

ES-MAC：一種無線感測網路的虛擬叢集式 MAC 層協定設計

竇奇、蕭簡浩、阮嘉翔
國立雲林科技大學電機工程學系

摘要 一本論文針對 Sensor-MAC (S-MAC)，一種採用共通式活動/休眠週期的無線感測網路 MAC 層協定，提出了一種加強型的 Enhanced S-MAC，稱之為 ES-MAC，並將其應用在無線感測網路較外層中低流量的部分。ES-MAC 分為三個階段：第一階段為建立虛擬叢集區及預設傳輸路徑；第二階段為資料收集與排序；第三階段為資料傳送。虛擬叢集區內的感測節點，利用微時槽(Mini-Slot)競爭的方式決定資料傳輸的順序。而在資料傳送階段則利用已建立的順序傳輸資料，以減少感測節點的碰撞與不必要的過度監聽，並增長節點的休眠時間。模擬結果顯示，ES-MAC 在不同流量與不同週期下，均較 S-MAC 有更低的平均封包延遲時間。¹

一、引言

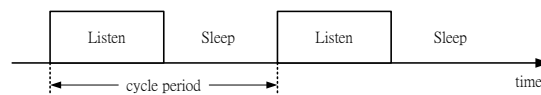
無線網路資料傳輸時，眾多的感測節點使用相同的傳輸媒介進行傳輸，所以如何利用多重存取(Multiple Access)技術，定義如何共享傳輸媒介，是十分重要的課題。多重存取技術大致可分為三類，分別為隨機式存取(Random Access)、排序式存取(Ordered Access)和組合式存取(Combinations Access)。隨機式存取：感測器節點在傳送資料時，對傳輸通道進行偵測，當通道狀態為空閒時，傳送端不需要等待，可直接發送資料給接收端，因此傳輸資料擁有即時性，此方式稱為載波偵測多重存取機制 CSMA (Carrier Sense Multiple Access)。排序式存取：將時間軸切割成數個相等長度的訊框(Frame)，並將訊框分為許多相等長度的時槽(Timeslot)，一個時槽分配給一個使用者，按照順序進行傳輸，因此可避免碰撞問題，此方式稱為分時多工存取 TDMA (Time Division Multiple Access)。組合式存取：結合隨機式存取和排序式存取技術，以增加網路傳輸效能，此方式稱做 TDMA/CSMA。

WSN 的 MAC 協定主要可分為以下三種[1]：競爭型(Contention-based MAC protocols)、排程型(Schedule-based MAC protocols)、混合型(Hybrid MAC protocols)。競爭型 MAC 協定主要使用單一通道進行傳輸，節點必須通過競爭取得通道的使用權，擁有使用權的節點才可傳輸資料。競爭型 MAC 協定經由動作/睡眠時間的調度、交握機制，提高通道的使用效率，其中以 S-MAC 為代表性的協定。排程型 MAC 協定以 TDMA 協定為主，也有其他的通道取得方式，如 FDMA 和 CDMA。排程型 MAC 協定可減少為了競爭通道所需的控制封包量，以及因碰撞重傳的能源消耗，藉由分配獨立的時槽/頻率/正交碼給感測節點，使節點只能利用特定的時槽/

頻率/正交碼取得通道，避免碰撞問題的發生。混合型 MAC 協定結合了競爭型 MAC 和排程型 MAC 的優點，其中以 Funneling-MAC 為代表。

二、相關文獻介紹

S-MAC[2]-[3]是一種架構在 IEEE 802.11 通訊基礎上，採用固定週期式活動/休眠機制的協定，如圖一所示。S-MAC 週期性的讓節點進入睡眠，可減少監聽所造成的能量消耗。節點間會協調成一個具有相同動作/睡眠週期的虛擬叢集(Virtual Cluster)，保證節點間的同步。S-MAC 採用虛擬載波監聽，以及 RTS/CTS 交換機制，避免資訊碰撞以及過度監聽。在 IEEE 802.11 中，每一個節點會監聽所有的鄰近節點，因此會監聽到不是指向本身的封包，這將造成不必要的能量消耗，特別在傳輸負載很大且當節點密度很大時。S-MAC 透過讓節點收到 RTS/CTS 以後，判斷節點是否可進入休眠，來避免節點的過度監聽。雖然 S-MAC 擁有避免碰撞、減少閒置監聽和增加網路的可擴充性等好處。但是仍存在某些缺點，例如實現上較複雜，且活動週期長度主要依靠訊息的傳輸速率，所以當網路流量較小時，仍會有閒置監聽時間過長的情形發生，而節點週期性的進入睡眠將造成資料傳輸延遲時間增加等。

