

在無線射頻系統中使用群組碰撞標籤的反碰撞辨識方法

An Anti-Collision Method for Using Collision Tags Grouping in RFID Systems

李銘城

龍華科技大學資訊管理系

leemc@mail.lhu.edu.tw

李之中

中華大學資訊管理系

leecc@chu.edu.tw

摘要—在無線射頻辨識系統中，當讀取器使用 FSA 方法進行標籤辨識時，標籤碰撞對辨識效率有重要的影響。因此，學界提出了許多的反碰撞方法來解決讀取器在辨識標籤過程所發生的碰撞問題。在這些反碰撞方法中，最常被學界所討論的方法為 Dynamic Framed-Slotted ALOHA (DFSA) 方法。然而，DFSA 方法在辨識過程中一個回合的辨識過程中，並未充分使用辨識標籤個數與框架中時槽個數相同時，辨識效率隨著辨識標籤個數減少而提高的特性，因此 DFSA 方法的辨識效能仍有增進的空間。本論文針對 DFSA 方法的未考慮之處，提出一個新的方法，稱為 Group Framed-Slotted ALOHA (GFSA)。DFSA 方法將當回合所有碰撞標籤在下一回合視為同一辨識群組，GFSA 方法則是將當回合同一時槽的碰撞標籤，在下一回合視為單一碰撞群組，以提升 DFSA 的辨識效率。我們對 GFSA 方法以數學分析與系統模擬的方式進行了效能評估。結果顯示，GFSA 方法的辨識效率為 42.7% 優於 DFSA 方法 36.7% 的辨識效率。

關鍵詞—無線射頻辨識，反碰撞方法

Abstract - In RFID systems, tag collision critically influences the throughput of tag identification when a reader uses an ALOHA-based method to identify tags. Researchers, therefore, proposed anti-collision methods to overcome tag collision. To conquer tag collision, the dynamic framed-slotted ALOHA (DFSA) is the most

common anti-collision method. DFSA has an important characteristic: in a cycle, the throughput increases as the identified tag quantity decreases when the tag quantity equals the frame size. However, most DFSA-based anti-collision methods neglect this characteristic, and therefore these methods can be further improved in terms of throughput. We propose a new anti-collision method which uses this characteristic to increase the throughput of tag identification. Our proposed method is called Group Framed-Slotted ALOHA (GFSA). In GFSA the collision tags on the current cycle don't appear as a whole on the next cycle. Instead, the collision tags on the current cycle are distinguished into several groups on the next cycle based on the slot of the tag collision event on the current cycle. We also derive the throughput using a probability model. Simulation results show that the throughput of GFSA, 42.7%, outperforms that of DFSA, 36.7%.

Index Terms - RFID, anti-collision method, Group Framed-Slotted ALOHA

一、概述

無線射頻辨識 (Radio Frequency Identification, 簡稱 RFID)[2] 是新一代用於辨

識資料的技術，有別於傳統條碼 (barcode) 接觸式的辨識方式，RFID 的優點在其可提供非接觸式且同時可以辨識多筆資料的技術，既簡便又具效率，所以廣泛的應用於各式資料處理，舉凡車輛自動識別管理、道路自動收費管理、航空行李包的自動管理、身份證 IC 卡、健保 IC 卡、生產線產品加工、動物識別晶片、物流業和倉儲自動管理等[1]。

RFID 系統主要包含二大部分：無線射頻辨識讀取器 (Reader)、無線射頻辨識標籤 (Tags) [2]。讀取器負責辨識其範圍內的所有標籤資料，並在其內含的資料庫系統進行相關的資料處理，通常讀取器採用非接觸的方式，同時處理多個標籤；我們稱讀取器的運作範圍為辨識空間(interrogation zone)。標籤內含一個唯一的身分辨識碼(EPC)[3]，在辨識過程中，讀取器發出辨識要求 (Request) 的訊號，辨識空間中的所有標籤在接收到訊號之後，回傳身分辨識碼，交由讀取器辨識[2]。

隨著 RFID 技術的發展，許多學者針對無線射頻辨識系統的問題進行研究[4]。其中，讀取器在標籤辨識過程中所發生的標籤碰撞 (collision) 直接影響讀取器辨識標籤的效率。因此，學界提出了許多反碰撞 (anti-collision) 方法 [7,10,11,13-15]來克服標籤碰撞所帶來的負面影響。在這些反碰撞方法中，最常被學界所討論的是簡單且具效率的 Framed-Slotted ALOHA 方法[5]，簡稱為 FSA 方法。FSA 方法由讀取器預定一段長度固定的框架時間，同時將此框架時間切割成若干個時間長度相同的時槽 (time-slots)，這些時槽的集合稱為框架 (frame)。每個時槽需預留足夠的時間供讀取器和標籤進行一次標籤辨識。讀取器在一個框架時間內進行標籤辨識的過程稱為回合。為了便於文章的說明，我們使用辨識效率 (throughput) 作為衡量績效的指標，辨識效率為當讀取器辨識完所有標籤時，每個時槽所能辨識的平均標

