Multirate Throughput Optimization With Fairness Constraints in Wireless Local Area Networks

Yu-Liang Kuo, Kun-Wei Lai, Frank Yeong-Sung Lin, Yean-Fu Wen, *Associate Member, IEEE*,
Eric Hsiao-Kuang Wu, *Member, IEEE*, and Gen-Huey Chen

***Abstract*—In 802.11-based wireless local area networks (WLANs), it is difficult to simultaneously attain both high throughput and fairness for multirate traffic. There is a performance anomaly when there are stations whose data rates are much lower than the other stations, in which the aggregate throughput of the high-rate stations drastically degrades. The problem of maximizing the total throughput while maintaining time fairness among the competing stations was studied previously by the same authors. However, our previous solution sacrificed the throughput of low-rate stations. In this paper, we extend our previous work by solving the same optimization problem while maintaining both time fairness and throughput fairness. The optimization problem is formulated as a mixed-integer nonlinear programming problem. The two fairness constraints are maintained by means of changing the channel access probability and transmission time among the competing stations, which can be realized by adjusting their minimum contention window sizes and medium access control (MAC) frame sizes, respectively. A penalty function accompanied with a gradient-based approach is used to solve the problem, and its effectiveness is verified by computational experiments. The proposed solution is also compared with our previous solution in terms of convergence speed and total throughput.**

***Index Terms*—Fairness, IEEE 802.11, multirate, optimization, penalty function.**

I. Introduction

In THE past few years, IEEE 802.11 [1] has widely been used in many hot spots and has become the *de facto* wireless local area network (WLAN) standard. The IEEE 802.11 standard specifies multiple modulation types to react against different channel conditions. Robust codes with more encoded bits (i.e., with a lower data rate) are transmitted to preserve their bit error rate below a specific threshold when the channel quality worsens. Hence, different modulation types are used to accommodate the tradeoff between the data rate and the bit error rate in different fading environments.

TABLE I
Performance Anomaly

	Data Rate (Mbps)		Throughput (Mbps)	
	station A		station B	
scenario 1	11	11	3.09	3.36
scenario 2	1	11	0.76	0.73

In [2], a performance anomaly was theoretically analyzed when a multirate traffic was present in IEEE 802.11b WLANs [3]. In the performance anomaly, the aggregate throughput of the stations with higher data rates dramatically degrades to the same level as that of the stations with lower data rates. Since the basic carrier-sense multiple access with collision avoidance (CSMA/CA) channel access method guarantees that the long-term channel access probabilities of all stations (with different data rates) are equal, the low-rate stations have more long-term channel occupancy time than the high-rate stations. If one low-rate station captures the channel, it will last for a long time and hence penalizes the aggregate throughput of the high-rate stations.

Table I shows two scenarios that were simulated with the ns-2 simulator [4]. In scenario 1, there are two stations that transmitted their data at 11 Mb/s. In scenario 2, one station transmitted its data at 11 Mb/s, and the other station transmitted its data at 1 Mb/s. In both scenarios, the traffic loads of the two stations were saturated, i.e., their queues always had packets ready to transmit, and their frame sizes were the same. Observe that in scenario 2, the throughput of station A was almost the same as that of station B. There was a performance anomaly because station B transmitted data at 11 Mb/s and station A transmitted data only at 1 Mb/s. This performance anomaly will also happen when the backward compatibility is supported in the 802.11 series of products. For example, IEEE 802.11g [3] is backward compatible with IEEE 802.11b [5]. The stations that use the IEEE 802.11g protocols transmit their data at higher data rates than the stations that use the IEEE 802.11b protocols. When they exist in the same network, the performance anomaly happens.

In addition, Fig. 1 shows the channel status over time when there are one low-rate station T_A and one high-rate station T_B. The CSMA/CA protocol guarantees that both high-rate and low-rate stations have the same long-term channel access

Manuscript received June 11, 2006; revised March 17, 2008, July 26, 2008, and September 13, 2008. First published November 25, 2008; current version published May 11, 2009. The review of this paper was coordinated by Prof. X. Shen.

Y.-L. Kuo is with Ruckus Wireless Inc., Taipei 114, Taiwan (e-mail: lkuo@ruckuswireless.com).

K.-W. Lai is with Foxconn International Holdings, Taipei 236, Taiwan (e-mail: terrylai@fihtdc.com).

F. Y.-S. Lin is with the Department of Information Management, National Taiwan University, Taipei 106, Taiwan (e-mail: yslin@im.csie.ntu.edu.tw).

Y.-F. Wen is with the Department of Management Information Systems, National Chiayi University, Chiayi 600, Taiwan (e-mail: yeanfu@mail.ncyu.edu.tw).

