

# IEEE 802.15.4 Beacon 模式之系統設計與實作

柯力群  
財團法人資訊工業策進會  
lcko@nmi.iii.org.tw

劉永之  
財團法人資訊工業策進會  
ulysses@nmi.iii.org.tw

## 摘要

IEEE 802.15.4 是一個針對低速率個人網路所設計的標準，在本篇論文的一開始，會先對這個標準做概要的介紹，接著會說明由我們自行設計的 802.15.4 Beacon-enabled Network 資料收發架構，然後會討論一些在實作這個標準上會遇到的取捨問題，最後對我們實作的系統進行效能分析的評估。透過我們的實作可以發現，在 802.15.4 中採取 Superframe 架構時，並非在任何情形下都能使用到所有的 GTS Slot，而另一方面，當其 SO 值過小時，即使採用 GTS 機制，其所保留的頻寬也可能發生不夠傳送任何標準封包的情形。最後，雖然在 Beacon-enabled 的架構下需要利用額外的 Preamble 機制來防止 Beacon Lost 的情形發生，但在我們的實作中，Preamble 對於系統工作週期幾乎不會造成任何影響。

**關鍵詞：** IEEE 802.15.4、Beacon-enabled Networks

**英文題目：** Design and Implementation of IEEE 802.15.4 Beacon-enabled Networks

## Abstract

In the beginning of this paper we introduce the basic concepts of IEEE 802.15.4 standard. In the second part, our implementation of beacon-enabled network architecture design is presented. Then we evaluate and analyze the performance of our Implementation. It can be shown that when we adopt superframe mechanism, only when the value of SO is larger than or equal to 3 can it obtain the complete GTS slots. On the other hand, for certain SO values, the system can hardly transmit any packets. Lastly, although we use preambles to avoid the happening of beacon lost, the using of preambles does not result in significant increase of its ordinary duty cycle.

**Keywords :** IEEE 802.15.4 、 Beacon-enabled Networks

## 一、IEEE 802.15.4 背景介紹

IEEE 802.15.4[2](以下簡稱為 802.15.4)為一針對無線個人區域網路所設計的介質存取控制(Media Access Control, MAC)層與實體層之低速率傳輸標準，目前的版本是於 2003 年所公佈。此標準具有之特性，包括分別在 2450、915、868 MHz 提供了 250kps、40kbps、20kbps 三種傳輸速率。同時它採用了 CSMA/CA 的通道存取方式，可形成點對點或是星狀拓樸的網路架構，並支援 Guaranteed Time Slots(GTS)的服務品質保證機制。目前對於 802.15.4 的論文探討多是在系統層的行為模擬方面[1][3][4]，在本篇論文中，我們將重點放在自行實作系統上。在論文的一開始，會先對 802.15.4 的標準做進一步的說明，接著再介紹由我們自行設計的 802.15.4 Beacon-enabled Network 架構，然後會對於一些在實作上遇到的取捨問題進行討論，最後對我們的系統進行整體性的效能分析評估。

## 二、IEEE 802.15.4 標準說明

在這個部份，我們會針對 802.15.4 的標準做概括性的介紹，同時對於本論文所用到的部份進行詳細說明。

(一) Beacon-enabled Network 與 Non-beacon Network

在 IEEE802.15.4 中的網路設備可分為 Full-function Device(FFD)與 Reduced-function Device(RFD)兩種。一個 FFD 具有 Personal Area Network (PAN) Coordinator、Coordinator 與 RFD 等功能，它可以跟其它的 RFD 或是 FFD 進行資料傳輸。RFD 是一個極其簡單的網路設備，只具備與 FFD 進行資料傳輸的功能。簡而言之，它一次只能跟單一 Coordinator 互相結合，同時也沒有大量的資料傳輸的需要，所以它的系統資源可以降到最低。

在 802.15.4 中允許採用一種稱為 Superframe 的架構，一個採用 Superframe 架構的 802.15.4 網路被稱為 Beacon-enabled Network，反之則被稱為 Non-beacon Network。如圖 1 所示，所謂的 Superframe 是一段由兩個 Coordinator 所發的

