

加強 IEEE 802.15.4 應用於災難偵測之通訊協定 Beacon Order Adaptation Algorithm (BOAA) 之省電效能的評估與比較

劉信賢

淡江大學電機工程學系

E-MAIL: 793350264@s93.tku.edu.tw

李維聰

淡江大學電機工程學系

E-MAIL: wtlee@fcu.edu.tw

張志豪

淡江大學電機工程學系

E-MAIL: 693390410@s93.tku.edu.tw

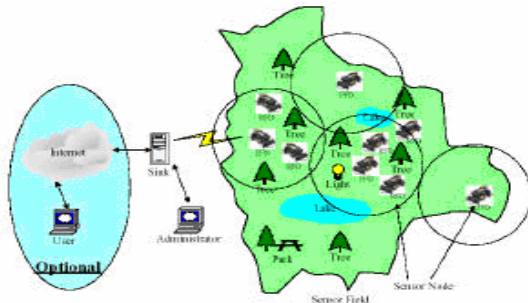
摘要

無線感測網路是近期資訊發展及微電腦科技下的衍生品，很多地方都可以看見它的影子，其應用的範圍很多，像人員的定位、氣象、土石流、海浪波幅、溫度、溼度及雨量上的監測等例子比比皆是，台灣是一個典型的海島國家，雨量豐沛，加上台灣山坡地過度開發、人口密集，土石流災害儼然成為台灣地區每年颶風豪雨季節民眾最關心的話題。另一面，IEEE 802.15.4 低速率無線個人區域網路 (Low-Rate Wireless Personal Area Network LR-WPAN) 發展已漸趨成熟，傳輸特性有低消耗功率、傳輸距離短、包含大氣、水文、山林、河流、保育動物，以及候鳥研究等領域環境都變化多端，非常適合建立感測資料，而且研究結果影響宏大。而在土石流偵測方面，由於主要是依據地聲以及土壤溼度來做判定【1】【2】，在下雨天時比較有可能會發生，故在絕大部分時間偵測元件是屬於非常低的資料流量，只有在雨天時才必須要密集的去偵測並回傳資料。因此，本篇論文主要是針對架構於 IEEE 802.15.4 LR-WPAN【6】通訊協定 Beacon Order Adaptation Algorithm (BOAA)【3】、【9】進行更深入的探討，以達到更好的延長偵測元件電池使用壽命的效果，並節省偵測元件重新部署之人力與經費。最後並新舊 BOAA 的功耗比較與分析，以求得最佳的省電方式。

關鍵字：IEEE 802.15.4、低速率無線個人區域網路、Beacon Order Adaptation Algorithm (BOAA)。

1、前言

在目前大部份對於 IEEE 802.15.4 通訊協定的應用上，主要是在無線感測網路。無線感測網路是分佈著大量感測器節點設備於一個待感測區域內，依照要求對特定項目來進行偵測，如溫度、溼度、壓力、流速、雨量等等，然後經由特定資料集中管理設備 (Sink Node) 將收集的資料提供給管理者來對這些資料進行分析及判讀，進而做出相對應之行動。無線感測網路環境如圖一所示。



圖一 無線感測網路的環境

而根據無線感測網路之特性，我們可以發現有許多特點值得考量及研究：

1、電力及生存時間

在無線感測網路中，電力是非常重要的一個關

鍵，由於感測器回收不易以及其使用方式特殊，使得感測器設備通常都使用電池做為電力提供，但如何使得感測器能降低耗電，甚至能自我充電，來延長其生存時間將是相當的重要。

2、位置的辨識

感測器網路中的感測設備，在進行感測任務時，若能夠對於網路中鄰近之感測設備之位置能正確掌握時，網路中各個節點將能夠更有效率的傳遞資訊。

3、資料處理與溝通的效能

對於每個感測器設備，常有可能重覆傳送所收集到的感測資料，或是無用的資料，因此若各個感測設備能夠減少資料傳輸，將資料整合後再傳輸真正有用資料流量小、體積小...等特性，其應用層面相當廣泛的資訊給資料集中管理設備，將能有效減少電力之損耗。

4、通訊協定

由於無線感測網路有許多特點，包括低功率、低傳輸速率、低傳輸距離、體積小等等特性，因此當感測設備將資料從某節點傳送到資料集中管理設備時，必須是可靠的傳輸，而當網路環境中有節點損壞時，必須動態調整網路使得資料仍可以正確而可靠的傳輸到資料集中管理設備。

而目前在無線感測網路的通訊協定，是採用 IEEE 802.15.4 低速率無線個人區域網路 (Low-Rate Wireless Personal Area Network, LR-WPAN)。本篇論文由下列部份組成。第二章將會對 IEEE 802.15.4 通訊協定做一個介紹。第三章則為本篇論文所提出的演算法。第四章將對本篇論文所提出的演算法做分析模擬與比較結果。第五章為結論與未來工作。

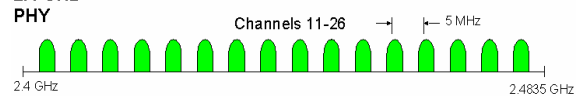
2、IEEE 802.15.4 通訊協定介紹

IEEE 802.15.4【6】是一個新的通訊協定規格，在 2003 年 2 月的 Draft 18(D18 版本)初步定義了短距低傳輸速率之實體層 (PHY) 與媒體存取層 (MAC)。而 802.15.4 LR-WPAN 有以下幾點特性：

1、操作頻帶有三種選擇：

●在 2.4 GHz ISM 頻帶中有 16 頻道，傳輸速率為 250 Kbps。

2.4 GHz



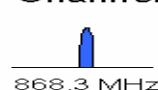
●在 915 MHz ISM 頻帶中有 10 頻道，傳輸速率為 40 Kbps。

