將自己的節點位址填入路由請求的 Last Hop 欄位(3)；如果不是空值則代表節點 i 不為來源端節點的鄰居節點。之後使用路由請求識別(RREQ ID)與來源位址(Originator Address)來確認此路由請求是否存在路由請求表(Route Request Table)中(4)，如果不存在則加入該路由請求於路由請求表中並初始化路由請求表中的資料(11)(12)(13)，以及初始化來源端節點的反向路徑存到路由表(Routing Table)表中(14)(15)，然後對 6LoWPAN 網路廣播此路由請求封包(16)；而如果路由請求存在路由請求表中，則判斷該路由花費與 Last Hop 資訊是否符合加入到往來源端

節點的反向路由表與路由請求表中(5)，如果符合則加入該路徑資訊到路由請求表與路由表(6)(7)；如果不符合則不需理會該路由請求封包(9)。

```

(1) update the RREQ(Cost);
(2) If ( RREQ>Last hop) is empty )
(3)   set RREQ>Last hop) = i ;
(4) If (match (RREQ ID , Originator) in Route_Request_Table) {
(5)   If (RREQ(Cost)<=Advertised_Forward_Cost) && (not match ( j , RREQ>Last
Hop ) in Route_Request_Table(RREQ ID, Originator) Reverse_Address_List) {
(6)     inset (j , RREQ>Last Hop) , RREQ(Cost)) into _
Route_Request_Table(RREQ ID,Originator) Reverse_Address_List;
(7)     inset j into Routing_Table(Originator) Next_Hop_List ;
(8)   }else{
(9)     discard the RREQ;
}
(10) }Else{
(11)   initial Route_Request_Table(RREQ ID, Originator);
(12)   inset (j , RREQ>Last Hop) , RREQ(Cost)) into _
Route_Request_Table(RREQ ID, Originator) Reverse_Address_List;
(13)   set Advertised_Forward_Cost = RREQ(Cost);
(14)   initial Routing_Table(Originator);
(15)   inset j into Routing Table(Originator) Next_Hop_List ;
(16)   forward the RREQ;
}

```

圖 8：路由請求階段的虛擬碼

(2) 路由回應階段

當目的端節點接收到來源端節點的路由請求後，將會對來源端節點單播(Unicast)發送出路由回應(RREP, Route Reply)封包，此時會將路由請求識別(RREQ ID)存放至路由回應封包中的路由回應識別(RREP ID)欄位，以及將路由回應封包中的 Last Hop 欄位設為空值。

當中間節點 i 收到節點 j 發送的路由回應後處理流程如圖 9 虛擬碼所示，先使用路由回應識別(RREP ID)與來源位址(Originator Address)來確認此路回應的路由請求是否存在路由請求表(Route Request Table)中(1)，如果不存在則不理會該路由回應封包(16)；如果路由請求存在路由請求表中，則節點 i 會先更新路由回應封包目前的花費(2)，接著判斷路由回應封包中的 Last Hop 欄位是否為空值(3)，如果是則代表節點 i 為目的端的鄰居節點，然後將自己的節點位址填入路由回應的 Last Hop 欄位(4)；如果不是空

值則代表節點 i 不為目的端節點的鄰居節點。接著判斷是否已收到路由回應封包(5)，如果沒則將該路由回應的路徑資訊加入到路由請求表中(6)(7)，以及在路由表中初始化往來源端節點的路由資訊(8)(9)；如果之前有收到過其他由目的端節點回應的路由回應，則判斷該路由花費與 Last Hop 資訊是否符合加入到往目的端節點的順向路由表與路由請求表中(10)，如果符合則加入該路徑資訊到路由請求表與路由表(11)(12)；如果不符合則不需理會該路由回應封包(14)。

