

無線區域網路上快速自適碰撞倒退演算法

韋宇航¹ 林政翰² 謝錫堃³ 黃文祥^{4,*}

¹ 國立成功大學 電腦與通訊工程所

^{2,3} 國立成功大學 電機工程學系

⁴ 國立高雄應用科技大學 電機工程系

Q36021282@mail.ncku.edu.tw¹

jhlin5@ee.ncku.edu.tw²

shieh@ee.ncku.edu.tw³

wshwang@mail.ee.kuas.edu.tw^{4,*}

摘要

近年來無線網路發展快速，目前被廣泛使用的則是 IEEE802.11 無線網路存取技術，裝置可以利用其預設的分散式協調功能(DCF)機制去競爭通道，透過競爭視窗(Contention Window)決定開始傳送資料前所需等待的時間，若發生碰撞則將競爭視窗調大，進而降低再次發生碰撞的機率，相反地，如果資料傳送成功則將競爭視窗降低，進而縮減倒數時間，使裝置能用更高的頻率去嘗試傳送資料，所以決定競爭視窗的成長或減少便成了影響無線網路傳輸效能的一個重要關鍵。在先前的研究，我們找出了較佳效能的碰撞率區間並提出基於碰撞率之倒退演算法(Collision Rate-based Backoff Algorithm, CRBA)，在本論文裡延續這個區間的研究，提出了快速自適碰撞倒退演算法(Rapidly Adaptive Collision Backoff Algorithm, RACB)，與先前的 CRBA 相比也有同等的優點，但卻能使碰撞率更快進入較佳的區間，而且更能適應變化快速的網路環境。

關鍵詞：IEEE 802.11、DCF、LILD、EIED

Abstract

In recent years, wireless networking is developed rapidly, IEEE 802.11 standard is the wireless access technologies that was adopted widely, wireless device can access wireless channels by the Distributed Coordination Function (DCF), the default medium access mechanism in IEEE 802.11, DCF decides the channel arbitration time from 0 to Contention Window size (CW size) randomly, whenever collision or data was transmitted successfully, it will adjust CW size by the backoff algorithm. The default backoff algorithm is Binary Exponential Backoff (BEB), when collision occurs, BEB will double the CW size to decrease the probability of collision, in opposite, while data was transmitted successfully, BEB will reset the CW size to the initial value. In this paper, we continues previous research, and proposed a new backoff algorithm, Rapidly Adaptive Collision Backoff

Algorithm (RACB). Same as CRBA, the backoff algorithm proposed in previous research, adjusting the CW size based on the transmission collision rate, the numerical results show that the RACB and CRBA have the same level system throughput、collision rate and channel utilization, but RACB making the collision rate into the best collision rate interval faster than CRBA, and using less time to transmit packets, comparing with any related backoff schemes, including BEB, Exponential Increase Exponential Decrease (EIED), Linear Increase Linear Decrease (LILD) and CRBA.

Keywords: IEEE 802.11、DCF、LILD、EIED

1. 前言

近年來無線網路發展快速，目前被廣泛使用的則是 IEEE802.11[1]無線網路存取技術，而 IEEE 802.11 的標準定義了實體層(Physical layer, PHY)與媒介存取控制層(Medium Access Control layer, MAC)協定，其中媒介存取控制層包含了兩種協調功能，分別為分散式協調功能(Distributed Coordination Function, DCF)與中樞協調功能(Point Coordination Function, PCF)，在 IEEE 802.11 的標準下 DCF 為預設使用的機制，而 PCF 則只是可選用的，所以本論文中就只針對 DCF 的機制去作探討。

在 DCF 機制裡是使用 CSMA/CA(Carrier Sense Multiple Access/Collision Avoidance)去做無線網路的存取，無線節點在傳送前要先探測目前無線通道的狀況，如果偵測到無線通道為閒置(Idle)時，就可以開始傳送資料；如果偵測到無線通道為忙碌(Busy)時，無線節點就會從競爭視窗大小(Contention Window Size)隨機選出一個等待時間，然後繼續偵測無線通道的狀態，直到為閒置時，才會開始遞減等待時間，一旦遞減到零時，就可以開始傳送資料，若同時有兩個以上的無線節點選到相同的等待時間，且同時於無線通道上傳送資料，就會發生碰撞(Collision)；在 DCF 機制中會使用倒退演算法去調整競爭視窗的大小，借此來降低碰撞的機率，所