Channels 1-10



●在歐洲 868 MHz 頻帶中有 1 頻道，傳輸速率為 20 Kbps。

Channel 0



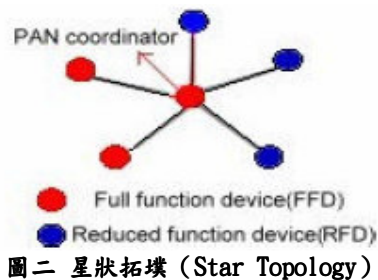
●操作頻率範圍：

BAND	COVERAGE	DATA RATE	CHANNEL(S)
2.4 GHz	ISM Worldwide	250 kbps	16
868 MHz	Europe	20 kbps	1
915 MHz	ISM Americas	40 kbps	10

- $F_c = 863.3$ in MHz, for $k = 0$
 $F_c = 906 + 2(k-1)$ in MHz, for $k = 1, 2, 3, \dots, 10$
 $F_c = 2405 + 5(k-11)$ in MHz, for $k = 11, 12, \dots, 26$
- 2、可選擇使用星狀 (Star) 或點對點 (Peer to Peer) 的拓樸方式。未來會支援叢集樹狀 (Cluster Tree) 架構。
 - 3、支援低延遲 (Low Latency) 設備。
 - 4、使用類似於 IEEE 802.11 CSMA/CA 方式競爭通。
 - 5、支援 64 位元延伸位址 (Extended Address) 與 16 位元短位址 (Short Address)。
 - 6、資料連結層中的回應封包 (Acknowledgement Frame, ACK Frame) 是可以選擇性使用。
 - 7、使用低能量消耗並有極低的執行週期。

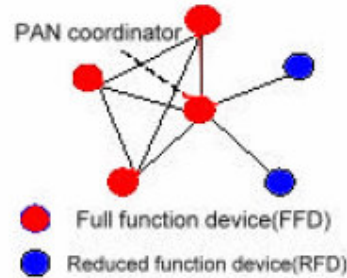
在 IEEE 802.15.4 中，要構成一個網路需要許多元件組成，其中規格中依設備對資料處理的能力，將其定義為兩類裝置：精簡功能裝置 (Reduced Function Device, RFD) 和完整功能裝置 (Full Function Device, FFD)。而數個裝置可構成一個網路架構，稱做個人操作範圍 (Personal Operation Space, POS)，而網路中的每個設備裝置都依照 IEEE 802.15.4 的通訊協定來進行運作及資料交換處理。而在一個 POS 中至少要有一個 FFD 來作 PAN 協調者 (PAN coordinator)。每個 PAN 都會有一個唯一的代表數值。此類低傳輸速率無線個人區域網路有兩種網路拓樸方式，一個是星狀拓樸 (Star Topology)，如圖二，一個是點對點拓樸 (Peer to Peer Topology) 如圖三。

●星狀拓樸有一個協調者和其他裝置所組成，其他裝置都只與協調者連線，由協調者處理所要做的事情；此網路拓樸方式基本上使用 64 位元延伸位址，此外，協調者可配置 16 位元短位址給裝置以節省頻寬的浪費。短位址的分配是當裝置與協調者進行連結 (Association) 時取得；其能源管理方式除了協調者因為要處理控制問題，例如：它要控制系統的同步問題 (如發 Beacon)、要轉換長短位址的對應或處理裝置彼此間的溝通等，所以需要較大能量，因此可以使用連接式電源，而其餘裝置基本上都是使用電池。

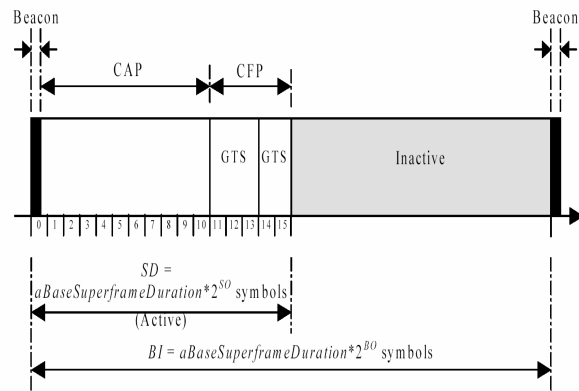


●點對點拓樸方式有一個協調者，但是其他非協調者的 FFD 裝置除了可以對協調者連結外，也可以對在其範圍內的其他裝置溝通，不過 RFD 裝置則只能

和 FFD 連結，而點對點的拓樸也可以升級成更複雜的拓樸方式運作，例如雜湊網路拓樸 (Mesh Networking Topology)、叢集網路拓樸 (Cluster Networking Topology) 等；此類網路可以是無基礎架構之無線區域網路，並可以自我建立路徑和自我更新，它也允許使用多重跳躍的方式使任一裝置可與其他裝置做溝通。



IEEE 802.15.4 的通訊協定架構是依照 OSI 模型來建立，其 PHY 包含 RF 傳送接收部分，用來控制低階傳輸，而 MAC (Medium Access Control) 控制所有類別之傳送，網路層 (Network Layer) 提供網路結構、操控及封包路徑等功能，而應用層 (Application Layer) 是依照裝置所需功能而制定，對於網路層和應用層這些屬於上層 (Upper Layers) 通訊協定的部分有另一組織 ZigBee 正在制定中，在 802.2 Type 1 LLC Sublayer，可透過服務標示收斂子層 (Service Specific Convergence Sublayer, SSCS) 存取 MAC 的資訊。LR-WPAN 除了可在內嵌式系統實行外，也可在外接式裝置 (如 PC, Notebook, PDA) 上實行。



