

## 無線感測網路之可應變式多點跳躍 MAC 排程機制

簡碩瑤 國立台中教育大學 資訊工程學系 vxellery@gmail.com 鍾銘山 雪霸國家公園	張林煌* 國立台中教育大學 資訊工程學系 lchang@mail.ntcu.edu.tw	吳忠宏 國立台中教育大學 永續觀光暨遊憩管理 研究所 hcwu@ms3.ntcu.edu.tw	林青 雪霸國家公園 徐嘉佑 國立台中教育大學 資訊工程學系 zuoi4321@hotmail.com
	李宗翰 國立台中教育大學 資訊工程學系 thlee@mail.ntcu.edu.tw		

### 摘要

近年來，無線感測網路(Wireless Sensor Networks, WSN)相關技術已發展日益成熟與進步，然而，在能源消耗與延遲的議題上仍然是主要的研究議題，因為感測節點在硬體上資源較為嚴苛，其運算速度、頻寬、能源等較為有限，其中節省電源與延遲時間是一個難以兼顧的需求，因此設計一個具節能且提供低延遲服務之解決方案是本論文欲解決的目標。

在 WSN 中，近年來在多媒體資料的傳送議題上逐漸受到矚目，而此類型的流量對於服務品質(Quality of Service, QoS)較為要求，其中碰撞不僅影響吞吐量，也影響著延遲時間，延遲時間的增加，則會影響即時性的流量類型，例如緊急訊息無法在短時間內送至伺服器端，而錯過黃金救援時間等。

本論文將延伸 RMAC(Routing enhanced MAC) 機制並結合無線感測網路技術，設計具節能且提供 QoS 保證傳輸排程機制。而具節能及 QoS 感知之多點跳躍無線感測網路之方法我們稱 AMH-MAC(Adaptation Multi-Hopping MAC)。基於 RMAC 機制，我們藉由動態調整傳輸功率、動態調整多點跳躍數量，達到降低延遲時間以及省能效益。最後使用學術界具公信力之 Ns-2(Network Simulator 2) 網路模擬器來展現我們改善了點對點延遲時間及能源效益，並且與 RMAC 機制做比較。

**關鍵詞：**RMAC、動態調整傳輸功率、排程、多點跳躍

### 1. 前言

無線網路的快速發展下，帶動許多無線網路相關技術興起，包含 IEEE 802.11[1]系列是目前最為盛行的無線網路標準，其主要應用於區域無線網路(Wireless Local Area Network, WLAN)中，而目前逐漸引起話題的無線感測網路(Wireless Sensor Network, WSN)，其應用範圍有環境感測、目標追蹤等，然而應用於不同服務類型的網路環境中，

WSN 具有低成本與低消耗功率之特性，由多個感測節點藉由無線通訊的方式進行溝通，適合應用在具臨時性且需要快速達到資訊傳播的場合，例如應用於山區急難救援、礦坑、農場監控等。

在 WSN 相關研究中，較廣為人知的 S-MAC[2]、T-MAC[3]、DSMAC[4]及 RMAC[5]等，皆為競爭基礎與具時間同步類型的多媒體存取控制(Medium access control, MAC)協定，其中 RMAC 利用跨層資訊如路由資訊，在傳送資料之前利用控制封包進行傳輸排程，使得資料封包能夠在一個循環(Duty cycle)時間內進行多點跳躍傳輸，而 S-MAC、T-MAC 等則是利用公平競爭方式進行資料傳輸，缺點是競爭者較多時會增加競爭失敗的機率，使得節點無法將剛產生的封包立刻送出，進而使得延遲時間增加，以及因競爭而導致的耗能，此外，由於時間同步機制，當節點不需要傳送封包時，其閒置時間亦會造成能源消耗。

B-MAC[6]、WiseMAC[7]、X-MAC[8]等 MAC 協定是一個 Low Power Listening (LPL)機制，之所以稱為 Low Power 的原因在於節點僅喚醒一小段時間，這段時間內沒有收到任何的前導封包，便接著進入睡眠來節省更多的能源，雖然能夠節省能源消耗，但因為無法在一個 cycle 內進行多點跳躍，使得延遲時間較長。

因此，在延遲的考量下我們選擇使用具時間同步機制與多點排程機制的 RMAC 協定作為機制改良的對象，我們結合類似 LPL 機制的方法降低 RMAC 閒置時間的能源消耗，藉由動態調整多點跳躍數量來增加跳躍的節點數以利降低延遲時間，以及動態調整傳輸功率的方法來降低傳輸的能源消耗，並設計動態調整競爭視窗，達到具節能且低延遲之多點跳躍傳輸排程機制。

