

位元競爭式的無線射頻辨識系統預防碰撞演算法

施再繁
朝陽科技大學資工所
助理教授
tfshih@cyut.edu.tw

許文力
朝陽科技大學資工所
研究生
s9627619@cyut.edu.tw

摘要

無線射頻辨識(RFID)系統是一種透過無線射頻(RF)技術的自動辨識系統，當多個標籤(tag)同時傳送它們的 Unique Identification (UID)給讀取器(reader)時，則會發生碰撞問題。在此論文中，我們提出了一個有效預防碰撞的方法，稱之為 bit competed algorithm (BCA)，來改善無線射頻辨識系統辨識上所發生的碰撞問題，進而提昇系統的辨識效能。在我們所提出的方法中，讀取器會將接收到請求傳送標籤的 UID 進行優先權競爭，所利用的技巧為以簡單布林代數的或運算(Boolean OR)進行逐位元(bit-by-bit)優先權競爭。我們進行大量電腦模擬分析，並和已被提出的防碰撞演算法進行比較，實驗的結果顯示我們所提出的方法可有效改善碰撞問題，提昇系統整體的效能，比其他方法有更佳的效能。

關鍵詞：無線射頻辨識系統、讀取器、標籤、碰撞、防止碰撞演算法

1. 前言

無線射頻辨識(RFID)系統是一種透過無線射頻(RF)技術達到自動辨識的系統，主要由兩種元件所構成：讀取器(reader)和標籤(tag)，讀取器與標籤是透過無線射頻(RF)技術經由兩者之間的無線頻道來進行溝通[1-3]，以辨識標籤上的 Unique Identification (UID)，因此，若有兩個或兩個以上的標籤同時傳送自身的 UID 給讀取器便會造成碰撞和辨識錯誤等問題。

RFID 系統中的碰撞問題主要可分為讀取器碰撞(reader Collision)[12-14]及標籤碰撞(tag Collision)[4-11]。當多個標籤同時回覆(Response)訊息至相同的讀取器所發生的碰撞，稱為標籤碰撞；而多個讀取器同時發送請

求(Request)訊息給同一個標籤，則會發生讀取器碰撞。

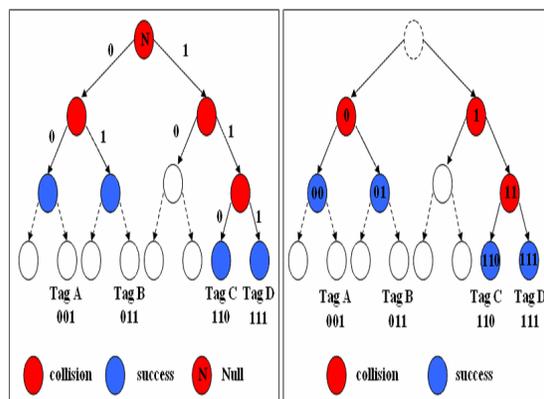
由於讀取器通常具有較佳的計算能力，因而彼此能透過事先的相互溝通與協調來預防碰撞的發生，故讀取器碰撞的問題較容易解決。反之，標籤碰撞因受限於標籤的計算能力，在預防碰撞演算法的設計上難度較高，故大部分的預防碰撞演算法的研究，也都著重於標籤碰撞問題上，但仍有可改善之處，在此論文中，我們將針對此議題進一步深入探討。

本論文之架構規劃如下：第二節將回顧先前的相關研究，第三節將描述我們所提出的預防碰撞演算法；第四節顯示模擬分析之結果，第五部份為我們的結論。

2. 相關研究

為了解決 RFID 系統在辨識上所發生的標籤碰撞問題，有許多的學者提出了各種預防碰撞的演算法，大致上可分為 Tree-based algorithm[4-7]以及 Aloha-based algorithm[8-11]兩種類型，分別說明如下：