籤個數。

FSA 方法具有兩個特性。第一個特性為使用 FSA 方法辨識 n 個待辨識標籤時，如果框架中時槽個數同樣為 n ，則該回合的辨識效率將是辨識 n 個標籤的最大值[6]。FSA 方法的第二個特性為在滿足第一個特性的條件下進行標籤辨識時，需辨識的標籤個數愈少辨識效率將愈高 [7]。舉例而言，使用 2 個時槽辨識 2 個標籤的標籤辨識效率高於使用 10 個時槽辨識 10 個標籤的標籤辨識效率。然而，FSA 方法並未充分運用上述的兩個特性。針對第一個特性，學者 Schoute[8] 提出 Dynamic Framed-Slotted ALOHA 方法，簡稱為 DFSA 方法。DFSA 方法具有動態調整每回合框架時槽個數與預測每回合待辨識標籤個數的能力，因此在每回合的辨識中，辨識效率有機會達到最佳值。儘管 DFSA 方法運用了 FSA 方法的第一個特性，但是 DFSA 方法並未積極運用 FSA 方法的第二個特性。DFSA 方法將當回合的所有碰撞標籤在下一回合中仍視為同一辨識群組進行標籤辨識。本論文修正 DFSA 方法，將當回合同一時槽的碰撞標籤在下一回合的辨識過程中，區分為若干個待辨識標籤數較小的碰撞群組，以充分利用 FSA 方法的第二個特性。冀望藉此有效的提升 DFSA 的辨識效率，我們將此一演算法稱之為 Group Framed-slotted ALOHA 方法，簡稱為 GFSA 方法。

本文的第二節介紹現存的反碰撞辨識方法，第三節說明 GFSA 方法，第四節為 GFSA 方法的效能分析，第五節為以系統模擬來印證 GFSA 方法的效能，第六節為本論文的相關研究，第七節為結論與未來工作。

二、現存的反碰撞辨識方法

(一) FSA 方法

FSA 方法[5]由讀取器預定一段固定時間進行辨識空間中的標籤辨識。FSA 方法同時將此

段時間切割成若干個固定大小的時間單位，稱為時槽 (time-slots)，此段固定時間中的時槽集合，稱之為框架 (frame)，讀取器使用一個框架進行標籤辨識的過程稱為回合。在框架中的每個時槽都預留足夠的時間供讀取器和標籤進行一次標籤辨識和回應。FSA 方法的標籤辨識過程如下：在每一回合的開始，讀取器發出辨識要求訊號，並告知此回合中框架的時槽個數 s_n (slot number)，框架中的時槽編號自 0 依序編至 s_n-1 。當標籤收到要求訊號後，使用均一分佈 $U(0, s_n-1)$ 隨機產生一個正整數作為時槽編號，並在該編號之時槽送出身份資料，供讀取器進行標籤辨識。如果該時槽中僅有該標籤進行標籤辨識，則該標籤成功完成辨識，讀取器送出辨識成功的訊號，自此該標籤不再參與標籤辨識；如果時槽中同時有其它標籤進行標籤辨識，則該標籤與其他標籤發生碰撞，讀取器回傳標籤辨識失敗訊息，碰撞標籤必須參加下一回合的標籤辨識。標籤辨識過程將逐個回合的進行直到所有標籤完成辨識為止。

(二) DFSA 方法

在使用 FSA 方法進行標籤辨識時，每回合待辨識標籤的個數將隨著逐次回合辨識的進行而減少。因此使用固定框架大小的 FSA 方法，無法隨著待辨識標籤個數的減少，同步調整每回合框架的大小，使得每回合的辨識效率都達到該回合的最佳值。因此我們需要一個具有動態調整每回合框架時槽個數與預測每回合待辨識標籤個數的能力的 FSA 方法，這個方法即為 DFSA 方法[8]。DFSA 方法在進行一個回合的標籤辨識之前，先行預估該回合中待辨識標籤個數，並將則該回合框架中的時槽個數設為該預估數大小。DFSA 方法的辨識過程與 FSA 方法的辨識過程幾乎完全相同，唯一不同之處為，在 DFSA 方法中當回合的標籤辨識完成時，此時讀取器統計發生標籤碰撞的時槽個數 c ，下一回合的框架中的時槽個數即為未完成辨識標

