

針對路燈監控系統的 Zigbee 繞徑方式研究

A Design of Zigbee Routing for Street Light Control System

Yeou-Der Chen (陳有德) Hsiao-Wen Yu (于曉雯) Ming-Shiou Wu (吳明修) Jih-Ching Chiu (邱日清)

Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, 804, Taiwan

Email: ytchen@ee.nsysu.edu.tw; ytchen@ee.nsysu.edu.tw; m973010028@student.nsysu.edu.tw; chiujihc@ee.nsysu.edu.tw

摘要—Zigbee 系統具有組建網路容易、設備擴充及升級容易、低功耗、價格便宜等優點，成為現今監控系統的主流。Zigbee 主要繞徑演算法為 Tree routing 與 On Demand routing，本論文以 On Demand Routing 演算法為基礎，改良 AODV-BR (Ad-hoc On-demand Distance Vector routing- Backup Route) 演算法，以提出 AODV-BN (Backup Node) 演算法。此演算法可以在不增加其他控制訊號負載前提下，使用原來 AODV 演算法中的 RREP 控制訊號來建置備份節點，並透過竊聽封包建立 Neighbor Table 取代週期性發送 Hello Message，因此可大量減少電力及頻寬的資源消耗，本論文提出的演算法可在有節點離開網路時，使用非常少量的控制訊號來使用備份節點修復路徑，以有效減少控制訊號在網路上造成碰撞，提高網路傳輸效能。最後以路燈監控之模擬實現來評估效能上的改善，經由模擬結果 AODV-BN 確實可減少路徑搜尋控制封包的數量近而改善 Throughput。

Abstract—Zigbee is a non-infrastructural wireless network which has low data rate and self-reconstruct capacity. There are two routing algorithms for Zigbee mesh network : Tree routing and On Demand routing. This paper attempts to using On Demand routing protocol to construct the data transfer path in symmetric street light structure. Like AODV(Ad-hoc On-demand Distance vector), we must reduce control message to find the route in limited bandwidth network. This paper proposes AODV-BN (Backup Node) according to AODV-BR. AODV-BN only use RREP message to find the

back up node and get the neighbor table by eavesdropping instead of sending Hello Message. So we can use very small amount of control messages to fix the broken route and reduce the collision. Finally, this paper use NS-2 (Network Simulation) to simulate the performance of AODV-BN. The simulation results present that our methods reduce the control message indeed and improve the throughput of the network.

關鍵詞—AODV、備份路徑、備份節點

Key word—AODV, backup route, backup node

一、簡介

Zigbee 是以 IEEE802.15.4 為基礎，發展出的短距離無線通訊技術，屬於無基礎架構無線網路，並具網路自我修復能力，若部分節點設備損壞，Zigbee 網路也可自我修復而將人力需求減低。目前 Zigbee 聯盟針對 Multi-hop[8]的網路架構提出兩種路由協議：樹狀路由及網狀路由。樹狀路由利用階層架構及節點位址即可將封包傳送至目的端節點，不需儲存路由表並做路由選擇；但當應用越來越廣，樹狀路由在大型網路自我組態前無法得知網路拓撲，不易決定 tree 參數，會造成節點位址無法取得的問題。而網狀路由則需維護路由表及路由發現表，繞徑方式類似無線隨意網路網狀路由繞徑方式，所需花費的時

間相對於樹狀路由較長，但在穩定具週期性的傳輸下，其效能表現與樹路由相當，故本論文以對稱式路燈架構下採用網狀路由之 On demand routing 演算法建立資料傳送路徑。一般 on-demand 型式的繞徑協定只在起始端節點及目的端節點間建立單一條封包傳送路徑，當路徑中某節點因電能耗盡時，路徑即發生斷裂，斷裂節點因為沒有其他路徑可以傳送封包，所以只能將這些封包丟掉或是暫存起來。並且重新啟動路徑搜尋機制來找尋新的路徑，如此將增加資料傳輸延遲及降低傳輸效能。故針對此問題本論文使用備份節點來解決，本論文修改 Sung-JuLee 和 Mario Gerla 提出的備份路徑的方法，在 RREP 封包中多加一個上一跳節點位址，即可在不增加其他控制訊號，使用原本就有的 RREP 控制訊號來達到備份節點的建置。