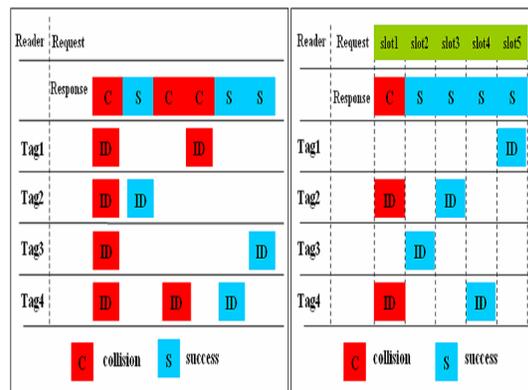
Tree-based algorithm：讀取器發送請求訊息，等待標籤回應自身的 UID，若回應標籤的資料位元相同，則造成碰撞狀態，讀取器以 bit by bit 的方式不斷發送詢問位元進行辨識，直到收到唯一的標籤回應即告完成辨識。在 Tree-based algorithm 中常見的方法有 Binary Tree 和 Query Tree 兩種演算法。Binary Tree 演算法是以一次辨識一個 Bit 為原則，如圖 1(a)，從最高位元至最低位元逐一辨識，每次 Reader 發送一個辨識位元值給所有的標籤，標籤收到這個 Bit 值後就和自身的 UID 相對位元的位元值做比對，如果相同就發出回應給 Reader，如果不同就進入 Sleep Mode，位元比對的動作會反覆持續到辨識出一個標籤後，Reader 就完成了一個回合的查詢，一直重複這樣的查詢，直到所有的標籤都被 Reader 所讀



(a) (b)
圖 1. Tree-based 演算法:(a) Binary tree 演算法 (b) Query tree 演算法

取為止，此方法不會造成飢餓問題存在，然而，標籤長度越長，可能會造成讀取過多的碰撞與延遲狀態，因而延伸出了許多改良之方法，來改善這些問題，如圖 1(b)中的 Query Tree Algorithm。Query Tree 演算法是以一次辨識一組位元串為原則，從最高位元至最低位元開始增加欲辨識位元串的长度，每次 Reader 會 Broadcast 一個位元串，收到該位元串之標籤，會由最高位元檢查自己的 UID 前幾個位元是否與該位元串相同，如果相同的話，標籤就會回應此標籤的 UID 給 Reader，若有超過一個以上的標籤回應給 Reader，就發生碰撞無法成功辨識唯一的標籤，此時 Reader 會再次 Broadcast 下一組位元串繼續進行下一回合的辨識，位元串的长度會隨著辨識次數不斷的增加，直到辨識出唯一的標籤為止。

Aloha-based algorithm：當進入通訊範圍內的標籤有資料需要傳送的時候，它會立即向讀取器傳送自身的 UID，唯有在某一個標籤傳送 UID 的過程中，完全沒有與其他的標籤發生碰撞時，該標籤才能優先被辨識進而讀取資料；反之，若有多個標籤同時傳送 UID 給讀取器時，便會發生碰撞，則標籤必需等待一段時間再重新傳送，直到讀取器收到唯一的標籤回應後即告完成辨識。在 Aloha-based algorithm 中常見的方法可分為 Pure-Aloha 演算法和 Slotted-Aloha 演算法兩種。Pure-Aloha 演算法的原理如下：在通訊範圍內的標籤若有資料需要傳送的時候，它會立即將自身的 UID 傳送給讀取器，如果沒有發生碰撞，則標籤可順利被辨識，反之，若有碰撞發生，則所有標



(a) (b)
圖 2. Aloha-based 演算法: (a) Pure-Aloha 演算法 (b) Slotted-Aloha 演算法

籤均無法被辨識，必需等待一段時間再重傳，直到所有標籤均被辨識完成。因為標籤可隨時傳送 UID 給讀取器的特性，因而發生碰撞的機率相對較大，使得系統的辨識效能較差，如圖 2(a)。此方法可能因標籤不斷發生碰撞無法順利傳送資料而造成飢餓問題，因此有學者提出了改良的方法，如圖 2(b) 中的 Slotted-Aloha 演算法，就是為了改善 Pure-Aloha 演算法中標籤可隨時傳送 UID 給讀取器，導致正在傳送資料的標籤因新標籤突然加入傳送行列，造成碰撞機率升高的特性而設計出來的改良方法。Slotted-Aloha 演算法會將傳輸時間分割成許多時槽(slot)，並限定標籤僅能於每一個時槽的起始點傳送 UID 給讀取器，如此可避免因新標籤突然加入傳送而和正在傳送資料之標籤發生碰撞的情況，可以有效降低標籤重疊傳送資料所造成的碰撞問題，使整個系統的效能提高一倍。讀取器負責控制時槽的個數及指派，讀取器會週期的性的發送一個請求訊息，收到該請求訊息之標籤，會利用讀取器所指派的時槽來傳送資料給讀取器，若同時有多個標籤在相同時槽傳送資料，就發生碰撞，此時標籤需退回並等待讀取器重新隨機指派的時槽來重傳資料，整個程序會反覆進行到辨識完成所有讀取範圍內欲傳輸資料之標籤。