以倒退演算法對無線網路的效能具有非常大的影響力。

原始的 DCF 機制是使用 BEB(Binary Exponential Backoff)倒退演算法，發生碰撞時競爭視窗大小會以指數的方式成長，每發生一次碰撞競爭視窗大小便成長一倍，直到變成最大值(Maximum Contention Window, CW_{max})；但一旦成功傳送資料，競爭視窗大小會直接調回最小值(Minimum Contention Window, CW_{min})。可是在無線節點數量增加的情況下，無線節點需要經過多次碰撞使競爭視窗大小到一定的值，封包才能成功傳送，但競爭視窗大小又會退回最小值，便又會開始持續的碰撞，使網路的傳輸效能低下。

從這裡可以看出倒退演算法其實扮演著非常重要的角色，左右著到底會不會發生碰撞，並且連帶地影響整體網路的傳輸效能，在之後的相關研究像是 [2] 中作者提出 EIED(Exponential Increase Exponential Decrease)倒退演算法，發生碰撞時也是如同 BEB 一般，將自身碰撞視窗的大小成長一倍，直到變成最大值(CW_{max})，可是為了避免像 BEB 一樣當資料傳送成功時，競爭視窗降得太快使得碰撞率上升，EIED 使用與碰撞發生時類似的處理方式，將競爭視窗大小每成功傳送便減為一半，而不像 BEB 一樣直接降為最小值(CW_{min})，這樣的作法相較 BEB 能有效地使碰撞率下降，使網路傳輸效能更加提升；但實際上競爭視窗大小都還是以指數的方式成長或減少，這樣的變動幅度還是相當的大，於是在 [3] 的作者提出了 LILD(Linear Increase Linear Decrease)倒退演算法。如同其名，當傳送資料發生碰撞或是傳送資料成功時，是以線性的方式去變化競爭視窗的大小，從概念上來說線性的變動幅度自然會比指數的變化來的小很多，發生碰撞時 LILD 會將競爭視窗大小增加 CW_{min} (或 1)，反之亦然，當傳送成功時則減少 CW_{min} (或 1)，這樣的作法競爭視窗大小變動並不大，相對來說非常穩定，所以 LILD 其實是非常適合在變化小的網路環境，也因為這樣的特性，在有許多節點競爭的網路環境下吞吐量會有較佳的表現，另一方面來說 EIED 因為變化較大就會比較適合在低流量的網路環境。但不論如何上述的 3 個機制都有一個問題就是當競爭通道的無線節點增加時，其吞吐量難免都會有明顯的下降；所以在我們先前的研究 [4] 找出了最佳的碰撞區間，使吞吐量可以趨近最佳化，並提出基於碰撞率之倒退演算法 (Collision Rate-based Backoff Algorithm, CRBA)，依碰撞率決定是否讓競爭視窗大小成長或減少，使碰撞率坐落在最佳的碰撞區間，不管節點數的多寡，皆能有非常好的網路傳輸效率。

綜觀以上 4 種機制，CRBA 似乎已經達到最佳的傳輸效能，但因為 CRBA 並不是每次發生碰撞或資料傳送成功就調整競爭視窗大小，而是經過幾筆傳送後視碰撞情況後再調整，所以其實在網路變動快速的環境底下是較不適合的，且因為這樣的機制

特性，即便已經知道了最佳的碰撞率區間，其碰撞率收斂至最佳區間其實需要經過不少次傳送才能達到，甚至在傳輸少量封包時所需的時間可能會比 BEB 或 EIED 來的長，所以我們這次延續之前最佳的碰撞率區間研究，並提出了快速自適碰撞倒退演算法 (Rapidly Adaptive Collision Backoff Algorithm, RACB)，同樣基於碰撞率調整競爭視窗大小，同時結合了 EIED 與 LILD 的優點，相較 CRBA 可以更快進入最佳的碰撞區間，同時保有相當的吞吐量，且不論傳輸少量或大量封包所需的時間也相對來的少。

本論文的章節安排如下，第二章介紹相關研究 BEB、EIED、LILD 演算法及 CRBA，第三章為快速自適碰撞倒退演算法(RACB)，第四章討論實驗與數據分析，最後是本篇論文的結論。