籤的預估個數—最接近 $2.39*c$ 的整數。

三、GFSA 方法

在本節中我們將依序介紹 GFSA 方法的框架結構，讀取器與標籤之間的傳遞訊息，以及讀取器與標籤的存取協定(access protocol)。

(一) 框架結構

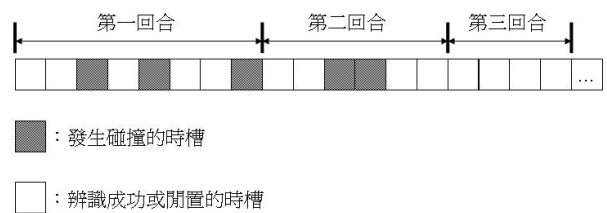


圖 1、框架結構

GFSA 方法的框架結構如圖一所示，每個回合的框架由多個時槽所組成，但是每個回合框架中的時槽個數並不固定。第一回合框架中的時槽個數為一個預設值，第二回合的框架大小則為第一回合碰撞時槽個數的兩倍，第三回合的框架大小則為第二回合碰撞時槽個數的兩倍，於此類推，直至標籤完成辨識為止。圖一中，第一回合至第三回合的框架大小依次為 8, 6, 4。第一回合預設的框架大小預設為 8，在第一回合的標籤辨識中，一共有 3 個時槽發生標籤碰撞，所以第二回合框架大小為 $2*3$ 等於 6。在第二回合的標籤辨識中，則是 2 個時槽發生標籤碰撞，所以第三回合框架大小為 $2*2$ 等於 4。

(二) 讀取器與標籤之間的訊息傳遞

在 GFSA 方法中，讀取器和標籤二者之間傳遞的訊息的格式為 (TYPE, VALUE)，其中 TYPE 為訊息種類，VALUE 為相關資料。GFSA 方法中所使用的訊息共有四個，其中前三個為讀取器以廣播方式傳遞給辨識空間內所有標籤的訊息，第四個訊息為標籤以多存擷取方式傳遞至讀取器。由讀取器傳遞給標籤的訊息為：

1. (Request, RandomRange)：告知所有標籤本回合辨識開始，在第一回合之 RandomRange 值為本回合標籤在選擇傳送標籤編號的時槽數，所使用的隨機整數 i 將服從均一分佈 $U(0, \text{RandomRange} - 1)$ 。
2. (Ack, Silent)：如果在某個時槽，只有單一個標籤發出信號，則讀取器成功接收標籤信號，身分辨識即告完成，讀取器發出此訊息告知標籤辨識成功，往後所有回合不須在發出身分信號。
3. (Ack, Offset)：如果有多個標籤在某個時槽發出辨識信號，則產生碰撞情況，無法完成辨識工作，讀取器發出此訊息給這組所有標籤進行內部 TagOffset 設定¹，此訊息告知同組標籤在下一回的 Offset 和 Offset + 1 二個時槽，隨機挑選作為資料回傳的時間點。
由標籤傳遞給讀取器的訊息為：
4. (Send, TagId)：標籤在挑選的時槽中傳送標籤內的辨識資料。

(三) 讀取器與標籤之間的存取協定

GFSA 方法的存取協定如圖 2 所示。在辨識工作的第一回合，首先根據標籤數量，輸入一個時槽大小整數 s_n ，並依此設定第一回合標籤隨機挑選的變數 RandomRange，每一標籤對內部 Offset 歸零，記錄當回合碰撞時槽發生個數的變數 collision 的值設為 1。

當標籤辨識開始時，首先由讀取器以廣播方式對所有標籤發出辨識通知 (Request, RandomRange)，其中 RandomRange 值為 s_n ，辨識空間中的標籤在接收到訊息 (Request, s_n) 後，立即根據 s_n 值產生一個服從均一分佈 (Uniform Distribution) $U(0, s_n - 1)$ 之整數隨機變數 I 。在第一回合中，此標籤將在本回合中的第 I 個時槽使用訊息 (Send, TagId) 傳送

```

1. input sn
2. For each Tag k
3.   TagOffset of Tag k = 0;
4. endFor
5. collision = 1;
6. RandomRange = sn;
7. do while (collision > 0)
8.   Reader send (Request, RandomRange);
9.   For each Tag k generate a random number I and
       I ~ U(0, RandomRange-1)
10.    TagOffset of Tag k = TagOffset of Tag k + I
11.  endFor
12.  collision = 0
13.  For each slot j
14.    For each tag k where TagOffset of tag k == slot j
15.      Tag k sends message (Send, TagId)
16.    endFor
17.    If Reader receive message (Send, TagId) //成功
18.      Reader send message (Ack, Silent)
19.    else //碰撞
20.      Offset = collision * 2
21.      collision = collision + 1
22.      Reader send (Ack, Offset)
23.      For each tag k which send (Send, TagId)
24.        TagOffset of Tag k = Offset
25.      endFor
26.    endIf
27.  endFor
28.  sn = collision * 2
29.  RandomRange = 2
30. endDo