IEEE 802.15.4 是應用類似 CSMA/CA 的方式競爭溝通，其中可以分為有信標網路 (Beacon-enabled Network) 與無信標網路 (Non Beacon-enabled Network)；無信標網路的協調者恆處在聽的狀態，在裝置要回傳資訊時會先彼此競爭，等通知協調者後，再傳送資料給協調者。而有信標網路中，含有超級訊框 (Superframe) 的架構 (圖四)，其固定將包含信標及超級訊框分為 16 個 slots，超級訊框持續時間 (Superframe Duration, SD) 與信標間距 (Beacon Interval, BI) 依照協調者使用信標級數 (Beacon Order, BO) 及超級訊框級數 (Superframe Order, SO) 來控制，其中 $BI = aBaseSuperframeDuration \times 2^{BO}$ ， $SD = aBaseSuperframeDuration \times 2^{SO}$ ，而 $aBaseSuperframeDuration$ 為一個時間常數，彼此關係是 $0 \leq SO \leq BO \leq 14$ ，如此可限制超級訊框持續時間會小於等於信標間距；協調者發送信標，除了用作同步化外，也包含網路相關資訊等；超級訊框

以有無使用保證時槽(Guaranteed Time Slots)來區別,有保證時槽的超級訊框可分成兩部分,一是競爭存取週期(Contention Access Period, CAP),另一是無競爭週期(Contention Free Period, CFP),而無保證時槽的超級訊框則全都是CAP。

一個superframe有16個slot,對於

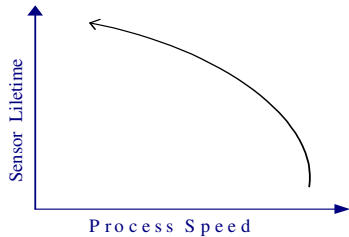
superframe中所定義的參數如下:

- BI(Beacon Interval) = aBaseSuperframeDuration x 2^{B0} symbols, ($0 \leq B0 \leq 14$).
- SD(Superframe Duration) = aBaseSuperframeDuration x 2^{S0} symbols, ($0 \leq S0 \leq B0 \leq 14$).
- aNumSuperframeSlots = 16 (fixed)
- aBaseSlotDuration = $2^{S0} \times 60$ symbols (variable)
- aBaseSuperframeDuration = aNumSuperframeSlots x aBaseSlotDuration = $16 \times 2^{S0} \times 60$ symbols(variable)

3、Beacon Order Adaptation Algorithm

Beacon Order Adaptation Algorithm (BOAA)

【5】-【9】是採取控制BO的方式來適當調整 Beacon 訊號發送頻率。BO 值大小直接影響到 BI 值。BO 減小 BI 時間變短,即表示 Beacon 訊號頻繁,可以傳送較多的資料,精準度高延遲時間短,其缺點是較耗電。反之,BO 加大 BI 時間變長,可以有較低耗電量,但是卻犧牲了效能。如圖五所示。



圖五 電池使用壽命與傳輸數率關係圖

在這方法中,協調者(coordinator)必須有一個 Buffer Matrix B,如圖六所示。此 Buffer Matrix B 為一個 $l \times n_{RFD}$ 矩陣,其中 n_{RFD} 為感測元件個數。每一列代表每一次 Beacon 訊號,每一行則分別對應到每一個 devices,用來記錄每一個 devices 在該筆 Beacon 訊號中是否有資料作傳送,有則設為 1,反之為 0。而為求得精確 Buffer Matrix B,協調者在發出 Beacon 訊號後便採用免競爭模式輪詢(polling)每個感測元件,如此可完全避免建立 Buffer Matrix B 時所產生碰撞問題。之後我們可以統計 Buffer Matrix B 所包含的訊息,並利用此數值來控制所需的傳輸速率。

$$B = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,j} & \cdots & b_{1,n_{RFD}} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,j} & \cdots & b_{2,n_{RFD}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{i,1} & b_{i,2} & \cdots & b_{i,j} & \cdots & b_{i,n_{RFD}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{l,1} & b_{l,2} & \cdots & b_{l,j} & \cdots & b_{l,n_{RFD}} \end{pmatrix}$$

圖六 Buffer Matrix B

其演算法如下:

```

Algorithm 1 recordMessageOccurence ( )
1:  i<=1
2:  while algorithm not abort do
3:    polling all devices in CFP
4:    for j=1 to nRFD do
5:      if RFD j send ack then
6:        mark message occurrence in bi,j
7:      else
8:        mark message absence in bi,j
9:        sleep until end of beacon interval
10:     end if
11:   end for
12:   evaluate N_MAX ( )
13:   evaluateBO ( )
14:   if i=lb then
15:     i<=1
16:   else
17:     i<=i+1
18:   end if
19:   for k=1 to nRFD do
20:     if N_MAXk !=0
21:       assign the sequence number from 1 to
nRFD according to
22:         value N_MAXk
23:     end if
24:   end for
25:   forward message in sequence
26:   sleep until end of beacon interval
27: end while

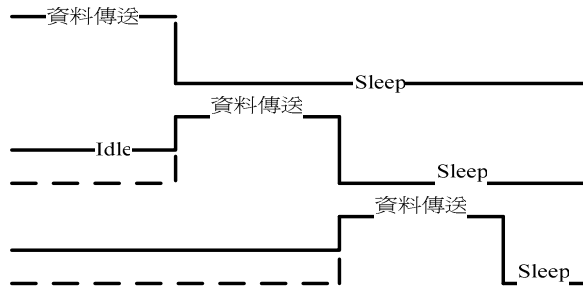
```

在免競爭模式(contention Free Mode)採用輪詢(polling)方法詢問每個 device 是否有資料傳送(演算法第 3 行),有則回應一個 ACK 訊號(演算法第 5 行),若無資料傳送則馬上進入 sleep mode(演算法第 9 行)。Coordinator 收到 ACK 訊號後,便將數值 1 填入所對應到的矩陣欄位(演算法第 6 行)。

