International Journal of Innovative Computing, Information and Control
Volume **8**, Number **8**, August **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 5425–5438

JOINT PRIORITY RANDOM EARLY DETECTION AND IEEE 802.11E EDCF TO SUPPORT QOS IN WLAN

Chien-Erh Weng[1], Ruei-Yi Li[2] and Chao-Hsing Hsu[3]

[1]Department of Electronic Communication Engineering
National Kaohsiung Marine University
No. 142, Haijhuan Road, Nanzih District, Kaohsiung City 81157, Taiwan
ceweng@mail.nkmu.edu.tw

[2]Institute of Communications Engineering
National Chung Cheng University
No. 168, University Road, Minhsiung Township, Chiayi County 62102, Taiwan

[3]Department of Electronic Engineering
Chienkuo Technology University
No. 1, Chiehshou North Road, Changhua City 500, Taiwan
chaohsinghsu@yahoo.com

Received April 2011; revised October 2011

Abstract. *During the past few years the widespread use of the wireless local area networks (WLANs) communication technique is one of the most popular technologies in data telecommunications and networking. With the increasing variety of multimedia applications, such as voice, video and web traffic, it is needed to develop a mechanism for the quality of service (QoS) to support different types of traffic in WLAN. IEEE 802.11e EDCF is a new wireless technology to support QoS. However, EDCF never discusses the queuing issue in a station (STA) with multiple services. In this paper, we propose priority random early detection (PRED) algorithm which integrated Random Early Detection (RED) with IEEE 802.11e EDCF to support QoS in WLANs. Simulation results show that PRED algorithm can provide better performance of access categories (ACs) with high priority than that EDCF with traditional queuing (FIFO) strategy. Besides, we performed the parameters tuning according to the traffic load. By dynamically tuning the parameters, we can prevent throughput decreasing due to the large number of collisions under heavy traffic load condition.*

Keywords: QoS, PRED, STA, EDCF, ACs, FIFO

1. **Introduction.** In WLAN, the medium access control (MAC) protocol is the key element that provides the efficiency in accessing the channel, while satisfying the QoS requirements of multiple flows. IEEE 802.11 distributed WLAN has become widely deployed since the contention-based MAC protocol is simple, robust, and allows fast installation with minimal management and maintenance costs. Although the contention-based MAC protocol fits for best effort traffic, it is unsuitable for multimedia services with QoS requirements. QoS is necessary for real-time applications such as web, voice or video transmissions. Even though IEEE 802.11 has mentioned a contention-free MAC protocol, it is hardly implemented due to several reasons, such as higher complexity and inefficiency for normal best effort traffic, lack of robustness, and the strong assumption of global synchronizations [1].

In order to support multiple services in WLAN, we must develop a suitable mechanism according to bandwidth, delay, packet loss, jitter, etc. A simple and effective scheme for improving the QoS performance in WLAN named as enhanced DCF (EDCF) is presented.

There have been many performance studies for EDCF with priority schemes. Deng and Chang [2] proposed a priority scheme by differentiating the backoff window: the higher priority class uses the window $[0, 2^{i+1} - 1]$ and the lower priority class uses the window $[2^{i+1}, 2^{i+2} - 1]$, where i is the backoff stage. Aad and Castelluccia [3] proposed a priority scheme by differentiating inter-frame spaces (IFS's), in which a higher priority class uses IFS, whereas a lower priority class uses a space that equals the sum of IFS and the maximum window size. In [4], Veres and Campbell et al. proposed priority schemes by differentiating the minimum backoff window size and the maximum window size. E. Ziouva and T. Antonakopoulos [5] presented an accurate analysis to compute the saturation throughput and delay performances. Wen and Weng [6] proposed a modified model from [5], and extended the model to support EDCF in ideal channel scenario. In [7], Y. Yan and C. Pan proposed an improved discrete three-dimension markov chain model. They considered elaborately the modified AIFS and backoff co-operation process in EDCF defined by the IEEE 802.11e standard. W. K. Lai et al. [8] proposed a novel scheme for the adaptation of the ratio of HCCA and EDCA periods to reduce the average delay and to improve the overall system throughput.

These performance studies never discuss the queuing issue in the station (STA) with multiple services. If each STA uses IEEE 802.11e EDCF to contend channel under heavy traffic load condition, the performance of each STA will be inefficient because of collisions.

In this paper, we propose a modified random early detection algorithm named as priority random early detection (PRED) to support QoS in WLAN. PRED provides a queuing algorithm in each STA and performs parameters tuning to get better performances in terms of higher throughput and lower delay for higher priority packets.

The rest of this paper is organized as follows. Section 2 presents the related work. A new algorithm that modified RED algorithm named as PRED including the descriptions of the adaptive tuning, optimal value is presented in Section 3. Section 4 analyzes simulation results to evaluate the effectiveness of our proposed algorithm. Finally, we conclude the paper in Section 5.

2. Related Work.

2.1. RED: random early detection algorithm.

RED algorithm solves congestion problem in packet-switched networks. RED can prevent congestion by controlling the average queue size. The strategy it adopted is dropping packets timely and ensuring that there will be always a buffer available for an incoming packet. The RED algorithm is given in Figure 1 [9]. RED computes the average queue size (avg) by using exponential weighted moving average (EWMA) at each packet arrival at the queue. If the estimated avg exceeds its minimum threshold ($\min_{th}$), then the dropping probability (P_a) that increases with avg until P_b reaches the maximum dropping threshold ($\max_p$) when avg reaches its maximum threshold ($\max_{th}$). If avg exceeds $\max_{th}$, the final dropping probability P_a is set to one. P_a and P_b are computed as follows:

$$P_a = \frac{P_b}{(1 - count * P_b)} \tag{1}$$

$$P_b = \frac{\max_p (avg - \min_{th})}{\max_{th} - \min_{th}}, \tag{2}$$

where $count$ is the number of un-dropped packets. From Equation (2) we know that P_b varies linearly in range $[0, \max_p]$, and P_a varies in range $[0, 1]$. The relationship between avg and P_a is shown in Figure 2.

```
Initialization:
    the average queue size $avg$ is 0
for each packet arrival:
    calculate the new $avg$
    if $min_{th} \le avg \le max_{th}$
        calculate the packet-marking probability $P_a$
        mark the arriving packet with $P_a$
    else if $max_{th} \le avg$
        mark the arriving packet
```