### 2. 文獻探討

S-MAC(Sensor-MAC)較廣為人知，至今仍有許多相關議題持續討論，而 S-MAC 的機制較為單純，主要運作流程為固定時間內喚醒、睡眠，其整個 Duty-cycle 可分為同步 - 資料 - 睡眠

(SYNC-DATA-SLEEP)期間。S-MAC 是由 IEEE 802.11 DCF 模式改良而成，他保有 IEEE 802.11 中用以解決隱藏節點 (Hidden Terminal) 問題的 RTS/CTS 機制與 CSMA/CA 機制。此外也新增時間同步功能，時間同步是避免節點因時間偏移，使得彼此無法順利溝通，除此之外，同步喚醒有利於傳送資料時能夠快速地找到目的節點，相較於非時間同步喚醒機制，更能降低等待節點喚醒的時間。

與 S-MAC 相同的機制有 T-MAC(Timeout MAC)、DSMAC(Dynamic Sensor MAC)、RMAC(Routing enhanced MAC)、DW-MAC[9](Demand Wakeup MAC)、P-MAC[10](Pipeline MAC)、AS-MAC[11](Adaptive Scheduling MAC)、PRMAC[12](Pipelined Routing enhanced MAC)等，其共同的特點為時間同步基礎的 MAC 機制，其中 RMAC、DW-MAC、P-MAC、AS-MAC 及 PRMAC 是具有多點傳輸排程機制，相較於同時僅能做點對點傳輸 S-MAC、T-MAC 與 DSMAC 等機制，單一週期的進行多點傳輸排程機制能有效降低延遲時間。

S. Du 等研究學者提出 RMAC 協定改良 S-MAC 使其達到多點跳躍傳送，RMAC 將 RTS/CTS 封包結合為單一封包，稱為 PION 訊框 (Pioneer Frame)，利用 PION 訊框在 DATA 期間依序傳送至其他節點告知接下來的資料傳送時間，各節點即可推算在 SLEEP 期間時需要在何時喚醒轉發資料，如此排成便可達到多點跳躍傳輸，其傳送流程示意圖如圖 1 所示。

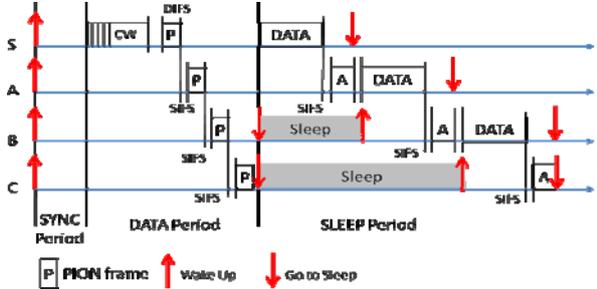


圖 1 RMAC 傳送流程示意圖

RMAC 機制中，將 Duty cycle 分為三個階段，第一階段為 SYNC 期間，此期間主要作為節點之間交換時間訊息所用，當節點同時在此階段喚醒時，節點會發送 SYNC 封包交換彼此的睡眠時間資訊，睡眠時間資訊作為節點調整 Duty cycle 時間的依據，也用於避免時間偏移。第二階段為 DATA 期間，此期間主要作為傳送排程控制封包使用，過去 S-MAC 藉由 RTS/CTS 來避免隱藏節點問題，而在 RMAC 中則將 RTS/CTS 進行結合，並命名為 PION 訊框，該封包包含了路由層資訊，例如最後目的地址、跳點數。PION 訊框功能主要作為預先傳輸排程使用，來源節點產生 PION 訊框並送給下一個跳點，當下一個節點收到時，便會透過路由層資訊尋找下一跳點位址接著送出，而送出 PION 的節點會藉由監聽(Overhearing)得知下一跳點是否已接收並繼續轉送。

綜合以上，在時間同步機制中，降低延遲的方法除了增加活動時間外，利用控制封包達到多點跳躍傳輸排程方法，能夠比 S-MAC、T-MAC 或 DSMAC 達到低延遲目標，本論文將延伸 RMAC 機制加以改良。

