An Adaptive MAC Protocol for Wireless Sensor Networks

Wen-Hwa Liao Hsiao-Hsien Wang
Department of Information Management
Tatung University, Taiwan
whliao@ttu.edu.tw e9312007@ms2.ttu.edu.tw

Abstract

無線感測網路的 MAC 通訊協定對於能源的消耗控制，扮演著很重要的角色，在有限制的能源前提下，如何有效地利用電力是很重要的議題，無線感測網路的 MAC 通訊協定最浪費電力的部分是在閒置 (idle) 等待的狀況，能夠讓裝置在閒置等待的這段時間適當的關閉無線網路訊號發射器是節省電力的重要關鍵，由於 PMAC [15] 的作法可以讓長時間沒有傳輸動作的感測器快速的進入休眠狀態，這樣的做法可以大為改善感測器過聽 (over hearing) 的問題，也可以使整個網路架構充分地適應實際的傳輸流量而不會產生過多的能源消耗，但是這種設計需要精確的時間同步機制。在感測網路架構中，要達到時間的同步卻是非常消耗能源、非常昂貴的一種機制，當達到目的的同時也會產生多餘的能源消耗，減少感測器的生存時間。另外，每次週期後與鄰居感測器的樣板 (pattern) 交換動作，不僅產生了額外傳輸資料的能源消耗，同時也伴隨著傳輸過程中，競爭、碰撞與樣板交換失敗等等的因素。在我們這篇文章中，將提出一種非同步的網路運作架構 AMAC (Asynchronous MAC protocol) 期待在 PMAC 的機體架構下，改善睡眠排程交換所造成的能源浪費與時間同步的問題。

1. Introduction

無線感測網路的架構下，感測器的設計以省電、價格低廉、體積小、且具有感測環境能力的裝置，為主要的設計目標，感測器本身就像是一台小型電腦，並配備了簡單的感測、運算、與無線傳輸等裝置，而感測裝置可以針對環境中我們所感興趣的事物 (例如：溫度、光源等) 做出偵測、感應的行為，並將所收集的資料先做簡單運算處理後，再透過無線傳輸裝置將資料回傳給資料收集器，最後，我們就可以根據資料收集器所收集的資料，了解我們所感興趣的環境的狀態，並且開發設計出我們所希望的特殊應用目的。

而感測器網路的運行方式，與目前傳統無基礎架構網路 (ad hoc network) 非常相似，不過，即使無基

礎架構網路的架構是最為接近無線感測網路的架構，但是，以現有無基礎架構網路的協定及演算法，大都無法直接應用到無線感測網路上。原因在於其中仍然存在著許多兩架構相異之處，而在這些限制與相異之中，感測器的電源能量問題是被討論最多，而且是無線感測網路的研究重點項目，並且有限的能源這個因素，也是感測網路與 ad hoc 網路兩者最大的差異。因此，在設計感測網路傳輸協定與演算法時，能源消耗的限制因素，大部分都會被融入設計的考慮之中。除此之外，每一個感測器裝置容易受到外在環境的影響而發生故障，所以當感測器故障時，需要擁有足夠的容錯能力，讓整個感測網路可以正常發揮應有的功能而不會因為受到某些感測器的故障而影響。因此，能量消耗、容錯能力、另外加上生產價格，這三者被視為設計感測網路的首要考量與研究重點。

由於感測器本身能夠攜帶的電源十分有限，大部分的裝置都僅有電池來供應所需要的電源，然而，感測器在偵測環境、資料運算、以及將資料傳輸給後端的無線資料感測器時，往往都會需要耗費為數可觀的電力，倘若沒有特別的省電機制，感測器本身的使用壽命勢必會大幅縮短，造成感測器裝置必須常常更換，我們知道要更換感測器或是感測器電池，是非常耗時與花費相當高的成本，甚至在某些特殊應用環境之下 (例如：戰場)，感測器的更換或是充電並不是那麼容易可以完成。因此，如何提供一個良好的省電機制，讓裝置可以運行的更長更久，可以說是無線感測網路所需要考量的重要研究議題 [3]。