E. H.-K. Wu is with the Department of Computer Science and Information Engineering, National Central University, Taoyuan 320, Taiwan (e-mail: yslin@im.csie.ntu.edu.tw).

G.-H. Chen is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: yslin@im.csie.ntu.edu.tw).

Digital Object Identifier 10.1109/TVT.2008.2009988

0018-9545/$25.00 © 2009 IEEE

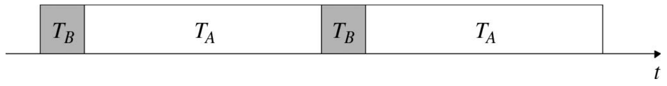


Fig. 1. Channel status for one low-rate station T_A and one high-rate station T_B.

probability. In the long run, the two stations seize the channel alternately. As observed from Fig. 1, the performance anomaly arises because the low-rate station will capture the channel for a long time whenever it seizes the channel. This motivates us to adjust the channel occupancy time for the stations.

It was shown in [2] that the performance anomaly can result in a considerable performance degradation. To avoid the performance anomaly, some approaches were proposed in the literature. In [6], a modified medium access control (MAC), which was named TF-802.11a protocol, was proposed, in which the behavior of the modified MAC was analyzed, and the analytical results were used to determine the average backoff size of the stations. It was suggested in [7] and [8] to adjust the MAC frame sizes for the stations according to their transmission data rates, which could be achieved by controlling the maximum transmission unit (MTU). The implementation issues for controlling the MTU were also addressed in [7].

In [9], a method was proposed, which was named idle sense, in which each station observed the average number of idle slots between the transmission attempts so as to dynamically adjust its contention window size. In [10], the minimum contention window sizes of the stations were assigned, which were inversely proportional to their transmission data rates. In [11]–[13], some approaches were suggested to avoid the performance anomaly by utilizing queuing theory and flow control mechanisms. Although the works previously mentioned could maintain time fairness, they could not guarantee the maximal total throughput. In the rest of this paper, when fairness is referred, it means a criterion for measuring the distribution of the available medium resources, e.g., channel occupancy time and throughput, among competing stations.

In [14], the problem of maximizing the total throughput while maintaining time fairness among the competing stations was studied. However, the solution of [14] sacrificed the throughput of low-rate stations. In this paper, we intend to maximize the total throughput while maintaining both time fairness and throughput fairness. The problem is formulated as a mixed-integer nonlinear programming problem, and a solution that is based on a penalty function accompanied with a gradient-based approach is proposed [15]. The effectiveness of the solution is verified by computational experiments. In addition, a performance comparison is made between the proposed solution and the solution of [14]. Useful guidelines are also provided for regulating the parameters needed for the proposed solution.

The rest of this paper is organized as follows. In Section II, the optimization problem is formulated. In Section III, a penalty function accompanied with a gradient-based approach is proposed to solve the problem. In Section IV, computational experiments are made to evaluate the performance of the proposed approach. In Section V, this paper concludes with some remarks.

II. Multirate Throughput Optimization

The system environment that we consider is a single wireless cell that is coordinated by an access point (AP). Each station that intends to transmit packets has to forward its packets to the AP, even if they are destined for a station located within the same cell. In addition, there is no hidden terminal problem. Without loss of generality, we assume that there are r modulation types with different data rates in the system, where $r \geq 1$. That is, each station can use the set $R = \{R_1, R_2, \ldots, R_r\}$ of data rates to transmit its data. The stations are categorized into different traffic classes according to their data rates. The stations with data rate R_k collectively form the traffic class k, where $1 \leq k \leq r$. We use n_k to denote the number of stations in the traffic class k. In addition, we let $n = \sum_{i=1}^{r} n_i$. A station is referred to as a class k station if it belongs to the traffic class k. The packets that are transmitted by class k stations are referred to as class k packets. We assume that each class k packet has length L_k, and that each class k station has the minimum contention window size W_k.

To quantify the time fairness among the different traffic classes, a time fairness index (TFI), which is obtained according to the definition of [16], is shown as

$$\text{TFI} = \frac{\left(\sum\limits_{k=1}^{r} n_k f_k\right)^2}{n \times \left(\sum\limits_{k=1}^{r} n_k f_k^2\right)}$$

where f_k is the long-term channel occupancy time for the class k stations. The computation of f_k can be found in [14]. On the other hand, let ρ_k be the aggregate throughput of all the class k stations. We compute ρ_k based on a renewal reward process [17] (refer to [14] for the details). Both f_k and ρ_k are evaluated as two functions of $W_1, \ldots, W_r, L_1, \ldots, L_r$. Similarly, there is a throughput fairness index (TPFI), which is given as

$$\text{TPFI} = \frac{\left(\sum\limits_{k=1}^{r} n_k(\rho_k/n_k)\right)^2}{n \times \left(\sum\limits_{k=1}^{r} n_k(\rho_k/n_k)^2\right)} = \frac{\left(\sum\limits_{k=1}^{r} \rho_k\right)^2}{n \times \left(\sum\limits_{k=1}^{r} \rho_k^2/n_k\right)}.$$

