

IEEE 802.11 無線網路倒退機制演算法之公平性探討

王昱淇

國立成功大學電腦
與通信工程研究所

[Sephiroth.Wang@gm
ail.com](mailto:Sephiroth.Wang@gm
ail.com)

林政翰

國立成功大學電腦
與通信工程研究所

[jhlin5@hpds.ee.ncku.
edu.tw](mailto:jhlin5@hpds.ee.ncku.
edu.tw)

梁浩銘

國立成功大學電腦
與通信工程研究所

[joeliang@hpds.ee.nc
ku.edu.tw](mailto:joeliang@hpds.ee.nc
ku.edu.tw)

謝錫堃

國立成功大學電腦
與通信工程研究所

shieh@ee.ncku.edu.tw

摘要

本論文主要探討 IEEE 無線網路的傳輸效能與公平性。我們為了觀察不同的倒退機制演算法，對於傳輸量與公平性的影響，以 BEB(Binary Exponential Back-off) 和 EIED(Exponential Increase Exponential Decrease)為例進行分析，並提出解決方法，使得兩個倒退機制可共存於一個無線網路中。由實驗結果證明，我們所提出的方法可以有效提供 BEB 和 EIED 共存於無線網路中。

關鍵詞：無線網路、802.11、BEB、EIED

1. 前言

IEEE 802.11[1]是目前使用最廣泛的無線區域網路技術。其中，所使用的存取傳輸媒介機制是以 CSMA/CA(Carrier Sense Multiple Access/Collision Avoidance)為主。當無線節點(Wireless Node)有資料要傳送時，會先偵測無線通道(Wireless Channel)是否為空閒，若為空閒則馬上可以傳送資料，反之，無線通道為忙碌時，無線節點必須等到無線通道轉成空閒才可以進行資料傳送。然而，若有兩個無線節點同時偵測到無線通道為空閒時，進而傳送資料，就會造成封包碰撞的情況。因此，802.11 針對此情況便提出了 BEB

(Binary Exponential Back-off) 倒退演算法來加以改進，以避免封包發生碰撞。

BEB 演算法會從競爭視窗中(Contention Window)隨機挑出一個亂數值，當亂數值倒數到 0 時，便把封包傳送出去。若封包是成功的傳送則把競爭視窗設為最小值 (Minimum Contention Window, CW_{min})，若發生碰撞則競爭視窗變成目前值的兩倍。此機制之缺點是當某一次封包成功的傳送，競爭視窗回到了最小值，但競爭節點數多的情況下，再次發生碰撞的機率是相當高，因此會降低網路傳輸效能。在相關的研究中，EIED(Exponential Increase Exponential Decrease)演算法就是針對 BEB 的問題做改善，以提高效能，EIED 與 BEB 的差別在於當成功把封包傳送出去時，競爭視窗不會回到最小，而會是當時競爭視窗的一半，因此較能適應當時無線環境的情況。然而，若要將這兩種演算法共存在同一無線環境下，可能會造成無線節點傳輸機會不公平的現象。所以我們希望藉由分析觀察出兩種機制對無線網路的效能影響，並提出適合的解決方法，讓兩者可以共存。

此份論文的大綱分為五大段。第二段中會先介紹兩種倒退機制演算法，分析其個別優缺點，且合併使用時會產生什麼問題。第三段則會針對共存之問題點提出有效解決方法。第四段是模擬實驗的結果分析。最後，是我們的結論與未來研究方向。

2. 倒退演算法

2.1 BEB 演算法

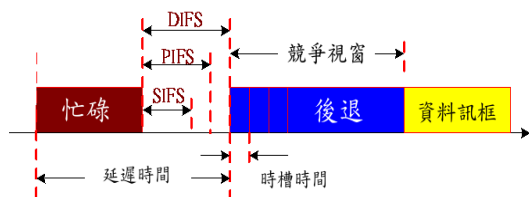


圖 1. 基本 DCF 運作模式

我們用以簡易的方式說明 DCF 的運作方式(圖 1)，剛開始無線節點要傳送資料時需先進行無線通道偵測，當發現無線通道正處於忙碌的狀態，於是無線節點便開始等待直到無線通道變成空閒為止。之後無線節點需再等待一段 DIFS(DCF InterFrame Space)時間，經過系統設定的一個時槽(slot)時間後便進入競爭視窗的部份，無線節點開始倒數計時，等待一段『倒退時間』(Back-off time)，同時也繼續偵測無線通道的狀態。如果自己的倒數時間比別的無線節點早結束，即可取得無線通道的使用權並開始傳送封包。若自己的時間比別的無線節點晚結束，別的無線節點先搶到無線通道的使用權，則在偵測到無線通道轉成忙碌時，就先凍結(frozen)自己的倒數，等到無線通道又轉為空閒時，且經過 DIFS 時間，然後再繼續剛才未完的倒數。直到倒數時間結束，才能取得傳送權而開始傳送封包。