根據以上的一些討論因素，我們知道，能量的消耗是感測器網路設計必須首先考量的因素，因此大多相關的研究幾乎以降低能量消耗為目標 [9]。睡眠機制是最有效節省能源的一個方式之一，但如何安排睡眠排程而不影響感測器網路的正常運作十分重要，尤其此排程不能影響感測器網路感測的功能，亦即是網路的感測涵蓋範圍 (sensing coverage) 不能降低太多，否則此網路便無法正常運作。此目標的限制因素在於感測器裝置上的演算法不能夠太複雜，卻又必須被要求具有較高效能，因而設計難度也就大幅提高。例如我們所熟知的 S-MAC [13]，

主要是將感測器的睡眠機制置入 IEEE 802.11 中的 MAC 傳輸協定中，目的用來節省能量的消耗。設計一個有效率的睡眠排程 (sleeping schedule) 為 S-MAC 的主要節省能源方法，其中這個方法的主要的精神是利用相互為鄰居的感測器，彼此同步建構出一個共同的睡眠排程，因此感測器彼此間的通訊變得非常的容易。此方法的最大缺點是感測器必須定期執行同步維護，一個具有經常性額外消耗能源的一種機制。

TDMA (Time-Division Multiple Access) 系統是一個非常有效率並且能完全避免碰撞的機制，因此許多學者建議使用 TDMA 作為無線感測網路的通道存取控制 [1]，但是在 TDMA 系統中必須是具備非常精確的時間同步 (clock synchronization) 機制，因為在時槽 (time slot) 的安排上必須是正確而沒有任何時間偏差的，如果使用不是非常精確的時間同步，則 TDMA 將無法避免碰撞的產生。但是這點在無線感測網路中，卻是一項比較困難且不合效益的限制條件。

在這篇文章的第二部份，我們將針對目前在無線感測網路所使用的各項重要的 MAC 協定加以說明，第三部分我們將所提出來的傳輸協定 AMAC，做詳細的說明與分析，第四部份定義出排程的產生規則與演算方法，第五部份針對我們所提出的方法加以模擬與驗證，最後再第六部份做出結論與未來的期望。

2. Related Work

我們都知道 S-MAC 傳輸協定是在無線感測網路中一個很特別的設計，他的設計重點在於將感測器週期性的進入睡眠狀態來取代過聽 (idle listening) 的問題，雖然 S-MAC 相較於 802.11 [4] 可以節省更多的能源，但是這種固定週期的設計對於網路流量的適應方面，並不是非常的良好，因為無論環境如何的變化，感測器傳輸協定並不會因此而改改變傳輸的模式，這種情況當感測器的運作週期高但傳輸流量小的時候，許多能源將會因此而浪費 [7] [8]，相反的，當運作週期低，傳輸流量高的時候，整個網路的傳輸能力就因此而被限制，資料的流通傳遞將會花上更多的時間 [12] [6] [10]。然而 Timeout-MAC (TMAC) [2] 協定適度調節了感測器的運作週期，當一個感測器在經過時間 T_A 後沒有任何活動的情況下，感測器便會重新進入睡眠模式。這樣的設計解決了要選擇多長的時間當做運作週期的問題，因此，在相同的交通流量條件之下，TMAC 可以與 S-MAC 有著相同的效能表現，但是卻較 S-MAC 擁有更少的能源消耗。

另外，在 Pattern-MAC (PMAC) 的設計理念中，感測器可以依據自己本身的資料傳輸狀況產生合適的睡眠排程，同時也可以藉由與鄰居感測器的排程資料交換 (pattern exchange) 達到擁有鄰近感測器的睡醒排程的目的。經過許多的實驗分析驗證結果，這樣的設計的確相較於 S-MAC 可以產生更高的資料穿透力 (throughput) 與省下更多的能源。這樣的方法其中隱含了兩個問題。第一，為了要增加鄰近感測器發現 (neighbor discovery) 的速度，因此必須要在睡眠排程產生之後再與鄰居感測器執行排程資料的交換，所有的感測器必須在排程交換週期 (PETF) 內進行資料的交換動作，所有的感測器競爭使用通道，因此勢必產生極大的資料碰撞問題，為了解決通道競爭所產生的資料碰撞，必須給予一段足夠的通道競爭時間 (contention window)，也因此對整個傳輸協定產生了過多的負擔 (overheads)，同時也造成了過多能源的消耗。第二，在無線感測網路架構的特性來看，要達到感測器時間的同步 (clock synchronization) 也要付出較多的能源消耗，時間的同步問題與鄰近感測器的發現問題，在傳輸協定的設計研究方面，也都是各界所一直想努力的改善目標。