Beacon 所限定的時間區段，每個單位都被稱為一個 Slot。在 Beacon-enabled network 下發 Beacon 的目的是要作同步化，任何設備想要進行資料傳輸都必須在屬於 Superframe 內的 Contention Access Period(CAP)區間內以 slotted CSMA/CA 的方式對傳輸通道進行存取，同時 CAP 區段可被平均的切割為 16 個 Slot 單位。

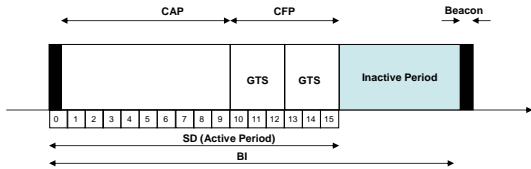


圖 1、採用 Superframe 架構時，CAP、CFP、SD、BI、與 Beacon 之關係圖

在 Beacon-enabled 架構下，Coordinator 亦可針對保留頻寬的需求來使用 Guaranteed Time Slot(GTS)機制形成 Contention-free Period(CFP)區間，在 CFP 區間內只有已於 Coordinator 註冊過要做頻寬保留的設備才可以進行資料傳輸，在一個 Superframe 中，CFP 會永遠被置於 CAP 的後面。

(二) MAC 層封包格式介紹

在 IEEE802.15.4 的標準中共有四種封包格式，分別是 MAC Command Frame、Data Frame、Beacon Frame 以及 Acknowledge(ACK) Frame，在此會針對本論文中所需的部份做簡單的介紹，細節部份請見[2]。

MAC Command Frame 的格式如圖 2，它的開頭是一個 MAC Header(MHR)，其內有一個 Frame Control 來分辨該封包之型別、所使用的定址模式等，一個 Sequence Number 以及一些 Addressing Field 的定址資訊。接在 MHR 後面的是 MAC Service Data Unit(MSDU)，MAC Command Frame 的 MSDU 包括有 Command Type 與 Command Payload，最後面接的則是 MAC Footer(MFR)，它包括了一個 16 位元的 Frame Check Sequence(FCS)。

|               |                 |                   |              |                 |     |
|---------------|-----------------|-------------------|--------------|-----------------|-----|
| Octets:2      | 1               | 4 to 20           | 1            | N               | 2   |
| Frame Control | Sequence Number | Addressing Fields | Command Type | Command Payload | FCS |
| MHR           |                 |                   | MSDU         |                 | MFR |

圖 2、MAC Command Frame 的封包格式

Data Frame 的格式如圖 3，它與 MAC Command Frame 的差別只有在 MSDU 的部份為單純的 Data Payload。

|               |                 |                   |              |     |
|---------------|-----------------|-------------------|--------------|-----|
| Octets:2      | 1               | 4 to 20           | Variable     | 2   |
| Frame Control | Sequence Number | Addressing Fields | Data payload | FCS |
| MHR           |                 |                   | MAC payload  | MFR |

圖 3、Data Frame 的封包格式

Beacon frame 的格式如圖 4，與一般 MHR 不同的地方在於其內是一個 Beacon Sequence Number 而非 Sequence Number。其 MSDU 則包括了跟 Superframe 設定相關的 Superframe Specification，含有 GTS 配置資訊的 GTS Field，有資料待收的 Pending Address Specification，以及一個 Beacon Payload。Superframe Specification(見圖 5)內包括有 Beacon order、Superframe Order 兩個用來決定 Superframe 長度的變數，跟 GTS 相關的最後一個 CAP Slot 位置，省電功能設定，以及跟 Coordinator 有關的 PAN Coordinator 位元與 Association Permit 位元。在 Beacon Frame 中兩個比較重要的不定長度的欄位是 GTS Field 與 Pending Address Field。GTS Field 的長度會與其 GTS List 有關，其總長度最多可達  $1+1+3*7=23$ bytes。Pending Address Field 的長度會與等待接收資料的 Device 個數有關，其長度最多可達  $1+8*7=57$ bytes。

|               |                 |                   |                          |            |                 |                |     |
|---------------|-----------------|-------------------|--------------------------|------------|-----------------|----------------|-----|
| Octets:2      | 1               | 4/10              | 2                        | Variable   | Variable        | Variable       | 2   |
| Frame Control | Sequence Number | Addressing Fields | Superframe Specification | GTS Fields | Pending Address | Beacon Payload | FCS |
| MHR           |                 |                   | MAC Payload              |            |                 |                | MFR |