2. 文獻探討

2.1 IEEE802.11 DCF

在 IEEE 802.11 無線網路存取技術底下是使用分散式協調功能(DCF)的機制，如圖 1 所示，並透過 CSMA/CA 做無線網路通道的偵測與存取，當偵測到通道為閒置時，經過 DIFS(Distributed Inter-Frame Space)長度的時間後就可以開始傳送資料；若發現通道為忙碌狀態時，無線節點就會從競爭視窗大小的範圍內隨機挑出一個值，做為所需等待的時間，並且持續地偵測通道的狀態，直到通道變為閒置時再經過 DIFS 長度的時間後，便開始遞減所需等待時間，當所需等待時間倒數至零時，就可以開始嘗試傳送資料，若是同時有兩個以上的節點同時倒數完畢，並在通道上一起傳送資料，就會發生碰撞，反過來說，若只有一個節點倒數至 0，且通道沒有雜訊等干擾，就會傳送成功，之後節點便會視傳送的結果是成功或碰撞，透過倒退演算法(Backoff Algorithm)去改變競爭視窗大小，再根據改變後的競爭視窗大小去隨機挑出新的所需等待時間，並重新競爭通道。

2.2 BEB 演算法

如圖 2 所示，BEB 演算法最一開始時競爭視窗的大小為 32，也就是最小值(CW_{min})，當碰撞發生時就將值成長 2 倍，每碰撞一次便成長一次，直到變為 1024，也就是最大值時(CW_{max})便停止成長，不論成長到多大，只要一傳送成功便將競爭視窗的大小設為最小值(CW_{min})；這樣的機制在許多節點競爭通道的情況時會造成網路傳輸效能低下，原因是需要經過無數次碰撞後才有機會傳送成功，一旦成功後競爭視窗大小又調回最小值(CW_{min})，又會持續發生碰撞，因碰撞率的攀升造成網路效能不佳。

2.3 EIED 倒退演算法

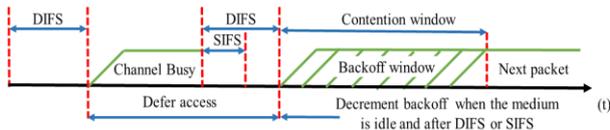


圖 1 DCF 機制

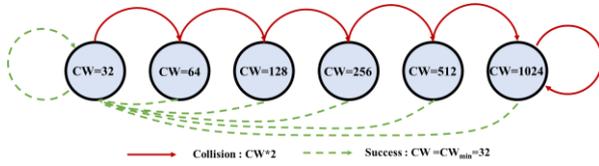


圖 2 BEB 演算法

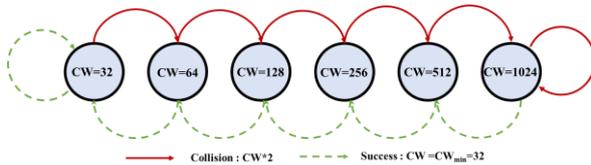


圖 3 EIED 演算法

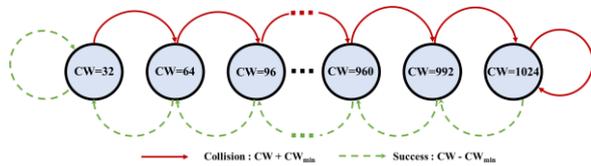


圖 4 LILD 演算法

表 1 CRBA 演算法

```

Start
  if(successful transmission)
    count += 1
  if(successful transmission)
    count += 1
    collision += 1

  if (count == 5)
    if (collision < 1)
      CW /= 2
      count = 0
      collision = 0
    else if (collision >= 1)
      CW *= 2
      count = 0
      collision = 0
  
```

如圖 3 所示，EIED 倒退演算法基本與 BEB 演算法相似，最一開始時競爭視窗的大小也同為最小值(CW_{min})，發生碰撞時也同樣會將競爭視窗大小成長 1 倍，且成長至最大值時(CW_{max})便停止成長，比較不同的是當傳送成功時不會直接將競爭視窗大小設為最小值(CW_{min})，而是每成功傳送一次便將競爭視窗大小縮減成一半，直到變為最小值(CW_{min})才保持不變，這樣的設計就能改善 BEB 演算法在大量節點競爭通道時遇到的高碰撞率問題；但實際上以指數的方式成長或減少競爭視窗大小，其變動幅度還是非常的大，尤其是在高競爭的網路環境下。