上述兩種類型的演算法，因為辨識程序上的不同，讀取器同時辨識過多標籤可能造成讀取過多閒置節點以及發生過多的碰撞，因無法有效預防碰撞的發生及不斷的延遲會造成系統辨識率的下降，為了改善目前 RFID 辨識之碰撞問題，我們在此提出一個防碰撞演算法

Bit competed algorithm(BCA)，以解決 RFID 系統辨識上之碰撞問題，藉此降低碰撞的次數，以有效改善系統效能。

3. Bit competed algorithm

在本論文中，我們試圖提出一個效能更好的預防碰撞演算法，以有效改善 RFID 辨識時所遇到的標籤碰撞問題。我們所提出的方法稱為 bit competed algorithm(BCA)，主要是藉由讀取器的計算能力，將所有參與傳輸競爭標籤所回傳的 UID，利用簡單的布林代數或運算 (Boolean OR) 進行逐位元 (bit-by-bit) 優先權競爭，以找出最高存取優先權的標籤。Reader 會週期性的傳送 Request Null 訊息給在讀取範圍內的標籤，所有接收到請求訊息且有資料欲進行傳輸的標籤，都會同時傳送自身的 UID 給讀取器以進行辨識，若此時回應標籤數大於 1，就代表有碰撞的情況發生，當碰撞發生時，就必須進行標籤存取優先權競爭，此時讀取器會將所有競爭標籤的 UID，由最高位元開始向最低位元以一個 bit 接一個 bit 的方式進行 Boolean OR 運算，來辨識出最高優先權的 UID，擁有此 UID 的標籤即為所辨識出來具最高存取優先權的標籤。

標籤存取優先權競爭的 Boolean OR 運算方法如下：每次 Reader 會將所有接收到標籤 UID 的相同資料位元取出進行 Boolean OR 運算，如果計算的結果為 1，所有此位元為 0 的標籤會自動放棄競爭，僅留下此位元為 1 的標籤，繼續進行下一個 bit 的競爭；如果計算的結果為 0，則無法分為高下，reader 會繼續取出所有競爭標籤 UID 的下一個 bit，再次進行 Boolean OR 運算，整個 Boolean OR 運算會一個 bit 一個 bit 反覆繼續向低位運算，直到可唯一辨識出一個標籤為止。以 UID 長度 3 個 bits 的標籤為例，若 bit 2 ~ bit 0 表由高至低的位元，則 Reader 進行 Boolean OR 運算的資料位元依序為：bit 2、bit 1、bit 0。若有新的一筆標籤參與競爭，則需要重新進行 Boolean OR 計算。

在圖 3 中，利用一個讀取器和多個標籤進行辨識為例，說明 BCA 演算法的運作方式和 BCA 演算法改善碰撞、減少讀取器和標籤之間詢問回應的次數以及所傳送的辨識位元數之情況，圖 3 中共有 Tag A、Tag B、Tag C、Tag D 四個標籤欲進行資料傳送，在此假設四個標籤的 UID 分別為：Tag A(0010)、Tag

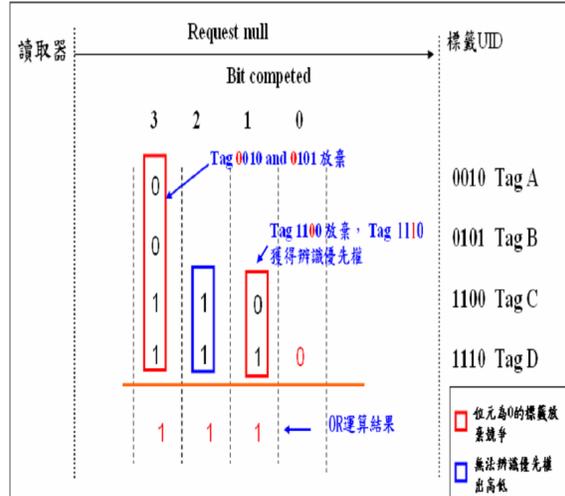


圖 3. BCA 競爭程序

B(0101)、Tag C(1100)、Tag D(1110)，其辨識過程如下：首先 Reader 傳送 Request Null 訊息給在讀取範圍內的標籤，並等待所有接收到請求訊息的標籤回應自身的 UID，因讀取器範圍內同時有四個標籤欲傳送資料，因此會造成碰撞的產生。此時讀取器會利用 BCA 將各個標籤的 UID 由最高位至最低位元以 bit by bit 的方式，進行 Boolean OR 運算來辨識標籤 UID 的優先權，優先權競爭的過程說明如下：