圖 4、Beacon Frame 的封包格式

|              |                  |                |                        |          |                 |                    |
|--------------|------------------|----------------|------------------------|----------|-----------------|--------------------|
| Bits:0-3     | 4-7              | 8-11           | 12                     | 13       | 14              | 15                 |
| Beacon Order | Superframe Order | Final CAP Slot | Battery life extension | Reserved | PAN coordinator | Association permit |

圖 5、Superframe Specification 的封包格式

(三) Superframe 架構

在 IEEE 802.15.4 中的 Beacon-enabled Network 主要是由 SO 與 BO 兩個參數來控制，當中的 BO 指的是 macBeaconOrder，SO 則是代表 macSuperframeOrder，這兩個參數會決定一個網路的 Superframe Interval 以及 Beacon Interval(請見圖 1)。其推斷公式為：

$$SD = aBaseSuperframeDuration * 2^{SO} \quad (1)$$

$$BI = aBaseSuperframeDuration * 2^{BO} \quad (2)$$

當中的 SO 與 BO 具有  $0 \leq SO \leq BO \leq 14$  的關係，aBaseSuperframeDuration 為一常數，其值為 160，SD 與 BI 的單位皆為 Symbol。IEEE 802.15.4 於 2450、915、868 MHz 之 Symbol Rate 分別是 62.5 (ksymbol/s)、40(ksymbol/s)、與 20 (ksymbol/s)。

(四) 資料傳輸模型

在 IEEE 802.15.4 中的 Beacon-enabled Network 有兩種資料傳輸模型，一種是由網路設備(Network Device)傳資料給 Coordinator，另一種是 Coordinator 傳資料給網路設備。

a、網路設備傳送給 Coordinator

在 Beacon-enable 的架構下，網路設備要傳輸資料給 Coordinator 前要先跟 Superframe 的 Beacon 同步，然後以 Slotted CSMA/CA 的方式將資料傳給 Coordinator，此時 Coordinator 可以回傳一個 (Optional) Acknowledgment。整個過程如圖 6 所示。

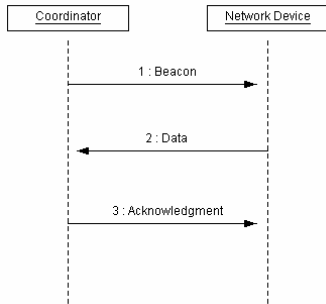


圖 6 網路設備傳送資料給 Coordinator 的傳輸模型

#### b、Coordinator 傳送給網路設備

在 Beacon-enable 的架構下，當 Coordinator 有資料要傳給網路設備時，它會在 Beacon 中附加 Data Pending 的資訊。此時網路設備要以 Unslotted CSMA/CA 的方式對 Coordinator 送 Data Request 的 MAC Command。Coordinator 收到後會回傳一 Acknowledgment，接著 Coordinator 同樣以 Unslotted CSMA/CA 回傳其所要求的 Data Frame，最後網路設備發出結束整個交易的 Acknowledgment。完整的過程如圖 7 所示。

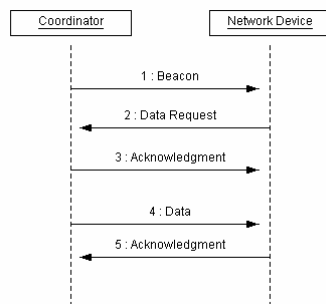


圖 7、Coordinator 傳送資料給網路設備的資料傳輸模型

### 三、系統設計

由於在使用 Beacon-enabled Network 時，系統對於時間的準確性具有高度的要求，為此，我們設計了一個適合此一特性的核心排程器—MOS (Mini Operating System)。在這段說明中，將會對 MOS 所使用的排程方法進行介紹，同時也會對於本系統採用的 Beacon-enabled Network 資料收發機制設計進行說明。