```

圖 2、GFSA 方法的存取協定

該標籤的編號。讀取器在預設時間之內，若能收到任一標籤的 (Send, TagId) 訊息，表示讀取器已經辨識出標籤編號為 TagId 的標籤，此時讀取器傳送訊息 (Ack, Silent) 給標籤 TagId；若多個標籤於該時槽傳遞訊息 (Send, TagId)，則讀取器將收到無法辨識的訊息，這意味著此一時槽為碰撞時槽，此時讀取器根據目前的 collision 設定 offset 值，再將訊息 (Ack, Offset) 以廣播方式傳送給所有參與此一時槽辨識的標籤。同組碰撞標籤收到 (Ack, Offset) 訊息的標籤，立即將訊息 (Ack, Offset) 中的 Offset 存入標籤中的變數 TagOffset 中。等待下一回合辨識的開始。

在第二回合之後，在每個回合開始時，讀取器以廣播方式對所有標籤發出辨識通知 (Request, RandomRange)，其中參數

¹標籤內部有一變數 TagOffset 記錄隨機時槽的起始位置，第一回合的初值設定為 0，第二回合以後則根據讀取器傳回之 collision 值而設定。

RandomRange 的值為 2。辨識空間中的標籤在接收到訊息(Request, 2) 後，根據 RandomRange 值產生一個服從均一分佈(Uniform Distribution) $U(0, 1)$ 之隨機整數 I ，並取出標籤中變數 TagOffset 的值，本回合標籤將在第 TagOffset+ I 個時槽送出訊息(Send, TagId) 進行標籤辨識。上述演算法的結束條件，是在某個回合之中，都沒有任何標籤產生碰撞。

(四) 範例說明

接下來，我們舉一個例子說明 GFSA 方法的存取協定。我們的例子如圖 3 所示，在第一回合的辨識中，在第 1 個時槽和第 3 個時槽分別發生碰撞現象，因此，第 1 個時槽中的 (t_1, t_2) 二個標籤得到碰撞的訊息，系統告知位移量為 0，下一回合的群組競爭位置在第 0, 1 二個時槽，所以 t_1, t_2 二個標籤將隨機挑選其中之一送出信號；第 3 個時槽的 (t_3, t_4, t_5) 三標籤碰撞，讀取器回應這組標籤的位移量為 2，其競爭位置在下一回合的第 2, 3 個時槽，則 t_3, t_4, t_5 將隨機挑選 2, 3 時槽其中之一作為發送信號的時間點。

假設第二回合中， t_1, t_2 分別挑選第 0, 1 時槽， t_3 挑選第 2 時槽， t_4, t_5 挑選第 3 時槽。前三個時槽辨識成功，讀取器回傳 Silent 訊號，第 3 時槽發生碰撞，讀取器再度回傳 Offset， t_4, t_5 繼續進入第三回合。

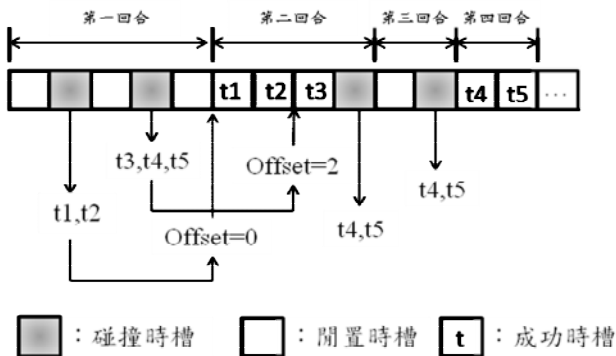


圖 3. 辨識碰撞之群組處理

四、GFSA 方法分析

在本節中，我們將針對 GFSA 方法進行分析，我們使用累積期望值(iterated expectation)計算辨識 n 個標籤所需要的所有時槽個數。我們首先令標籤數與時槽數相等並以 n 表示。令 N 為一個隨機變數表示於某時槽中進行傳送辨識資料的標籤個數。則 $P(N=t)$ 可表示某時槽中有 t 個標籤選擇於該時槽傳送辨識訊息的機率。 N 的機率密度函數可表示為

$$P(N=t) = \binom{n}{t} \left(\frac{1}{n}\right)^t \left(1 - \frac{1}{n}\right)^{n-t}, \quad t=0, 1, \dots, n \quad (1)$$