步驟一、讀取器將接收到的各個標籤中的 UID 最高位元 (bit 3) 取出，分別為 0、0、1、1，然後再將此四個 bits 進行 Boolean OR 運算，計算的結果為 1，其中 Tag A(0010) 和 Tag B(0101) 的最高位元 (bit 3) 為 0，因此 Tag A 和 Tag B 在這一輪會認輸而放棄繼續競爭的資格，等待下次再接收到請求訊息時，才會重新加入競爭，而 Tag C(1100) 和 Tag D(1110) 的最高位元 (bit 3) 為 1，因此 Tag C 和 Tag D，可繼續進行次高位元的競爭。

步驟二、接下來欲進行競爭的標籤只剩 Tag C(1100) 和 Tag D(1110)，在這輪會取出這兩個標籤的次高位元 (bit 2)，分別為 1、1，做 Boolean OR 運算，計算的結果為 1，然而因為兩個標籤的次高位元均為 1，無法分出高低，故將會繼續進行下一位元 (bit 1) 的競爭。

步驟三、取出 Tag C(1100) 和 Tag D(1110) 的第三高位元 (bit 1)，分別為 0 和 1，再做 Boolean OR 運算，得到計算的值为 1，因 Tag C 的第三高位元 (bit 1) 為 0，因此 Tag C 在此輪會認輸而放棄繼續競爭，而 Tag D 的第三高位

元(bit 1)為 1，成為最後的贏家，讀取器因而成功的辨識出唯一標籤 Tag D (1110)，若能在較高位元分出優先權高低，則所需進行的 Boolean OR 運算次數也越少，可越快辨識出要存取的標籤。

步驟四、再以相同的方式將未完成辨識之標籤，重覆進行 Boolean OR 計算，直到完成所有標籤 UID 的辨識。當有 n 個標籤同時傳送自身的 UID 時，讀取器和標籤所花費的詢問回應次數就等於 n-1 次，能有效改善讀取器和標籤之間的請求及回應次數。

4. 模擬實驗

在此節中，我們將以電腦模擬的方式，驗證我們所提出之預防碰撞演算法(BCA)能有效減少碰撞的發生，進而提高 RFID 系統的辨識率。然而，不同的預防碰撞演算法有不同的特性，不同的預防碰撞演算法會對某一類的排程比較有利。當選擇在某一種狀況要使用那一種方法時，就必須考慮各種演算法的不同特性、適用場合與時機。因此為了公平性，我們將以相同類型 bit by bit 的 Tree-based 演算法中的 Binary Tree Algorithm (BTA)和 Query Tree Algorithm (QTA)兩種演算法來做為模擬分析和效能比較的對象。

在模擬實驗中，將針對主要影響辨識效能的標籤數量加以進行模擬分析，將測量不同的標籤數量對碰撞次數、讀取器請求次數與標籤回應次數(Request-Response)以及傳送的請求、回應的總位元數所造成的影響，並與先前的相關研究 Binary Tree Algorithm (BTA)和 Query Tree Algorithm (QTA)做比較，評估我們所提出之 BCA 演算法對系統整體辨識率的改善情況。在模擬程序中，每回模擬的總標籤數量將以亂數產生，標籤數量範圍為 100~1000 個標籤為限，所辨識的標籤 UID 位元長度設為 12 bits。模擬的結果分別顯示如下：圖 4(a)顯示標籤數量與碰撞次數的關係，與 Binary Tree Algorithm (BTA)和 Query Tree Algorithm (QTA)做比較，我們的方法(BCA)能有效的減少碰撞發生的次數，主要是因 BCA 演算法在每次查詢中均可經由 Boolean OR 運算，唯一辨識出一個標籤進行資料傳輸，而在 BTA 和 QTA 的辨識標籤存取過程是以一問一答的方式向下查詢，因而不保證 Reader 在每次查詢中均可唯一辨識出一個標籤，反而常常需要經過多次碰撞之後才能唯一辨識出一個標籤，如