Figure 1. The RED algorithm

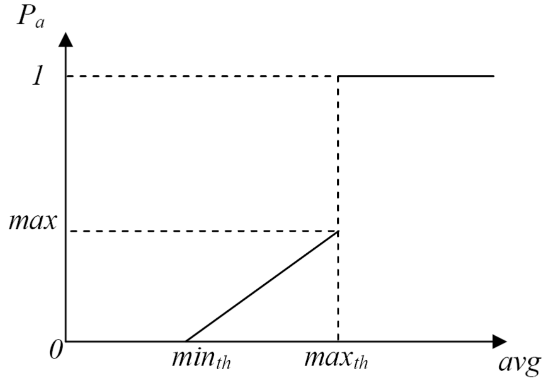

Figure 2. The relationship between *avg* and P_a

2.2. **Overview of the IEEE 802.11e EDCF.** EDCF is a feasible scheme for different QoS requirements. EDCF specifies four default access categories (ACs). Each STA contends for the channel access and independently starts its backoff depending on its associated AC. Each AC uses arbitration interframe space (*AIFS*[AC]), $CW_{\min}$[AC] and $CW_{\max}$[AC] instead of the DIFS time, $CW_{\min}$ and $CW_{\max}$ of the DCF.

The contention method of EDCF is the same as that in DCF. Each STA having a frame to transmit must wait for the channel to be idle without interruption for a period *AIFS*[AC], and then it should start a random backoff process with its own *CW*[AC]. For each time slot interval, during which the channel stays idle, the random backoff value is decremented. When the backoff counter reaches zero, the frame is transmitted. *AIFS*[AC] is calculated as follows:

$$AIFS[\text{AC}] = AIFSN[\text{AC}] * aSlotTime + aSIFSTime, \tag{3}$$

where *AIFSN* [AC] is shown in Table 1, *aSIFSTime* is the duration of a short interframe space, *aSlotTime* is the time interval that the backoff counter uses as time unit the duration a station needs to detect the transmission of a frame from any other station, and the backoff time is calculated as follows:

$$\text{backoff time} = random\_integer * aSlotTime, \tag{4}$$

where *random_integer* is uniformly and randomly chosen in the range $(0, CW[AC])$, instead of $(0, CW - 1)$ in the DCF. Initially, *CW* of each AC is equal to $CW_{\min}$[AC]. After each collision, *CW* is doubled up to:

$$CW_{\max}[\text{AC}] = 2^m * (CW_{\min}[\text{AC}]), \tag{5}$$

where m is called the maximum backoff stage. Once it reaches $CW_{\max}$[AC], it remains at this value until it is reset [1].

Table 1. AIFSN for each ACs

AC	AC_BK	AC_BE	AC_VI	AC_VO
AIFSN	7	3	2	2

3. **PRED: priority random early detection algorithm.** In the proposed PRED algorithm, the packet which enters the queue will use EDCF to contend the channel. The structure of PRED is shown in Figure 3. We offer corresponding $\min_{th}$[AC] and $\max_{th}$[AC] according to each AC. The packets with higher priority have bigger $\min_{th}$[AC] and $\max_{th}$[AC]. On the contrary, the packets with lower priority have smaller $\min_{th}$[AC] and $\max_{th}$[AC]. By using the corresponding $\min_{th}$[AC], $\max_{th}$[AC], AIFS[AC], $CW_{\min}$[AC] and $CW_{\max}$[AC], PRED can support QoS requirements in WLAN.

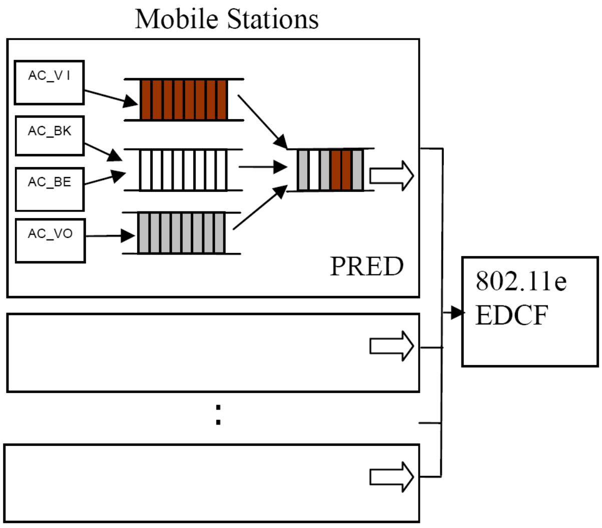

Figure 3. The structure of PRED

3.1. **Adaptive tuning of PRED algorithm.** This section shows the tuning method of parameters according to the traffic load condition. Under heavy traffic load condition, the competing STAs will increase and collisions will increase too. If we still use the same parameters in heavy traffic load condition, the throughput of packets with high priority will decrease because of colliding. So, in order to maintain suitable QoS requirement under the heavy traffic load condition, we need to adjust the PRED algorithm according to the traffic load condition.

The adaptive tuning is shown in Figure 4. q[AC] is a threshold that judges whether the arriving packet can perform PRED algorithm or not. If the queue size of a STA is greater than q[AC], the arriving packet will be dropped. Otherwise, the arriving packet can perform PRED algorithm. The initial value of q[AC] is equal to the queue size of the STA. After collision, q[AC] is decreased as:

$$q[\mathrm{AC}] = q[\mathrm{AC}] - tune[\mathrm{AC}], \tag{6}$$

where $tune$[AC] is a constant value and $tune[0] > tune[1] > tune[2] > tune[3]$. On the contrary, if after several consecutive successful transmissions (*uncollide_time*), q[AC] is increased with $tune$[AC]: $q[\mathrm{AC}] = q[\mathrm{AC}] + tune[\mathrm{AC}]$.