二、相關研究

目前已有許多針對無線隨意網路所提出的繞徑演算法被提出[2][3][4]，這些演算法可以粗略的分為兩種不同的類型：Reactive 和 Proactive。Proactive 會預先尋得所有起始端節點及目的端節點之間的路徑，而不管這些路徑是否有需要用到。而 reactive 則採取 on-demand 型式的繞徑演算法，是在需要傳送資料到目的端節點時才會進行路徑搜尋，這樣可以節省不需使用路徑的搜尋及儲存負擔。

On-demand 型式的繞徑演算法主要有下列幾種：

AODV[1]只建置和維護需要用到的路徑，有三大主要程序：Route Request、Route Reply、Route Maintenance。當節點有資料封包要傳送到某個 Destination 時，Source node 會先查詢 Routing table 中是否有到此 Destination 的路徑資訊，若有則根據此資訊將資料封包傳送到 next hop node。若沒有則啟動 Route Request 程序，以 Flooding 方式廣播 RREQ 封包，其中以 Source

node IP 及 Broadcast id 唯一識別此 RREQ 封包，避免同一個 RREQ 封包重複廣播，同時建立 Reverse Route，讓 Destination 回應的 RREP 封包可循此路徑送到 Source Node。

RREQ 傳送的過程中若中間節點的 Routing table 有到達此 Destination 的路徑，則此 node 就可以回覆 RREP 封包。若沒有則由 Destination 回應 RREP。RREP 是以 unicast 方式傳送，並於傳送過程中建立 Forward route。

路徑維護方式大致分為 Routing Table 內資訊的維護以及路徑斷裂時的處置。Routing table 有一欄位記錄 expire time，當此路徑一段時間沒有被使用或更新時，便會從 Routing Table 中移除。而路徑斷裂時有兩種方式處置，第一是直接傳送 RERR 給 source node，沿途經過的 Node 會依序將此路徑的資訊清除，RERR 回到 Source Node 後，再由 Source Node 重新建立一條到達 Destination 的路徑。另一種作法為啟動 Local Repair 機制，當斷裂點離 Destination Node 較近時，由斷裂點直接發處 RREQ 封包來建立到 Destination Node 的路徑。

在 AODV 中每個節點會保留自己的序號 SN(S) 及其他目的節點的序號 SN(D)來達到 loop free 的能力，此 Sequence number 均為單調增加。當一節點收到 RREQ 或 RREP 時，會比較繞徑封包中攜帶的 SN(D)與路徑表中的 SN，若繞徑封包的 SN(D)大於路徑表中的 SN 時才更新路徑表資訊。另外當路徑斷裂或過期時，斷裂節點會將無法到達的 SN(D)加一來避免迴圈

AODV-BR 的路徑演算法在 route request phase 和 AODV 一樣，沒有做任何修改，主要是在 route reply phase 時建構備份路徑。節點藉由竊聽鄰居節點的 RREP 封包，選擇最佳的路徑並將鄰居節點紀錄成到目的節點的 next hop，建立在 alternate table 中。

當主要路徑斷裂時，才會使用此 Alternate

table 中的備份路徑，並向 source node 方向傳送 RERR 將原本主要路徑以及依附其上的備份路徑拆掉。

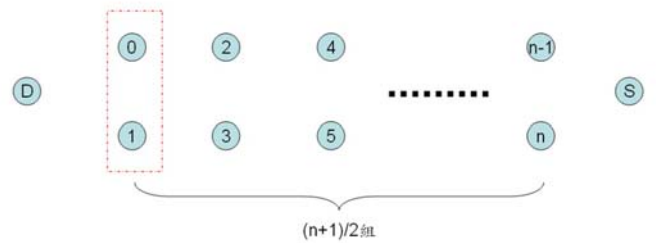
當主要路徑斷裂時(如 MAC 層沒有收到 ACK 時，或一段時間沒有收到鄰居節點的 hello message)，節點會在資料封包標頭註明此資料封包是因為路徑斷裂，並將封包做一步式的廣播(one hop data broadcast)給周圍鄰居節點，接收到此封包的節點會查看自己的備份路徑記錄表是否有到達目的節點的紀錄，若有則以 unicast 方式傳送到下一節點。