2.4 LILD 倒退演算法

如圖 4 所示，LILD 倒退演算法作法與 EIED 倒退演算法相似，初始時競爭視窗的大小同為最小值(CW_{min})，成長的上限也同為最大值(CW_{max})，但發生碰撞或傳送成功時，是以線性的方式去增加或減少競爭視窗大小，每碰撞一次便將競爭視窗大小增加最小值(CW_{min})，而傳送成功則減少最小值(CW_{min})，與 BEB 演算法和 EIED 倒退演算法相比，其競爭視窗大幅度非常小的，在穩定的網路環境及高競爭的網路環境是相較適合使用的倒退演算法，不會因競爭視窗大小的劇烈變化導致突然的高碰撞率或拉長傳送所需等待時間；但即便較適合高競爭的網路環境，其吞吐量還是不可避免的會隨著競爭節點數的增加而遞減。

2.5 CRBA

在先前的研究裡我們針對不同節點數固定其競爭視窗大小去對 DCF 做實驗，發現不論在多少節點數下面，倘若沒有任何雜訊或通道不穩定的因素，碰撞率介於 0.1~0.2 時皆能擁有 80% 以上的通道使用率，整體系統吞吐量趨近最大值，並基於這樣的結果提出了 CRBA，演算法如表 1，當發生碰撞或傳送成功時不會直接調整競爭視窗大小，而是以嘗試傳送資料 5 次作為週期，經過一次週期統計的碰撞率來決定競爭視窗大小的成長或減少，其成長與減少的方式與 EIED 相同，當碰撞率大於等於 0.2 時將競爭視窗大小成長 1 倍，直到變為最大值(CW_{max})，而碰撞率小於 0.2 時則將競爭視窗大小縮減一半，直到變為最小值(CW_{min})，利用這樣的方式使碰撞率介於最佳的碰撞區間，以達到趨近最佳的網路傳輸效能；但由於每 5 次才調整一次競爭視窗大小，在最初傳送資料時會不斷的發生碰撞，若只需傳輸少量資料其所需花費的時間甚至會比 BEB、EIED、LILD 倒退演算法還來的長，而且也因為需 5 次嘗試傳輸資料才改變競爭視窗大小，在變動大的網路環境下來說還是相當不適合的。

3. 快速自適碰撞倒退演算法(RACB)

3.1 碰撞率指標

在這裡我們延續先前的研究，當碰撞率高於 0.2 時碰撞率過高會使得吞吐量下降，若碰撞率低於 0.1 時又因等待傳送時間拉得更長，使得吞吐量有開始下降的趨勢，所以最佳的碰撞區間是介於 0.1 到 0.2 這個範圍，為了解決先前 CRBA 的缺點，也就是不適用於變化快速的網路環境，及一開始碰撞率過高的問題；基本的想法也是以碰撞率作為指標去調整競爭視窗的大小，其中我們的碰撞率指標會參考過去的碰撞情形，如方程式 1 所示，其中 CR_T 為當下碰撞率指標，可以是單次也可以是多次的統計碰撞率，w_T 則為當下碰撞率指標(CR_T)的權重

值， CR_{T-1} 則是前一次的碰撞率指標的值。

$$CR_T = (1 - w_T)CR_{T-1} + w_T CR_T \dots(1)$$

3.2 快速自適碰撞倒退演算法(RACB)

同樣基於最佳碰撞區間去調整競爭視窗大小，使用碰撞率指標(CR_T)來決定該如何對競爭視窗大小作調整，目的也是為了讓網路的碰撞率坐落在 0.1~0.2 的區間，為了使演算法可以適用在快速變動的網路環境，這邊碰撞率指標的統計週期為 1 次傳送，即發生碰撞碰撞率指標(CR_T)等於 1，反之傳送成功即為 0，表 2 介紹的為本論文提出的快速自適碰撞倒退演算法(RACB)，同時採用了 EIED 與 LILD 的概念去對競爭視窗做調整，由於最佳碰撞區間為 0.1~0.2，其中間值約莫等於 0.15，所以希望碰撞率可以接近 0.15，理論上競爭視窗大小與碰撞率有反比的性質，競爭視窗變為 2 倍，其碰撞率大約為原先的 2 分之 1，所以我們以 0.15 的 2 倍與 2 分之 1 作為使用 EIED 的方式調整競爭視窗大小的邊界，也就是 0.3 與 0.075，當碰撞率指標大於等於 0.3 時便將競爭視窗大小變為 2 倍，而碰撞率指標小於 0.075 時則將競爭視窗大小調整為原先的一半，若碰撞率指標趨近 0.15 時我們希望競爭視窗大小只需做小幅度的微調，使之維持在 0.15 上下，所以當碰撞率指標介於 0.075~0.3 之間時就使用 LILD 的方式去調整，以 0.15 做為增加或減少競爭視窗大小的分水嶺，小於 0.3 又大於等於 0.15 時將競爭視窗大小增加最小值(CW_{min})，相反的，若大於等於 0.075 又小於 0.15 時便將競爭視窗大小文減少最小值(CW_{min})，如此使得碰撞率可以落於最佳碰撞區間達到趨近最佳吞吐量，而且快速的調整競爭視窗大小可以避免 CRBA 前期的高碰撞率，且更能適應變化快速的網路環境。