### 3. 動態調整多點跳躍傳輸機制

在本章節將描述我們提出的方法，其中主要為動態調整多點跳躍傳輸之方法、基於 RSSI 之動態調整傳輸功率之方法及基於動態調整競爭視窗之 QoS 設計。

Duty cycle 中共分為三個期間，其中 SYNC 期間與 DATA 期間是必須喚醒的時間，前者是為了定期地交換睡眠時間資訊以達到時間同步的目的，然而，後者則不論節點需要傳送資料與否都會醒來，倘若一條路徑上所有的節點都沒有資料需要進行傳送時，則在 DATA 期間喚醒是一種閒置能源的浪費。其中 RMAC 主要目的是達到單一週期進行多點跳躍排程機制，其中關鍵的 DATA 期間長度組成為：

$$T_{DATA} = CW \mid DIFS \mid durPION \mid N \cdot (SIFS \mid durPION) \quad (1)$$

$T_{DATA}$  為 DATA 期間之時間長度，CW 為競爭視窗， $durPION$  為 PION 訊框之傳送時間長度， $N$  值代表  $N$  個 PION 訊框數量； $N$  值為主要影響 DATA 期間長度的關鍵因素之一，其餘數值為固定大小，因此我們可知道  $N$  值的大小影響 PION 訊框在 DATA 期間能夠繞送多少至節點。

在 K.T. Cho 等研究學者[13]的研究中，發現 RMAC 在不同路徑下使用不同的  $N$  值時能達到延遲時間最佳化，藉由最佳化的  $N$  值帶來更低的延遲時間。然而，該作者研究中並未提出動態改變  $N$  值的方法，由於節點可能同時服務兩個以上的路徑，而不同的路徑亦需要不同的  $N$  值，因此我們提出動態調整  $N$  值的方法，其示意圖如圖 2 所示。

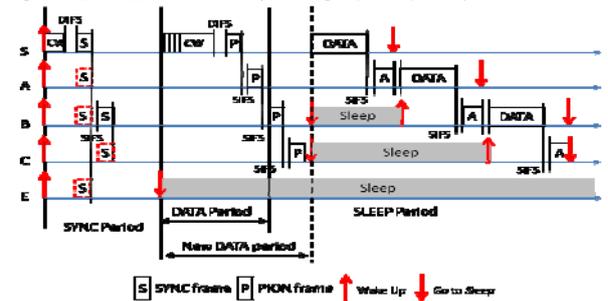


圖 2 AMH-MAC 由新  $N$  值計算出新 DATA 期間之運作流程示意圖

圖 2 為節點從 Sync 封包夾帶的  $N$  值重新計算 DATA 期間長度，其中  $N$  值代表 PION 訊框繞送數量也意味著跳點數量， $N$  值的增加延長節點的 DATA 期間長度、縮短睡眠時間，使得封包能夠利用較長的活動時間繞送 PION 訊框，但也使得節點必須喚醒更長的時間來處理封包；然而，縮短的睡眠時間使節點能夠更快地來到下一次 Duty cycle，繼續將封包轉送至 Sink 端，延遲時間的降低，卻帶來能源消耗的增加，這是一個耗能與延遲之間的取

捨。

綜合上述，在動態調整多點跳躍的部分，首先我們保留 Sync 封包功能並額外增加 N 值、路徑 ID 及 RSSI 值欄位，如圖 3 所示。

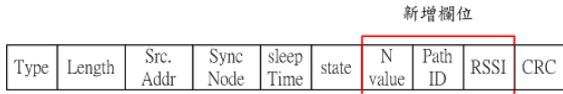


圖 3 新增後的 Sync 封包格式

路徑 ID 用於判斷接收 Sync 封包是否與自身路徑 ID 相符，相符的路徑 ID 代表節點需要在 DATA 期間喚醒進行傳送/接收封包，若路徑 ID 不符合，節點便不會在 DATA 期間喚醒，而是選擇繼續睡眠直到下一個週期開始；需要傳送/接收的節點將依據 Sync 封包中的 N 值來調整自身的 DATA 期間長度，延長的 DATA 期間將佔去一部份睡眠時間，換句話說，我們在固定長度的週期時間內，延長活動時間、縮短睡眠時間達到低延遲目標。

### 3.1 基於 RSSI 之動態調整傳輸功率

Sync 封包是一個廣播封包，當節點收到 Sync 封包時，將會持續廣播直到 SYNC 期間結束，然而，在節點密度較高、訊號覆蓋範圍內具多個節點的情況下，一個 Sync 封包能讓多個節點同時接收，並由訊號覆蓋重疊的相鄰節點進行傳送，使得轉送 Sync 封包變得較沒效率。