AODVjr[5]為 AODV 的精簡版，和 AODV 在不同的節點移動速度下幾乎有相同的 packet deliver 表現。它移除了 AODV 規格中的目的節點序號、hop count、hello messages、rerr、precursor list 項目。沒有了目的節點序號，則必須由目的節點對 RREQ 做回應，而不能由中間節點查詢路由表後直接回應 RREP，且 Destination node 只回應最快收到的 RREQ，所以可得到一條最快的路徑。且因為只有 Destination 可以回應 RREP 給 source node，所以 hello messages、RERR、precursor list 都不再需要。

三、設計理念

在對稱式路燈架構中，因分佈節點兩兩對稱，因此考慮若其中某一節點離網，另一節點能夠馬上替代它修復斷裂路徑，以節省重找路徑所須耗費的電量、及產生的大量控制封包造成的封包碰撞，提高 throughput。

考慮若路燈節點分佈如圖一所示，相鄰兩節點為一組，共有 $(n+1)/2$ 組。資料由節點 S 傳送到節點 D，假設節點離網機率為 P_e ，則一組節點均離網的機率為 P_e^2 ，故任一組節點至少有一點正常即可通行，可通行機率為 $1 - P_e^2$ ，若每組均可通行則 S 到 D 點才可做資料傳輸，此時的機率為 $(1 - P_e^2)^{(n+1)/2}$ ，由此可知，若不考慮其他造成路徑失敗的原因如干擾、碰撞等，節點 S 到 D 的成功機會最大不會超過 $(1 - P_e^2)^{(n+1)/2}$ 。



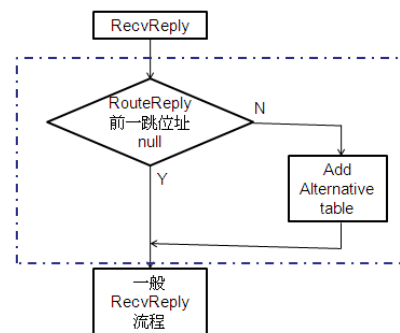
圖一、對稱路燈架構圖

為了提高路徑搜尋成功的機率，減少控制封包的數量可以有效減少封包碰撞的機會。在前面提到的 AODV-BR 繞徑協定中利用 RREP 控制封包來建立備份路徑是個滿理想的方法，它可在不增加額外控制訊號及簡易的路徑維護機制做到備份路徑的使用，但此演算法是用在因節點移動造成主要路徑上 link 斷裂但節點仍正常運作的情形。在對稱式路燈架構下，若發生監測節點電能耗盡或損壞時，AODV-BR 繞徑協定則無法找到備份路徑來傳送資料，故本論文擴充 AODV-BR 演算法使其跨過損壞節點，並進一步修復斷裂的主要路徑。

四、AODV-BN 機制設計

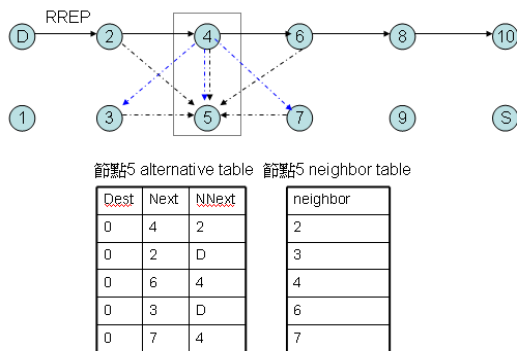
(一) Backup Node 備份節點建置

與 AODV-BR 相同，備份節點在 RREP 階段時，利用無線傳輸廣播特性接收鄰近節點的 RREP 封包，並將資訊記錄在 Alternative table 中，不過我們在 RREP 封包增加一個前一 hop 節點的 IP，並在 AODV RecvReply 流程中加入竊聽 RREP 封包的流程，如圖二框中所示。