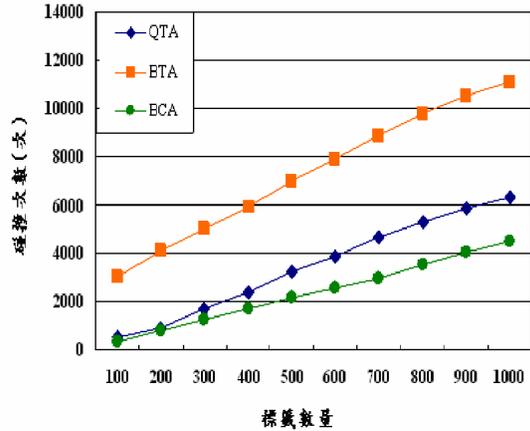


圖 4.(a)標籤總數與碰撞次數關係

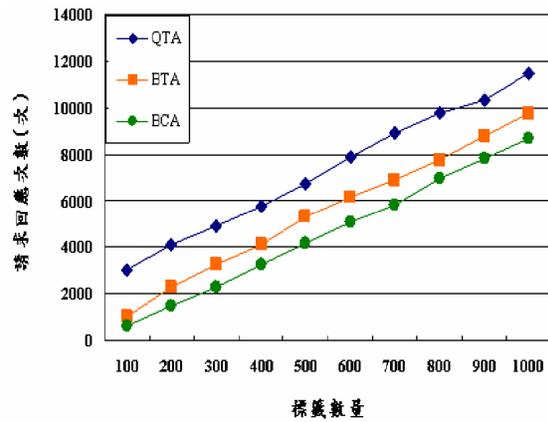


圖 4.(b)標籤總數與請求回應次數關係

先前我們在相關研究中所提及的例子一樣，當讀取器要詢問讀取範圍內的標籤 UID 時，讀取器發送請求訊息，等待標籤回應自身的 UID，若回應標籤的 UID 位元相同，則造成碰撞狀態，讀取器以 bit by bit 的方式不斷向下發送詢問位元進行辨識，直到收到唯一的標籤回應後即告完成辨識。故 BCA 的碰撞次數會遠低於 BTA 和 QTA，若有 k 個標籤欲進行優先權競爭，BCA 可以保證在經過 k-1 次碰撞即可將此 k 個標籤辨識完成，即碰撞次數主要受參與辨識優先權競爭的標籤個數所影響。

圖 4(b)顯示增加標籤數量時對讀取器請求次數與標籤回應次數(Request-Response)所造成的影響，由圖 4(a)的實驗數據可看出我們的方法能有效的降低碰撞的次數，讀取器與標籤之間的溝通次數也因碰撞次數降低而減少，因

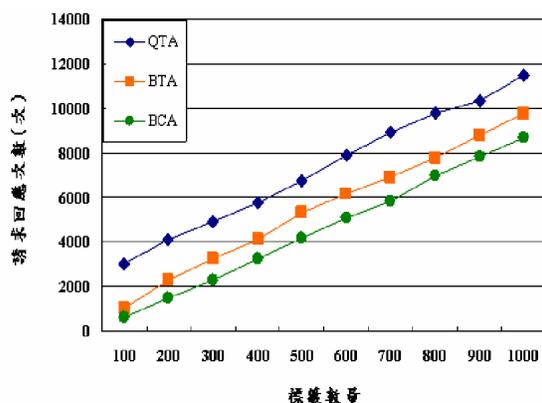


圖 4.(c)標籤總數與請求-回應總位元(bits)次數關係

由 BCA 演算法的辨識存取方法與先前學者所提及的 BTA 和 QTA 一問一答的辨識存取方式不同，主要 BCA 演算法藉由 reader 的計算能力一次將所有參與的標籤 UID 接收進行簡單的布林代數或運算(Boolean OR)進行逐位元(bit-by-bit)優先權競爭，以找出最高存取優先權的標籤，故採用 BCA 演算法時，讀取器請求次數與標籤回應次數遠小於 BTA 和 QTA，請求次數與標籤回應次數受改變標籤數量所造成的影響亦較 BTA 和 QTA 演算法為小。