再來我們要探討如何產生倒退時間。首先在 $[0, CW-1]$ 範圍取一個整數亂數值，CW 簡稱為競爭視窗，而倒退時間=亂數值 \times 時槽時間。競爭視窗 CW 隨著重傳次數(m)增加而以指數成長，直到所設定的最大競爭視窗(Max Contention Window, CW_{max})為止，一般可將競爭視窗表示成

$$\begin{cases} CW = 2^i CW_{min} & i < m \\ CW_{max} = 2^m CW_{min} & i = m \end{cases} \quad (1)$$

圖 2 為 BEB 演算法競爭視窗的行為模式，假設 CW_{min} 為 32， CW_{max} 為 1024，當無線環境不佳，封包一直傳送失敗時，競爭視窗就會以兩倍速度成長。反之，當成功傳送封包時，則會把競爭視窗重設為最小值 CW_{min} 。

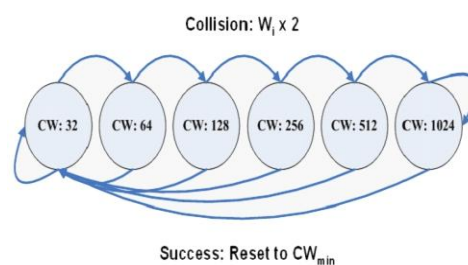


圖 2. BEB 演算法競爭視窗行為模式

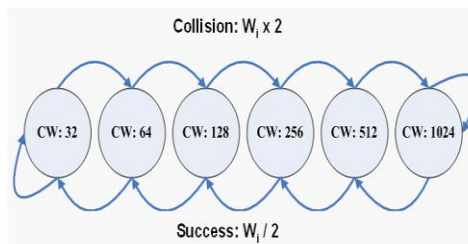


圖 3. EIED 演算法競爭視窗行為模式

2.2 EIED 演算法

EIED 演算法與 BEB 演算法不同之處在於當成功傳送封包時，競爭視窗並不是回歸到 CW_{min} ，而是回到上一個的競爭視窗大小。圖 3 中即是 EIED 演算法競爭視窗的行為模式。我們假設現階段的競爭視窗為 CW_{old} ，若在此階段中傳送成功，則新競爭視窗以數學表示

$$CW_{new} = CW_{old} / 2 \quad (2)$$

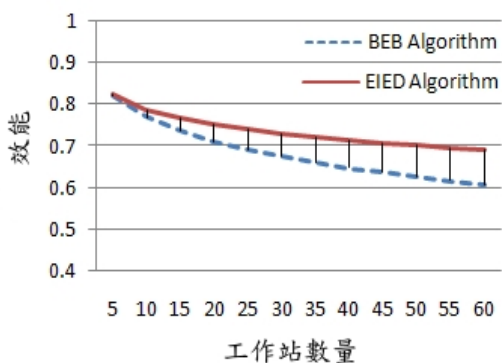


圖 4. 分別使用 BEB 與 EIED 演算法的效能(Throughput)

在圖 4 中可以看到 EIED 演算法的效能是較 BEB 佳,原因在於 EIED 考慮到在實際情況中,當無線節點會連續一直發生碰撞,表示該段時間網路上有大量的無線節點都在競爭傳送通道。因此,若僥倖的搶到傳送權而把競爭視窗設為最小值,對於當時忙碌的網路環境而言,會再次發生碰撞的機率可說是極高,所以 EIED 才會將新的競爭視窗設為上次的一半。

2.3 兩種演算法的合併

在我們實驗的環境下,假設無線節點裡有些使用 BEB 演算法,有些採用 EIED 演算法,探討此時的效能會有怎樣的問題變化。圖 5 中的實驗環境共有 20 台無線節點,可以看到當使用 EIED 的無線節點越多時,其效能會越佳,代表效能是由 EIED 所提昇。