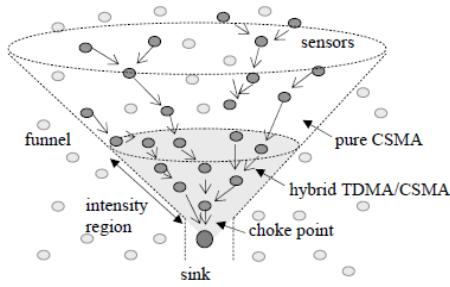


圖一：S-MAC 協定週期性機制示意圖(參考文獻 [2])

TDMA 協定[4]的基本概念是每個感測節點都使用單一的傳輸通道，並將此頻道分為數個傳送時槽(Time Slot)，各感測節點使用特定的時槽傳送/接收資料，其他的時間，節點則進入睡眠狀態，以減少能源的消耗。TDMA 協定對整個網路的同步要求非常嚴格，網路的延遲問題會隨著網路同步的精準度而變化，但是無線感測網路是由許多獨立感測節點所組成，所以要達成網路的同步非常不容易。

Funneling-MAC[5]-[6]協定是一種混合 CSMA/TDMA 的協定。Funneling-MAC 協定將 WSN 分為兩層，距離 Sink 較近的感測節點容易發生封包遺失、碰撞、壅塞等問題，因此將此區域定義為強度區(Intensity Region)，而在強度區之外則為一般區，如圖二所示。在強度區內的感測節點採用 TDMA/CSMA 的混合協定，由於 Sink 擁有比一般感測節點更強大的運算能力和能量，因此使用 Sink 來管理強度區的傳送排序。在一般區內距離 Sink 較遠的感測節點則採用 pure CSMA 協定。

¹ 本研究由國科會贊助，計畫編號 NSC 101-2221-E-224-064.

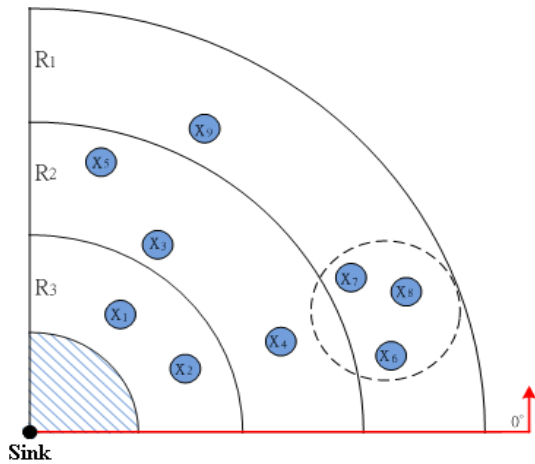


圖二：無線感測網路的漏斗效應(參考文獻 [5])

三、改良型 S-MAC 協定設計

本論文提出一種加強型的 S-MAC 協定，稱為 ES-MAC (Enhanced Sensor-MAC)，本 ES-MAC 以 S-MAC 為基礎架構，除了感測節點週期性的醒來外，另外加入許多新的機制。ES-MAC 協定可分為三個階段：

第一階段為建立虛擬叢集區，以及預設各感測節點的傳輸路徑。各感測節點將與其鄰近節點(Neighbor Nodes)構成虛擬叢集區(Virtual Cluster, VC)，虛擬叢集區內的節點擁有相同的競爭及休眠週期。本論文由 $0^\circ \sim 90^\circ$ 逆時針掃描感測節點編號，以建立 WSN 各層的虛擬叢集區，如圖三所示。以建立 R₁ 層的虛擬叢集區為例，由 0° 逆時針向上掃描到第一個感測節點為 x₆，而 x₆ 的感測範圍中包含了 x₇ 及 x₈ 號節點，並且 x₇ 號及 x₈ 號互相在彼此的感測範圍內，因此將 x₆、x₇ 和 x₈ 號節點編在同一個虛擬叢集區內。已經編入 VC 的節點，不會再編入其他的 VC，我們以此方式建立各層的虛擬叢集區。各感測節點會自動搜尋傳輸範圍內下一層的鄰近節點，且自行選定距離 sink 最近的下一層鄰近節點為目的節點，我們以此方式預設各節點之傳輸路徑。