#### (一) 核心排程器設計

MOS 的工作排程是以『執行緒』的概念為基

礎。每個 Task 都是透過 API 的登記後排入排程內。每個 Task 都使用自己的資料結構來登記排程相關的資訊，例如優先順位、要執行的 Callback Function 等。登記完 Task 之後，MOS 將依照先登記先執行的『Queue』方式依序執行。在 MOS 裡沒有實作 Round-Robbing 式的排程處理方式，每個 Task 一定要執行完畢後才換到下一個 Task 執行。而每個 Task 都有相對應的優先權 (Priority)，在 MOS 裡優先權共分五級，分別是 Emergency、Top、High、Medium、與 Low。在實作上則是採用一個優先順位等級對應一個 Queue 的方式處理，如圖 8 所示：

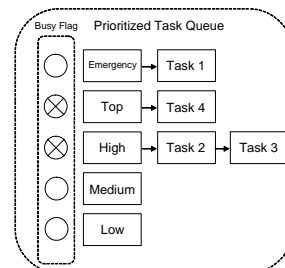


圖 8、MOS 排程器之示意圖

當 MOS 在實際執行時，會以 Timer Interrupt Service Routine (TISR) 來觸發，期週期為 320u 秒，TISR 首先會檢查跟 Timer 相關的 Timer Event，所謂的 Timer Event 是一種由使用者註冊的提醒通知，告知使用者之前設定的等待時間已經到期，同時執行與之對應的 Callback Function。處理完 Timer Event 後，TISR 會從目前已登記的 Task Queue 中挑選一個 Task 來執行。這裡所謂的『執行』也是去呼叫該 Task 事先所登記好的 Callback Function。在實作上，每一個 Callback Function 都應該採用 State-Machine 的方式來撰寫。這樣不只是為了能有比較大的彈性，同時 MOS 也需要依靠 Task 執行的狀態來判斷一個 Task 是否已經執行結束：當 MOS 發現一個 Task 已經結束時，便會自動將之從 Task Queue 中移除。

由於 Callback Function 是在 TISR 中被呼叫，Callback Function (的不同 State) 的執行時間不可能剛好符合 TISR 的時間，因此會發生兩種狀況：

1. 在下次 TISR 發生前，Callback Function 就已經結束回傳。

2. 在下次 TISR 發生前，Callback Function 尚未結束。

在 2 的狀況中，發生 TISR 時若前一個 task 尚未執行完畢，就不應該重複執行，以免干擾目前正在執行的工作。MOS 藉由登記/檢視 Busy Flag 來作為判斷的依據。由於每個 Priority 相對應的 Task Queue 都只會執行最早登記的 Task，所以每個 Priority Level 對應一個 Busy Flag 就能順利運作。在多 Task 的狀況，則會視優先順位而定：MOS 會先挑選高優先順位的 Task，而不論低優先順位的 Task 是否正在執行。也就是說，高優先

順位的 Task 會取代 (中斷) 低優先順位 Task 的執行。

在本系統中，由於跟 Beacon 相關的 Beacon Task 對於即時性的要求十分嚴謹，故在此被賦予最高的 Emergency 優先權，而跟接收資料相關的工作(RX)因為不可被其他 Task 中斷，所以在此設計中擁有第二的 Top 級優先權。再來是 MAC Command 相關的封包傳送，本系統給予 High 等級的優先權。而一般與 MAC 上層的介面做資料交換的工作，在此採用 Normal 優先權。最後。對於時間要求並不必非常精確的工作，如等待傳送中的封包，則被設定為 Low 優先權。

## (二) Beacon-enabled Network 收發機制設計

在這段中將對於本系統在採用 Beacon-enabled Network 時的資料收發機制。

### a、Beacon-enabled Network 之資料接收機制

圖 9 是本系統所使用的 Beacon-enabled Network 之資料接收機制。一般而言，採用 Beacon-enabled 的資料接收機制都是 End Device。在接收機制啟動之後，首先要跟系統註冊一個優先權為 Emergency 的 Beacon Task 並設定預計之喚起時間(Preamble)，當時間到的時候，系統會開啟接收天線(RX)，同時註冊一個 Support Timer 的 Timer Event 以及與之對應的 Timeout Task，這個 Timeout Task 是負責監控 Beacon 是否有在預計的時間內被收到。