由圖 4(c)可看出各種預防碰撞方法，在進行辨識時所傳送的請求-回應總位元數會隨標籤數量的增加而遞增，由圖中可看我們的方法所傳送的請求-回應總位元數低於其他方法，即我們的方法因辨識所產生的額外負擔(overhead)較低。此乃因我們的方法能有效減少讀取器請求次數與標籤回應次數(如圖 4(b)所示)，因為 BCA 演算法將所有參與的標籤 UID 接收來加以進行逐位元(bit-by-bit)的布林代數或運算(Boolean OR)優先權競爭，越快找出最高存取優先權的標籤則傳送的請求-回應總位元數也會隨著請求與回應的次數下降而減少。

5. 結論

在本論文中，我們提出一個效能更好的預防碰撞演算法，以有效改善 RFID 辨識時所遇到的標籤碰撞問題，進而提昇無線射頻辨識系統的效能。我們所提出的方法稱之為 bit competed algorithm(BCA)，主要是藉由讀取器的計算能力，將所有參與傳輸競爭標籤所回傳的 UID，利用簡單的布林代數或運算(Boolean

OR)進行逐位元(bit-by-bit)優先權競爭，每次發生碰撞時，最少會找出一個獲得優先辨識的標籤。我們所提出的方法可有效降低碰撞次數和辨識的額外負擔(overhead)，同時，碰撞次數和 overhead 亦較不受改變標籤 UID 位元長度影響，我們利用電腦模擬來比較我們的演算法和其他演算法的效能，實驗的結果顯示我們的演算法比 Binary Tree Algorithm 和 Query Tree Algorithm 有更好的效能。

6. 參考文獻

- [1] D. Shih, P.L. Sun, D.C.Yen and S. M.Huang, "Taxonomy and Survey of RFID Anti-collision protocols," **Computer and communications**, vol.29, pp. 2150-2166, 2006.
- [2] K. Finkenzeller, (2003), **RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification**, 2nd ed. New York: Wiley.
- [3] Tang, Zhijun and He, Yigang, "Research of Multi-access and Anti-collision Protocols in RFID Systems," **Anti-counterfeiting, Security, Identification**, IEEE International Workshop on pp.377 - 380, April 2007.
- [4] J. H. Chio, D. Lee, H. Lee, "Query tree-based reservation for efficient RFID tag anti-collision," **IEEE Communications Letters**, vol. 10, no. 4, pp. 236-238, April 2006.
- [5] Jihoon Myung and Wonjun Lee, "Adaptive Splitting Protocols for RFID Tag Collision Arbitration," **MobiHoc'06**, Florence, Italy, pp. 202-213, 2006.
- [6] Sung Hyun Kim and PooGyeon Park, "An Efficient Tree-Based Tag Anti-Collision Protocol for RFID Systems," **IEEE Communications Letters**, vol.11, pp.449 - 451, May 2007.
- [7] T. Wang, "Enhanced binary search with cut-through operation for anti-collision in RFID systems," **IEEE Communications Letter**, vol. 10, no. 4, pp.236-238, April 2006.
- [8] C. S. Kim, K. L. Park, H. C. Kim, and S. D. Kim, "An efficient stochastic anti-collision algorithm using bit-slot mechanism," **PDPTA04**, 2004.
- [9] J. Cha and J. Kim, "Novel Anti-collision Algorithms for Fast Object Identification in

- RFID System,” in **Proc. of Parallel and Distributed System**, vol. 2, pp. 63-67, Jul. 2005.
- [10] J. Park, M. Chung, and T.-J. Lee, “Identification of RFID Tags in Framed-Slotted ALOHA with Robust Estimation and Binary Selection,” **IEEE Communications Letters**, vol. 11, no.5, pp. 452-454, 2007.
- [11] S. Lee, S. Joo, and C. Lee, “An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification,” in **Proc. MobiQuitous2005**, pp. 166-172.
- [12] J. Kim, W.Lee, J. Yu, J. Myung, E. Kim, C.Lee, “Effect of Localized Optimal Clustering for Reader Anti-Collision in RFID Networks:Fairness Aspects to the Readers, ” **Proceedings 14th International Conference on Computer Communications and Networks**, pp. 497-502, 2005.
- [13] Junius Ho, Daniel W. Engels, and Sanjay E. Sarma, “HiQ: A Hierarchical Q-Learning Algorithm to Solve the Reader Collision Problem,” **SAINTW’06**, 4 pp, 2006.
- [14] Shailesh M.Birari “Mitigating the Reader Collision Problem in RFID Networks with Mobile Readers”Dissertation for Master of Technology, **Indian Institute of Technology, India**, 2005.