在排程中有很明確指定出醒來的時間機制之下，所有進入睡眠模式的感測器必須同時在排程中約定的時間醒來，並且週期性的醒來執行這樣的步驟，如此才能確保感測器之間有相互通訊的機會，像是 IEEE 802.11 中所制定的省電模式 (Power Saving Mode, PSM) 所有參予的感測器擁有相同的時間週期，並且沒有任何時間偏差，每一個感測器共同醒來的時間也是被安排好的，這樣的機制設計非常適合單步式 (single hop) 網路架構，所有的感測器都可以彼此聽到對方的信號，但這樣的設計並不太適合多步傳輸式 (multi hop) 網路的網路架構，像是 ad hoc 網路架構或是 WSN 網路架構。因為這樣的分散式網路架構明確地隱含著時間無法一致的問題，這也表示著這樣的結果必然無法在動態的網路結構上有很好的表現 [5]。舉例來說，在時間同步的網路架構下，若兩個子網路 (sub-networks) 擁有不同的睡眠排程，並且排程中沒有任何的重疊醒來時間，這樣的結果會導致感測器永遠沒有辦法發現對方，即使在實際的地理位置上兩感測器的位置是非常相鄰的。

在非同步的傳輸協定方面 [11]，作者提出幾種睡眠策略並且建議 IEEE 802.11 PSM 必須支援非同步的睡眠排程方式，非同步排程方式最大的特色就是不需要進行時間的同步校正，每一個感測器進入睡眠模式時，只需要按照自己的規劃時程動作，其中醒來的時間必須要與鄰居感測器部分的重疊，好讓彼此“聽”的見對方的訊息。但 TDMA 的架構

下，每一個時槽都必須精準的對齊，都不能容許任何的時間偏差，要達到這樣的時間同步，其實並不適合分散式架構的無線感測網路，因而產生了一個想法，是否可以將 PMAC 傳輸協定改成以非同步的模式來運作，我們參考 quorum-based 的非同步架構，以 PMAC 的運作理念加以實現，提出一個非同步傳輸協定 AMAC，期待能讓現有的網路架構變得更簡單、更有單性、同時也更具有效益的一種新的 MAC 傳輸協定。

相較於固定式的睡眠排程週期，爲了要達成不需要時間同步的目的，通常要讓感測器醒來的頻率變得比較頻繁，這也是必然的一種現象。但是這樣的架構可以使得網路的建置變得比較簡單，同時也可以適應高度動態變化的網路結構。換句話說，非同步的睡眠策略也可以視爲能源消耗交換網路可靠度，一體兩面的問題。在設計非同步協定的考量方向，將是如何減少進入睡眠的裝置醒來的次數所產生的能源消耗與縮短鄰居感測器發現的時間所造成的資料傳遞延遲。

3. Protocol Overview

有鑒於時間同步模式運作之下必須定時的調教感測器時間的偏差量，所產生的能源消耗也十分可觀，我們提出一個非同步的協定 AMAC；依據 PMAC 的觀念之下，巧妙的利用了睡眠時間的長度變化來達到適應環境的能力，在我們設計的傳輸協定中，保留了原來的睡眠排程產生規則，而將原來排定的“醒”的時間（字元 1），以一連串醒的時間（字串 1111...1）來取代，將原來的“睡”（字元 0），以第一個醒來時間加上一連串的睡眠時間加以取代（字串 10000...0），這樣做的目的是希望感測器僅依據自己本身的交通傳輸量來產生睡眠排程規劃，不論與鄰居感測器的時間偏差量有多大，仍然可以確保兩個感測器之間最少有兩個時槽的機會可以彼此溝通。依據 quorum 的睡眠排程陣列法，在陣列中任取一行、列的選擇下，能確保最低重疊區域的特性，可以將這樣的想法所產生的睡眠排程策略，應用在不需要時間同步機制的網路架構上，同時也保留了 PMAC 設計的精神，如何讓睡眠排程具有環境適應的能力，我們將在以下的章節中，針對 AMAC 協定的實行方式加以詳細地說明。