為了解決較無效率的轉送 Sync 封包，我們亦設計動態調整傳輸功率之方法。我們在 Sync 封包中新增 RSSI 值欄位，當 SYNC 期間時，節點互交換彼此的 Sync 封包，建立鄰居節點清單，而 RSSI 值也會保留至清單中，使節點之間能夠透過鄰居節點清單中的 RSSI 值調整傳輸功率。廣播 Sync 封包時以最大的傳輸功率傳送，讓 Sync 封包一次傳送給覆蓋範圍內多個節點，而節點收到 Sync 封包後藉由 RSSI 值與轉送之門檻值來決定是否繼續傳送，以此方法能有效減少廣播次數，且仍達到時間同步的功能；而資料封包及排程封包的傳輸，則從路由層得知下一個跳點後，在 MAC 層比對鄰居節點清單中下一個跳點之 RSSI 值，並依此值調整傳輸功率，如圖 4 所示。

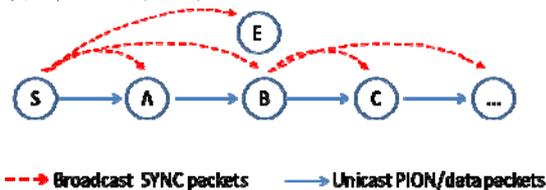


圖 4 Sync 封包傳送及功率調整傳送示意圖

當節點 S 需要傳送資料時，在 SYNC 期間以最大功率廣播 Sync 封包，而路徑上的節點收到此封包後，在距離節點 S 較遠的節點 B 則繼續轉送 Sync 封包，而距離節點 S 較近的節點 A 則將封包收入，檢查路徑 ID 若與自身路徑 ID 相同則記錄 N 值，節點 B 也在檢查路徑 ID 及記錄 N 值後接著傳送；其中節點 E 雖然收到 Sync 封包，但該封包與自身的

路徑 ID 不符，因此不會記錄任何資訊，在 DATA 期間到來時，由於沒收到任何需要在 DATA 期間喚醒並轉送的通知，而切換至 SLEEP 期間，節省 DATA 期間的閒置聆聽耗能。

## 4. 效能分析

為了量測我們提出的方法，我們使用網路模擬器(Network Simulator Ver. 2, Ns2)來分析、模擬。主要參數設定如表 1 所示，其中 Propagation 模型部分我們使用 Free Space 模型。耗能設定參考 Xbee Pro 900Mhz[14]模組設定。

表 1 各參數設定

Parameters	S-MAC	RMAC	AMH-MAC
Bandwidth	20Kbps		
Rx Power	95.7mW		
Idle Power	95.7mW		
Sleep Power	0.00825mW		
Sync Range	100m	100m	200m
Data/PION Range	100m	100m	100m
Transmitted Power(200m)	Null	Null	24.5dBm
Transmitted Power(100m)	18dBm		
SYNC period(ms)	55.2		
DATA period(ms)	104.0	168.0	168.0
SLEEP period(ms)	3025.8	4241.8	4241.8
Long SLEEP period(ms)	Null	Null	4409.8
Total cycle(ms)	3158.0	4465.0	4465.0
Duty cycle	5%		
N value	Null	4	4

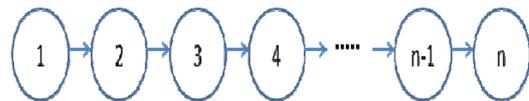


圖 5 鏈狀拓樸

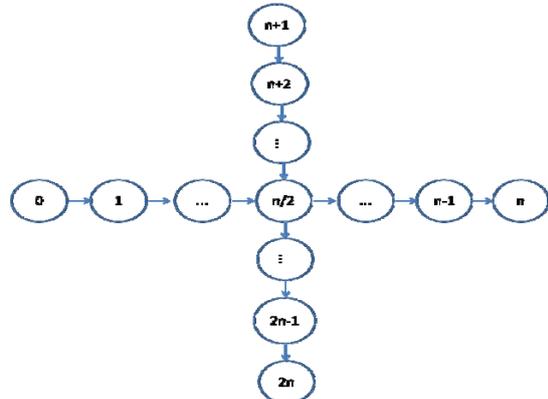


圖 6 交叉拓樸

我們延續 RMAC 機制進行改良,在方法比較上將與 RMAC 互相比較,來驗證我們提出的方法,此外並與 S-MAC 進行比較,由於 S-MAC 是一個較為知名的 WSN MAC 協定,且該協定也已被實作在各個嵌入式平台中,是一個非常適合做為比較的對象。