路由花費以圖 10 表示為說明範例，在圖中有兩條路徑分別為路徑一(S,A,B,D)與路徑二(S,C,E,F,D)，以下對兩種路由花費使用的計算方法分別說明：

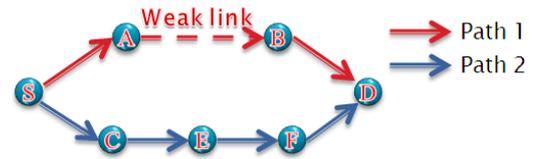


圖 10：路由花費計算說明圖

```

(1) If (match (RREP ID, Originator) in Route_Request_Table) {
(2)   update the RREP(Cost);
(3)   If ( RREP(Last hop) is empty )
(4)     set RREP(Last hop) = i ;
(5)   If (Route_Request_Table(RREP ID, Originator) Forward_Address_List is empty){
(6)     inset (j, RREP(Last Hop), RREP(Cost)) into _
       Route_Request_Table(RREP ID, Originator) Forward_Address_List;
(7)     set Advertised_Reverse_Cost = RREP(Cost);
(8)     initial Routing_Table(Destination);
(9)     insert j into Routing_Table(Destination) Next_Hop_List;
(10)  }Elseif (RREP(Cost) <= Advertised_Reverse_Cost) && (not match (j, RREP(Last
      Hop)) in Route_Request_Table(RREP ID, Originator) Forward_Address_List {
(11)   inset (j, RREP(Last Hop), RREP(Cost)) into _
       Route_Request_Table(RREP ID, Originator) Forward_Address_List;
(12)   insert j into Routing_Table(Destination) Next_Hop_List;
(13)  }Else{
(14)   discard the RREP;
      }
(15) }Else{
(16)   discard the RREP;
      }
}

```

圖 9：路由回應階段的虛擬碼

而本文提出的多路徑演算法在路由花費(Route Cost)使用 8 位元來計算，計算上採用兩種計算方式，一是單純以跳躍次數(Hop Count)來做計算，找尋跳躍次數最短的路徑作為傳輸，在之後的模擬上使用此路由計算的多路徑演算法表示為 MLOAD。另一種是 LQI 計算方式，使用節點接收到的 LQI 值來判斷鏈路的強鍵性，將 LQI 較差的鏈路視為弱鏈結(Weak Link)，在路徑選擇上將會盡量避免使用弱鏈結的路徑，之後使用 LQI 為路由計算的多路徑演算法表示為 MLOAD-LQI，之後模擬將會評估此兩種路由花費的差異性。

MLOAD 演算法單純使用跳躍次數來做計算，以找尋到最短的跳躍次數路徑來作為傳輸。因此，路徑一的跳躍次數為 3 次跳躍(Hop)，而路徑一為 4 次跳躍，在路徑的選擇上 MLOAD 演算法會採用跳躍次數較少的路徑一來作為資料傳輸路徑。

MLOAD-LQI 的路由計算使用到 LQI 值做判斷，以找尋到鏈結性較好的路徑為傳輸路徑。在計算上是以弱鏈結數量與排除落鏈結的跳躍次數來作位元組合併，弱鏈結數量為高 4 位元，排除弱鏈結的跳躍次數為低 4 位元，將此兩數值組合成一個位元組來表示。因此，路徑一的路徑上有一條弱鏈結所以路由花費為 18 (00010010B)，而路徑二的路徑上無弱鏈結所以路由花費為 4 (00000100)，在路徑的選擇上 MLOAD-LQI 演算法會採用路徑二作為資料傳輸路徑。

(五) 路由維護

本文提出的多路徑路由演算法路由維護上，延用原有設計於 6LoWPAN 網路的 LOAD 路由演算法概念，在路由失效後使用原有 LOAD 的路由錯誤(RERR, Route Error)封包格式。而在