The performance anomaly arises when the channel occupancy time for the low-rate stations is much more than the channel occupancy time for the high-rate stations. Hence, the work in [14] maximized the total throughput by adjusting the channel occupancy time among the different traffic classes, where the problem was formulated as a mixed-integer nonlinear programming problem as

$$P' = \text{maximize} \sum_{k=1}^{r} \rho_k$$

subject to

$$\begin{aligned}
&\text{TFI} = a, && a \in [0,1]\\
&L_{\min} \le L_k \le L_{\max}, && k = 1,2,\ldots,r\\
&W_{\min} \le W_k \le W_{\max}, && k = 1,2,\ldots,r\\
&L_k \text{ and} W_k \text{ are integers}, && k = 1,2,\ldots,r.
\end{aligned}$$

The objective is to maximize the total throughput of all the stations. The first constraint requires that the TFI should maintain a fixed value a. The second constraint (third constraint) requires that the length of each class k packet (the minimum contention window size of each class k station) should be bounded within a range from $L_{\min}$ to $L_{\max}$ (from $W_{\min}$ to $W_{\max}$).

When the channel time occupied by each station is equal (i.e., a time-fair situation), we have $f_i = 1/n$ for all $1 \le i \le r$, and hence, TFI $= 1$. On the other hand, suppose that the traffic class k has the fewest stations among all the traffic classes. When the channel time is exclusively occupied by the class k stations (i.e., a time-unfair situation), we have $f_k = 1/n_k$, because each class k station has the same opportunity to capture the channel, and $f_i = 0$ for all $i \ne k$. Hence, TFI $= n_k / \sum_{i=1}^{r} n_i$.

In the foregoing problem formulation, only the time fairness is considered, and as a consequence, the total throughput of the low-rate stations will be much smaller than the total throughput of the high-rate stations. The following problem formulation, also mixed-integer nonlinear programming, can also maintain the throughput fairness:

$$P = \text{maximize} \sum_{k=1}^{r} \rho_k$$

subject to

$$\begin{aligned}
&\text{TFI} = a, && a \in [0,1]\\
&\text{TPFI} = b, && b \in [0,1]\\
&L_{\min} \le L_k \le L_{\max}, && k = 1,2,\ldots,r\\
&W_{\min} \le W_k \le W_{\max}, && k = 1,2,\ldots,r\\
&L_k \text{ and} W_k \text{ are integers}, && k = 1,2,\ldots,r.
\end{aligned}$$

The second constraint, i.e., TPFI $= b$, is added to guarantee throughput fairness among the different traffic classes, where b is a constant.

III. Penalty Function and a Gradient-Based Approach

The problem of computing P' was solved in [14] by using a penalty function accompanied with a gradient-based approach. In this section, the problem of computing P is similarly solved with a modified penalty function, where the objective function is changed to

$$Q = \sum_{k=1}^{r} \rho_k - \mu \times \left[(a - \text{TFI})^2 + (b - \text{TPFI})^2\right]$$

where μ is called the *penalty multiplier*, and $\mu \times [(a - \text{TFI})^2 + (b - \text{TPFI})^2]$ is called the *penalty function* [15]. A penalty algorithm is proposed whose execution will invoke a gradient-based algorithm to solve a relaxed problem as

$$Q_{\max} = \text{maximize } Q$$

subject to

$$\begin{aligned}
&L_{\min} \le L_k \le L_{\max}, && k = 1,2,\ldots,r\\
&W_{\min} \le W_k \le W_{\max}, && k = 1,2,\ldots,r.
\end{aligned}$$