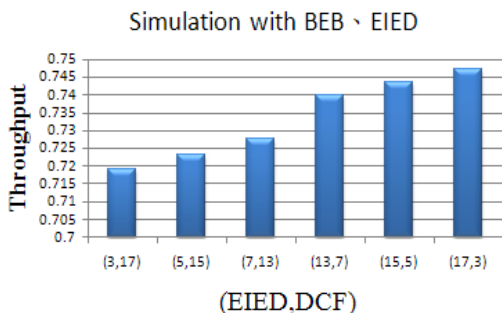


圖 5. EIED 與 BEB 演算法合併使用的效能(Throughput)

接下來我們要探討的是當兩者演算法合併使用時,其傳輸量的公平性。我們以三種情況來討論。(圖 6~8)

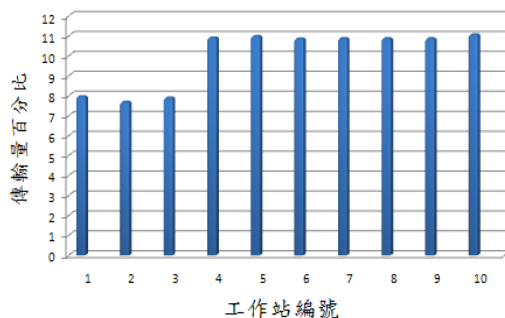


圖 6. EIED 與 BEB 演算法比例為 3 : 7

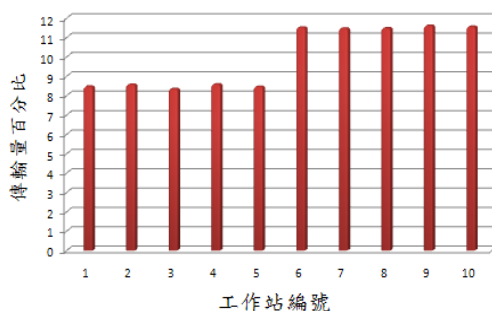


圖 7. EIED 與 BEB 演算法比例為 5 : 5

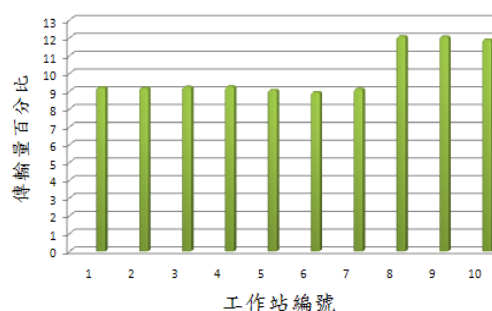


圖 8. EIED 與 BEB 演算法比例為 7 : 3

不管使用 EIED 與 BEB 演算法的比例為何,其傳輸量都無法到達公平性,且使用 EIED 演算法的無線節點,傳輸量一定比較少。之前我們也提過 EIED 在傳送成功時會把競爭視窗設為此次的一半,而 BEB 演算法則是將競爭視窗設為 CW_{min} ,顯然的設為 CW_{min} 一定比較快倒數到零,所以使用 BEB 演算法,其拿到傳輸權的機會一定比 EIED 來得高。其問

題的關鍵就在於 EIED 演算法的競爭視窗容易長的太快，而回不到較小的值。

3. 改善方法

3.1 修改傳送機會(Transmission Opportunity, TXOP)

傳統我們使用 BEB 和 EIED 這兩種演算法，都預設每次拿到傳送權時，皆只能傳送一個封包。為了改善 EIED 的傳輸量，我們參考了 IEEE 802.11e[2]的傳輸的模式，修改其傳送機會(TXOP)，設定 EIED 每次拿到傳送權時可以傳送一個以上的封包量。但此方法有個缺失，當使用兩種演算法的無線節點比例不同，就無法達到公平性。圖 9 中使用 EIED 演算法的無線節點每次拿到傳送權皆能傳送 3 個封包，在 EIED 與 BEB 比例為(15,25)時，編號 1~15 為 EIED，16~40 為 BEB，可達到較高的公平性，最大與最小傳輸量差距約 0.4%~0.5%。若比例為(35,5)時，其傳輸差距量可達 0.7%~0.8%，公平性就不是那麼準確了。