路由有效的判斷上，使用 Link Layer 的 ACK 機制來判斷鏈路是否有效，當每次傳送封包時均要求對方節點須回應 ACK 封包來確認封包是否正確傳遞到。如果收到回應的 ACK 封包則判斷出此鏈路路由失效，並對此時傳輸資料的來源端節點回傳路由錯誤封包通知此路由已失效。當中間節點收到路由錯誤封包後，會先刪除往此鏈路的路由資訊，並判斷是否還有替代路由可以使用，如果有的話則不需再傳遞路由錯誤封包通知來源端節點；如果沒有的話則繼續傳遞路由錯誤封包通知源端節點。當來源端節點全部的替代路徑都失效後，此時才會重新作路由找尋流程，對 6LoWPAN 網路廣播路由請求封包找尋新的路由資訊。

四、模擬和結果分析

(一) 模擬環境

在模擬環境上，我們使用網路模擬軟體 NS2[14][15]來做模擬測試，版本採用由 NS2 源碼釋放網頁[16]下載的 NS2.34 版本。在模組套件上使用由紐約市大學(City University of New York)於 2003 年到 2004 年開發的 IEEE802.15.4 模組套件[17]，此 IEEE802.15.4 模組套件包含實體層與鏈結層的網路模擬。並且使用卡內基梅隆大學(Carnegie Mellon University)於 1997 年到 1998 年開發的 AODV 模組套件來修改。

在 NS2 網路模擬工具上我們修改了原有的 AODV 模組將其修改為 LOAD 模組，移除了 AODV 中的 Hello Message 與 Destination Sequence Number 與 Precursor List 等相關機制，並使用 IEEE802.15.4 模組中 Link Layer 的 LQI 資訊來做路由計算。以及將 LOAD 模組修改為本文的多路徑 MLOAD 模組，在後續模擬上使用兩種路由花費計算來做模擬，分別為單純使用跳躍次數(Hop Count)的 MLOAD 路由方法與使用 LQI 計算的 MLOAD-LQI 路由方法。

以下為模擬節點移動速度對路由效能影響的結果分析，此模擬分析是要模擬節點在移動速度上對路由效能的影響。在模擬上我們變動節點的移動速度由 0~3.5m/s 來做調整，模擬在節點移動速度低和高時導致路由失效的效能，相關模擬參數請參考表 2。

表 2：節點移動速度模擬參數

模擬參數	參數設定值
Propagation	TwoRayGround(2D)
Radio range of a node	25 m
Channel capacity	250k bits/sec
(MAC) protocol	IEEE802.15.4 Distributed Coordination Function
Traffic pattern	5 CBR/UDP
Size of data packet	70 bytes
Data rate	1 packet/sec
Simulation area	100m x 100m
Number of nodes	50 nodes
Maximum speed	0~3.5 m/s
Number of simulations	50
Simulation time	500 seconds

(二) 模擬結果與分析

在模擬結果的分析上，我們主要針對三個路由演算法 LOAD、MLOAD 以及 MLOAD-LQI 來做模擬，並且比較了分析項目分別如下：

(1) 平均每次路由找尋的花費(Average Per Route Discovery Overhead)

平均每次路由找尋的路由花費為計算在每次路由找尋時所需花費的路由封包數量，計算方式為路由找尋封包總數除以由找尋次數總數，單位為每次多少路由封包(packets)。

圖 11 為變動節點移動速率後的模擬分析圖，X 軸代表節點的移動速率分別為 0~3.5 公尺/每秒，模擬當節點移動速率越大而導致路由失效頻繁的對效能的影響。Y 軸代表模擬的分析出的平均每次路由找尋路由花費，計算在每次的路由找尋時平均需要花費的路由封包數量。