依據上式，任一時槽發生碰撞的機率可以下式表示為

$$P(N \geq 2) = 1 - P(N=0) - P(N=1) \quad (2)$$

因此，如果 n 是個夠大的整數，在任意一個時槽中，在已知該時槽發生碰撞的條件下，該碰撞是由 t 個標籤所造成的條件機率為

$$P(N=t|N \geq 2) = \frac{P(N=t)}{1 - P(N=0) - P(N=1)} \approx \frac{1}{t!(e-2)} \quad (3)$$

在 GFSA 方法中，如果當回合某個時槽由 t 個標籤發生碰撞，則下一回合將提供兩個時槽供該 t 個標籤進行下一回合的辨識。令 $f(t)$ 表示讀取器完成 t 個碰撞標籤辨識工作所需的時槽總數，則 $f(t)$ 可表示為

$$f(t) = \begin{cases} 0, & t=0,1 \\ \sum_{i=0}^{t-1} C_i^t (f(i) + f(t-i)) + 2, & t=2, \dots, n \end{cases} \quad (4)$$

使用 GFSA 方法進行 n 個標籤辨識，在第一回合所使用的平均時槽個數為 n ，而自第二回合開始至所有標籤完成辨識所需要的時槽數量可使用累積期望值表示為 $\sum_{t=2}^n f(t)P(N=t|N \geq 2)$ ，因此使用 GFSA 方法進行 n 個標籤辨識所需的

時槽總數 n_s 為

$$n_s = n + \sum_{t=2}^n f(t)P(N=t|N \geq 2) \quad (5)$$

式(3)中等號左邊的第二項，經過運算後約略等於 $1.337942 * n$ 。因此因此使用 GFSA 方法進行 n 個標籤辨識所需的時槽總數 n_s 為可進一步簡略表示為

$$n_s = 2.337942 * n \quad (6)$$

而 GFSA 方法進行 n 個標籤辨識的辨識效率 S_{GFSA} 可使用下式表示為

$$S_{GFSA} = \frac{n}{n_s} \quad (7)$$

將式(6)代入式(7)得到

$$S_{GFSA} = \frac{n}{2.337942n} \approx 0.4277 \quad (8)$$

五、系統模擬

在本節中，我們使用系統模擬的方式對本論文所提出的 GFSA 方法與 DFSA 方法進行效能評估，評估指標為讀取器完成標籤辨識時的辨識效率。辨識效率的定義為辨識成功的時槽數除以辨識完成所使用的總時槽數。實驗中所使用的模擬參數，標籤總數為 100 至 2000，間隔 100。在 DFSA 方法中，使用 $2.39 * c$ 來估計辨識空間尚未完成的標籤個數，同時也是下一回合的框架個數。實驗的模擬程式在 Windows XP 工作平台上，使用 JAVA 語言進行開發。

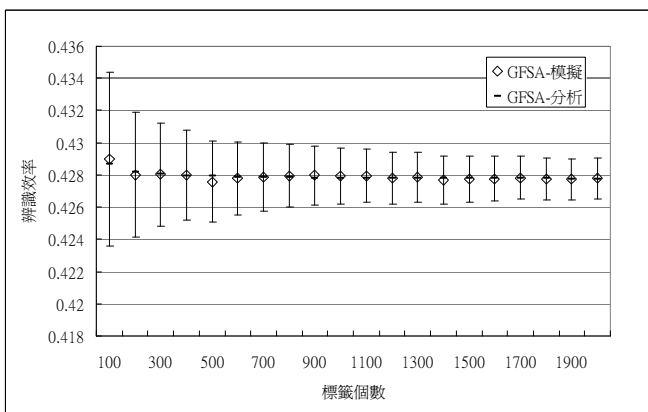


圖 4、GFSA 方法的效能分析

在進行 GFSA 與 DFSA 兩種方法的效能評估之前，我們先驗證第四節中 GFSA 方法的分析結果。我們的驗證方式為在不同的標籤個數條件下分別根據式(7)進行 GFSA 方法的數值實驗與使用隨機方式進行 GFSA 方法的系統模擬實驗。實驗結果如圖 4 所示。在系統模擬實驗中，我們針對不同標籤個數分別進行 1000 次的模擬實驗，取得信心水準為 99% 的辨識效率區間，並以誤差線表示於圖 4。在圖 4 中我們發現所有 GFSA 方法在不同標籤個數下的數值實驗結果都落在系統模擬實驗 99% 信心水準下的辨識效率區間中。這個結果可以說明我們對 GFSA 方法的效能分析的正确性。