3.1. Protocol Details of AMAC

有關非同步運作架構的研究資料也非常眾多，但大多數都局限於本身固定式的睡眠排程策略，並沒有辦法對於實際面臨網路交通流量的多寡來適度調節、反應。在現行的非同步運作方法中，大多努

力的焦點都集中在如何能縮短感測器顯著等待的時間，而又能確保彼此間能有機會相互通訊，當然縮短感測器顯著等待的時間，這對於能源要求極高的感測網路架構上是很重要的。AMAC 的設計理念就是希望以非同步的 MAC 協定爲主要的運作架構，同時還能如 PMAC 般的能夠依據交通負載狀況而產生不同的睡眠排程，我們選擇了採用 quorum 矩陣的方式來實現這個目的。

3.2. The Concept of Asynchronous Protocol

在分散式運算的網路架構中，每一個感測器都必須視爲一個獨立運算的個體，讓裝置進入睡眠狀態是達到節省能源的最好方式，但對一個合作型態的網路架構而言則是一項嚴重的挑戰，因爲裝置之間必須相互溝通，交換資料。這裡明白地指出了兩個重要的設計目標，第一、如何讓沒有任何傳輸需求的裝置能進行長久的睡眠，以達到節省最大能源的目的。第二、如何能讓裝置在需要資料傳輸時，快速的能與其他裝置溝通，而不會因爲無法發現其他裝置的狀況而造成資料傳輸上的時間等待延遲，在時間同步的狀態下，每一個裝置的動作時程都是可以知道的，因此，感測器所需要等待傳輸的時間不會是無限制的，所以也不會有鄰居感測器發現的問題，只要大家的時間標準不要造成偏差，依據排程在指定的時間上進行傳輸，原則上是一定可以完成資料傳遞的。在非同步的設計中，不可避免的要花費較多的醒來時間，雖然如此，少去了同步模式定期所必要執行的時間同步而言，以一些醒來時間來換得非同步的傳輸架構而言，同時也提高了網路的彈性，這樣的想法仍然是非常良好的設計。

3.3. Asymmetric AMAC design

以 WSF [14] 而言，是一個非常良好的對稱式 (symmetric design) 非同步網路架構，所謂對稱式架構指的是所有的感測器擁有相同的醒來時間 (duty cycle)，相反地，非對稱式架構指的是每個感測器所醒來的時間總量不一定相同。在環境適應的前提下，如果把感測器清醒的時間總量加以固定，似乎並不是很恰當的設計，因爲事件的發生是沒有辦法預測的，也許這個時間傳輸量資料的需求很小，但下個時間馬上會發生極高的傳輸需求。因此，設計一個非對稱的睡眠排程架構，在非同步的傳輸模式之下，這樣的組合應該是比較恰當的無線感測網路設計。

我們知道，在 quorum 的架構下，進入睡眠模式的感測器必須定期的醒來發送訊號 (beacon)，發送

訊號數量為 $O(\frac{1}{n})$ ，在非同步的傳輸模式下必須保

持一定的醒來時間以便讓需要傳輸的感測器有機會進行資料傳輸，quorum 的設計概念非常適合大範圍的分散處裡架構，像是無線感測網路架構就會顯得非常的適合。我們發現在矩陣中任意兩組行、列所形成的集合，可以保證最少有兩個相互交疊的機會，如果將每一個矩陣單位視為一個時間，這也就意味著這兩個矩陣所選擇的集合最少有兩個時間單位是一相同的。而非同步的架構代表著每個感測器都有著各自的運算時間，也就是每個感測器的時間偏差最大可以是一個完整週期。以 Figure 1. 為列，感測器 A 選擇 $r1, c1$ 所形成的清醒時間與感測器 B 選擇 $r2, c2$ 所形成的清醒時間，兩組集合在時間 2 及時間 9 兩個時間共同醒來，也就表示當 A, B 感測器在時間 2, 9 可以共同醒來進行資料通訊。