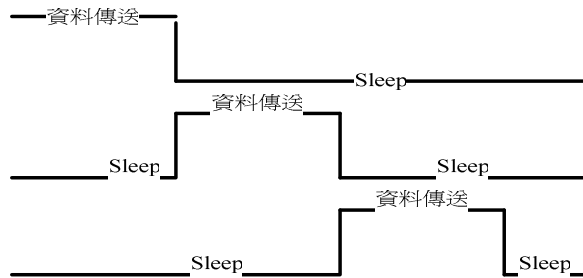
而由於每個 devices 所要傳送的資料格式皆為一樣,因此所需之資料傳送時間皆為固定,如圖 4.5, 4.6 所示。由圖七中我們可以發現,有資料待傳輸之 devices 須等待 medium 空閒時才有機會傳輸資料,在 idle 等待過程浪費不少電量,因此在演算法第 19-24 行中我們根據 N_MAX_Matrix 裡將有資料待傳送之 devices 元素由大至小排列,並給於一 sequence number, devices 則根據 sequence number 順序傳送資料,其於時刻皆處於 sleep mode 以節省 power consumption,如圖八所示。

有資料傳送之 devices 在接收到自身 sequence number 後,便依循 IEEE 802.15.4 CSMA/CA 免競爭方式爭取 medium 使用權,資料傳送完畢立即進入 sleep mode(演算法第 25、26 行)。而 evaluateN_MAX () 與 evaluateBO () 兩個函數(演算法第 12、13 行)分別是用來計算 N_MAX 與下個 BO 值。

在下一個段落中我們將引述介紹 evaluateN_MAX () 與 evaluateBO () 兩個函數,並將針對本論文所提出的方法做詳細模擬與分析。



圖七 原 BOAA 演算法 device 資料傳輸情形



圖八 改良後 BOAA 演算法 device 資料傳輸情形

●evaluateN_MAX () 函數：

evaluateN_MAX () 主要是根據上個段落中提到之 Buffer Matrix B，計算 Buffer Matrix B 中每一行元素和，並且定義一個參數 N_MAX，其所代表的意義為 n_{RFD} 行元素和最大者，數學表示式【4】如下：

$$N_MAX = \max \left(\dots, \sum_{k=1}^{l_b} b_{k,m-1}, \sum_{k=1}^{l_b} b_{k,m}, \dots \right)$$

而在 IEEE 802.15.4 無線個人區域網路應用於災難偵測與評估【3】中，BOAA 演算法對於 evaluateN_MAX () 函數共有四個演算法來計算 Buffer Matrix B 每一行元素和，在此我們選擇其中較好一組應用在 Algorithm 1 中，其演算法我們轉錄如下，共選擇一種計算 N_MAX 的方法：

```

Algorithm 2C : (Weight algorithm)
1 : for i=0 to ∞ (send beacon)
2 :   for j=0 to nRFD do
3 :     for k=0 to lb do
4 :       if k=0 then
5 :         n[j]=n[j]+Buffer_Matrix_B[(i%lb+1)%lb][j] ×WEIGHT
6 :       else
7 :         n[j]=n[j]+Buffer_Matrix_B[(i%lb+1)%lb][j]
8 :       end if
9 :     end for
10 :    if n (j) >N_MAX then
11 :      N_MAX<=n (j)
12 :    end if
13 :  end for
14 : end for

```

在上述 Algorithm 2C 演算法中，Buffer Matrix B 為一個 l_b×n_{RFD} 矩陣。我們給定一個 WEIGHT 值 (演算法第 5 行)，在每一次 Beacon 訊號，最新一列 Buffer Matrix B 元素乘上 WEIGHT 值後計算每一行元素和 (演算法第 4、5、6、7 行)，並在 n_{RFD} 筆元素和中取出最大值等於 N_MAX。

●evaluateBO () 函數：

N_MAX 是一個非常重要的參數，其意味著目前網路資料量大小，故我們可以根據 evaluateN_MAX () 函數所求得 N_MAX 值，配合以下方法來計算 BO 值：

1、Algorithm 2.D (Table Algorithm)：

在本方法中，我們建立一個表格，每個範圍 N_MAX 值分別對應到不同 BO 值，根據這個各表格來控制 BO 大小。如表一所示。

表一 N_MAX 對照表

N_MAX 值	動作
N_MAX ≤ 0	BO = 14, 251s
0 < N_MAX ≤ 1	BO = 13, 126s
1 < N_MAX ≤ 2	BO = 12, 63s
2 < N_MAX ≤ 3	BO = 11, 31s
3 < N_MAX ≤ 4	BO = 10, 16s
4 < N_MAX ≤ 5	BO = 9, 7.86s
5 < N_MAX ≤ 6	BO = 8, 3.93s
6 < N_MAX ≤ 7	BO = 7, 1.96s
7 < N_MAX ≤ 8	BO = 6, 0.98s
8 < N_MAX ≤ 9	BO = 5, 0.49s
9 < N_MAX ≤ 10	BO = 4, 0.24s
10 < N_MAX ≤ 11	BO = 3, 0.12s
11 < N_MAX ≤ 12	BO = 2, 61ms
12 < N_MAX ≤ 13	BO = 1, 30ms
N_MAX ≥ 14	BO = 0, 15ms

2、Algorithm 2.E：

在本方法中，我們建立一個表格，每個範圍 N_MAX 值分別對應到不同 BO 值，根據這個各表格來控制 BO 大小。其中 C_MAX 為矩陣 buffer Matrix B 行的最大值，我們將其分割為 14 等分。如表二所示。