表 2 CRBA 演算法

```

Start
  if(successful transmission)
    collision = 0
  else
    collision = 1
   $CR_T = (1 - w_T) CR_T + (w_T) collision$ 

  if ( $CR_T \geq 0.3$ )
     $CW *= 2$ 
  else if( $CR_T < 0.075$ )
     $CW /= 2$ 
  else if( $CR_T \geq 0.15$ )
     $CW += CW_{min}$ 
  else
     $CW -= CW_{min}$ 

```

4. 實驗結果與分析

表 3 IEEE802.11 參數值

CW_{min}	32
CW_{max}	1024
Packet payload	8184 bits
MAC header	272 bit
PHY header	128 bit
ACK	112bit+PHY header
Channel bit rate	1 Mbit/s
Slot time	50 μ s
DIFS	128 μ s
SIFS	28 μ s
Propagation Delay	1 μ s

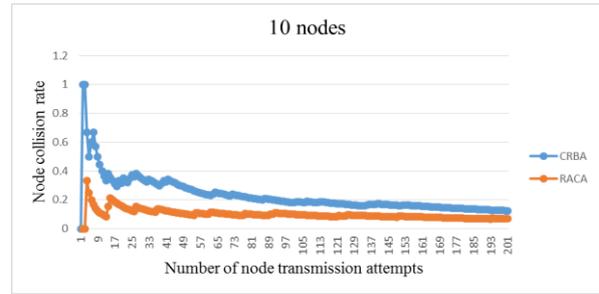


圖 5 碰撞率收斂情形(10 節點)

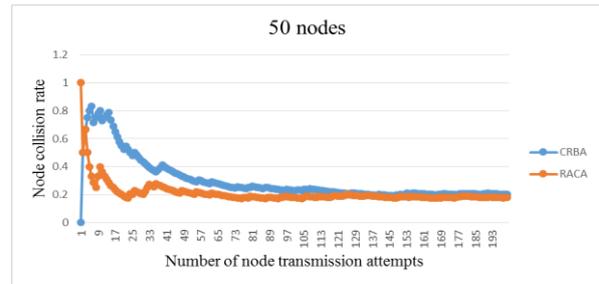


圖 6 碰撞率收斂情形(50 節點)

本章節主要對於這篇論文提出的快速自適碰撞倒退演算法(RACB)與文獻探討提到的倒退演算法 BEB、EIED、LILD、CRBA 進行效能分析比較，先比較 RACB 與 CRBA 碰撞率收斂情形，再對前述演算法做吞吐量、碰撞機率及通道空閒時間以及傳輸定量封包平均所需時間之比較。表 3 為 IEEE802.11 無線區域網路環境模擬之參數值，設定無線節點數量從 10 到 50，而 RACB 中當下碰撞率指標(CR_T)之權重值(w_T)設定為 0.2，並假設模擬網路環境為飽和狀態(Saturation state)且沒有隱藏節點的問題下進行模擬。

碰撞率收斂情形我們取較少節點數時與多節點數時的情況來比較，圖 5、圖 6 分別為 RACB 與先前研究提出的 CRBA 在節點數為 10 與 50 時之比較，不難看出 RACB 碰撞率收斂至 0.1 到 0.2 之間所需要嘗試的次數是遠少於 CRBA 的，不論在節點數只有 10 或是高達 50 時皆是如此。