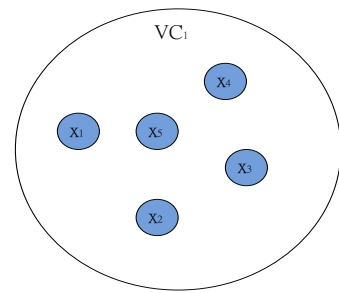


圖三：建立各層的虛擬叢集區示意圖

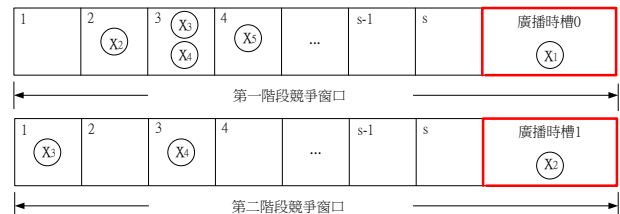
第二階段為資料收集與排序階段，一開始所有感測節點均處於活動狀態，在此階段包含了節點間微時槽的競爭機制。在此機制下，感測節點會利用競爭窗口(Contention Window)來競爭時槽，並且發送喚醒封包(Wakeup Packet)，來通知下層目的節點是否必須轉傳資料或可進入休眠。喚醒封包中包含封包標頭，由訊框控制、序列號碼和位置欄位組成，其中裝載了來源節點資訊(Source ID)及來源節點在此週期內必須傳送之資料量

(Data Records)。下層節點便可據此封包判斷在此週期內是否繼續保持醒著或可進入休眠。感測節點本身若沒有資料需要進行傳送，而且沒有接收到上層節點發送的喚醒封包時，節點在此階段便可進入睡眠狀態，以節省感測節點的能量消耗。此階段為整個協定最重要的核心，在虛擬叢集區內的節點，隨機選擇競爭窗口中的一個時槽數(Mini-Slot Number)，並在該 mini-slot 送出 Request 封包，且會有一個節點負責監聽 VC 內所有節點的競爭狀況，並且將競爭結果廣播給所有 VC 內的節點。時槽競爭成功的節點，便可發送喚醒封包來通知下層目的節點是否必須轉傳資料。利用競爭時槽便可建立各節點傳輸順序，且可確保在資料傳輸階段不會發生碰撞問題。競爭窗口將分為第一階段與第二階段競爭窗口。

在第一階段競爭窗口中有 s 個時槽，VC 內的節點可隨機選擇一個 mini-slot 傳送 Request 封包。第一階段競爭窗口包含一個監聽節點的廣播時槽。監聽節點將由 VC 內編號最小的節點，開始輪流擔任。當第一階段有兩個以上的節點選擇到相同的時槽，這些節點可在第二階段競爭中重新競爭時槽。二階段競爭窗口機制的運作可以圖四虛擬叢集區為例。圖中將監聽時槽的責任依序分配給節點 x₁、x₂、x₃、x₄、x₅；如圖五所示，我們將廣播時槽 0 分配給節點 x₁，將廣播時槽 1 分配給節點 x₂。在第一階段競爭窗口，節點 x₂ 選擇時槽 2，節點 x₃、x₄ 選擇時槽 3，節點 x₅ 選擇時槽 4，傳送 Request 封包。由於在第一階段節點 x₃、x₄ 選擇到同一時槽，因此無法成功傳送 Request 封包；所以在第二階段競爭窗口時，節點 x₃、x₄ 重新選擇時槽傳送 Request 封包，結果節點 x₃ 選擇時槽 1，節點 x₄ 選擇時槽 3。本機制將廣播時槽輪流分配給節點 x₁ 與 x₂，可確保 x₁ 與 x₂ 至少有一次競爭時槽的機會，並且給予兩階段競爭窗口以增加節點成功傳送 Request 封包的機會。經過上述步驟便可建立出每個 VC 在目前週期內各節點的傳送順序，以圖五為例，節點傳送順序為：x₂-x₅-x₃-x₄，根據此順序節點便可傳送喚醒封包給下層目的節點。