Both algorithms are hereinafter elaborated, where Q is represented by $g(W_1,\ldots,W_r,L_1,\ldots,L_r)$ (recall that ρ_k, TFI, and TPFI are evaluated as functions of $W_1,\ldots,W_r$, $L_1,\ldots,L_r$).

```
Penalty Algorithm
  Step 1: Initialization.
    1) $t \leftarrow 0$.
    2) Set an initial penalty multiplier $\mu_0 > 0$.
    3) Set an initial point $x_{(0)} = (x_1, x_2, \ldots, x_{2r})$, where
       $W_{\min} \le x_i \le W_{\max}$ and $L_{\min} \le x_{i+r} \le L_{\max}$ for
       $1 \le i \le r$.
    4) Set an escalation factor $\beta > 1$.
  Step 2: Relaxed Problem Optimization.
    1) Beginning from $x_{(t)}$, solve the relaxed problem with
       $\mu = \mu_t$ to produce the point $x_{(t+1)}$ by invoking the
       gradient-based algorithm.
  Step 3: Stopping.
    1) If TFI is sufficiently close to $a$ and TPFI is suffi-
       ciently close to $b$, then return $x_{(t+1)}$.
  Step 4: Advance.
    1) $\mu_{t+1} \leftarrow \beta \times \mu_t$.
    2) $i \leftarrow i + 1$.
    3) Go to Step 2.
```

```
Gradient-Based Algorithm
  Step 1: Initialization.
    1) $i \leftarrow 0$.
    2) $y_{(i)} \leftarrow x_{(t)}$.
    3) Set a feasibility tolerance $\varepsilon > 0$ and a threshold
       $max\_throughput > 0$.
  Step 2: Step Size.
    1) Set a step size $\lambda_i > 0$.
  Step 3: Gradient.
    1) Calculate the gradient
       $\nabla g(y_{(i)}) = \left[\frac{\partial g(y_{(i)})}{\partial W_1}, \ldots, \frac{\partial g(y_{(i)})}{\partial W_r}, \frac{\partial g(y_{(i)})}{\partial L_1}, \ldots, \frac{\partial g(y_{(i)})}{\partial L_r}\right]$.
  Step 4: Stationary Point.
    1) If
       $\max\left\{\left|\lambda_i \times \frac{\partial g(y_{(i)})}{\partial W_1}\right|, \ldots, \left|\lambda_i \times \frac{\partial g(y_{(i)})}{\partial W_r}\right| \times \left|\lambda_i \times \frac{\partial g(y_{(i)})}{\partial L_1}\right|, \ldots, \left|\lambda_i \times \frac{\partial g(y_{(i)})}{\partial L_r}\right|\right\}$
       or $g(y_{(i)}) > max\_throughput$, then $\{x_{(t+1)} \leftarrow y_{(i)}$;
       return $x_{(t+1)}\}$.
```

TABLE II
VALIDATION OF ANALYSIS AND SIMULATION RESULTS

(n_1, n_2, n_3, n_4)	$(W_1, W_2, W_3, W_4, L_1, L_2, L_3, L_4)$	analysis total throughput (Mbps)	simulation total throughput (Mbps)
(1, 1, 1, 1)	(32, 32, 32, 32, 1024, 1024, 1024, 1024)	1.6734	1.6428
(2, 2, 2, 2)	(64, 64, 64, 64, 512, 512, 512, 512)	1.3362	1.3021
(4, 4, 4, 4)	(256, 128, 64, 32, 256, 256, 512, 512)	2.0074	1.9992
(8, 8, 8, 8)	(512, 256, 128, 64, 512, 512, 512, 512)	1.9289	1.9018
(10, 10, 10, 10)	(256, 256, 64, 64, 256, 256, 256, 1024)	2.3096	2.2988

```
Step 5: Direction.
  1) $\Delta y_{(i)} \leftarrow \nabla g(y_{(i)})$.
Step 6: New Point.
  1) $y_{(i+1)} \leftarrow y_{(i)} + \lambda_i \times \Delta y_{(i)}$.
  2) If $g(y_{(i+1)}) < g(y_{(i)})$, then $\{\lambda_i \leftarrow \lambda_i/2$; go to Step 3$\}$.
Step 7: Advance.
  1) $i \leftarrow i + 1$.
  2) Go to Step 2.
```

The penalty algorithm finds an approximate solution to the problem of computing P by iteratively invoking the gradient-based algorithm. Given a penalty multiplier μ_t and a point $x_{(t)}$, the gradient-based algorithm produces a new point $x_{(t+1)}$. The penalty multipliers used start with a small initial value, i.e., μ_0, and grow with each iteration. The point $x_{(t)}$ consists of $2r$ coordinates $x_1, x_2, \ldots, x_{2r}$ that are taken as tentative values for $W_1, \ldots, W_r, L_1, \ldots, L_r$, respectively. The obtained point $x_{(t+1)}$ is a feasible solution to the relaxed problem with $\mu = \mu_t$. The finally obtained point can guarantee that the resulting TFI is sufficiently close to a, and the resulting TPFI is sufficiently close to b.