圖二、建立備份結點示意圖

在對稱式路燈架構下，每個節點定時會將監測資訊傳回匯集節點，故節點 5 會竊聽到來自四周鄰居節點的 RREP，並將相關資訊登錄在 Alternative table 中，例如收到節點 4 的 RREP 封包，會將目的節點紀錄在 Alternative table 中 dest 欄位、傳送 RREP 封包的節點紀錄在 next 欄位、並將前一 hop 節點紀錄在 nnext 欄位。且同一鄰居只紀錄最新一筆。作法如下圖三黑色虛線所示。同理，節點 4 發送的 RREP 也會被鄰近結點竊聽並紀錄，如圖十一藍色虛線所示。另外因各節點均會定時回報監測資訊給彙集節點，故可在鄰居節點傳送資料時竊聽並紀錄以獲得鄰居節點資訊。此方式可避免主動詢問鄰居節點資訊需定時發送 hello message 造成額外訊息傳送負擔。



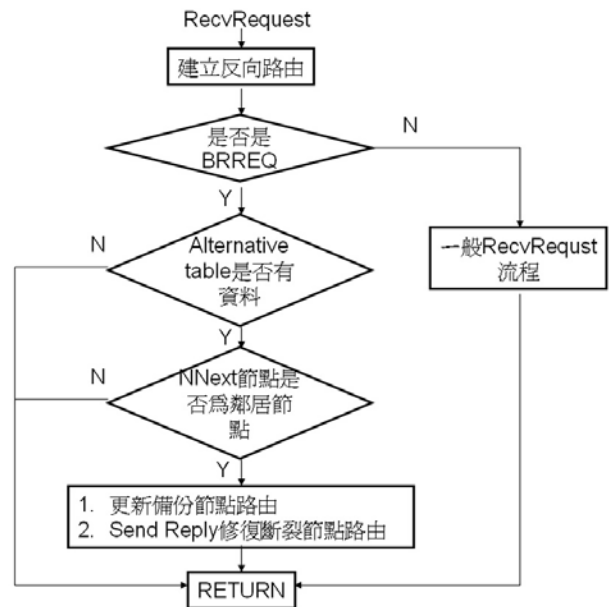
圖三、竊聽 RREP 封包來源

(二) 備份節點使用

當節點 4 離網時，主要路徑即告斷裂，此時節點 6 會單播 TTL=1，目的節點為 0，下一跳節點為節點 4 的 BRREQ (Backup RREQ) 控制訊號，並在此 BRREQ 封包的標頭上註明此資料封包是因為路徑斷裂，而鄰近節點在接收到標頭上有註明之 BRREQ 封包後，查詢 Alternative table 是否有 dest 為目的節點(節點 0)、next hop 為 BRREQ 廣播節點下一跳(節點 4)的紀錄，若有則檢查 neighbor table 是否有節點 2。若節點 2 在鄰居節點表中則節點 4 回應 RREP 給節點 6。斷裂

節點可根據收到的 RREP 封包更新 routing table。檢查 neighbor table 可避免節點 6 和節點 7 回應 RREP 造成 loop。控制流程如圖四所示。

若斷裂節點在發出 BRREQ 一段時間後若沒有收到 RREP 的回應，則啟動原始 AODV 路徑斷裂時的機制，如 Local Repair 或向 Source node 發送 RERR，如此當節點損壞時可立即由備份節點取代損壞節點位置做資料轉傳並修復路徑，避免路徑重新建立所造成的大量路徑搜尋封包可能造成的碰撞及提高網路 throughput，及降低封包丟棄率。



圖四、備份節點修復斷裂路徑流程圖

(三) 單向路徑搜尋建立雙向路由

在 AODV 路徑廣播搜尋的過程中，網路上所有接收到路徑搜尋訊息的節點會在繞路表中建立回傳來源節點的回傳路徑資訊，但只有在目的節點回傳路徑上才可建立雙向路徑，其他節點建立的回傳路徑則會在一段時間後自動刪除。在路燈網路架構中，因匯集節點固定且只有匯集節點跟路燈節點之間會做資料交換，可由匯集節點向末端結點啟動路徑搜尋，並參考自主學習路由協定[7]，保留這些暫存到匯集節點的單向路徑，將來那些只有單向路徑的節點即可以單播方式