表二 N_MAX 對照表

N_MAX 值	動作
N_MAX ≤ 0(C_MAX)/14	BO = 14, BI=251s
0(C_MAX)/14 < N_MAX ≤ (C_MAX)/14	BO = 13, BI=126s
(C_MAX)/14 < N_MAX ≤ 2(C_MAX)/14	BO = 12, BI=63s
2(C_MAX)/14 < N_MAX ≤ 3(C_MAX)/14	BO = 11, BI=31s
3(C_MAX)/14 < N_MAX ≤ 4(C_MAX)/14	BO = 10, BI=16s

$4(C_MAX)/14 < N_MAX \leq 5(C_MAX)/14$	$B0=9, BI=7.86s$
$5(C_MAX)/14 < N_MAX \leq 6(C_MAX)/14$	$B0=8, BI=3.93s$
$6(C_MAX)/14 < N_MAX \leq 7(C_MAX)/14$	$B0=7, BI=1.96s$
$7(C_MAX)/14 < N_MAX \leq 8(C_MAX)/14$	$B0=6, BI=0.98s$
$8(C_MAX)/14 < N_MAX \leq 9(C_MAX)/14$	$B0=5, BI=0.49s$
$9(C_MAX)/14 < N_MAX \leq 10(C_MAX)/14$	$B0=4, BI=0.24s$
$10(C_MAX)/14 < N_MAX \leq 11(C_MAX)/14$	$B0=3, BI=0.12s$
$11(C_MAX)/14 < N_MAX \leq 12(C_MAX)/14$	$B0=2, BI=61ms$
$12(C_MAX)/14 < N_MAX \leq 13(C_MAX)/14$	$B0=1, BI=30ms$
$N_MAX \geq 14(C_MAX)/14$	$B0=0, BI=15ms$

4、模擬與分析結果

在本段落中，我們針對於之前所討論之BOAA演算法做一些模擬與效能上的評估。

● 模擬環境：

在本篇論文中，我們所採用的是星狀拓撲

【3】，如圖二所示。由於所要感測的環境所需傳送之資料量很小，如溫度、溼度、地聲、震動...等，因此我們假設頻寬滿足所有感測元件在同一個Beacon訊號內發送完個別所要傳送的資料，即不會因為競爭不到頻寬而須延遲到下一個Beacon訊號。

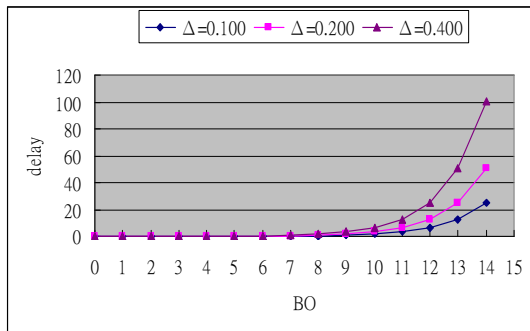
● 變數設定

我們定義以下幾個參數作為效能評估的依據：

Δ ：單位Beacon內有資料傳送次數，由系統亂數決定。在本篇論文中我們取 $\Delta=\{0.100, 0.200, \dots, 1.000\}$ 。

δ ：傳輸延遲。

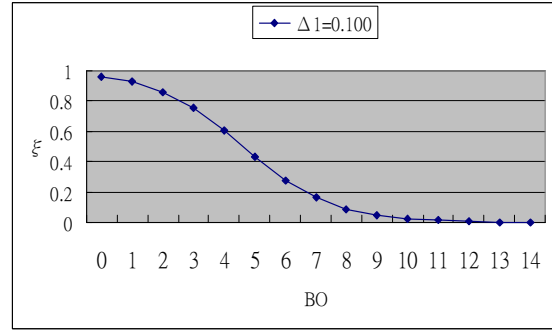
$\delta = BI \times \Delta$ ，越大則代表延遲時間越長。圖九為在不同 Δ 下所產生不同延遲分佈圖。



圖九 BO-Delay關係圖

ξ ：功率消耗。

$\xi_{avg} = P_{rx} / P_{total}$ ，越大則代表在接收模式下所佔的百分比越高。其中 P_{rx} 為接收模式下所消耗功率， P_{total} 為全部的消耗功率。圖十為在 $\Delta=0.100$ 下所產生功率消耗百分比分佈圖。



圖十 BO-Power Consumption關係圖

● 參數設定：

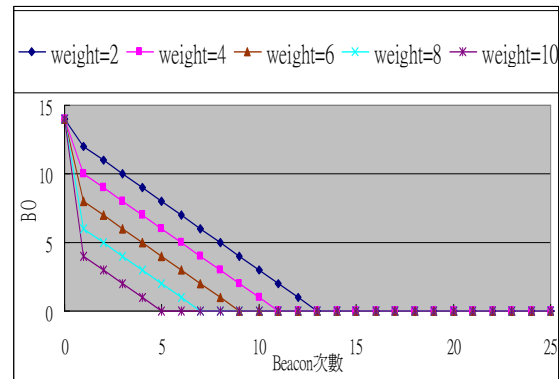
在本章中的模擬實驗，所採用的模式是遵循著IEEE 802.15.4 MAC layer所定義之slotted CSMA/CA演算法為基礎所設計的。並參考【3】【4】【6】【7】【8】【9】設定下列數值，如表三所示：

表三 模擬環境參數

參數	數值	參數	數值
P_{idle}	30mw	T_{Beacon}	0.1ms
P_{tx}	31mw	T_{ack}	0.1ms
P_{rx}	35mw	T_{data}	0.2ms
P_{sleep}	$3 \mu w$		