Under slight traffic load condition, the change of q[AC] is slight. However, under heavy traffic load condition, the gap of q[AC] will increase. Therefore, the packets with lower priority will be dropped first and the packets of higher priority can still maintain the QoS requirement even under heavy traffic load condition.

```
for each AC
    Initialization:
        $q$[AC] = $queue\_size$
    for each packet arrival
        if current queue size > $q$[AC]
            drop the arriving packet
        else perform PRED
    end for
    if collision
        reset $uncollide\_time$
        $q$[AC] = $q$[AC] − $tune$[AC]
    else
        $uncollide\_time$ = $uncollide\_time$ − 1
        if $uncollide\_time$ = 0
            reset $uncollide\_time$
            $q$[AC] = $q$[AC] + $tune$[AC]
    end if
```

Figure 4. The adaptive tuning of PRED

Even CW[AC] reaches $CW_{\max}$[AC], it still can be doubled after several consecutive collisions ($cont\_c$). This method can solve the problem of colliding under heavy traffic load condition.

3.2. **Collision resolution with doubling CW in EDCF.** In fact, due to the nature of IEEE 802.11e EDCF and in particular due to the dynamic adjustment of q[AC], a resolution for decreasing the collision probability is necessary. The drawback of IEEE 802.11e EDCF is that the contention window will be reset to the initial value ($CW_{\min}$[AC]) after each success transmission, regardless the traffic load condition. This method is effective under slight traffic load condition. However, under heavy traffic load condition, it is ineffective since the collision probability increase.

In the above section, the packets with higher priority can maintain the QoS requirement even under heavy traffic load condition. However, the problem is that the collision probability will increase since $CW_{\max}$[AC] of higher priority packets is small. Therefore, we need to resolve this problem. The resolve procedure is shown in Figure 5. Even CW[AC] reaches $CW_{\max}$[AC], it still can be doubled after several consecutive collisions ($cont\_c$). This method can solve the problem of colliding under heavy traffic load condition.

```
when $CW$[AC] = $CW_{max}$[AC]
    if the number of consecutive collisions = $cont\_c$
        double $CW$[AC]
(where AC can be 2 or 3)
```

Figure 5. The resolve procedure

3.3. **Optimal value of *cont_c* in PRED algorithm.**

In this section we will discuss with optimal *cont_c*. If *cont_c* is too small, the packets with higher priority will be allowed to increase their CW[AC] easily, and as a consequence they will not be able to achieve the desired QoS requirement. On the other hand, if *cont_c* is too large, the collision probability will be very high. Therefore, we will derive the optimal value of *cont_c* from the analysis.

In the analysis we use similar procedures and index in [10]. Let p be the probability that a transmitted packet collides, τ be the probability that a STA transmits in a randomly chosen slot time, $b(t)$ is defined as a stochastic process that presents the value of the backoff counter for a given station at slot time t. We assume that each STA has $m+n$ stages of backoff delay and that $s(t)$ is the stochastic process representing the backoff stage i at time t. The value of the backoff counter is randomly chosen in the range $(0, W_i - 1)$, where $W_i = 2^i W_{\min}$ and depends on the STA's backoff stage i. It is possible to model the bi-dimensional process $\{s(t), b(t)\}$ with the discrete-time markov chain model of PRED depicted in Figure 6. The transition probabilities are listed as follows:

$$\begin{cases} P\{i,k|i,k+1\} = 1, & k \in [0, W_i - 2], i \in [0, m+n] \\ P\{0,k|0,0\} = \frac{(1-p)}{W_0}, & k \in [0, W_0 - 1] \\ P\{i,k|i-1,0\} = \frac{p}{W_i}, & k \in [0, W_i - 1], i \in [1, m] \\ \qquad\qquad\qquad\;\; = \frac{p'''}{W_i}, & k \in [0, W_i - 1], i \in [m+1, m+n] \\ P\{i-1,k|i,0\} = \frac{p'}{W_i - 1}, & k \in [0, W_{i-1} - 1], i \in [1, m+n] \\ P\{i,k|i,0\} = \frac{p''}{W_i}, & k \in [0, W_i - 1], i \in [1, m-1] \\ \qquad\qquad\quad = \frac{1-p'-p'''}{W_i}, & k \in [0, W_i - 1], i \in [m, m+n] \\ P\{m+n,k|m+n,0\} = \frac{1-p'}{W_{m+n}}, & k \in [0, W_{m+n} - 1] \end{cases} \tag{7}$$

Then, we can construct corresponding transition equations of markov chain model, where $p' = (1-p)^c$, $p'' = 1 - p - p'$, $p''' = p^{cont\_c}$, and we can aggregate the state (i,k) into a single state $(i,0)$, so it is easy to get that

$$\begin{cases} \rho_1 = \frac{p}{p'} \to b_{i,0} = \rho_1^i b_{0,0}, & 0 \le i \le m \\ \rho_2 = \frac{p'''}{p'} \to b_{i,0} = \rho_2^{i-m} b_{m,0} = \rho_2^{i-m} \rho_1^m b_{0,0}, & m+1 \le i \le m+m \end{cases} \tag{8}$$

For each $k \in [0, W_i - 1]$, $b_{i,k}$ also has the relationship

$$\begin{cases} b_{i,k} = \frac{W_i - k}{W_i}[(1-p)b_{0,0} + p'b_{i+1,0}] & i = 0 \\ b_{i,k} = \frac{W_i - k}{W_i}[pb_{i-1,0} + (1-p-p')b_{i,0} + p'b_{i+1,0}] & 0 < i < m \\ b_{i,k} = \frac{W_i - k}{W_i}[pb_{i-1,0} + (1-p'-p''')b_{i,0} + p'b_{i+1,0}] & i = m \\ b_{i,k} = \frac{W_i - k}{W_i}[p'''b_{i-1,0} + (1-p-p')b_{i,0} + p'b_{i+1,0}] & m < i < m+n \\ b_{i,k} = \frac{W_i - k}{W_i}[p'''b_{i-1,0} + (1-p')b_{m+n,0}] & i = m+n \end{cases} \tag{9}$$