建立到匯集節點的雙向路徑。如此可避免發起耗電且易造成大量封包的廣播搜尋。

五、模擬與驗證

(一) 模擬環境與設定介紹

本論文以 ns2[6] 模擬實驗來驗證以及比較 AODV、AODV-local repair、AODV-BN 等路徑搜尋法在本論文提出的路燈架構下，當節點數量變化時廣播次數、封包碰撞及效能進行比較。

本模擬環境作以下的設定：

1. 本模擬比較的三種路徑搜尋法均採用類似 AODV-jr 的方式，必須由目的節點回應 RREP，以建立相同的比較基準。因為若可由中間節點回應 RREP 將造成路徑集中，使一條路徑中同時包含同一組備用節點，造成 AODV-BN 失敗。

2. 為使本模擬單純只統計節點離網的效能，排除各繞徑協定在建立路由時耗費的時間不同。模擬實驗 10000 秒時間，前 1000 秒用於事先建立各點到彙集節點的路徑及 Alternative table、neighbor table 資料，並且一旦建立了 routing table 及 neighbor table 的紀錄後即不會 time out，只會做 update。統計效能時不包含事先建立各節點到彙集節點路徑時的封包數。

3. 在模擬時間 2000 秒到 10000 秒的時間內，每 40 秒會有一節點隨機離網，但同時間只有一節點會離網，其機率分佈採用 uniform distribution。

(二) 模擬實驗

在實驗的過程中若當路徑中包含同一組節點時此備份節點則無法使用，如此將會啟動 local repair 機制或重新由原始節點啟動路徑搜尋機制。

本論文以一個對稱式路燈架構，模擬路寬 15 公尺，路燈距離 30 公尺，每個節點訊息廣播半徑為 40 公尺，傳輸模型採用 Two Ray Ground，

節點 0 為匯集節點，從最遠節點傳送 CBR/UDP 的資料到彙集節點。並設定 CBR 的 packet size 為 80bytes，模擬節點隨機離網的情況，並比較不同路燈數各種繞徑演算法的效能。

首先下圖分別是各種路徑搜尋法在上述模擬環境下對不同節點個數之路徑搜尋次數、RREQ 廣播次數、訊息碰撞次數、端對端平均延遲及傳遞訊息量的模擬實驗結果。名詞定義如下：

路徑搜尋次數：來源節點因無到目的節點的路徑時所發起的路徑搜尋次數。

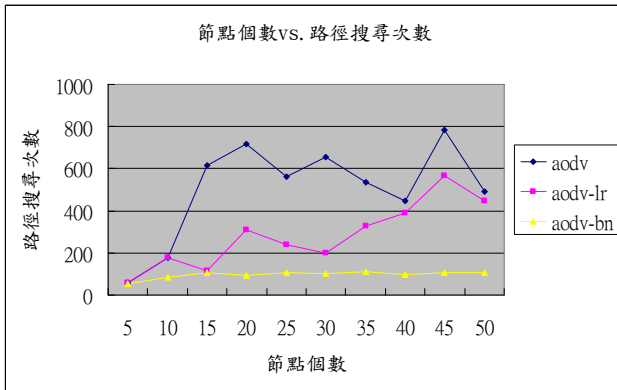
RREQ 廣播次數：包含來源節點路徑搜尋所發的 RREQ 次數、中間節點轉發 RREQ 次數、及路徑斷裂修復時所發的 RREQ 次數。

端對端平均延遲：所有 CBR 封包由來源節點到目的節點的平均延遲時間。

由實驗中可看出 AODV 協定因路徑斷裂重新尋找須發送大量 RREQ 封包，常造成封包碰撞而失敗，需反覆重新做尋找路徑的動作。使用 AODV-Local repair 可縮短尋找路徑時的 hop，有效減少了 RREQ 的數量及碰撞次數，而本論文提出的備份節點演算法可再進一步的大量減少 RREQ 數量，由實驗數據可看出 AODV-BN 在靜態路燈架構下其所發出的 RREQ 數量及所造成的碰撞數都遠較其他繞徑協定減少許多，並且成功傳輸的資料量也較其它協定優異。