我們提出的 AMH-MAC 其 Duty cycle 時間參數基本上 RMAC 大致相同,並額外新增一項 Long SLEEP 睡眠參數,該參數為 DATA 期間與 SLEEP 期間之總和,主要作為節點在未有資料傳送或接收時,進入睡眠的時間長度;而在表 1 中,其 N 值是採用 RMAC 原有的預設值,在此我們亦使用相同的 N 值。資料流量的部分我們使用固定流量(Constant bit rate, CBR)來產生需要的資料量,而網路拓樸方面我們以鏈狀拓樸與交叉拓樸來進行模擬,如圖 5、圖 6 所示。

本實驗模擬時間 5500 秒共 24 跳點數,首先我們主要量測項目有延遲、能源消耗,以每 100 公尺設置一個節點的情境進行模擬,並與 RMAC、S-MAC 進行比較。

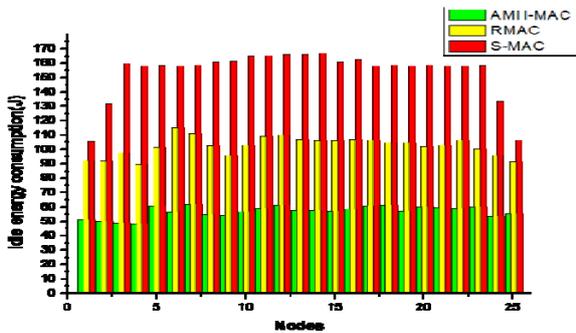


圖 7 平均閒置耗能

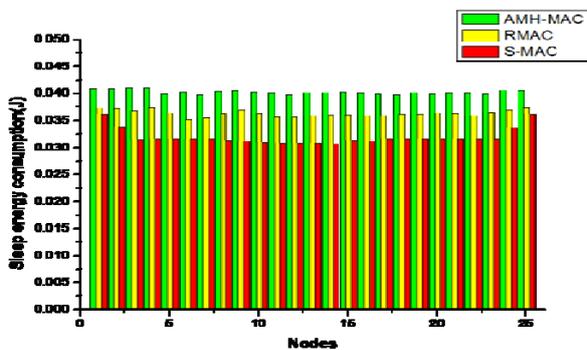


圖 8 平均睡眠耗能

圖 7 可看出閒置耗能中 AMH-MAC 較其他機制消耗較少能源,其中 S-MAC 由於單一 cycle 時間內僅做一個跳點傳輸,即使節點在不需傳輸或接收情況下仍在 DATA 期間喚醒,閒置時間的增加也促使能源消耗增加。而 RMAC 雖能在單一 cycle 時間內進行多點跳躍排程,當節點未有任何封包傳輸或接收時,仍會於 DATA 期間喚醒增加閒置能源消耗,並且增加額外接收封包的耗能風險。S-MAC 於 DATA 期間之時間長度較 RMAC 短,但傳送封包至 Sink 所花費之 cycle 數較 RMAC 長,使其閒置耗能較 RMAC 高。

睡眠的耗能中,AMH-MAC 在 SYNC 期間藉由路徑 ID 控制節點的喚醒時機,因此,非封包傳輸路徑的節點便於 DATA 期間進入睡眠,圖 8 中可看出 AMH-MAC 睡眠能源消耗皆比其他機制要來的高,代表 AMH-MAC 睡眠時間較長;S-MAC、RMAC 則是固定的睡眠時間,節點在 DATA 期間未收到或傳輸任何排程封包時,進入 SLEEP 期間後開始睡眠。

表 2 AMH-MAC 與 S-MAC、RMAC 之各狀態能源消耗

	AMH-MAC	S-MAC	RMAC
Idle energy consumption(J)	56.426	153.644	102.0316
Sleep energy consumption(J)	0.04	0.032	0.036
Avg. residual Energy(J)	940.162	843.314	894.654
Tx energy consumption(J)	0.969	0.759	0.89
Rx energy consumption(J)	2.404	2.251	2.387