By Equation (8), we can get:

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0} \quad 0 < i < m+n. \tag{10}$$

The probability conservation relation states that:

$$\sum_{i=0}^{m+n} \sum_{k=0}^{W_i - 1} b_{i,k} = 1 \Rightarrow \sum_{i=0}^{m+n} b_{i,0} \frac{W_i + 1}{2} = 1. \tag{11}$$

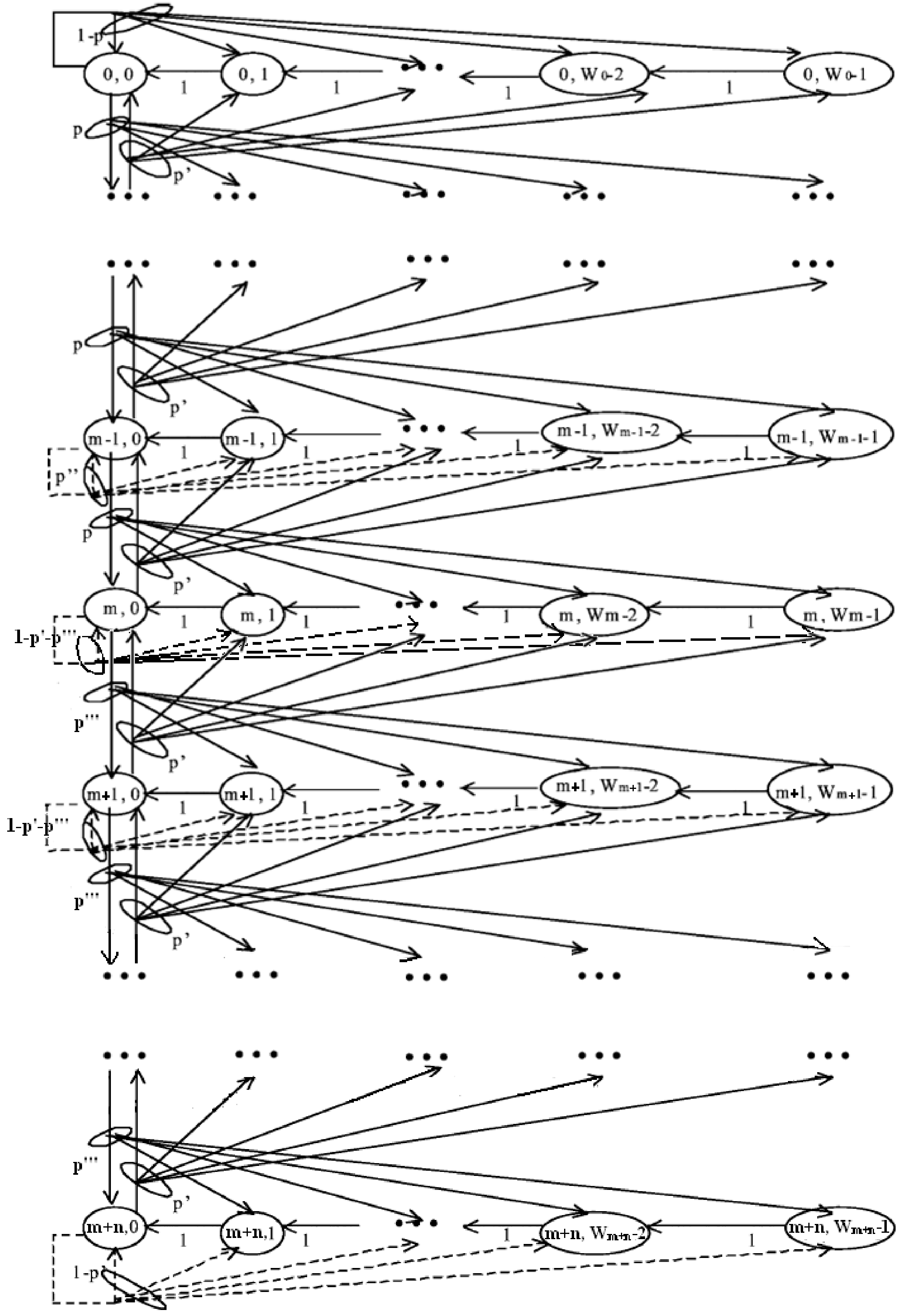

Figure 6. Markov chain model of PRED

In Equation (11), $b_{i,0}$ can be computed using Equation (8) and the backoff window size at the backoff stage i can be shown as:

$$\begin{aligned} W_i &= 2^i W \quad & i \le m' \\ &= 2^{m'} W \quad & i > m'. \end{aligned} \tag{12}$$

Replacing Equation (11) with Equation (8) and Equation (12), we can derive this formula within two cases. One is $m + n \le m'$ and the other is $m + n > m'$. Therefore, we can get the value of $b_{0,0}$ as

$$\begin{aligned} \tfrac{1}{b_{0,0}} &= T_1\left(\rho_1, \rho_2, m, n\right) & m + n \le m' \\ &= T_1\left(\rho_1, \rho_2, m, m' - m\right) + T_2 & m + n > m', \end{aligned} \tag{13}$$

where

$$\begin{aligned} T_1\left(\rho_1, \rho_2, m, n\right) &= \tfrac{1-\rho_1^{m+1}}{2(1-\rho_1)} + \tfrac{W\left(1-(2\rho_1)^{m+1}\right)}{2(1-2\rho_1)} + \tfrac{\rho_1^m \rho_2\left(1-\rho_2^n\right)}{2(1-\rho_2)} + \tfrac{\rho_1^m W 2^{m+1} \rho_2(1-(2\rho_2)^n)}{2(1-2\rho_2)} \\ T_2 &= \tfrac{\rho_1^m \rho_2^{m'-m+1}\left(1-\rho_2^{m+n-m'}\right)\left(2^{m'} W+1\right)}{2(1-\rho_2)}. \end{aligned}$$