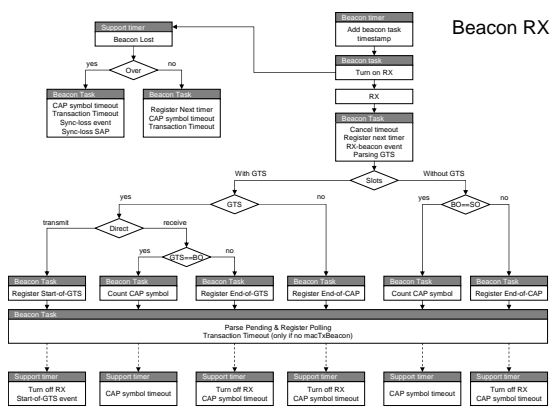


圖 9、本系統用於 Beacon-enabled Network 之資料接收機制設計

若在 Support Timer 所設定的等待時間內並未收到任何 Beacon 訊號，則 Timeout Task 會紀錄有 Beacon Lost 的情形發生，同時累積 Beacon Lost 次數(當 Beacon Lost 次數超過設定值時會進入 Sync-loss 的狀態)，接著它會註冊下一個要被喚起接收 Beacon 的 Task。若在 RX 開啟的時間內有順利接收到 Beacon 訊號，則 Beacon Task 會主動把之前設定的 Timeout Task 取消，同時註冊下一次要被喚起接收 Beacon 的 RX Task，然後就開始解析位於 Beacon 內的 GTS 封包。透過解析 GTS 封包，End Device 可以得知自己是否有使用 GTS。

若該網路自己並未使用 GTS 機制，它接著會判斷是否 BO 等於 SO。當 BO 等於 SO 的時候，代表這個網路沒有 Inactive 區間，所以 RX 必須一直保持開啟。如果 BO 不等於 SO，則它只需要注意 CAP 的運作區間即可，此時它會註冊一個 End-of-CAP 的 Timer Event。

若該網路有使用 GTS 機制，則它首先要判斷自己是否有使用到 GTS 的功能。若沒有，則它一樣只需要注意到 CAP 的運作區間即可，此時它也是用 Support Timer 註冊一個 End-of-CAP Event。若有，則它還要去判斷其所保留的是一個接收或是一個發送區間。若為接收，判斷是否 BO 等於 SO。當 BO 等於 SO 的時候，代表這個網路沒有 Inactive 區間，所以 RX 必須一直保持開啟。如果 BO 不等於 SO，則它只需要注意到 GTS 結束即可，此時它會註冊一個 End-of-GTS 的 Timer Event。若為發送，則要用 Support Timer 註冊一個 Start-of-GTS Event，如此便會在該 GTS 區間開始前關閉 RX 並執行 Start-of-GTS Event 來處理使用者設定的 GTS 功能。

### b、Beacon-enabled Network 之資料發送機制

圖 10 是本系統所使用的 Beacon-enabled Network 資料發送機制。一般而言，只有 PAN Coordinator 或 Coordinator 才會採用 Beacon-enabled 的資料發送機制時。

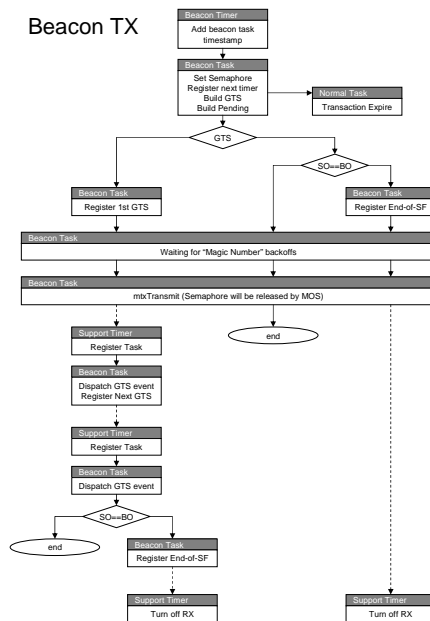


圖 10、本系統用於 Beacon-enabled Network 之資料發送機制設計

在此資料發送機制中，Coordinator 同樣要先跟系統註冊一個 Beacon Task 以及該被喚起的時間(Preamble)。當 Coordinator 被喚起時，它會先註冊下一次該要被喚起的時間，然後開始建立 GTS Frame 與 Pending Data Address List 以及完整的 Beacon Frame，接著它將整個 Beacon Frame 發送