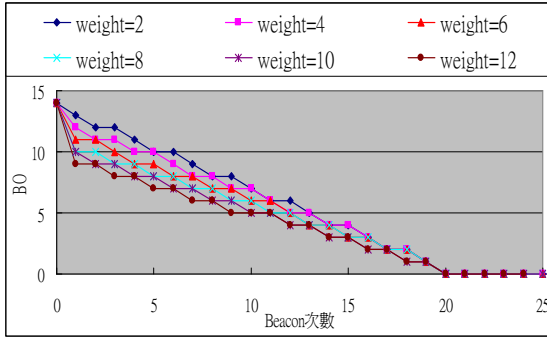
1、感測元件瞬間反應速度

在本小節中我們主要是要探討weight值大小對於瞬間反應時間的影響。圖十一的模擬實驗中我們搭配Algorithm 2.C (Weight Algorithm)及Algorithm 3.D (Table Algorithm)演算法，上述的BO初始值定為14，buffer Matrix $B=0$ ， $\Delta=1.000$ ， $l_b=20$ ， $weight=\{2, 4, 6, 8, 10\}$ ，即表示從原本傳輸速率從 $\Delta=0.000$ 瞬間變為 $\Delta=1.000$ ，模擬結果如圖十一所示。從圖十一我們可以發現weight值愈大則瞬間反應速度愈快。以 $weight=2, 10$ 為例，當 $weight=10$ ，BO從14變為4，Beacon Interval (BI)由251秒減少為0.24秒，比 $weight=2$ ，BO從14變為12，Beacon Interval (BI)由251秒減少為62.91秒瞬間反應數度高出很多。



圖十一 weight值對瞬間反應影響

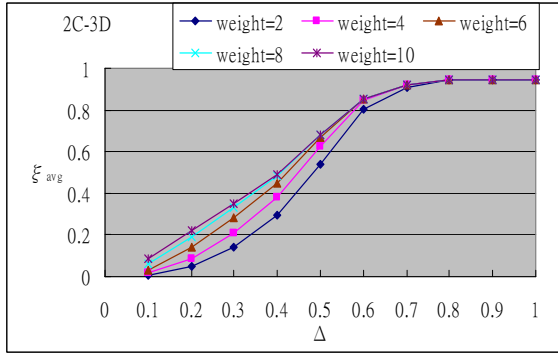
圖十二的模擬實驗中我們搭配Algorithm 2.C (Weight Algorithm)及Algorithm 3.E演算法，模擬條件設定與圖十一模擬實驗相同。由圖十二我們可以發現，除瞬間反應能力較差外，還需要更多的Beacon訊號才能使BO由14降為0。



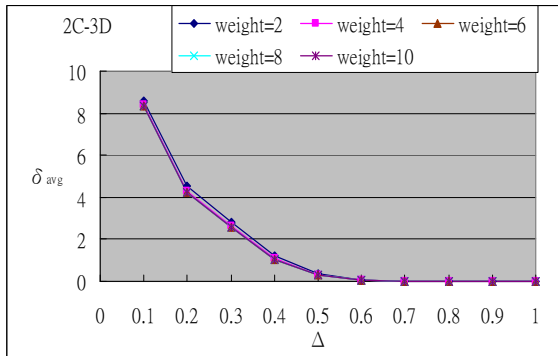
圖十二 weight值對瞬間反應影響

2、平均功率消耗及平均延遲對 Δ 分布圖

圖十三、圖十四為改良後BOAA演算法algorithm 2C搭配algorithm 3D平均功率消耗及平均延遲對傳輸速率 Δ 分布圖。在本次實驗中我們設定weight={2, 4, 6, 8, 10}, $l_b=20$ 。圖十三、圖十四顯示在相同傳輸速率 Δ 下, weight值愈高則平均消耗功率越大, 但有較低平均時間延遲。在相同WEIGHT值下, 傳輸速率 Δ 愈高, BO值愈小, 平均消耗功率越大, 平均時間延遲越短。

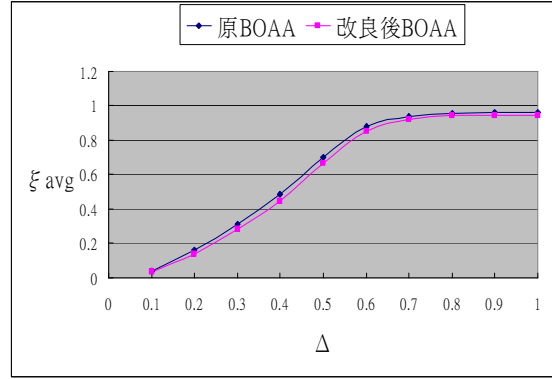


圖十三 algorithm 2C搭配 algorithm 3D演算法不同WEIGHT值平均功率消耗對 Δ 分布圖

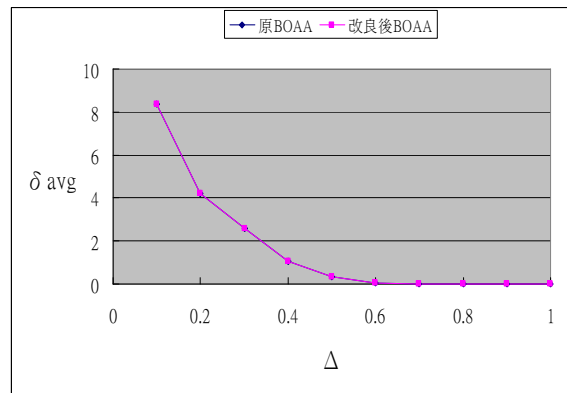


圖十四 algorithm 2C搭配algorithm 3D演算法不同WEIGHT值平均延遲對 Δ 分布圖

圖十五、圖十六為改良後BOAA演算法與原BOAA演算法algorithm 2C搭配algorithm 3D平均功率消耗及平均延遲對傳輸速率 Δ 分布圖。在本次實驗中我們設定weight=6, $l_b=20$ 。由圖十五中我們可以發現改良過後的BOAA演算法在平均消耗功率上有較好表現, 根據不同的網路流量會有約0.78%~1.36%改善。而由圖十六中我們可以發現改良過後的BOAA演算法其平均延遲時間不會有太大的影響。