We can express the probability τ that a STA transmits in a randomly chosen slot time. As any transmission occurs when the backoff timer is equal to zero, regardless of the backoff stage, it is

$$\begin{aligned} \tau &= \sum_{i=0}^{m+n} b_{i,0} = \sum_{i=0}^{m} b_{i,0} + \sum_{i=m+1}^{m+n} b_{i,0} \\ &= b_{0,0}\left(\tfrac{1-\rho_1^{m+1}}{1-\rho_1} + \tfrac{\rho_1^m \rho_2\left(1-\rho_1^n\right)}{1-\rho_2}\right). \end{aligned} \tag{14}$$

For convenience of the following discussions, τ can be modified as:

$$\tau = f_M(c, cont\_c, p), \tag{15}$$

where c is the consecutive successful transmissions. In general, τ depends on the conditional collision probability p. To find the value of p that a transmitted packet encounter a collision, is the probability that, in a time slot, at least one of the $N - 1$ remaining STAs transmit. At steady state, each remaining STA transmits a packet with probability τ. That yields

$$p = 1 - (1 - \tau)^{N-1}. \tag{16}$$

The normalized system throughput S can be defined as:

$$S = \frac{E[\text{payload information transmitted in a slot time}]}{E[\text{length of a slot time}]}. \tag{17}$$

The average amount of payload information successfully transmitted in a slot time is $P_s P_{tr} E\left[L\right]$, $E[L]$ is the average packet payload size, and $P_s P_{tr}$ is the successful transmission probability in a slot time. The average length of a slot time is readily obtained considering that, the slot time is empty with probability $1 - P_{tr}$, it contains a successful transmission with probability $P_s P_{tr}$, and it contains a collision with probability $\left(1 - P_s\right) P_{tr}$. Hence, we get

$$S = \frac{P_s P_{tr} E\left[L\right]}{\left(1 - P_{tr}\right)\sigma + P_s P_{tr} T_s + \left(1 - P_s\right) P_{tr} T_c}, \tag{18}$$

where T_s is the average time the channel is sensed busy because of a successful transmission or collision, and T_c is the average time that the channel has a collision. T_s and T_c can be computed as shown in Equation (19), where δ is the propagation delay, and $E[L*]$ is the average length of the longest packet payload involved in a collision. In this paper, all the packets have the same fixed size, so $E[L] = E[L*] = L$.

$$\begin{cases} T_s = AIFS + PHY_{hdr} + MAC_{hdr} + E\left[L\right] + \delta + SIFS + ACK + \delta \\ T_c = AIFS + PHY_{hdr} + MAC_{hdr} + E\left[L*\right] + SIFS + ACK. \end{cases} \tag{19}$$

In order to get the optimal values of c and *cont_c*. We use the following method to determine the optimal values of c and *cont_c*. Let us rewrite Equation (18) as

$$S = \frac{E\left[L\right]}{T_s + \frac{(1-\tau)^N \sigma + \left[1-(1-\tau)^N - N\tau(1-\tau)^{N-1}\right] T_c}{N\tau(1-\tau)^{N-1}}}. \tag{20}$$

The analytical model is very convenient to determine the maximum achievable saturation throughput. As T_s, T_c, $E[L]$, and σ are constants, the throughput is maximized when the following quantity is maximized:

$$\frac{N\tau\left(1-\tau\right)^{N-1}}{\left(1-\tau\right)^N \sigma + \left[1-\left(1-\tau\right)^N - N\tau\left(1-\tau\right)^{N-1}\right] T_c}. \tag{21}$$

From Equation (21) we can get the optimal value of τ as

$$\left(1-\tau\right)^N \sigma + \left[1 - N\tau - \left(1-\tau\right)^N\right] T_c = 0. \tag{22}$$

If N is too large, we get:

$$\left(1-\tau\right)^N \approx 1 - N\tau + \frac{N\left(N-1\right)}{2}\tau^2 \tag{23}$$

Let $T_c' = T_c/\sigma$ be the normalized average collision length. We can obtain the optimal value of τ as

$$\tau_0 = \frac{\sqrt{\frac{\left[1+2(N-1)(T_c'-1)\right]}{(N-1)}}}{\left(N-1\right)\left(T_c'-1\right)}. \tag{24}$$

When we get the optimal value of τ, we can obtain the optimal values of c and *cont_c* according to Equation (15) and Equation (16).

4. **Performance Evaluation.** In this section, the parameters of our analysis are as follows: Table 2 shows the parameters setting of EDCF. AC(3) has the minimum value of $CW_{\min}$, $CW_{\max}$ and AIFS, respectively. The packet length is $(36 + 2340)/2 = 1188$ bytes, data rate is 27Mb/s and time slot is 9 μs [11].

Table 2. The parameters setting of EDCF

AC	$CW_{\min}$	$CW_{\max}$	AIFS
AC(0)	15	1023	9
AC(1)	15	1023	5
AC(2)	7	15	4
AC(3)	3	7	4

We simulate the throughputs of IEEE 802.11e EDCF. From the result as shown in Figure 7, we can observe that the performance of IEEE 802.11e EDCF is efficient under slight traffic load condition, but it is inefficient when the number of STAs is large. That is because the collision probability increases under heavy traffic load condition.