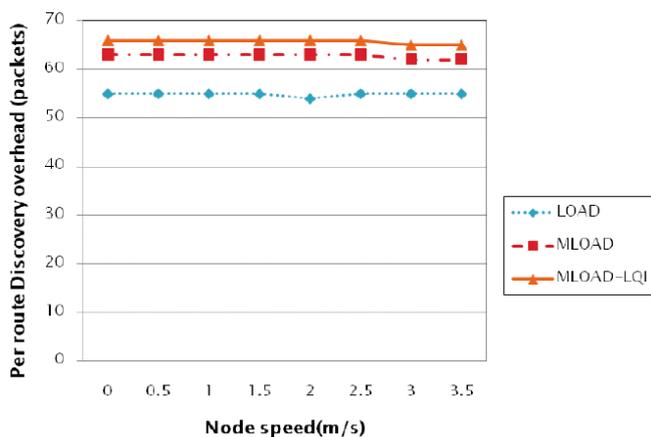


圖 11：變動移動速度的平均每次路由找尋花費

圖中可以看到隨著移動速率的提高，對三種演算法的平均每次的路由找尋花費都沒明顯影響。單路徑演算法 LOAD 由於在每次找尋路由時只找尋單一條路徑，所以在路由建立上只需花費回應一條路徑的 RREP 封包，而多路徑演算法 MLOAD 與 MLOAD-LQI 由於必須同時找尋三條路徑，在回應上需比單路徑 LOAD 來的多兩條路徑回應上的花費，因此多路徑演算法在平均每次路由找尋的花費比單路徑演算法 LOAD 來的花費高。而在多路徑演算法 MLOAD 與 MLOAD-LQI 每次找尋的路由花費上，由於 MLOAD 採用的是跳躍次數較短的路徑，因此平均在每次路由找尋的回應封包會比 MLOAD-LQI 的花費略少一些。

(2)平均每秒路由找尋的路由花費(Average Routing Overhead)

平均每秒路由找尋的路由花費為計算平均每秒針對路由找尋所花費的路由封包數量，計算方式為平均每次路由找尋的平均花費乘以平均每秒的路由找尋次數，單位為每秒多少路由封包(packets/s)。

圖 12 為變動節點移動速率後的模擬分析圖，X 軸代表節點的移動速率分別為 0~3.5 公尺/每秒，模擬當節點移動速率越大而導致路由失效頻繁的對效能的影響。Y 軸代表模擬的分析出

的平均每秒路由找尋路由花費，計算在每秒鐘平均路由花費的路由封包數量。

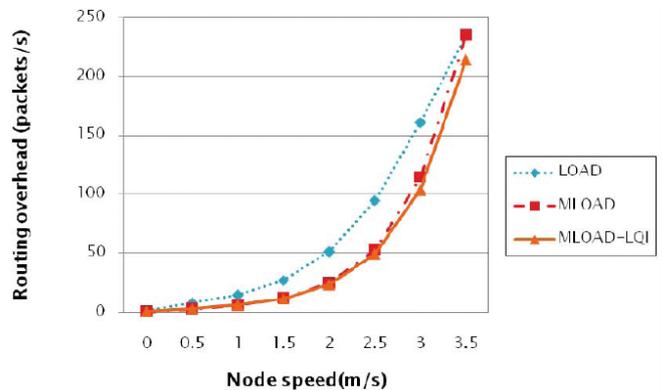


圖 12：變動移動速度的平均每秒路由找尋路由花費

圖中看在低速率時採用多路徑的方式明顯減少了在低功率網路中的路由花費，但在較高速率時由於路由失效頻繁，單路徑與多路徑的演算法兩者的差異較小。而多路徑的 MLOAD 與 MLOAD-LQI 在 2.5 公尺每秒之前的路由花費差異不大，平均 MLOAD 較 MLOAD-LQI 來的花費少，但到了 2.5 公尺每秒以後由於 MLOAD-LQI 的路由較為強健，在路由失效機率較 MLOAD 來的低，所以在平均每秒的路由花費上開始明顯的降低了路由找尋的花費。

五、結論

在這篇論文中，我們修改了於 IPv6 協議的低功率無線個人區域網路所使用的 LOAD 路由演算法，成功提出一個多路徑於使用 IPv6 協議低功率無線個人區域網路的距離向量路由演算法方法(MLOAD, Multipath 6LoWPAN Ad Hoc On-Demand Distance Vector Routing)。降低了原有單一路徑的 LOAD 路由演算法在每次因路徑失效而對低功率網路進行的路由找尋次數，以節省在 6LoWPAN Mesh 網路中因路由找尋而使