Observe the objective function of the relaxed problem again. When the value of μ increases, the value of $[(a - \text{TFI})^2 + (b - \text{TPFI})^2]$ will decrease so as to obtain a large objective value (i.e., $Q_{\max}$). The principle of the penalty algorithm is to gradually increase the value of μ so that the solution, i.e., $x_{(t+1)}$, can converge to a point in which the resulting TFI and TPFI are sufficiently close to a and b, respectively. Usually, when the penalty algorithm starts with a large initial value of μ, the obtained objective value is small, although it can quickly converge to a point with TFI $\approx a$ and TPFI $\approx b$. This is why the penalty algorithm is iterative and starts with a small initial value of μ. The escalation factor β is used to increase the value of μ in each iteration.

On the other hand, given μ_t and $x_{(t)}$, the gradient-based algorithm performs an iterative binary search to solve the relaxed problem. Initially, set $y_{(0)}$ to $x_{(t)}$ and choose a large step size λ_0. At each iteration i, the vector $\nabla g(y_{(i)})$, which is called the *gradient* (or *direction*) of ascent of $g(W_1, \ldots, W_r, L_1, \ldots, L_r)$ at $y_{(i)}$ [15], is calculated. The execution will terminate and return $x_{(t+1)}$ if a locally maximal point is found, or $g(y_{(i)}) > max\_throughput$ (i.e., the if-condition of Step 4 is satisfied), where $x_{(t+1)}$ is set to $y_{(i)}$. Otherwise, a new point $y_{(i+1)}$ is set to $y_{(i)} + \lambda_i \times \nabla g(y_{(i)})$. If the resulting objective value of $y_{(i+1)}$ is smaller than the resulting objective value of $y_{(i)}$, i.e., if $g(y_{(i+1)}) < g(y_{(i)})$, then a binary search (i.e., $\lambda_i \leftarrow \lambda_i/2$) is performed for a locally maximal point. The iteration $i + 1$ is initiated when $g(y_{(i+1)}) \geq g(y_{(i)})$.

IV. COMPUTATIONAL EXPERIMENTS

The experiment environment is described as follows. The IEEE 802.11b standard [3] was adopted as the physical layer, which specifies four modulation types with data rates of 1, 2, 5.5, and 11 Mb/s, respectively. The two-way handshaking mechanism (i.e., DATA-ACK) was used for the data transmission between stations. There were four stations whose data rates were 1, 2, 5.5, and 11 Mb/s, respectively, i.e., $(n_1, n_2, n_3, n_4) = (1, 1, 1, 1)$ and $(R_1, R_2, R_3, R_4) = (1, 2, 5.5, 11)$. We set $L_{\min} = 41$, $L_{\max} = 2304$, $W_{\min} = 32$, and $W_{\max} = 1024$. All of the computational experiments were carried out on a PC equipped with an Intel Pentium D CPU 3.4 GHz and 512-MB RAM.

To validate the total throughput obtained by the analytical model, ns-2 was used. Table II shows the total throughputs, which were obtained by both analysis and simulation, for the five randomly generated instances. It can be observed that both of the results obtained by analysis and simulation are close.

Table III(a) shows the penalty form for the problem of computing P, where $\mu_0 = 1$, $\beta = 8$, $\varepsilon = 10^{-9}$, $a = 0.7$, $b = 0.7$, $max\_throughput = 2.5$, and $x_{(0)} = (32, 32, 32, 32, 41, 41, 41, 41)$ are assumed. When $t = 0$, the penalty multiplier is set to μ_0, and the gradient-based algorithm generates a solution with TFI $= 0.5183$ and TPFI $= 0.4570$, which violates the constraints of TFI $= 0.7$ and TPFI $= 0.7$. Then, the penalty multiplier increases by a factor of $\beta = 8$, i.e., $\mu_1 = \beta \times \mu_0$, and the iteration of $t = 1$ starts. The execution continues until both TFI and TPFI sufficiently approach 0.7. Finally, when $t = 4$, the execution terminates with the total throughput (i.e., the value of P) 2.1490 Mb/s. The total throughput decreases when t increases (except $t = 0$). Table III(b) shows the penalty form for the problem of computing P' with the same values of μ_0, β, a, and $x_{(0)}$, but $max\_throughput = 4$. Similarly, the total throughput (i.e., the value of P') decreases when t increases (except $t = 0$).