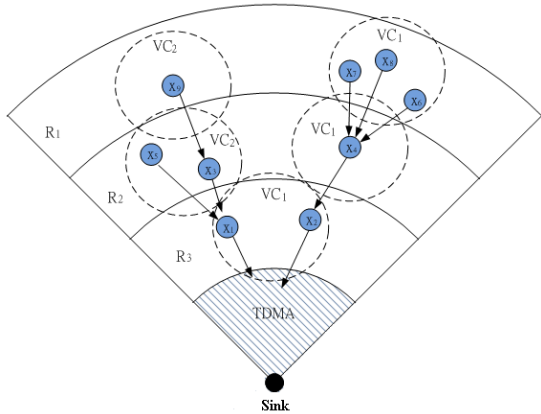
圖四：虛擬叢集區內節點示意圖



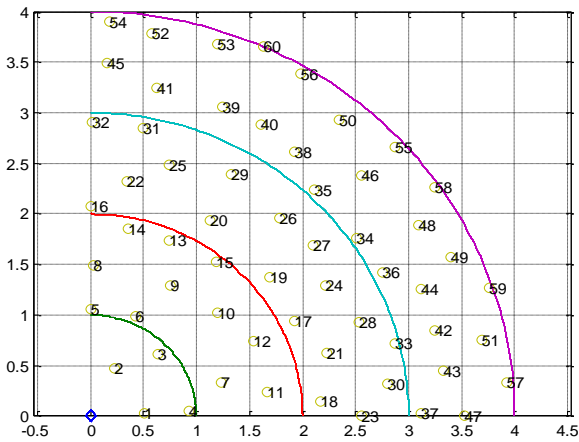
圖五：時槽競爭以及節點隨機選擇時槽示意圖

第三階段為資料傳輸階段(Packet Transmission Phase)，在此階段各感測節點將利用前一階段所建立的傳送順序，進行傳輸/接收動作，由最外層向內將資料傳送至 Sink。本論文假設封包資料欄位可存放 k 筆資料。

本協定為了避免虛擬叢集區間互相干擾，因此將傳輸時間分為單號虛擬叢集區傳輸時間及雙號虛擬叢集區傳輸時間。以下圖六為例，R1 層中擁有兩個虛擬叢集區 VC1、VC2。VC1 使用單號傳輸時間傳送資料，VC2 使用雙號傳輸時間傳送資料，並且利用各虛擬叢集區的傳送順序表，將資料傳送至 R2 層節點後，R1 層節點便可進入休眠。以此類推，R2 層傳送資料至 R3 層，利用此方式直至將資料傳送至 TDMA 層(或指 Sink)。



圖六：網路中虛擬叢集區分佈示意圖



圖七：隨機佈點示意圖

四、系統模擬結果

本論文以 MATLAB 模擬程式，實現 ES-MAC 協定，並針對不同流量及不同週期，進行模擬分析。模擬的參數設定如表 I：

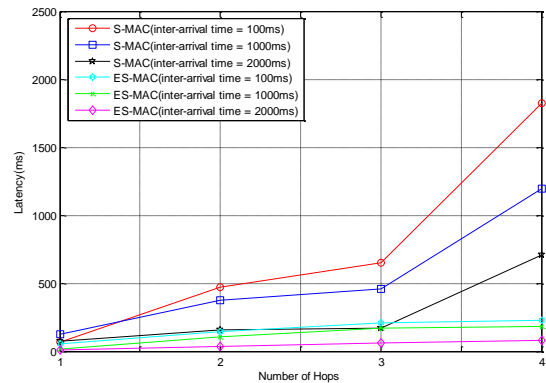
表 I Default Setting of ES-MAC Parameter

Parameter	Default
週期時間(T)	600ms、800ms、1000ms
封包傳送時間(t_p)	1.44 ms (0.00144 s)
封包可裝載的資料量(k)	3
感測器節點數量(n)	60
傳輸速率(T_{rate})	250Kbps
資料平均到達間隔時間(T_{ia})	100ms、1000ms、2000ms

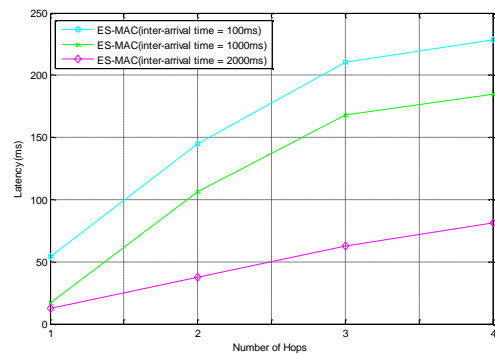
本論文模擬隨機佈建 60 顆感測器節點，並將感測器節點隨機佈建在一個半徑為 4 個單位的漏斗型無線感測器網路內，如圖七所示，其中 \diamond 為 Sink，數字為節點的 ID 編號。