Figure 8 shows the throughputs of PRED without adaptive tuning. The curves (AC(n) with FIFO) represent EDCF with traditional queuing (FIFO). The curve (AC(n) with

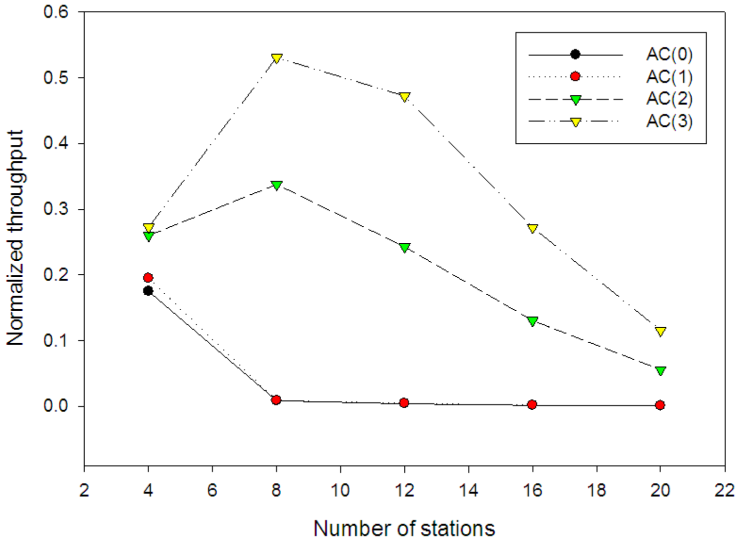

Figure 7. Throughputs of EDCF

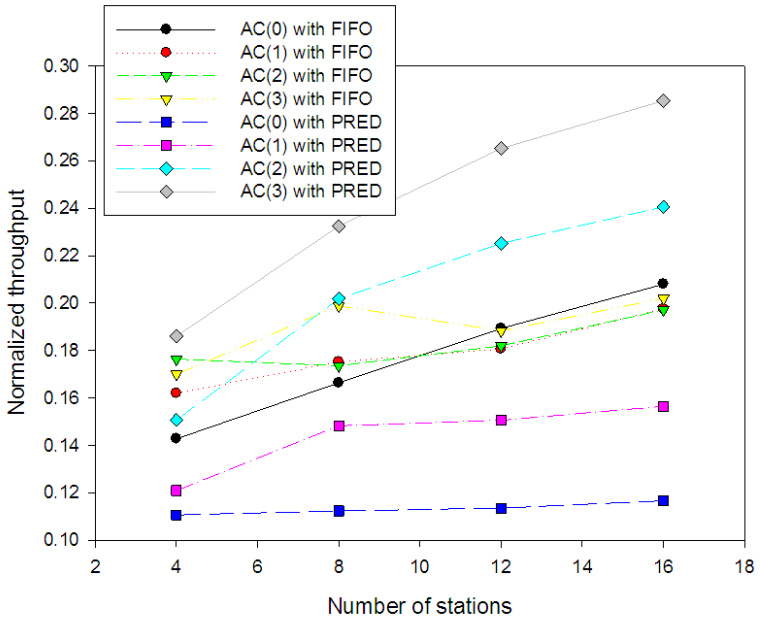

Figure 8. Throughputs of FIFO and PRED

PRED) represent the EDCF with PRED. From the simulation result, AC(3) with PRED gets better performance than that with FIFO. This is because PRED can support higher throughput and lower delay for higher priority ACs. From Figure 9 we can observe that by utilizing the PRED algorithm the dropping rate of AC(3) and AC(2) is reduced. Therefore, higher priority ACs have more chances to access the channel.

Figure 10 shows the throughputs of adaptive tuning of $q[\mathrm{AC}]$. The values of $tune[\mathrm{AC}]$ are setting as follows: $tune[0] = 0.05$, $tune[1] = 0.035$, $tune[2] = 0.015$ and $tune[3] = 0$. The value of $uncollide\_time$ is 50. The throughput of AC(3) without the tuning of $q[\mathrm{AC}]$ is decreasing under heavy traffic load condition, but the AC(3) with the tuning of $q[\mathrm{AC}]$ is increasing under heavy traffic load condition. Therefore, the adaptive tuning of $q[\mathrm{AC}]$ according to the traffic load condition is necessary.