Figure 1. Quorum-Based Protocol

許多的證明顯示 quorum 的方法不論時間偏差量有多少，都能確保最少有兩個完整的時槽可以完整的重疊。

3.4. Different matrix size on Quorum interval

Quorum 的設計是以一個 $n \times n$ 的列優先 (row-major) 二維陣列，總共具有 n^2 個時間區間組成的睡眠排程週期，所選定的列區間，可以視為一段連續醒來的時間，而所選定的行區間，則是為了避免讓連續睡眠的時間大於連續醒來的時間，因此，行區間的目的便是為了讓每個連續醒來時間之內一定有一個醒來的時間相重疊。這也隱含著矩陣中列的大小增加，並不會影響這個方法的特性，也就是說，在固定矩陣行寬的情況下，矩陣列的大小改變，彼此間還是存在著最少有兩個時間機會重疊的機會。在 Figure 2. 中，矩陣 A 是由兩個列長為 x (pattern x) 所形成的矩陣，矩陣 B 是由列長為 y (pattern y) 所形成的矩陣，在行寬都是 n 的情況下，兩個矩陣所選出的清醒區間 (quorum intervals) 仍然具有上述的特性，若 $x = \frac{y}{2}$ 則代表 A 矩陣相較於 B 矩陣只需要一半的鄰居發現時間。這樣的特性

也相當直覺可以觀察，在矩陣 A 中，可以將它視為選擇了一行兩列的 $n \times y$ 矩陣，由於較矩陣 B 多選擇了一列，醒著的時間幾乎變長了一倍，也就意味著在矩陣 A 中，鄰近感測器與其作資料傳輸的機會也增加了一倍。這樣的結果對於期待設計一個非對稱式的睡眠排程來說是很有用的，只要能控制列的數量，便能夠控制睡眠排程中醒著的時間，也就能將感測器本身依據環境狀況加以調整睡眠排程策略。

Figure 2. 相同行寬下，不同大小列長的矩陣仍然具有 Quorum 的特性

3.5. Pattern Generation of AMAC

首先我們觀察 PMAC，依據感測器是否有資料要傳輸作為睡眠策略的產生依據，利用睡眠時間的長短調節來達到適合網路流量的目的，在沒有任何傳輸流量的情況下，PMAC 的排程如下：

$$1, 01, 0^21, 0^41, \dots, 0^{\delta}1, 0^{\delta}01, \\ 0^{\delta}0^21, 0^{\delta}0^31, \dots, 0^{m-1}1.$$

原來的的方式，利用字元 "0" (代表感測器睡眠時間)、字元 "1" (代表感測器醒著時間) 的排列組合所形成的字串，當作是感測器睡眠排程的依據，在這樣的前提之下，我們所提出的傳輸協定 AMAC，試著將原本的字元 "0"、字元 "1" 以一個字串加以替代。即以 n 個字元 1 所形成的字串 "1111...1" 取代原來的字元 "1" 以一個 1 加上 $n-1$ 個 0 所形成的字串 "1000...0" 取代原來的字元 "0"。重新調整過的結果如下：

$$(1111 \dots 1), (1000 \dots 0)(1111 \dots 1), (1000 \dots 0)^2(1111 \dots 1), \\ , (1000 \dots 0)^4(1111 \dots 1), \dots, (1000 \dots 0)^{m-1}(1111 \dots 1)$$

我們選擇任何一個排程樣板，以 quorum 矩陣的方式加以排列，可以得到如 Figure 3. 般的陣列組合，每一個睡眠排程週期時間長度總共有 N 個時槽 ($n \times m$)，依據 AMAC 規則所產生出來的樣板，當樣板的時槽數量小於總睡眠排程 N 時，則重複現有的

樣版，直到填滿全部的時間 N 為止。這樣特殊排列的目的是為了讓新的睡眠排程能夠符合 quorum 矩陣的特性（即醒來的時間在陣列中為一行與列的集合），如此便能讓這樣的睡眠排程運作在非同步的模式架構中。