去。在發送完 Beacon Frame 之後，它接著會判斷其本身是否也有使用 GTS 機制。若無，則它可以在等待超過 CAP 時間後將天線關閉。若有，則它要註冊其所擁有的第一個 GTS 任務起始時間。當時間到的時候它要負責處理該任務並註冊下一個未完成的 GTS 任務，如此反覆到直完成所有 GTS 工作為止。

#### 四、實作系統規格

在這個部份我們會對本實作所採用的開發平台以及一些系統設計規格做更進一步的說明。

##### (一) 實作開發平台

在本實作中所採用的開發平台是由 Chipcon 公司所出的 CC2420 DBK，在此平台上具有一顆運作時脈為 8Mhz 的 Atmel Atmega128 微處理器，其內具有 4K 的 SRAM 以及 128K 的 Flash 記憶體，在此平台上另還有一顆外接的 32K SRAM 供程式開發使用。本平台的收發晶片為 Chipcon 相容於 2.4GHz 之 IEEE 802.15.4 標準的 CC2420。CC2420 為業界第一顆支援 IEEE802.15.4 的射頻晶片，同時也是目前業界使用的主要平台。

##### (二) 本系統內 SO 與 BO 對於 GTS Slot 頻寬之影響

在 IEEE802.15.4 的標準中，決定 Superframe 參數的 SO 與 BO 具有  $0 \leq SO \leq BO \leq 14$  的關係。本系統在實作 GTS 上為符合規範，在使用 Superframe 機制時保證提供至少 440 個 Symbol 的 CAP 時間，故在本平台上 SO 對於 GTS Slot 個數 K 之限制可計算如下

$$K = \left\lfloor \frac{960 * 2^{SO} - 440}{60 * 2^{SO}} \right\rfloor \quad (3)$$

當中  $0 \leq SO \leq 14$ ，且可總結如表一，由表一中可以看出當 SO 大於等於 3 之後才可使用 IEEE802.15.4 的 GTS Slot 上限共 15 個。

表一、不同的 SO 值對於可供 GTS 使用之 Slot 個數造成的影響

| SO          | 0 | 1  | 2  | 3  |
|-------------|---|----|----|----|
| GTS Slot 個數 | 8 | 12 | 14 | 15 |

本系統在每個 GTS Slot 傳送完資料後會保留 54 個 Symbol 的 ACK 等待時間與 40 個 Symbol 的

Long Inter-frame Spacing(LIFS)時間，所以對於不同的 SO 來說，每個 GTS 可進行的最大封包傳輸個數 N 與其封包大小 P Bytes(含所有 MAC 標頭檔)具有以下關係

$$N = \left\lfloor \frac{60 * 2^{SO}}{P * 2 + 94} \right\rfloor \quad (4)$$

當中  $0 \leq SO \leq 14$ ，而圖 11 中則是分別以 P=30、60、90、120 的前提下來描繪傳輸封包個數 N 在  $SO=0 \sim 6$  的情形下的最大限制。由圖中可以看出當 SO 小於 3 時，幾乎無法傳送任何資料封包，但在超過 3 之後，可傳輸之封包數則開始大幅上升。