The experiments in Fig. 2 investigate the effect of the escalation factor β on the two problems of computing P and computing P', where $\mu_0 = 1$, $\varepsilon = 10^{-9}$, $a = 0.7$, $b = 0.7$, and $x_{(0)} = (32, 32, 32, 32, 41, 41, 41, 41)$ are assumed. We set $max\_throughput = 2.5$ and $max\_throughput = 4$, respectively, for the problem of computing P and the problem of computing P'. As shown in Fig. 2(a), when the value of β increases, the time requirement decreases, where the time requirement is the average time required to complete the computation. The

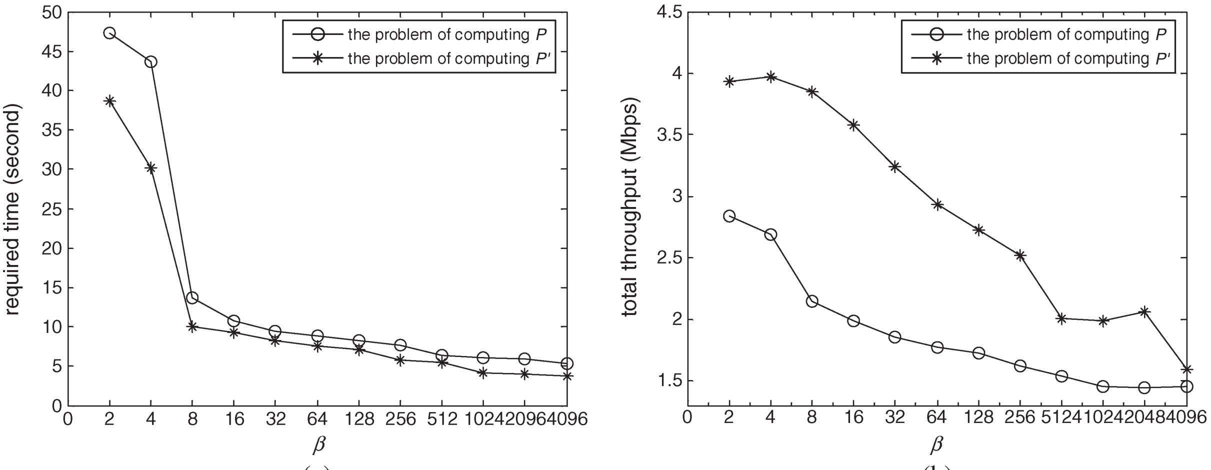
TABLE III
PENALTY FORMS FOR (a) THE PROBLEM OF COMPUTING P AND (b) THE PROBLEM OF COMPUTING P'

(a)

t	TFI	TPFI	total throughput (Mbps)
0	0.5183	0.4570	2.7460
1	0.6982	0.4466	2.8108
2	0.7020	0.6985	2.1514
3	0.6995	0.6997	2.1493
4	0.7000	0.7000	2.1490

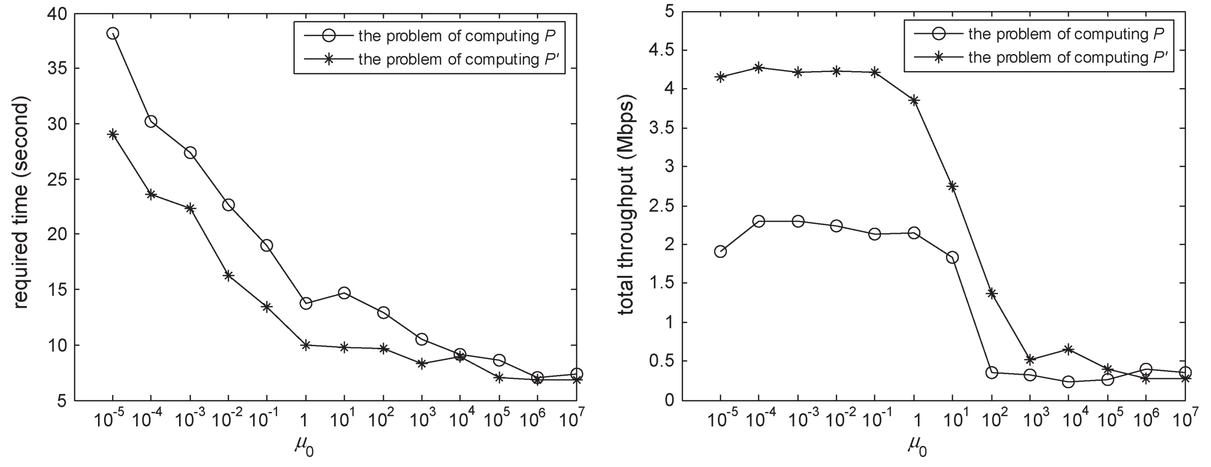
(b)

t	TFI	total throughput (Mbps)
0	0.9785	2.2751
1	0.4724	4.9681
2	0.4710	4.3134
3	0.6760	4.1761
4	0.6960	4.0703
5	0.6994	3.9121
6	0.6998	3.8832
7	0.7000	3.8543



(a) (b)

Fig. 2. Effect of β on (a) the convergence speed and (b) the total throughput.