Figure 3. AMAC 依據 Quorum 特性所產生出來睡眠排程矩陣

針對這樣的方式，舉例來作說明，若某個感測器 A 具有整個睡眠週期 $N=16$ ($n=4, m=4$)，睡眠樣板為 (0^21) ，依據 AMAC 的睡眠排程產生方式，一個樣板週期為 001，轉換之後實際結果為：0001、0001、1111、0001，用這個樣板所產生出實際上的睡眠排程為 0001、0001、1111、0001，如 Figure 4. 中所顯示出實際的排程結果。其中，設計睡眠排程週期的目的是希望在經過一段時間之後，可以調整睡眠樣板，以符合實際網路的傳輸需求。在我們的排程產生規則中，當感測器發生感測到事件發生而產生傳輸需求時，立即將該感測器執行排程產生，這時候無論目前的感測器處於何種排程樣板，均會立即恢復到長時間醒來的狀態，如此可以應付隨之而來的傳輸需求。同時鄰居感測器也不需要彼此的資訊交換，每個感測器均保持獨立運作的狀態。

Figure 4. AMAC 利用樣板: 001 所產生的實際排程

3.6. Analysis of AMAC

在 AMAC 的方法中，所設計的矩陣不一定要是方陣形式 ($n=m$) 但是設定過大或者過小的 n 與 m 對於網路的穩定性都會有不良的影響，由實驗中可以得知，若 $n > m$ 很多，則感測器睡醒的切換時間比

較長，對環境的適應能力也比較差，若 $n < m$ 則會形成感測器切換過於頻繁，容易造成額外的能源消耗，許多研究資料證明，感測器在切換睡眠模式時也是能源消耗的來源之一。另一方面，就醒來的時間分析：AMAC 協定在一個完整週期中，總共醒來的時間近似為 $m+n-1/m*n$ 當矩陣越趨於方陣的設

計時 ($n \approx m$)，則醒來的時間分析為 $O(\frac{1}{n})$ ，在整個

睡眠週期 N 固定的情況下，選擇 n 的大小為 $\sqrt{N}$ ，則醒來的時間將會最少，這也就代表著可以省下更多的能源。若矩陣設計的越大，則在網路流量較輕的狀態下，醒來的時間比睡眠的時間，相對的就少很多了。Figure 5. 中呈現出在相同大小的睡眠排程下，利用不同的樣板所形成不同比例的睡眠排程，但這些不同的排程，不論時間偏差多少，彼此間仍都具有部分清醒時間重疊的機會。清醒的時間可以減少資料的傳遞延遲，睡眠的時間可以減少過多的能源消耗。

Figure 5. 不同的樣板值所產生的排程結果

3.7 事件觸發重新排程機制

在 AMAC 的方法中，我們利用事件的觸發這樣的機制來重新調整感測器的睡眠排程，當事件發生的越頻繁時，感測器將會有較多的清醒時間，藉以擔任資料傳輸的任務，不論目前執行的睡眠樣板為何，一旦事件發生，感測器依據原訂的睡眠排程計畫，嚐試傳送資料，等到資料傳送成功，便立即將睡眠樣板更改為 '1'，並且重新產生睡眠排程。資料接收端在完成資料傳輸後，亦即改變睡眠樣板為 '1' 同時也重新產生睡眠排程，如 Figure 6. 所示，這樣的作法對於有連續資料傳輸需求的情況時，可以大大降低傳輸上的延遲時間，並且經常傳輸的感測器組合將會具有相同的睡眠排程。

original schedule:

event trigger adjust schedule:

Figure 6. 事件觸發重新排程機制

4. Pattern Generation Algorithm

感測器 j 的樣板產生規則 P_{n+1}^j 整理如下：

- (1) 01: if $P_n^j = 1$ 同時感測節點 j 在 '1' 之後的時間沒有任何資料需要傳送;
- (2) $0^{2m}1$: if $P_n^j = 0^m1$ ($0 < m \leq \delta/2$) 同時感測節點 j 在 '1' 之後的時間沒有任何資料需要傳送;
- (3) $0^{m+1}1$: if $P_n^j = 0^m1$ ($\delta \leq m < N-1$) 同時感測節點 j 在 '1' 之後的時間沒有任何資料需要傳送;
- (4) 0^m1 : if $P_n^j = 0^m1$ ($m=N-1$) 同時感測節點 j 在 '1' 之後的時間沒有任何資料需要傳送;;
- (5) 1: 若感測節點 j 有資料要傳送時, 不管之前的位元狀態為何。

首先, 將演算法中所定義的參數整理如下:

N : 完整睡眠週期長度

n : 矩陣列寬

P : 睡眠樣板

S : 睡眠排程結果

Function of SCHEDULE_GENERATION (P):

```
{
  For ( $x=1$ ;  $x < N$ ;  $x++$ ){
    For ( $y=1$ ;  $y < P.Size$  and  $x < N$ ;  $y++$ ){
      If ( $P.[y]='0'$ )
         $S = S + '1000...0'$ ; //1+n-1 個字元'0'
      If ( $P.[y]='1'$ )
         $S = S + '1111...1'$ ; //n 個字元'1'
       $x=x+n$ ;
    }
  }
}
```

Return S ;

}

5. Simulation Experiments

為了證明我們所提出的 AMAC 真的能夠對於環境的變化有良好的調整機制, 我們設計一個模擬感測器運作的環境加以驗證, 以下針對我們所規劃的環境條件加以說明。

為了讓結果更貼近實際感測網路的應用環境, 我們模擬了 100 個無線感測器裝置動作結果, 每個感測器的運作週期 (duty cycle) 為 100 個時槽 (time slot), 採用長、寬為 10×10 個時槽的排程矩陣加以規畫, 依據 AMAC 規則產生感測器的睡眠排程。執行時間設定為 1000~10000 個時槽, 每經過一個完整週期之後 (100 個時槽) 重新依照網路狀況調整排程樣板, δ 用來控制網路的安定性, 另外我們定義了兩個參數: α : 即無線感測裝置在每個時槽發生資料傳輸需求的機率, β : 當產生資料傳輸需求時, 連續發生資料傳輸需求的機率。利用這兩個參數模擬出無線感測網路的應用環境, 因為我們發現在真實的應用環境中, 感測事件的發生往往都不是很平均的情況, 也就是說, 也與很長一段時間都沒有任何傳輸需求, 但下一刻又發生連續很多資料需要傳輸的情況。

首先, 我們驗證了在固定排程下的睡眠策略與 AMAC 兩相比較, 當 AMAC 偵測到傳輸需求的時候, 不論現在的排程樣板為何, 立即改為第一個樣版重新執行排程, 這裡所謂的第一個樣板也就是不含有任何睡眠時間的排程, 由於 AMAC 的運作方式採用時間非同步設計, 感測器之間並不需要進行任何的資料交換, 像是如 PMAC 的 pattern exchange 那樣的需求, 所以每一個感測器可以獨立的再任何的時間進行重新排程的動作。我們將連續事件發生機率設定為 $\beta=0\%$ (每次僅傳送一個時槽的資料量) 測器偵測到事件的機率設定為千分之 2 到 14, 計算出每個事件的平均等待傳輸時間如 Figure 7. 所表示, 採用 AMAC 的協定之後的傳輸等待時間大為減少, 這表示當感測器感測到事件訊息時, 資料停留在緩衝區的時間並不會很久, 可以減少資料延遲的時間。

Figure 7. 具有環境適應能力的協定可以降低傳輸等待時間

我們發現當資料傳輸需求越高的時候，感測器應該提高清醒的時間比例來增加資料的傳輸量，在固定排程的設計之下，感測器始終保持一定的清醒時間，但這樣的設計在傳輸需求增加時，相對的資料傳輸的等待時間也隨之增加，在實驗結果中如 Figure 8. 發現，AMAC 在當感測環境傳輸需求增加時，會更密集的觸發感測器重新睡眠排程，藉以提高清醒時間的比例，降低資料傳輸的等待時間。

Figure 8. 越高的傳輸需求促使 AMAC 提高清醒的時間比例

再來，我們知道在無線感測網路的應用環境中，事件的發生往往不一定成常態分布狀，應該是突然會出現集中在某一時間，形成區域性的資料傳輸需求型態，為了模擬這樣的情況，我們將事件發生的機率 α 加以固定，利用每次發生事件時，連續的資料需求 β 來產生不同的資料量，來加以驗證。我們將事件發生需求固定在千分之 1，資料連續傳輸的機率設定從 0% 到 90%，發現具有環境適應能力的 AMAC 仍舊能夠反應突然變化的傳輸需求，Figure 9. 中明白的顯示不論在同一時間的資料傳輸需多少個時槽，都可以在很短的等待時間下，將資料迅速的傳遞。當資料不均勻的出現時，AMAC 會將目前的排程改以醒著機率較高的排程取