AMH-MAC 所剩能源皆比 S-MAC、RMAC 高,其總體耗能分別較 S-MAC、RMAC 節省約 9.7%、4.6%,詳細耗能比較如表 2 所示,在閒置能源消耗的比較上,AMH-MAC 能較 S-MAC 節省約 9.7%、較 RMAC 節省約 4.6%,閒置能源消耗佔總消耗中是最大的能源消耗,有效節省閒置能源消耗,便能提升整體的剩餘能源;而在睡眠能源消耗部分,AMH-MAC 則較 S-MAC 節省 0.001%、較 RMAC 節省 0.0004%,睡眠的耗能功率較小,但從結果中可看出,由於路徑切換機制,使 AMH-MAC 具有較多的睡眠能源消耗,代表節點睡眠時間較長  
傳送耗能方面 AMH-MAC 較 S-MAC 高出約 0.021%、較 RMAC 高出約 0.008%,接收耗能部分則較 S-MAC 高出約 0.015%、較 RMAC 高出 0.002%;主要原因在於 SYNC 期間的轉送與接收能源消耗,雖然在 SYNC 期間降低廣播次數,但造成較遠距離的節點需要不斷的轉發,造成能源消耗;從模擬結果中可得知藉由路徑控制切換節點於 DATA 期間的喚醒與睡眠,能夠減少閒置能源消耗、提升節點剩餘能源。

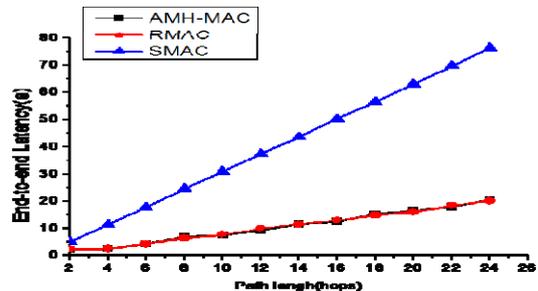


圖 9 以 24 hops 在鏈狀拓樸下的延遲比較

當 AMH-MAC 與 RMAC 使用相同 N 值時,會有相似的延遲表現。AMH-MAC 中影響延遲的關鍵因素在於睡眠時間長度;SYNC 期間我們喚醒多個

節點，當 N 值大於我們預設值 4 的時候，新的 N 值會計算出新的 DATA 期間，然而睡眠時間將會縮短，使得 RMAC 與 AMH-MAC 在相同的 N 值且 N 值大於 4 的時候，延遲時間會有所差距，而在圖 9 中，由於使用預設的 N 值，因此在延遲上兩者的結果會較為類似。