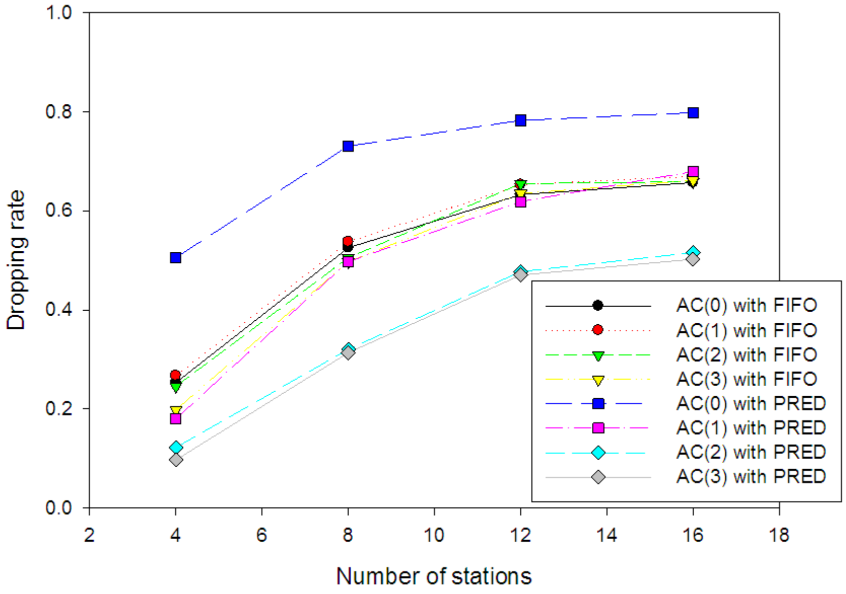

Figure 9. Dropping rate of EDCF with FIFO and EDCF with PRED

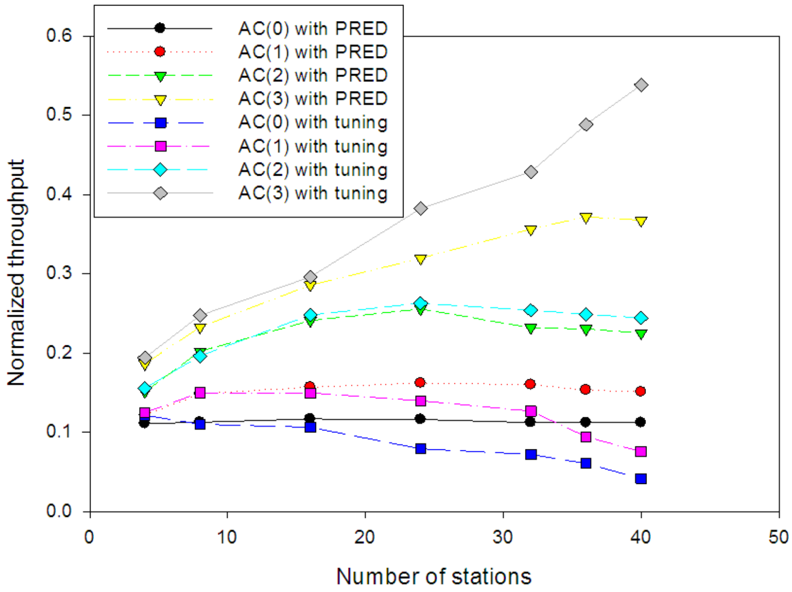

Figure 10. Throughputs for PRED and tuning $q[\mathrm{AC}]$

Figure 11 shows the throughputs for tuning the values of c and *cont_c*. The parameters are set as follows: the number of competing STAs ($N = 40$), $T_c = 430$ μs. Therefore, $c = 200$ and $cont\_c = 75$. In the simulation, we only double CW[3] after *cont_c* consecutive collisions. After doubling CW[3], the range of backoff time is larger than that of IEEE 802.11e EDCF and the collision probability will decrease. Therefore, the throughput of AC(3) will increase.

Figure 12 discusses the relationship between average delay and *cont_c*. From Figure 12 the average delay of AC(2) and AC(3) with doubling CW[3] are shorter than that without doubling CW[3] ($cont\_c \rightarrow \infty$) and average delay of AC(0) and AC(1) have less

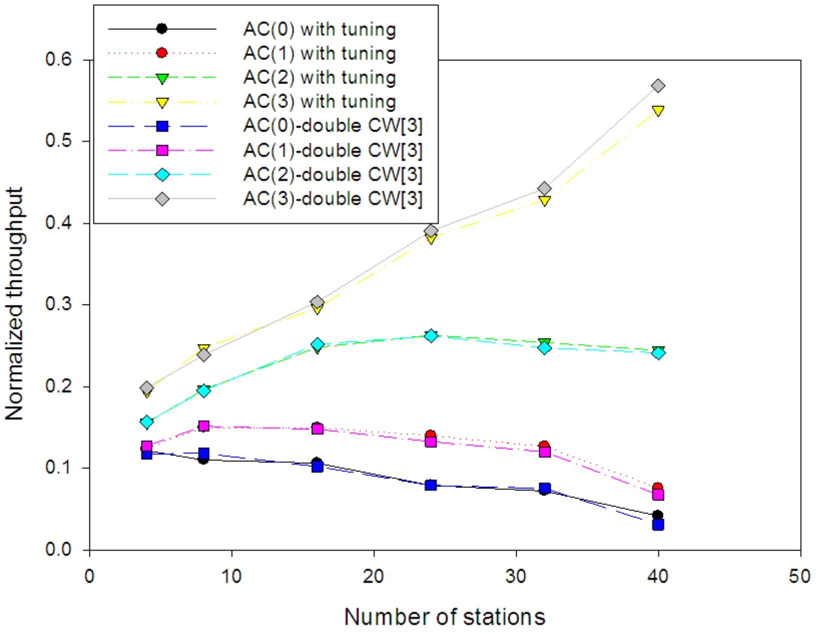

FIGURE 11. Throughputs for tuning q[AC] and doubling CW[3]

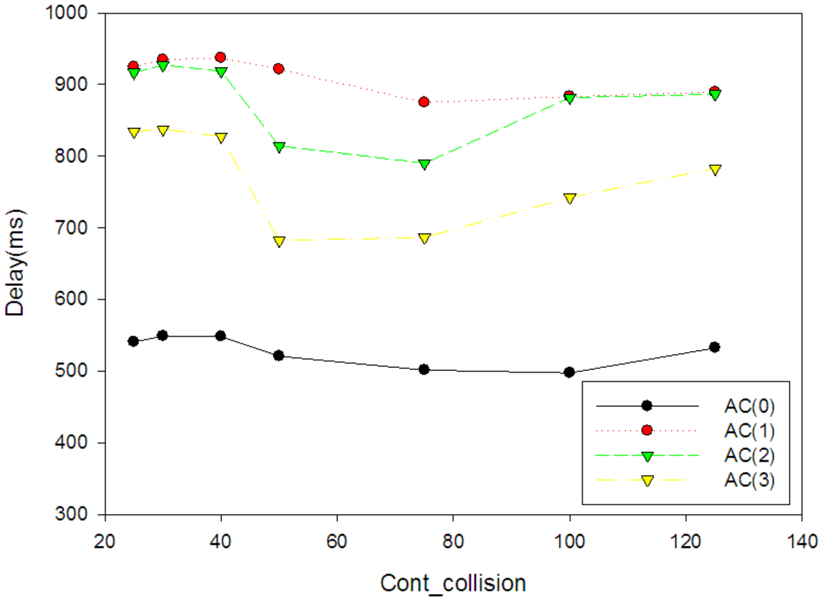

FIGURE 12. The relationship between *cont_c* and average delay

concern with *cont_c*. The reason is that AC(2) and AC(3) have more opportunity to enter queue than AC(0) and AC(1) under heavy traffic load condition. Therefore, average delay of AC(2) and AC(3) have much relate to *cont_c*. After doubling CW[3], the collision probability will decrease and packets can be transmitted in time. Therefore, average delay of AC(2) and AC(3) will decrease. However, when the value of *cont_c* is too small (< 50), CW[3] will increase quickly and the range of backoff time will increase too. Therefore, the value of *cont_c* cannot be chosen too small.