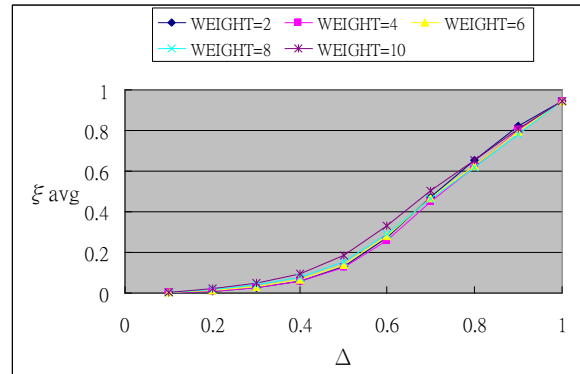


圖十五 改良後與未改良BOAA演算法algorithm 2C搭配 algorithm 3D演算法平均功率消耗對 Δ 比較圖

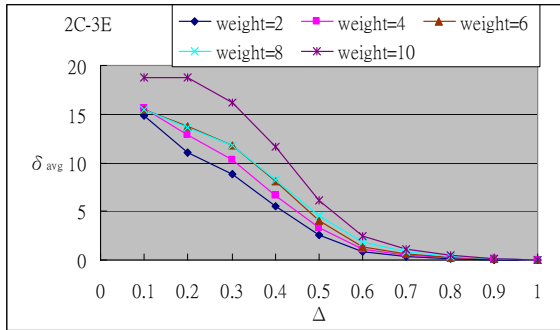


圖十六 改良後與未改良BOAA演算法algorithm 2C搭配 algorithm 3D演算法平均傳輸延遲對 Δ 比較圖

圖十七、圖十八為改良後BOAA演算法algorithm 2C搭配algorithm 3E平均功率消耗及平均延遲對傳輸速率 Δ 分布圖。在本次實驗中我們設定weight={2, 4, 6, 8, 10}, $l_b=20$ 。比較圖十三與圖十七我們可以發現本次模擬實驗擁有更低的平均消耗功率, 然而缺點從圖十四、圖十八可得知是其平均延遲時間較大, 以 $\Delta=0.100$ 為例, 平均延遲時間差距約為7秒, 且在WEIGHT=10時平均延遲時間明顯增加, 約為19秒。

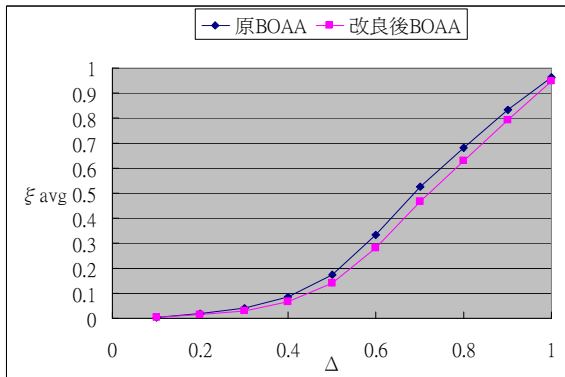


圖十七 algorithm 2C搭配algorithm 3E演算法不同WEIGHT值平均功率消耗對 Δ 分布圖

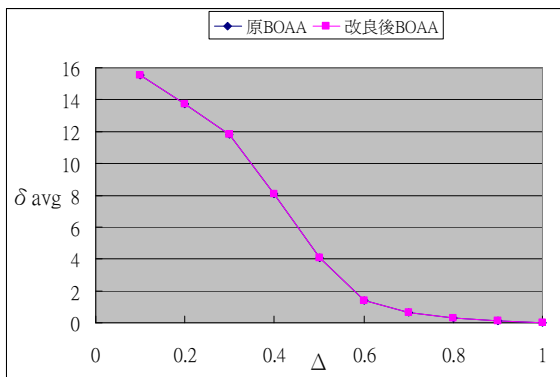


圖十八 algorithm 2C搭配algorithm 3E演算法不同WEIGHT值平均延遲對 Δ 分布圖

圖十九、圖二十為改良後BOAA演算法與原BOAA演算法algorithm 2C搭配algorithm 3E平均功率消耗及平均延遲對傳輸速率 Δ 分布圖。在本次實驗中我們設定weight=6, $l_b=20$ 。由圖十九中我們可以發現改良過後的BOAA演算法在平均消耗功率上有較好表現, 根據不同的網路流量會有約0.14%~5.69%改善。而由圖二十中我們可以發現改良過後的BOAA演算法其平均延遲時間不會有太大的影響。



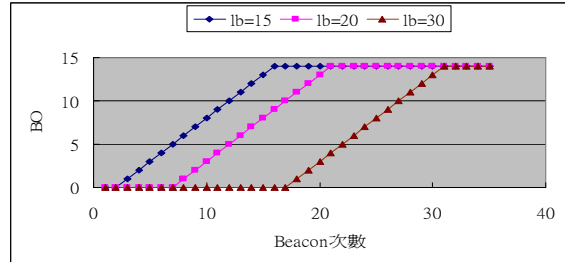
圖十九 改良後與未改良BOAA演算法algorithm 2C搭配algorithm 3E演算法平均功率消耗對 Δ 比較圖



圖二十 改良後與未改良BOAA演算法algorithm 2C搭配algorithm 3E演算法平均傳輸延遲對 Δ 比較圖

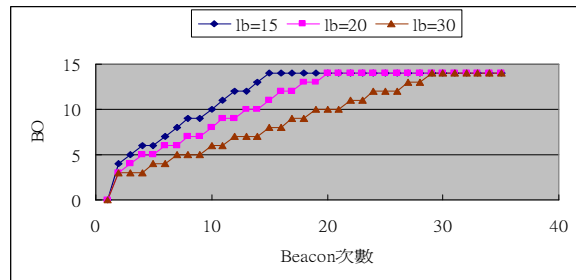
3、Buffer Matrix B大小對BO值及消耗功率影響
在本小節中我們主要是要探討Buffer Matrix