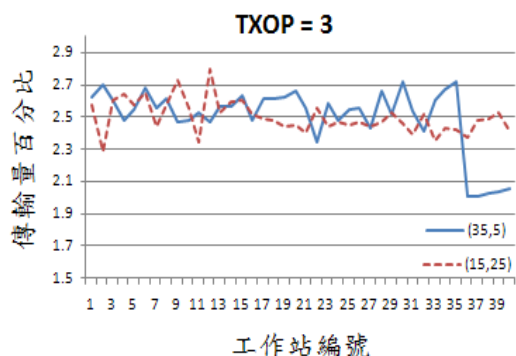


圖 9. 修改 TXOP 來改善公平性

另外，每次傳送的封包量皆只能為整數值，較難以去找出一個平衡點。因此在 3.2 節中我們提出了較為適合的方法，可以不用考慮使用兩種演算法的無線節點

比例，僅只要解決當無線節點數量提昇與下降時的影響。

3.2 動態調整倒退視窗倒數倍率

正因為 EIED 演算法的競爭視窗可能比 BEB 來得大，而搶到的傳送權機率較小，所以改善公平性的另一方法就是加快其競爭視窗倒退的倍率。簡言之，封包成功的傳送後，預設新的競爭視窗會是現階段的一半，為了加快其倒退速度，可以將新的競爭視窗設為現阶段的四分之一或其他更低的倍率。

我們依循著模擬出來的結果，推演出無線節點與倒退速度倍率的一種關係。預設的倒數倍率皆固定為 2，這裡把無線節點的因素考慮進來。若倒數倍率為 k、無線節點個數為 n，則

$$k = \lceil n \div 10 \rceil + 2 \quad (3)$$

倒數的倍率會隨著無線節點數量的不同，而作相對應的改變，且此方法並不會有無線節點比例的問題。圖 10 就明顯的看出即使使用兩種演算法比例不同，最大與最小傳輸量都維持在 0.2%~0.3%之間。

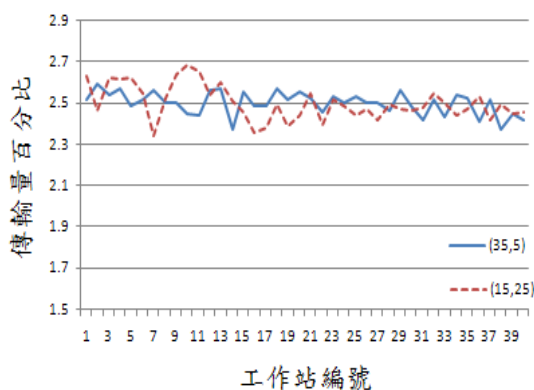


圖 10. 動態調整倒退視窗倒數倍率 (40 個無線節點)

4. 實驗結果

接下來我們會以動態調整倒數倍率的方法來量測網路傳輸的效能，包含公平性、平均一個封包傳送幾次才成功、每個封包平均選到的競爭視窗大小。表格 1 是此實驗中的一些參數。

表格 1. 實驗參數

最小競爭視窗 CW_{min}	32
最大競爭視窗 CW_{max}	1024
SIFS	28 μ s
DIFS	128 μ s
Slot Time	50 μ s
Propagation Delay	1 μ s
封包大小	1000byte
Data Rate	1 Mbit/s
Control Rate	1 Mbit/s

4.1 公平性的量測

公平性是我們此次實驗中優先考慮的重點，第 3.2 節中我們有模擬過 40 個無線節點時的傳輸量公平性，這邊我們針對較少的無線節點來分析，來確認是否仍合乎公平性。圖 11(a)與圖 11(b)中無線節點數量採用 10 台與 20 台，(i,j)分別為 EIED 與 BEB 的節點數目。最大與最小的傳輸量可達 0.3%~0.4%左右，無線節點數量少時有些微的不公平，但還在可以忍受的範圍，差距約在 0.1%。

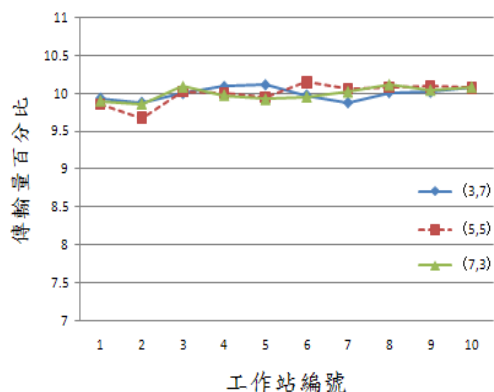


圖 11(a). 動態調整倒數倍率 (10 個無線節點)