1. 節點個數 vs. 路徑搜尋次數

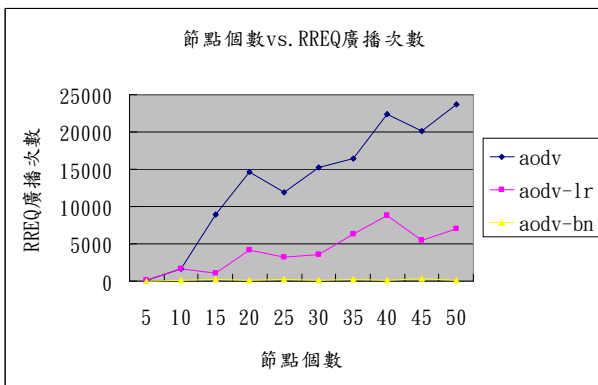
在圖五中我們比較三種協定在不同的節點個數下，發起路徑搜尋的次數。因 AODV-LR 只有在斷點離目的節點 hop count 小於二分之一 network diameter 時才有作用。故在 5 個節點及 10 個節點時 AODV-LR 和 AODV 運作方式是一樣的。



圖五、節點個數 vs. 路徑搜尋次數

2. 節點個數 vs. RREQ 廣播次數

由圖六我們比較三種協定的 RREQ 廣播次數，AODV 因為每次路徑斷裂均須由 Source node 重新找尋路徑，經過較多的中間節點轉發及較多碰撞造成的重新搜尋路徑，故發送的 RREQ 廣播次數最多。AODV-lr 若較接近目的節點，可由斷裂節點處發送路徑找尋控制訊號，故可減少 RREQ 廣播次數。但 AODV-BN 在單一節點離網時，發出的 BRREQ TTL 只有 1，可將修復路徑所發出的控制封包限制範圍在 1 hop 以內。故所發出的 RREQ 廣播次數最少。



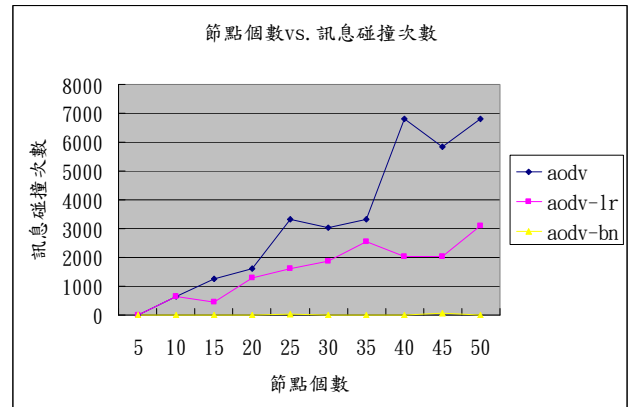
圖六、節點個數 vs. RREQ 廣播次數

3. 節點個數 vs. 訊息碰撞次數

在圖七中，因 AODV 由 Source node 重新找路須經過較遠的路徑，且發送的控制訊號較多，故造成較多的封包碰撞。AODV-lr 較少，而

AODV-BN 因為找尋路徑控制訊號只在 one hop 的範圍內發送，故幾乎不會發生碰撞。

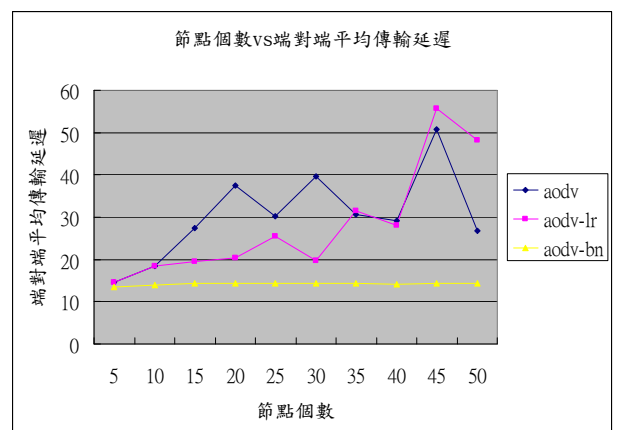
但實驗中若同次路徑搜尋封包重複碰撞並重送時，會逐漸延長重送間隔，反而造成 RREQ 封包總數減少，碰撞總數也較少的情形。