N 值的大小將影響 AMH-MAC 與 RMAC 延遲表現，因此我們針對 N 值進行延遲的模擬。

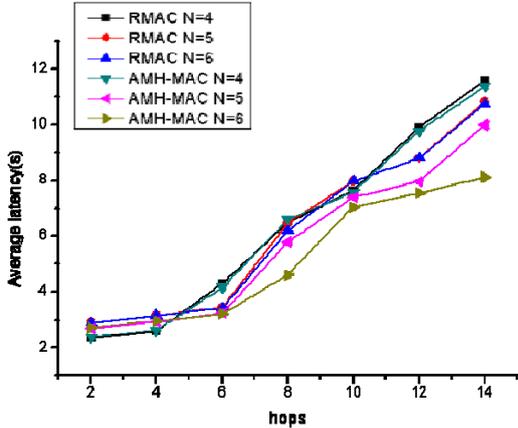


圖 10 以 14 hops、N 值為 4、5、6 之延遲比較

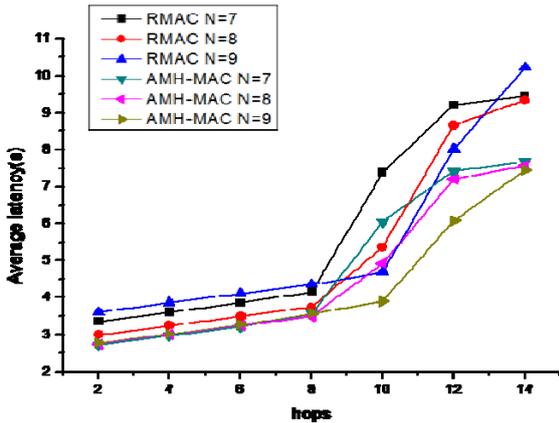


圖 11 以 14 hops、N 值為 7、8、9 之延遲比較

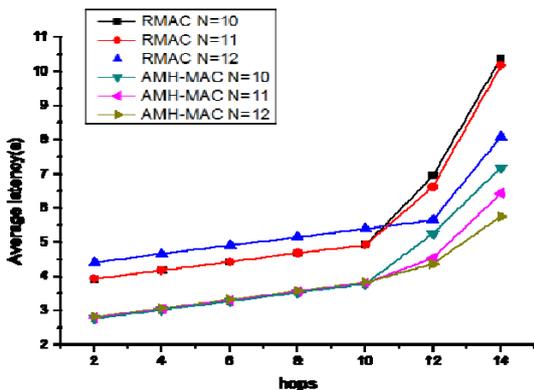


圖 12 以 14 hops、N 值為 10、11、12 之延遲比較

由圖 10、圖 11、圖 12 中可看出，隨著 hop 數增加，AMH-MAC 與 RMAC 相同 N 值比較下，其延遲時間皆較 RMAC 低，綜觀分析，AMH-MAC 在各 N 值的表現下，延遲平均較 RMAC 低。

交叉拓樸的分析中，我們模擬 21 個節點數、

每條鏈路 11 個節點，以預設的 N 值 4 來模擬延遲及節點能源消耗。

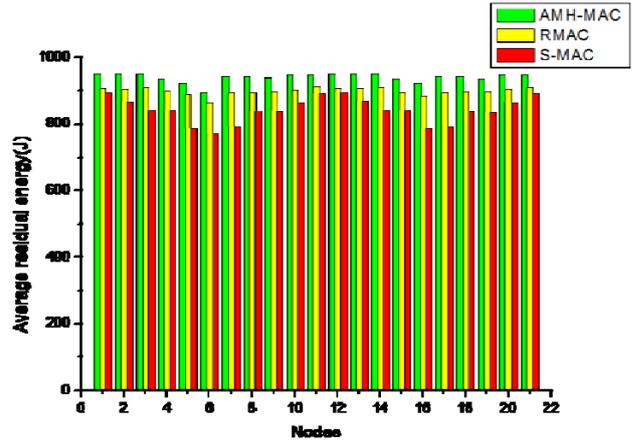


圖 13 交叉拓樸之平均剩餘節點能源

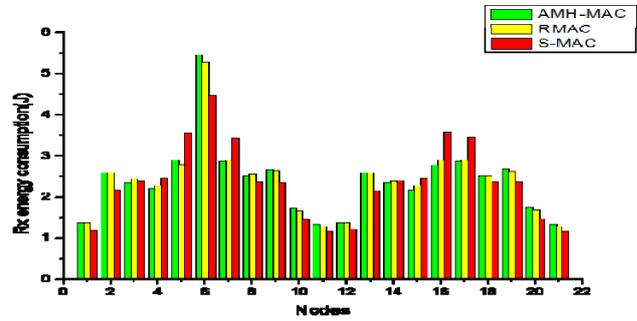


圖 14 交叉拓樸之平均接收能源消耗

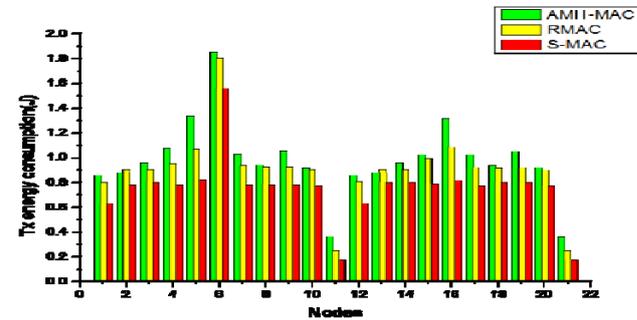


圖 15 交叉拓樸之平均傳送能源消耗

分析結果如圖 13、圖 14、圖 15 所示，其剩餘能源中可看出 S-MAC 能源消耗是三者之中較多的 MAC 機制，其中節點 6 為中間節點，其剩餘能源較其他節點少，因其必須轉送來自兩條路徑的封包，因此傳送、接收、閒置耗能都較其他節點高。而鄰近中間節點的鄰居節點，其耗能也會較高，主因在於隱藏節點的問題。

表 3 交叉拓樸之延遲時間比較表

	Avg. delay(s)	Max. delay(s)	Min. delay(s)
AMH-MAC	11.854	20.152	5.666
S-MAC	38.242	53.097	29.323
RMAC	11.91	32.971	5.265

表 3 為平均延遲時間的比較，我們將兩條鏈路之延遲時間相加取平均值；其中可看出 AMH-MAC 與 RMAC 極為相似，主要是因為在 N 值的部分同樣設為 4，具有相同的 Duty cycle 時間，然而，在