圖 7 為快速自適碰撞倒退演算法(RACB)與 CRBA、BEB、EIED、LILD 的吞吐量效能曲線圖，可以看出當節點數量越多，所有演算法吞吐量都會無可避免地降低，其中 BEB 倒退演算法最為明顯，

節點數越多，下降的幅度越大，而 EIED 因為發生碰撞不直接將競爭視窗大小調為最小值(CW_{min})，避免了高頻率的碰撞，所以吞吐量是優於 BEB 的，雖說圖 7 中 LILD 表現較差，但吞吐量的下降率卻隨著節點數增加而趨緩，如果在更多節點數的環境下其實效能是會優於 EIED 的；由於 CRBA 與 RACB 是基於最佳碰撞率區間去調整競爭視窗大小，不論節點數的多寡其吞吐量起碼都可達到 80%，也就是趨近於 DCF 分析之最大效能。

圖 8 為碰撞機率分析曲線圖，可以看出不論是哪個倒退演算法，在節點數增加的情況下碰撞率都會因此增加，只是成長幅度的差異，其中 BEB 演算法碰撞率為最高的，而 LILD 和 EIED 次之，而 RACB 與 CRBA 的碰撞率則落於 0.1 至 0.2 之間，與兩個演算法的設計理念相符，也因如此才能擁有接近 DCF 分析的最佳吞吐量。

圖 9 為通道空閒時間曲線圖，為了達到 DCF 分析結果的最大值，也就是使碰撞率盡量落於 0.1~0.2 的區間，RACB 與 CRBA 的競爭視窗大小會成長至較大的值，通道空閒時間會比其他演算法多了 1 倍至 2 倍，但吞吐量卻可以接近 DCF 分析的最佳狀況，代表著其實通道利用率是較佳的，而不是看似通道空閒時間較短，但是實際上浪費在不斷地碰撞。

圖 10 與圖 11 則為平均成功傳送定量封包所需時間的分析圖，分別為 10 個節點與 50 個節點環境下的比較，有成功傳送 100~500 個封包的情況，在節點數只有 10 時可看出 CRBA 相較其他演算法需要花費較多的時間，因為節點數較少時競爭通道情形是沒那麼嚴重的，所以不論是 BEB、EIED 或是 LILD 皆會有較好的表現，因為 CRBA 前期碰撞要收斂至 0.1~0.2 的時間被每 5 次傳輸才調整競爭視窗大小給拖長，所以在 10 個節點的情況是最差的，但當傳輸節點達到 50 時，因為競爭通道情況嚴重，BEB、EIED 及 LILD 的表現就會明顯差了很多，比較起來各有好壞，但本篇論文提出的 RACB 不論在低節點數或高節點數皆擁有較低的傳輸時間，兼具了文獻探討提及之演算法的優點。