4.1 資料傳輸延遲分析

本節中我們以本論文提出的加強型 ES-MAC 與傳統的 S-MAC，在不同流量及不同週期的情況下，比較資料傳輸的延遲時間。以 T 為 800ms 為例，圖八中的六條曲線分別表示 ES-MAC 和傳統 S-MAC，在 T_{ia} 分別為 100ms、1000ms、2000ms 的情況下，平均封包延遲時間的模擬結果。由圖八可知，ES-MAC 在各流量下都擁有比 S-MAC 低的平均封包延遲時間，因為 S-MAC 在傳送封包前都必須競爭通道，且於多跳網路中，封包可能無法在一個活動週期內傳送至 Sink，必須經過休眠週期後才可繼續傳送，因此造成較大的封包延遲；而 ES-MAC 只需要在週期開始時透過競爭建立傳送的順序，便可順利將封包傳送至 Sink，避免傳送控制封包所造成的延遲時間，所以可以有效的降低封包延遲時間。ES-MAC 在 T=800ms 時，不同流量下之平均傳輸延遲的放大圖，如圖九所示。圖中我們觀察到， $T_{ia}=100ms$ 時，S-MAC 三跳時的封包延遲時間為 652ms，而實施 ES-MAC 三跳時的封包延遲時間為 210ms，明顯的減少許多。且在四跳情況下，只要資料平均到達的間隔時間縮短，S-MAC 平均封包延遲時間均大於一個週期時間(T=800ms)。



圖八：S-MAC 和 ES-MAC 在 T=800ms 下，不同流量的平均傳輸延遲時間

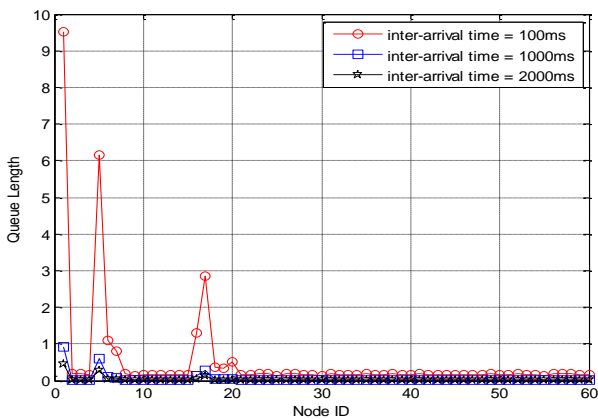


圖九：ES-MAC 在 T=800ms 時，不同流量下的平均傳輸延遲

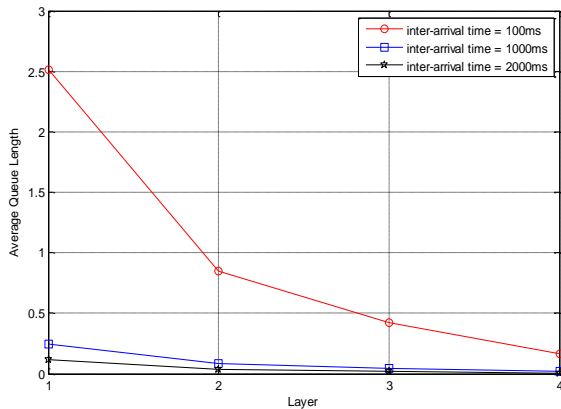
4.2 佇列長度(Queue Length)分析

本節模擬在不同流量及不同週期的情況下，分析各節點中的佇列長度。圖十的三條曲線分別表示週期在 1000ms， T_{ia} 分別為 100ms、1000ms、2000ms 的情況

下，各節點中的平均佇列長度。其中節點 1、5、6、7、16、17 和 20 擁有較大的佇列長度，原因在於這些節點擁有較多的上游節點，導致節點中存放較多的資料。這些擁有較大佇列長度的節點，由於必須傳送大量的資料，因此會花費較多的傳輸時間，導致內層已經接收完資料的節點，必須等待很長的時間才可傳送資料，也因此內層節點將消耗較大的能量。擁有較大佇列長度的節點相較於其他節點將花費較長的傳輸時間，所以也會消耗較大的能量，導致節點較快死亡。圖十一的三條曲線分別表示在 T 為 1000ms， T_{ia} 分別為 100ms、1000ms、2000ms 的情況下，各層中的平均的佇列長度。圖中我們可以發現由外層向內層，當越靠近 Sink 擁有越大的平均佇列長度。原因在於資料由外層向內層不斷的累積，導致內層的節點相較於外層擁有較大佇列長度。