最大延遲的部分，AMH-MAC 使用路徑控制，使得在交叉拓樸情境下，兩條路徑能夠較為平均的傳送，而 RMAC 則是於 DATA 期間進行公平競爭，其固定 CW 長度亦是造成延遲上升的原因之一；S-MAC 由於每次 Duty cycle 僅做一次跳點傳送，在交叉拓樸情境下，遇到隱藏節點、競爭等問題使得其延遲時間較其他兩者高出許多。

## 5. 結論及未來研究

在過去的研究中，如 S-MAC 在單一 cycle 中以 1-hop 進行傳送，當需要繞送至 Sink 的節點數越多時，受到睡眠時間的影響，使得延遲時間越來越高；而亦有相關研究透過調整 Duty cycle 時間使延遲較 S-MAC 低，睡眠時間的降低，能夠帶來較低的延遲。然而降低睡眠時間對於延遲降低仍然有限，因此，在 RMAC 機制中提出單一 cycle 進行多點跳躍排程機制，節點之間使用共同的 Duty cycle，透過排程封包讓節點在睡眠時間進行資料傳送，減少 Duty cycle 回合數能夠有效降低延遲時間。

因此，我們提出 AMH-MAC (Adaptation Multi-Hopping MAC) 用於多點跳躍傳輸排程機制，並透過路徑控制、動態調整 N 值，達到降低延遲時間及能源消耗的目標；從實驗結果可明顯看出，在延遲表現上，具有同樣 N 值但因睡眠時間的縮短，使得延遲時間能夠較 RMAC 低；而能源消耗表現上，由於路徑的控制，使節點能夠節省閒置能源消耗，達到省能效益。

未來我們將針對交叉拓樸分析中，模擬各 N 值及更多 Hop 數之延遲與耗能分析，使整體機制更為完善。

## 6. 致謝

本計劃之執行承蒙國科會計畫(編號：NSC 101-2221-E-142-003、NSC 102-2221-E-142-009)，及教育部網路通訊人才培育先導型計畫-102 年度重點領域學程推廣計畫(發文字號：第 1020035480A 號)之支持，特此致謝。

## 參考文獻

- [1] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical (PHY) layer specifications: IEEE, 2012.
- [2] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks", Proceedings of INFOCOM 2002, pp. 1567-1576, 2002.
- [3] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", Proceedings of First International Conference on Embedded Networked Sensor Systems (SenSys 2003), pp. 171-180, Nov. 2003.
- [4] P. Lin, C. Qiao and X. Wang, "Medium Access Control With A Dynamic Duty Cycle For Sensor

Networks", Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), vol. 3, pp. 1534-1539, Mar. 2004.

- [5] S. Du, A. K. Saha, and D. B. Johnson, "RMAC: a routing-enhanced duty cycle MAC protocol for wireless sensor networks," Proceedings of INFOCOM 2007, pp. 1478-1486, May 2007.
- [6] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks", Proceedings of The Second ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 95-107, Nov. 2004.
- [7] A. El-Hoiydi, and J.-D. Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks." Algorithmic Aspects of Wireless Sensor Networks. Springer Berlin Heidelberg, 2004, 18-31.
- [8] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks", Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys), pp. 307-320, 2006.
- [9] Y. Sun, S. Du, O. Gurewitz and D. Johnson, "DW-MAC: a low latency, energy efficient demand wakeup MAC protocol for wireless sensor networks", Proceedings of the Ninth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 53-62, May 2008.
- [10] Y.Z. Zhao, M. Ma, C.Y. Miao and T.N. Nguyen, "An energy-efficient and low-latency MAC protocol with Adaptive Scheduling for multi-hop wireless sensor networks", Computer Communications, vol. 33, pp. 1452-1461, 2010.
- [11] F. Tong, W. Tang, R. Xie, L. Shu and Y.C. Kim, "P-MAC: A Cross-Layer Duty Cycle MAC Protocol Towards Pipelining for Wireless Sensor Networks", Proceedings of IEEE International Conference on Communications (ICC 2011), Jun. 5-9, 2011.
- [12] T. Canli and A. Khokhar, "PRMAC: Pipelined Routing Enhanced MAC Protocol for Wire-less Sensor Networks", Proceedings of IEEE International Conference on Communications (ICC 2009), Jun. 14-18, 2009.
- [13] K.T. Cho and S. Bahk, "Optimal Hop Extended MAC protocol for wireless sensor networks", Computer Networks, 2012.
- [14] Xbee Data Sheet, <http://www.digi.com>