B大小對BO值及消耗功率影響。在本模擬實驗中我們搭配Algorithm 2.C (Weight Algorithm) 及Algorithm 3.D (Table Algorithm) 演算法, 並設定Buffer Matrix $B=1$, $\Delta=0.000$, weight=4, 即表示從原本傳輸速率從 $\Delta=1.000$ 瞬間變為 $\Delta=0.000$, 模擬結果如圖二十一所示。圖二十一顯示當傳輸速率從 $\Delta=1.000$ 瞬間變為 $\Delta=0.000$, l_b 值愈大需要更多次的Beacon訊號才能將BO值控制在符合目前的傳輸速率, 即較為耗電, 但相對而言能夠紀錄更多筆網路資料, 得到較精確的數值。



圖二十一 l_b 大小對BO值影響

圖二十二模擬實驗為搭配Algorithm 2.C (Weight Algorithm) 及Algorithm 3.E演算法, 其於條件均與圖二十一模擬實驗相同。由圖二十二可得知, BO由0增加為14所需的Beacon訊號次數相同, 斜率較小, 即較省電。故在相同條件下搭配Algorithm 3E演算法, l_b 值越大, 能夠紀錄更多筆網路資料, 得到較精確的數值, 而當BO從0增加為14的耗電量比搭配Algorithm 2.D演算法低。



圖二十二 l_b 大小對BO值影響

5、結論與未來工作

在BOAA演算法中, 主要針對災難偵測方面提出幾個演算法, 使偵測元件能夠自我控制流量, 在效能與能量消耗間做一個評量, 以達到延長偵測元件電池使用壽命的目的。而在本論文中, 我們主要是針對這些演算法加以改良, 並用程式做了許多的模擬來驗證。從我們的模擬結果中可以得知改良後的BOAA演算法搭配Algorithm 2C及Algorithm 3D與Algorithm 2C及Algorithm 3E演算法, 對於平均功率有較好數值呈現, 然而在瞬間反應順速度、平均時間延遲方面並無太大影響。

改良後BOAA演算法搭配Algorithm 2C及Algorithm 3D演算法, Weight值愈大瞬間反應數愈快, 平均消耗功率卻不會因而急速上升, 改良後BOAA演算法根據不同的網路流量會有約0.78%~1.36%改善, 而平均延遲時間卻無太大改變, 最大值約只有0~9秒, 對於災難偵測方面也有相當程度的反應時間。 l_b 值愈大能夠紀錄更多筆網路資料, 得到較精確的數值, 平均延遲時間也較短, 但相對而言更為耗電, 也更佔記憶體資源, 而根據我們實驗結果得知 l_b 值約在15~20間, 平均延遲時間與平均消耗功率能取得較好平衡點。

另一方面, 改良後BOAA演算法搭配Algorithm 2C及Algorithm 3E演算法, 擁有更低平均消耗功率, 改良後BOAA演算法根據不同的網路流量會有約0.14%~5.69%改善, 缺點是平均延遲時間較長, 當

weight值由8上升為10時平均延遲時間有較明顯增加，約由15秒增加為18秒，如圖十八所示，故此種演算法搭配weight值應小於等於8。此外，此種演算法搭配瞬間反應能力也較差。而lb值愈大平均消耗功率較低(差距很小)，卻有較低平均延遲時間，如圖十七所示，缺點則是浪費硬體資源。故此演算法搭配應視實際需求及硬體資源，以達到最大效益。至於在實際應用上，我們根據實際情形可以選擇適合的演算法配。

本篇論文對 Beacon Order Adaptation Algorithm (BOAA) 演算做了許多模擬分析及驗證，但是仍有許多方面值得我們再去深入探討的。

- 在不同的應用：

本篇論文除了在土石流災難偵測方面外，在其他的災難偵測方面是否適用。針對不同的應用我們未來將做更詳細探討與研究。

- 網路效能研究：

IEEE 802.15.4 通訊協定在網路流量壅塞的狀況下，其網路效能並不理想。如何針對這個通訊協定，讓它的碰撞解決演算法能有適應性去針對不同的網路狀況下去改變其參數設定，或是採用一個更有效率的碰撞解決演算法，我相信一定能大大的提升本篇論文的準確性。

6、參考文獻

【1】92年度交通部山區道路邊坡監測系統自動化及緊急臨時通報系統研發

【2】行政院農業委員會水土保持局土石流觀測示範站建立與系統維護工程

【3】IEEE 802.15.4 低速率無線個人區域網路應用於災難偵測之研究與評估

【4】A. El-Hoiydi and J.-D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks.

【5】Christian C. Enz Amre El-Hoiydi Jean-Dominique Decotignie Vincent Peiris. WiseNET: An Ultralow-Power Wireless Sensor Network Solution. 0018-9162/04/\$20.00 © 2004 IEEE.

【6】IEEE 802.15.4 Standard Information technology.

【7】Jian Ma, Min Gao, Qian Zhang, Lione M. Ni and Wenwu Zhu. Localized Low-Power Topology Control Algorithms in IEEE 802.15.4-based Sensor Networks. Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICSCS' 05) 1063-6927/05 \$20.00 c 2005 IEEE.

【8】Jianliang Zheng and Myung I. Lee, The City College of CUNY. Will IEEE 802.15.4 Make Ubiquitous Networking a Reality? A Discussion on a Potential Low Power, Low Bit Rate Standard.

【9】Mario Neugebauer, Jom Plonnigs, Klaus Kabitzsch. A New Beacon Order Adaptation Algorithm for IEEE 802.15.4 Networks. 0-7803-8801-1/05/\$20.00 (~)2005 IEEE..