圖十：ES-MAC 在 $T=1000ms$ 時，不同流量下各節點中的佇列長度



圖十一 ES-MAC 在 $T=1000ms$ 時，不同流量下各層的平均佇列長度

五、 結論與未來研究方向

本論文提出了一種加強型的 Enhanced S-MAC (ES-MAC)，並將其使用於無線感測網路較外層中低流量的部分。ES-MAC 分為三個階段：第一階段為建立虛擬叢集區及預設傳輸路徑；第二階段為資料收集與排序；第三階段為資料傳送。虛擬叢集區內的感測節點，利用微時槽競爭窗口的方式決定資料傳輸的順序。而在資料傳送階段則利用已建立的順序傳輸資料，以減少感測節點不必要的過度監聽，並增長節點的休眠時間。

本論文針對封包延遲時間進行模擬分析，比較在不同週期分別為 600ms、800ms、1000ms 時，以及資料平均

到達的間隔時間分別為 100ms、1000ms 和 2000ms 的情況下，傳統 S-MAC 和 ES-MAC 的封包延遲時間。透過模擬的結果可知，ES-MAC 擁有比傳統 S-MAC 較短的封包延遲時間。因為 S-MAC 在傳送封包前都必須競爭通道，且於多跳網路中，封包可能無法在一個動作週期內傳送至 Sink，必須經過睡眠週期後才可繼續傳送，因此造成較大的封包延遲，而 ES-MAC 只需要在週期開始時透過競爭建立傳送的順序，便可順利將封包傳送至 Sink，避免傳送控制封包所造成的延遲時間，所以可以有效的降低封包延遲時間。

ES-MAC 雖然可以有效減少封包延遲時間，但是在傳輸路徑的選擇上，容易造成特定節點擁有較大的佇列長度，導致花費較多的傳輸時間以及消耗較大的能量。因此，本論文未來將針對如何改進資料傳輸的路徑，達到平衡各層節點佇列長度的目的。本論文目前只針對封包延遲時間進行模擬分析，在未來將考慮感測節點能量受限之情況下，針對無線感測網路的生命週期(Life Cycle)進行研究。另外，本論文中並沒有限制節點暫存器的大小和封包延遲限制(Delay Bound)。論文中我們記錄了不同流量及不同週期的情況下，各層中每個節點的平均佇列長度，以及各層的平均佇列長度，可作未來設定各層中節點暫存器大小的參考。

參考文獻

- [1] Q. Jian, Z. H. Gong, P. D. Zhu and C. M. Gui, "Overview of MAC protocols in wireless sensor networks," *Journal of Software*, vol. 19, 2008, pp. 389-403.
- [2] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *IEEE INFOCOM'2002*, vol. 3, New York, USA, pp. 1567-1576.
- [3] W. M. Song, Y.-M. Liu and S. E. Zhang, "Research on SMAC protocol for WSN," in 4th International Conference Wireless Communications, Networking and Mobile Computing, 2008, pp. 1-4.
- [4] S. Mishra and A. Nasipuri, "An adaptive low power reservation based MAC protocol for wireless sensor networks," *IEEE International Conference on Performance Computing and Communications*, 2004, pp. 731-736.
- [5] G.-S. Ahn, E. Miluzzo, S. G. Campbell, Andrew T. Hong, and F. Cuomo, "Funneling-MAC: A Localized, Sink-Oriented MAC for Boosting Fidelity in Sensor Networks," in 4th ACM Conference on Embedded Networked Sensor Systems (SenSys), 2006, Boulder, Colorado, USA, pp. 293-306.
- [6] G.-S. Ahn, E. Miluzzo and Andrew T. Campbell, "Demo Abstract: A Funneling-MAC for High Performance Data Collection in Sensor Networks" in ACM Conference on Embedded Networked Sensor Systems (SenSys), 2006, Boulder, Colorado, USA, November 1-3, pp. 345-346.