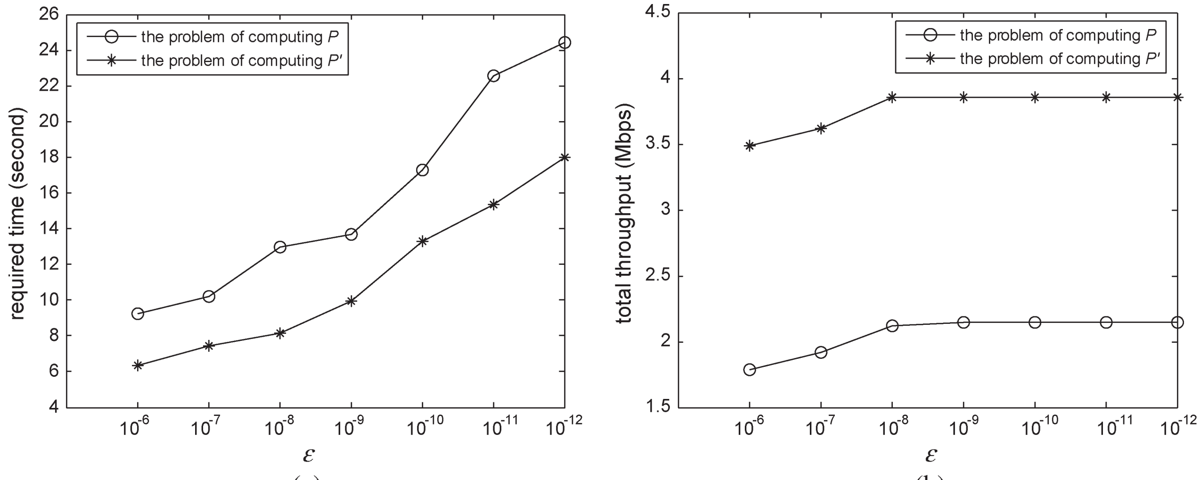
(a) (b)

Fig. 3. Effect of μ_0 on (a) the convergence speed and (b) the total throughput.

curves for both problems have the same tendency, but the problem of computing P consumes more time than the problem of computing P'. The reason is that the problem of computing P has one more constraint than the problem of computing P', which incurs more computation time.

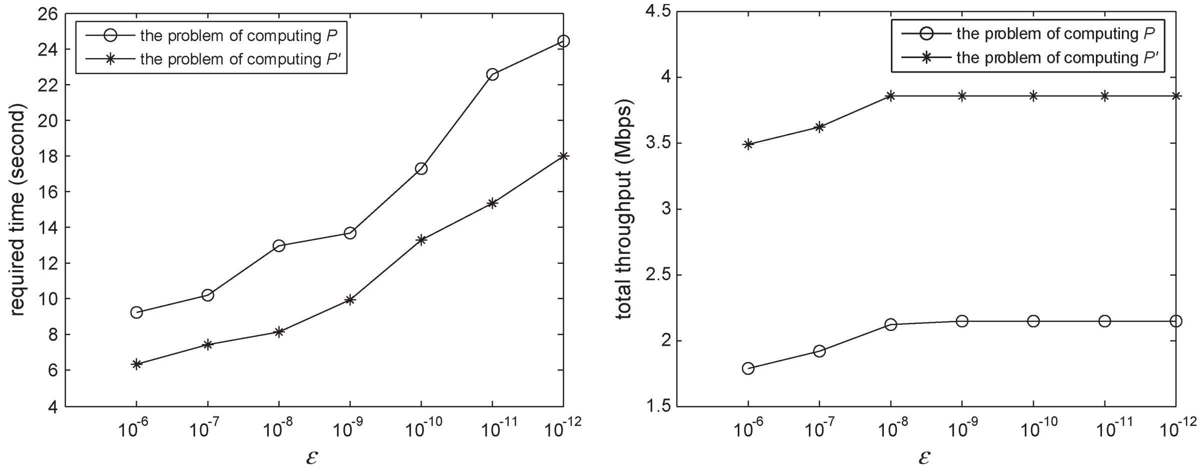
Fig. 2(b) exhibits that the total throughput decreases when the value of β increases. The problem of computing P has a lower total throughput than the problem of computing P' as a consequence that it needs to maintain the throughput fairness. It is suggested to choose a moderate value of β. A small β will induce a slow convergence. Although a large β can induce a fast convergence, it will induce a small total throughput at the same time.

The experiments in Fig. 3 investigate the effect of the initial penalty multiplier μ_0 on the two problems of computing P and P', where $\beta = 8$, $\varepsilon = 10^{-9}$, $a = 0.7$, $b = 0.7$, and



(a) (b)

Fig. 4. Effect of ε on (a) the convergence speed and (b) the total throughput.