代，因此對於突然而來的較高資料傳輸需求也可以很快的適應。

Figure 9. 在不同機率下發生連續事件的傳輸等待時間

在無線感測網路的應用環境中，大部分的時間並不是在資料傳輸方面，而是在感測事件的發生，而大多數的應用環境的事件通常是很久才會發生一次，因此，我們仍舊必須考量在網路長時間不傳輸資料情形下的睡眠狀況，以我們的實驗假設條件而言，當網路沒有任何資料傳輸需求的時候，依據 AMAC 的排程規則，可以讓感測器很快的進入較高比例的睡眠狀況，如 Figure 10. 所呈現的結果，在很短的時間之內 (大約經過十個排程週期)，感測器可以進入睡眠比例約為 80% 的睡眠排程中，但是一但偵測到有任何的事件發生，仍然可以在很短的等待時間之下，馬上對目標感測器進行資料傳輸，這樣的設計對於能源的節省，有很大的幫助，同時也不會造成資料傳輸等待時間過長的情況。

Figure 10. AMAC 在短時間快速進入高睡眠比例的狀態

6. Conclusion

經過實驗分析證明結果發現，AMAC 的確對環境的變化，有非常良好的適應能力，在無線感測網路的架構之下，時間非同步運作的模式是比較接近

分散運算的精神，不但不需要考慮時間的偏差對整個網路運作所造成的影響，同時也使裝置省去了時間同步所需執行額外的同步機制，同時對於感測器的加入、移除也更加的具有彈性。

如何讓網路能在不需要傳輸資料的時候執行長時間的睡眠以節省能源以及在資料傳輸需求產生時，很快的完成資料傳輸的動作，減少資料傳輸延遲，這是我們設計傳輸協定所努力的方向與目標，如何在兩者之間做有效的取捨、協調，也是設計一個良好的傳輸協定時所必須要考量的重要因素。

References

- [1] S. Cho, K. Kanuri, J. W. Cho, J. Y. Lee, and S. D. June, "Dynamic Energy Efficient TDMA-based MAC Protocol for Wireless Sensor Networks," *IEEE Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS-ICNS)*, 2005.
- [2] T. V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *ACM Embedded Networked Sensor Systems (SenSys)*, 2003.
- [3] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: A Survey," in *IEEE Communications*, Vol. 44, No. 4, April 2006, pp 115-121.
- [4] LAN MAN Standards Committee of the IEEE Computer Society, "IEEE Std 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," *IEEE*, 1999.
- [5] G. Lu, B. Krishnamachari, and C. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks," *IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2004.
- [6] M. J. Miller, C. Sengul, and I. Gupta, "Exploring the Energy-Latency Trade-Off for Broadcasts in Energy-Saving Sensor Networks," *IEEE International Conference Distributed Computing Systems (ICDCS)*, 2005.
- [7] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-Efficient MAC: Energy-Efficient Collision-Free Medium Access Control for Wireless Sensor Networks," *ACM Conference On Embedded Networked Sensor Systems (SenSys)*, 2003.
- [8] Q. Ren and Q. Liang, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *IEEE Global Telecommunications Conference (GLOBECOM'05)*, 2005.
- [9] Y. E. Sagduyu and A. Ephremides, "The Problem of Medium Access Control in Wireless Sensor Networks," *IEEE Wireless Communications*, 2004.
- [10] C. Suh and Y. B. Ko, "A Traffic Aware, Energy Efficient MAC Protocol for Wireless Sensor Networks," *IEEE International Symposium on Circuits and Systems Circuits and Systems (ISCAS)*, 2005.
- [11] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks," *IEEE Computer and Communications Societies (INFOCOM)*, 2002.
- [12] X. Yang and N. H. Vaidya, "A Wakeup Scheme for Sensor Networks: Achieving Balance Between Energy Saving and End-To-End Delay," *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2004.
- [13] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *IEEE INFOCOM*, 2002.
- [14] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous Wakeup for Ad Hoc Networks," *ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, 2003.
- [15] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2005.