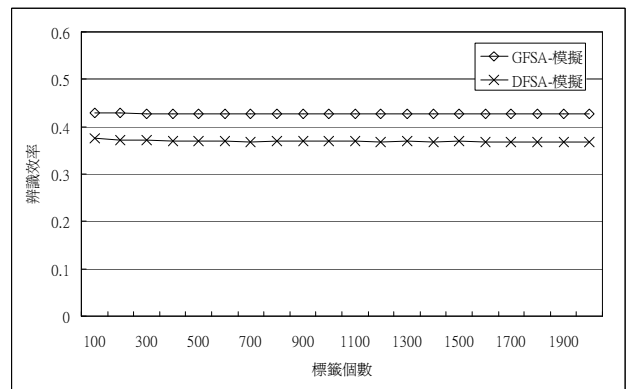


圖 5、標籤數量對辨識效率的影響

接著我們進行 GFSA 與 DFSA 兩種方法的效能評估，我們分別針對 GFSA 與 DFSA 兩種方法針對讀取器在辨識空間中辨識不同個數標籤，完成標籤辨識時的辨識效率進行系統模擬實驗。其實驗結果如圖 5 所示。其中 X 軸為辨識空間中的標籤個數，其範圍由 100 至 2000，每次實驗標籤個數增加 100，Y 軸為完成標籤辨識時的辨識效率。在圖 5 中顯示 GFSA 方法的辨識效率在不同的標籤個數下都維持在 0.427 附近，大於 DFSA 方法的 0.367 附近。

GFSA 方法優於 DFSA 方法的主要因為在辨識過程中，GFSA 方法將同一時槽中發生碰撞

的標籤視為同一群組，並在下一回合提供二個時槽供此群組內的標籤進行標籤辨識，而非如 DFSA 方法將當回合的所有碰撞標籤視為同一群組。GFSA 方法，將同一時槽發生碰撞的標籤視為一個群組，實際上就是將辨識標籤分為一些標籤個數較少的群組，在分別針對這些群組進行標籤辨識，以利用辨識標籤個數與框架中時槽個數相同時，辨識效率與被辨識標籤個數成反比的特性，增加辨識效率。而 GFSA 方法優於 DFSA 方法的另一個原因為 GFSA 方法對同一群組的標籤在下一回合提供二個時槽供此群組內的標籤進行標籤辨識。為了說明使用 2 個時槽的原因，我們以辨識空間中有 1000 個標籤待辨識為例說明使用 2 個時槽的原因。我們針對 DFSA 方法使用 1000 個時槽辨識 1000 個標籤進行系統模擬實驗，在重覆 1000 次實驗之後，其第一回合完成後的結果如表 1 所示。再表一中，第一個欄位為時槽中發生碰撞的標籤個數，第二個欄位則為第一回合中發生第一欄位個標籤碰撞時槽的平均個數，第三個欄位則為當標籤發生碰撞時，發生第一欄位個標籤碰撞時槽的條件機率，第四個欄位則為發生第一欄位個標籤碰撞時，使用兩個時槽完成標籤辨識所需的平均時槽個數。

從表 1 中我們觀察到，在標籤個數和時槽個數均為 1000 的情形下，在已知標籤發生碰撞的條件下，同一時槽碰撞標籤數為 2 所佔的比例為 68.7% 為最高。而在已知標籤發生碰撞的條件下，其他同一時槽碰撞標籤個數則與發生的比例成反比。另一個觀察則為使用 2 個時槽辨識 2 個標籤，其辨識效率可達 50% 為最高，而在已知標籤發生碰撞的條件下，其他同一時槽碰撞標籤個數則與辨識效率成正比。

如果分別以 $r(i)$ 和 $p(i)$ 表示表 1 中 i 個標籤碰撞的比率和次數， $s(i)$ 表示 GFSA 演算法處理 i 標籤所需的時槽總數，第二回合以後，GFSA 方法的辨識效率 S_{GFSA} 為

$$S_{GFSA} = \sum_{i=2}^{10} \frac{f(i) * i}{s(i)} = 0.47973。$$

而 DFSA 方法在第二回合以後的辨識效率 S_{DFSA} 為

$$S_{DFSA} = \frac{1}{e} = 0.36788$$

在第一回合辨識效率相同的條件下，顯然 GFSA 方法的辨識效率 S_{GFSA} 大於 DFSA 方法的辨識效率 S_{DFSA} 。

表 1 使用 GFSA 進行標籤辨識第一回合完成後的結果(時槽個數與標籤個數均為 1000)