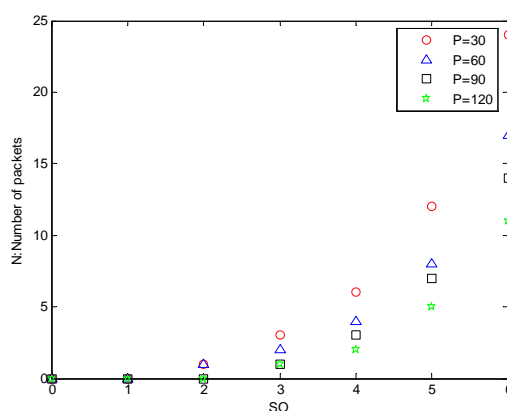


圖 11、當  $0 \leq SO \leq 6$  時，封包大小 P=30、60、90、120 對於封包個數 N 所造成的影響

##### (三) 使用 Beacon Network 時 Preamble 對於工作週期之影響

在使用 Beacon Network 時，對於工作週期影響最重要的參數就是 SO 與 BO。而除此之外，為了讓 Beacon 的同步動作可以順利進行，每個 Device 都必須在 Coordinator 發出 Beacon 之前就預先開啟(即需要一段 Preamble 時間)收發器以避免硬體時脈產生的誤差去影響 Beacon 的同步動作。在本論文所採用的平台上，其時脈誤差為  $\pm 30\text{PPM}$ ，而每次系統中斷觸發的週期為  $320\mu$  秒，所以為了實作上的方便，本系統的最小的 Preamble 單位也是  $320\mu$  秒，表二中整理了在不同的 BO 參數下，本實作對於 Preamble(SO)的設定。

表二、在不同 SO 值下本實作對於 Preamble 之設定

| SO           | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|--------------|---|---|----|----|----|----|----|
| Preamble(SO) | 1 | 2 | 3  | 6  | 12 | 24 | 48 |

由於每一個 Preamble 的時間為  $320\mu$  秒，也

就是 20 Symbol，故可以依此來在不同 SO 時，Preamble 對於原本工作週期的增加量有多大。在此在將其整理於表三中，從表三可以看出當 SO 大等於 2 之後對於工作週期的增加幾乎不會有任何影響(影響小於百分之一)。

表三、對於不同的 SO 值，本實作之 Preamble 對於工作週期(Duty Cycle, DC)造成的增加量

|        |     |     |     |     |     |     |
|--------|-----|-----|-----|-----|-----|-----|
| SO     | 0   | 1   | 2   | 3   | 4   | 5   |
| DC (%) | 2.8 | 1.0 | 0.5 | 0.3 | 0.1 | 0.1 |

## 五、結論

透過我們的實作結果，我們可以發現在 802.15.4 中採取 Superframe 架構時，一直要等到 SO 值大於或等於 3 之後才可能完全使用 15 個 GTS Slot 的系統上限。而由於我們的實作完全合乎標準規範，所以相信其他人的實作結果也會是如此。在另一方面，當 SO 值小於 2 時，我們可以發現幾乎無法利用標準的 MAC Data Frame 來傳送任何封包。表四列出了在不同 SO 值的情形下，其對應的封包大小上限。

表四、不同的 SO 值下，MAC 封包大小的上限

|             |    |     |    |     |
|-------------|----|-----|----|-----|
| SO          | 0  | 1   | 2  | 3   |
| 封包上限(bytes) | NA | 13* | 73 | 127 |

其中當 SO=1 時，依照[2]的規定可以用 SIFS(Short Inter-frame Spacing)來取代 LIFS，在此情形下，其封包之上限則會是 18 Bytes(採用 SIFS 的 MPDU 大小上限)。最後，雖然在 Beacon-enabled 的架構下需要利用額外的 Preamble 機制來防止 Beacon Lost 的情形發生，但在我們的實作中，Preamble 對於系統工作週期幾乎不會造成任何影響。

## 六、誌謝

作者在此要感謝財團法人資訊工業策進會與中華民國經濟部的“環境感知建構與普及運算技術引進計劃”對於本論文研究經費之贊助。同時也要感謝財團法人資訊工業策進會網路多媒體研究所的先進行動技術中心對於本研究的支持。

## 七、參考文獻

[1] Bruno Bougard, Francky Catthoor, Denis C. Daly, Anantha Chandrakasan, Wim Dehaene, “Energy Efficiency of the IEEE 802.15.4 Standard in Dense Wireless Microsensor

Networks: Modeling and Improvement Perspectives”, in DATE’05, 2005.

[2] IEEE 802.15.4 Task Group, “IEEE 802.15.4-2003 Standard”, IEEE, 2003.

[3] Jelena V. Mistic, Vojislav B. Mistic, Shairmina Shafi, “Performance of IEEE 802.15.4 Beacon Enabled PAN with Uplink Transmissions in Non-Saturation Mode - Access Delay for Finite Buffers”,. BROADNETS’04, 2004.

[4] Mario Neugebauer, Jörn Plönnigs and Klaus Kabitzsch, “A New Beacon Order Adaptation Algorithm for IEEE 802.15.4 Networks”, In Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN 2005), Istanbul, Turkey, January 2005.