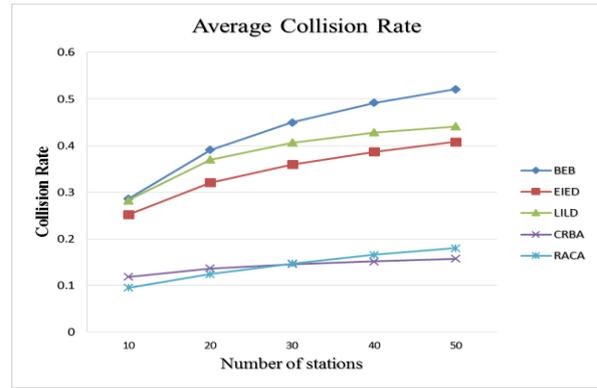


圖 8 不同節點數的碰撞率

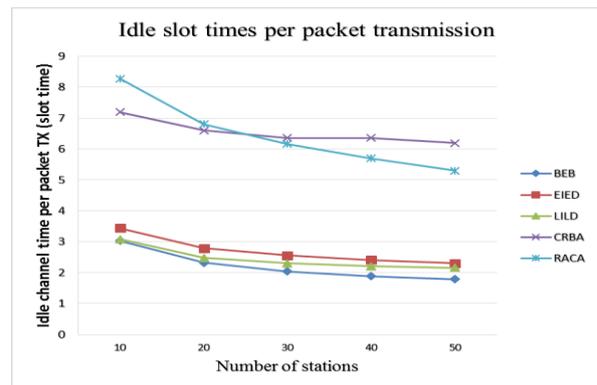


圖 9 不同節點數的通道空閒時間

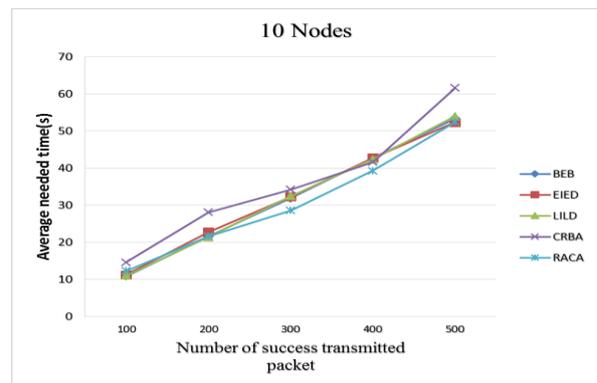


圖 10 成功傳輸定量封包所需時間(10 節點)

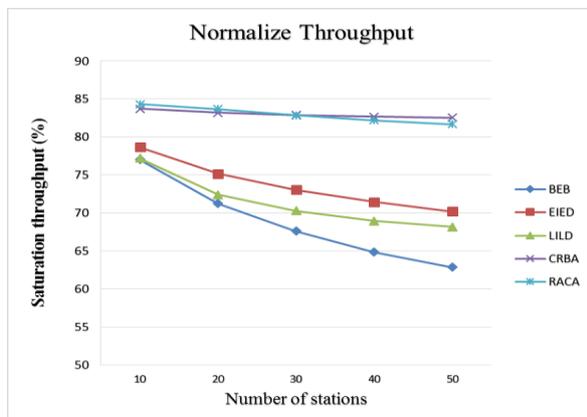


圖 7 不同節點數的吞吐量

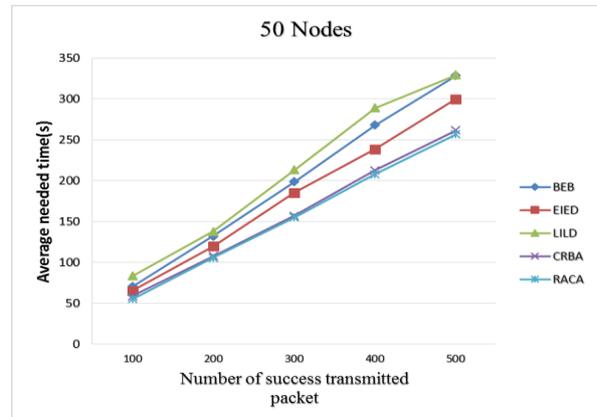


圖 11 成功傳輸定量封包所需時間(50 節點)

5. 結論與討論

在本論文中，我們延續先前最佳碰撞率區間的實驗結果提出了快速自適碰撞倒退演算法 (RACB)，同樣基於碰撞率調整競爭視窗大小，但會參考先前網路的碰撞情形，且每次傳輸便調整，不同於先前研究提出的 CRBA 每 5 次傳輸才調整，所以碰撞率收斂至最佳區間所需的嘗試傳輸次數也相對少了許多，當碰撞率大於或小於最佳區間時大幅度的調整競爭視窗大小，使碰撞率更快進入最佳碰撞區間，當碰撞率落於最佳碰撞區間時，則對競爭視窗大小做細部的微調，降低競爭視窗大小的變化幅度，使碰撞率可以維持在最佳碰撞區間；其吞吐量與 CRBA 相似，皆接近理論上最佳的結果；但在平均成功傳輸封包所需的時間，不論在少節點或是多節點競爭通道時，皆能擁有較短的傳輸時間。

6. 誌謝

本研究成果由科技部計畫編號 NSC 101-2221-E-151-020-MY3 及科技部計畫編號 NSC 103-2221-E-006-144-MY3 補助支持，特此誌謝。

參考文獻

- [1] IEEE standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification, IEEE Standard 802.11, June 1999.
- [2] N. Song, B. Kwak, J. Song, and L. E. Miller, "Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm," The 57th IEEE Semiannual Spring VTC, vol. 4, April 2003.
- [3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," Proceeding ACK SIGCOMM'94, London, England, 1994.
- [4] 韋宇航, 林政翰, 謝錫堃, 柯志亨, "無線區域網路上基於碰撞率之倒退演算法", 2013 全國計算機會議(NCS 2013), 亞洲大學, 2013.