時槽中碰撞標籤個數- i	平均時槽個數- $p(i)$	碰撞發生時的比率- $r(i)$	平均時槽個數 $s(i)$
0	367.363	0	0
1	368.688	0	0
2	183.802	0.687	3.99022
3	61.238	0.229	6.67758
4	18.18	0.068	10.14854
5	3.718	0.014	13.93768
6	0.483	0.0018	17.75314
7	0.059	0.00022	21.50326
8	0.008	0.00003	25.28784
9	0.001	0.000004	29.15328
10	0	0	32.95008

六、相關研究

與 RFID 標籤辨識最有關的研究為在網路領域中有關多重擷取(Multi-Access)方面的研究 [13]。在封包廣播網路的早期研究中，多使用 ALOHA、時槽 ALOHA (Slotted ALOHA) 或框架 ALOHA (Framed-Slotted ALOHA，簡稱 FSA)，來處理多重擷取所發生的碰撞。但是 FSA 在框架大小與需傳送的封包個數相同時，此時效率可達最佳值的特性。因此為了充分發揮 FSA 的

此一特性，學者提出每回合框架大小可隨著實際需傳送封包數量而進行調整的方法的動態框架 ALOHA (Dynamic Framed-Slotted ALOHA，簡稱 DFSA)[6]。在[6]中，學者 Schoute 對 DFSA 進行了效能分析。在這個分析中，作者以數學分析證明在 FSA 中，如果框架大小與需傳送的封包個數相同時，此時 FSA 效率為最佳值。作者也同時提出使用 $2.39 * c$ 來動態調整下一回合的框架大小，其中 c 為當回合發生碰撞的時槽個數，使得 DFSA 可以得到最佳的傳輸效率。

在 RFID 系統中進行標籤辨識也可使用以 ALOHA 為基礎的方法進行。在[2]中，作者對使用 ALOHA 方法來進行標籤辨識做了一個基礎的說明。在後續的研究中，學者使用 DFSA 或是 DFSA 的變化形式作為 RFID 系統中的標籤辨識方法[7,10,11,13-15]。在使用 DFSA 方法進行標籤辨識時需要進行多個回合，而使得 DFSA 達到最佳效率的方法就是將每一回合的框架大小設為辨識空間中待辨識標籤個數。但是在標籤辨識過程中，讀取器無法得知辨識空間中的標籤個數，因此 DFSA 要在經過多個回合之後，才有機會逐漸接近最佳效能。因此許多研究提出標籤個數的估計方法，希望能在短時間內使得辨識標籤個數與框架大小相近儘快使得 DFSA 達到最佳效能。在[13]中，學者 Vogt 在飛利浦半導體公司「I-code」系統中提出一個利用動態框架 ALOHA 的標籤辨識方法。在 Vogt 的方法中，作者也提出利用讀取結果向量與期望值向量之間距離進行標籤估計。在其他研究中[10,15,16]，學者 Cha 與 Kim [14] 與學者 Kodilam 與 Nandagopal [10] 分別提出標籤個數估計方法，將使用這些標籤個數估計方法獲得之標籤估計個數設為動態框架 ALOHA 初始回合的框架大小，然後使用動態框架 ALOHA 進行標籤辨識。學者 Cha 與 Kim [14]所提出的估計方法是利用框架大小和衝突率，來預測未辨識的標籤個數。而學者 Kodilam 與

Nandagopal [10] 根據辨識過程中的時槽閒置率、成功辨識率和衝突率，推算標籤數和時槽數的負載比 (load factor)，藉以更快速預測標籤數。在[15]中，學者 Chen 提出一個使用最大後驗機率決策(maximum a posteriori probability decision)進行標籤估計的方法。學者 Chen 的估計方法所估計出的標籤個數與實際標籤個數差異在 4% 以下。學者 Chen 的估計方法搭配動態框架 ALOHA 其辨識效率可以達到框架式 ALOHA 通訊協定的理論最佳值。

在業界的相關技術中，EPCglobal 提出 EPC Class 1 Gen 2 標準[3]，在這個標準中使用變化形式的框架時槽 ALOHA 進行標籤辨識。在這個方法中，使用 Q 演算法來決定下一週期的框架大小。Q 演算法維護一個變數 Q_{fp} ，在標籤辨識過程中每當時槽成功辨識使標籤， Q_{fp} 不變，新增常數 C ($0.1 < C < 0.5$)。當時槽為空置， Q_{fp} 減去常數 C 。當時槽發生碰撞， Q_{fp} 增加常數 C 。如果 Q_{fp} 值的改變造成 $\text{round}(Q_{fp})$ 的值發生改變，則立即進行新一回合的辨識，新一回合的框架大小為 $2^{\text{round}(Q_{fp})}$ 。學者 Lee 等人[16]修正了 Q 演算法中的參數調整方式，使得 Q 演算法更精確估計下一週期的框架大小，而達到最佳辨識效率，這個方法被稱為 Q^+ 。