Figure 13 shows the overall throughput of each method. Under heavy traffic load condition, PRED with tuning and double CW[3] method can reduce the collision probability than PRED with tuning and PRED. Therefore, the overall throughput of PRED with tuning and double CW[3] is increase even under heavy traffic load condition.

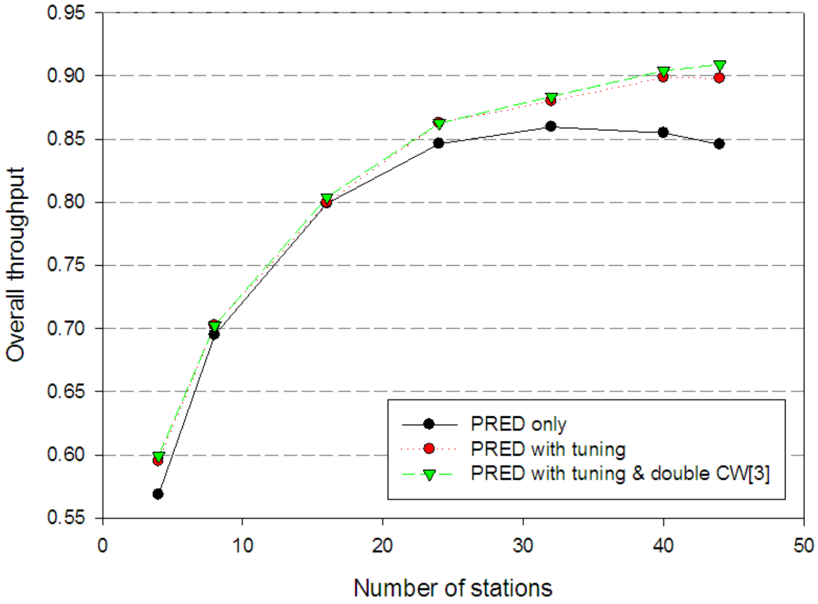

FIGURE 13. Overall throughputs

5. **Conclusions.** In this paper we have proposed an efficient algorithm for QoS requirements of priority application in WLAN named as PRED. PRED provides a queuing algorithm for the priority of packets within each STA. For the channel access outside each STA, PRED introduces parameters c and $cont\_c$ to modify the original IEEE 802.11e EDCF, it obtains higher throughput especially under heavy traffic load condition. Moreover, PRED performs tuning of parameter q[AC] according to the traffic load condition. Therefore, it can guarantee the QoS requirements of priority application under changing traffic load. Furthermore, even there are a lot of collisions occurred under heavy traffic load, PRED can support higher overall throughput.

There are numerous packet scheduling algorithms in wired networks. Besides the usage of RED, we still could survey others queuing algorithm. On the other hand, RED offers some control parameters, $\max_{th}$, $\min_{th}$ and $\max_p$ to tune RED's dynamics according to requirements. However, the impact of the choice of values of individual parameter on the queue's performance is dependent on the values of the other. Thus a judicious choice of parameter values is complicated. Works investigating this issue are numerous [12-15]. Therefore, we may joint these packet scheduling algorithms with IEEE 802.11e EDCF in the future.

Acknowledgment. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Y. Xiao, QoS guarantee and provisioning at the contention-based wireless MAC layer in the IEEE 802.11e wireless LANs, *IEEE Wireless Communications Magazine*, vol.13, pp.14-21, 2006.

[2] D. J. Deng and R. S. Chang, A priority scheme for IEEE 802.11 DCF access method, *IEICE Trans. Communications*, vol.E82-B, no.1, pp.96-102, 1999.

[3] I. Aad and C. Castelluccia, Differentiation mechanisms for IEEE 802.11, *Proc. of the IEEE Information Communications (INFOCOM)*, 2001.

[4] A. Veres, A. T. Campbell, M. Barry and L. H. Sun, Supporting service differentiation in wireless packet networks using distributed control, *IEEE J-SAC*, vol.19, no.10, pp.2081-2093, 2001.

[5] E. Ziouva and T. Antonakopoulos, CSMA/CA performance under high traffic conditions: Throughput and delay analysis, *Computer Communications*, vol.25, pp.313-321, 2002.

[6] J.-H. Wen and C.-E. Weng, Performance improvement of IEEE 802.11e to support QoS in saturated conditions, *Proc. of International Conference on Communications and Networking in China (CHINACOM)*, 2009.

[7] Y. Yan and C. Pan, An improved analytical model for IEEE802.11e enhanced distributed channel access, *Proc. of International Symposium on Information Technology Convergence (ISITC)*, 2007.

[8] W. K. Lai, C.-S. Shieh and C.-S. Jiang, Adaption of HCCA/EDCA ratio in IEEE 802.11e for improved system performance, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4177-4188, 2009.

[9] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking*, vol.1, pp.397-413, 1993.

[10] C. Wang and B. Li, A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF, *IEEE Transactions on Vehicular Technology*, vol.53, pp.1234-1246, 2004.

[11] IEEE-SA Standards Board, *IEEE Standard for Message Sets for Vehicle/Roadside Communications*, 1999.

[12] K. Chandrayana, B. Sikdar and S. Kalyanaraman, Scalable configuration of RED queue parameters, *Proc. of IEEE Workshop on High Performance Switching and Routing*, pp.185-189, 2001.

[13] T. Ye and S. Kalyanaraman, Adaptive tuning of RED using on-line simulation, *Proc. of IEEE Global Telecommunications Conference*, vol.3, pp.2210-2214, 2002.

[14] W. C. Feng, D. D. Kandlur, D. Saha and K. G. Shin, A self-configuring RED gateway, *Proc. of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.3, pp.1320-1328, 1999.

[15] T. Bonald and M. May, Drop behavior of RED for bursty and smooth traffic, *Proc. of IEEE International Workshop on Quality of Service*, pp.257-259, 1999.