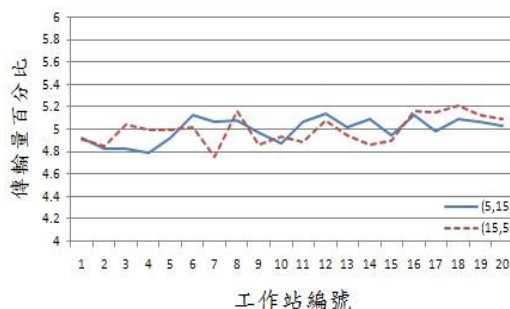


圖 11(b). 動態調整倒數倍率 (20 個無線節點)

4.2 平均一個封包傳送幾次才成功與每個封包平均選到競爭視窗大小

我們將使用 EIED 演算法的無線節點數量固定為 10 台，其餘無線節點則使用 BEB 演算法。由圖 12 可以發現隨著無線節點的數量越多，其因為碰撞越容易發生，因此要成功傳送一個封包的次數也就會隨之提昇。同樣的在圖 13 裡，無線節點數量提昇，導致碰撞機率上升，平均選到的競爭視窗值也會越來越高。

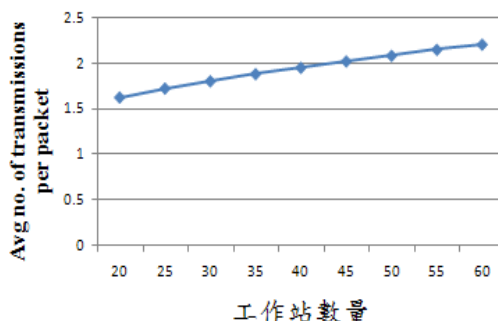


圖 12. 平均一個封包傳送幾次才成功

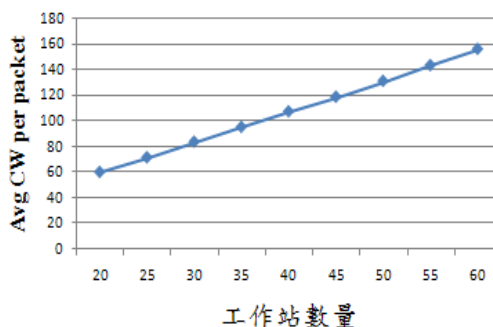


圖 13. 每個封包平均選到競爭視窗大小

5. 結論與未來

最後我們提出的動態調整倍率的方法，可以有效的改善當使用 EIED 與 DCF 演算法時，彼此傳輸量的公平性。若某個無線節點中只能使用 EIED 演算法時，就不會導致自己傳輸量比其他無線節點來得少。但此方法的缺點在於效能，雖然可以達到公平性，但卻無法讓效能提昇。但整體而言，此方法所得之效能仍會比完全使用 BEB 演算法時來得高。

未來仍須改進的在於當使用 EIED 的無線節點極高於 BEB 時，在不考慮其公平性下，其效能原本應該會很高，但套用我們的方法後，效能反而被拉低，之後我們會繼續針對當 EIED 與 BEB 使用比例不同時，進一步的探討如何把公平性與效能做到一個最佳平衡的狀態。

參考文獻

- [1] IEEE Standard for Wireless LAN medium Access Control (MAC) and Physical Layer (PHY) Specifications, Nov. 1997. P802.11
- [2] IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, 2005
- [3] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function”, IEEE Journal on Selected Area in Communications, vol.18, no.3, pp. 535-547, 2000.
- [4] N. Qiang, I. Aad, C. Barakat and T. Turletti, “Modeling and analysis of slow CW decrease in IEEE 802.11”, in Proceedings of IEEE Conference Personal, Indoor Mobile Radio Communications (PIMRC 2003), pp. 1717-1721, Sept. 2003.
- [5] N. O. Song, B. J. Kwak, J. Song, and L. E. Miller. “Enhancement of IEEE 802.11 Distributed Coordination Function with Exponential Increase Exponential Decrease Backoff Algorithm”, in Proc. of VTC 2003 (Spring), New Orleans, 2003.
- [6] I. Vukovic and N. Smavatkul, “Saturation throughput analysis of different backoff algorithms in IEEE 802.11”, in Proceedings of the IEEE Conference on Personal, Indoor and Mobile Radio Communications (PIMRC 2004), Barcelona, Spain, Sept. 2004.
- [7] 簡榮宏、廖冠雄，無限區域網路，全華科技圖書股份有限公司，2007。