七、結論與未來展望

本論文針對 Framed-Slotted ALOHA 演算法提出一個有效的改善方案，稱為 Group Framed-Slotted ALOHA (GFSA)，GFSA 將當回合(cycle)辨識過程中在同一個時槽發生碰撞的標籤視為一個標籤群組，在下一回合中，GFSA 將分配二個時槽 (slots)，專供辨識此一標籤群組中的標籤。因此在辨識過程中，讀取器將紀錄發生碰撞的次數，每當發生碰撞時，由讀取器送出一個代表位移的 Offset 信號，其值為單一回合中碰撞次數的二倍，告知同群組標籤下一回合的隨機位置是在 Offset 開始的二個時槽

(time-slots) 其中之一；如果再度發生碰撞，則重複上述群組動作，直到所有標籤辨識完畢為止。藉由碰撞標籤的群組化，縮小每一個碰撞標籤的隨機範圍，可以有效的提升辨識率。我們對 GFSA 方法以數學分析與系統模擬的方式進行了效能評估。結果顯示，GFSA 方法的辨識效率為 42.7% 優於 DFSA 方法 36.7% 的辨識效率。

從以上的結果可以看出，如果標籤個數可以事先估計，則藉由對發生衝突的標籤，進行群組的處理，不論原本的標籤個數多寡，DFSA 演算法均可得到明顯的改善。但是，極有可能對於辨識範圍內的標籤個數，無法事前取得，或者個數隨著時間改變，也可能大幅度改變，則如何快速的動態預測辨識範圍內的標籤總數，是未來研究的重點。

參考文獻

- [1] 曾煜棋，潘孟鉉，林致宇編著，無線區域及個人網路，知城圖書出版社，2006。
- [2] K. Finkenzeller, *RFID Handbook*, 2nd ed. New York: Wiley, 2003
- [3] Class 1 Generation 2 UHF Air Interface Protocol Standard EPCglobal, 2005.
- [4] Dong-Her Shih, Po-Ling Sun, David C. Yen, and Shi-Ming Huang, "Taxonomy and survey of RFID anti-collision protocols," *Computer Communications*, Vol. 29, Issue 11, pp. 2150-2166, 2006
- [5] J. E. Wieselthier, A. Ephremides, and L. A. Michaels, "An exact analysis and performance evaluation of framed ALOHA with capture," *IEEE Trans. Commun.*, vol. 37, no. 2, pp. 125-137, Feb. 1989
- [6] W. Szpankowski, "Packet switching in multiple radio channels: Analysis and stability of a random access system," *Comput. Network*, vol. 7, pp. 17-26, 1983
- [7] Jeong Geun Kim, "A Divide-and-Conquer Technique for Throughput Enhancement of RFID Anti-collision Protocol," *IEEE Communications Letters*, Vol. 12, No. 6, pp. 474-476, 2008.
- [8] F. C. Schoute, "Dynamic frame length ALOHA," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 565-568, Apr. 1983.
- [9] Robert M. Gray and Robert M. Gray, and Lee D. Davisson *An Introduction to Statistical Signal Processing*, Cambridge University Press, 2004.
- [10] Murali Kodialam, Thyaga Nandagopal, "Fast and Reliable Estimation Schemes in RFID System," *MobiCom*, 2006
- [11] Quaikubg Tong, Xuecheng Zou, Dongsheng Liu, Yifei Dai, "Modeling the Anti-collision Process of RFID System by Markov Chain," *Wireless Communications, Networking and Mobile Computing*, pp.2054-2057, Sep 2007.
- [12] Andrew S. Tanenbaum, *Computer Networks*, Fourth Edition, Prentice-Hall,
- [13] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Int. Conf. Pervasive Computing*, 2002, pp. 98-113.
- [14] J.-R. Cha and J.-H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in *Proc. Int. Conf. Parallel and Distributed Systems Computing*, 2005, vol. 2, pp. 63-67.
- [15] Wen-Tzu Chen, "An Accurate Tag Estimate Method for Improving the Performance of an RFID Anticollision Algorithm Based on Dynamic Frame Length ALOHA," *IEEE Transactions on Automation Science and Engineering*, Vol. 6, No. 1, pp.9-15, 2009.
- [16] D. Lee, K. Kim, and W. Lee, "Q+-Algorithm: An Enhanced RFID Tag Collision Arbitration Algorithm," *Lecture Notes in Computer Science, Ubiquitous Intelligence and computing*, 4611, pp23-32, 2007