圖七、節點個數 vs. 訊息碰撞次數

4. 節點個數 vs. 端對端平均傳輸延遲

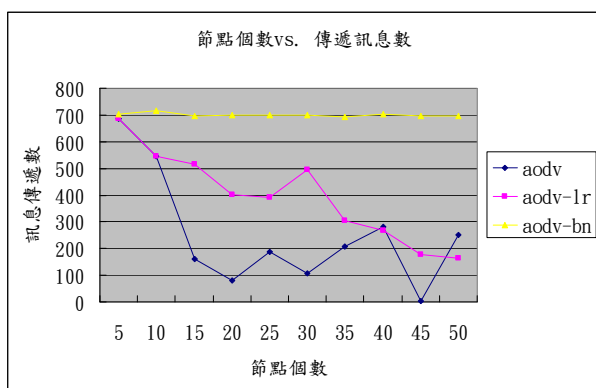
圖八為節點個數 vs. 端對端平均傳輸延遲，當路徑斷裂時由於 AODV-BN 可在最短的時間內修復路徑並將資料傳送到目的端，故其端對端平均傳輸延遲時間最短，而 AODV-LR 也較優於 AODV。但實驗中若 AODV-LR 有較多次的失敗導致重新啟動路徑搜尋時，則 AODV-LR 有時也會有較差的表現。



圖八、節點個數 vs. 端對端平均傳輸延遲

5. 節點個數 vs. 傳輸訊息數

圖九為節點個數 vs. 傳輸訊息數，AODV-BN 因為可以在短時間內修復斷裂路徑，其訊息傳遞數非常地穩定維持一定傳輸量。驗證 AODV-BN 確實可減少路徑搜尋控制封包的數量近而改善 throughput。



圖九、節點個數 vs. 傳輸訊息數

六、結論

在這篇論文中，我們修改 AODV-BR 的方法，提出一種適合對稱路燈架構的 Routing algorithm AODV-BN，利用兩個一組的節點互相做為備份節點，在不增加控制封包的情況下，快速修復斷裂路徑，並可減少搜尋路徑時的控制封包數量及改善封包碰撞情形。

原先的 AODV 因為路徑斷裂時沒有作修復動作，須要由 Source node 重新搜尋路徑，而模擬結果的效能表現較 AODV-LR 差。AODV-BN 進一步將修復斷裂路徑的動作控制在 1 個 HOP 的範圍內，更有效的提供連線路徑的穩定性，提供網路維持一個穩定的 throughput。

七、參考文獻

- [1] C. E. Perkins, Royer, and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing", Internet Draft, draft-ietf-manet-aodv-13.txt, February 2003.
- [2] C.E. Perkin, and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobil Computers", In Proc, ACM SIGCOMM Conference (SIGCOMM 94), pp. 234-244, August 1993
- [3] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153-181. Kluwer Academic Publishers, 1996.
- [4] D.B. Johnson. "Routing in Ad Hoc Networks of Mobile Hosts", Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 158-163, IEEE Computer Society, Santa Cruz, CA, December 1994.
- [5] I. Chakeres, and L. Klein-Berndt, "AODVjr, AODV Simplified", Mobile Computing and Communications Review, Volumn 6, Number 3, pp. 1-2, July 2002.
- [6] ns2 simulation: <http://www.isi.edu/nsnam/ns/>
- [7] R. Peng, S. Mao-heng, Z. You-min, "Zigbee Routing Selection Strategy Based on Data Services and Energy-balanced Zigbee Routing", 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06), pp.400-404, 2006.
- [8] Zigbee Alliance, Network Layer Specification 1.0, Dec. 2004.