(a) (b)

Fig. 5. Effect of the number of stations on (a) the convergence speed and (b) the total throughput.

$x_{(0)} = (32, 32, 32, 32, 41, 41, 41, 41)$ are assumed. We set $max\_throughput = 2.5$ and $max\_throughput = 4$, respectively, for the problem of computing P and P'. The curves for both problems have the same tendency. Fig. 3(a) shows that the convergence becomes faster when the value of μ_0 increases. Fig. 3(b) shows that the total throughput dramatically drops when the value of μ_0 increases from 10 to 100. The reason is hereinafter explained.

Since IEEE 802.11b has the maximum data rate of 11 Mb/s, the maximum value of P for $\sum_{k=1}^{4} \rho_k$ is 11. At the first iteration (i.e., $t = 0$) of the gradient-based algorithm, the objective value of the relaxed problem is evaluated as $Q_{\max}$ for $\sum_{k=1}^{4} \rho_k - \mu_0 \times [(a - \text{TFI})^2 + (b - \text{TPFI})^2]$. When the value of μ_0 is as large as 100 (much larger than $\sum_{k=1}^{4} \rho_k$), the gradient-based algorithm will produce a solution whose resulting TFI and TPFI are very close to 0.7 so as to make the value of $Q_{\max}$ as large as possible. Consequently, the penalty algorithm will quickly converge. The objective value thus obtained is usually small (as described in the second last paragraph of Section III).

The experiments in Fig. 4 investigate the effect of the tolerance ε on the two problems of computing P and P', where $\beta = 8$, $\mu_0 = 1$, $a = 0.7$, $b = 0.7$, and $x_{(0)} = (32, 32, 32, 32, 41, 41, 41, 41)$ are assumed. We set $max\_throughput = 2.5$ and $max\_throughput = 4$, respectively, for the problem of computing P and P'. The curves for both problems have the same tendency. Fig. 4(a) shows that the time requirement increases when the value of ε decreases. The reason is that it takes more computation time for the gradient-based algorithm to obtain a solution close to a stationary point. Fig. 4(b) shows that the total throughput increases when $10^{-9} \le \varepsilon \le 10^{-6}$ and remains stable when $\varepsilon < 10^{-9}$. The reason is that the gradient-based algorithm cannot reach a stationary point (i.e., a locally optimal point) unless the value of ε is sufficiently small (e.g., $\varepsilon < 10^{-9}$).

The experiments in Fig. 5 investigate the effect of the number of stations on the two problems of computing P and P', where $\beta = 8$, $\mu_0 = 1$, $\varepsilon = 10^{-9}$, $a = 0.7$, $b = 0.7$, and $x_{(0)} = (32, 32, 32, 32, 41, 41, 41, 41)$ are assumed. We set $max\_throughput = 2.5$ and $max\_throughput = 4$, respectively,

TABLE IV
TOTAL THROUGHPUTS WITH 16 EXTREME POINTS $x_{(0)}$

initial point								total throughput (Mbps)	
W_1	W_2	W_3	W_4	L_1	L_2	L_3	L_4	problem of computing P	problem of computing P'
32	32	32	32	41	41	41	41	2.1490	3.8543
32	32	32	32	41	41	2304	2304	2.0345	3.7861
32	32	32	32	2304	2304	41	41	1.7989	3.2236
32	32	32	32	2304	2304	2304	2304	2.2190	4.1830
32	32	1024	1024	2304	2304	2304	2304	2.1203	2.4387
32	32	1024	1024	41	41	2304	2304	2.0302	3.9817
32	32	1024	1024	2304	2304	41	41	2..0234	1.7654
32	32	1024	1024	41	41	41	41	2.1342	2.7877
1024	1024	32	32	41	41	41	41	1.8343	4.0869
1024	1024	32	32	41	41	2304	2304	1.9789	3.3256
1024	1024	32	32	2304	2304	41	41	1.6345	4.0193
1024	1024	32	32	2304	2304	2304	2304	2.0355	1.8934
1024	1024	1024	1024	41	41	41	41	2.1914	3.6323
1024	1024	1024	1024	41	41	2304	2304	2.2345	3.2021
1024	1024	1024	1024	2304	2304	41	41	1.6400	2.9670
1024	1024	1024	1024	2304	2304	2304	2304	2.0934	2.7235

TABLE V
PERFORMANCE COMPARISON

	a	b	class 1 (Mbps)	class 2 (Mbps)	class 3 (Mbps)	class 4 (Mbps)	total throughput (Mbps)
P	0.7	0.4	0.0209	0.0256	0.7331	2.8246	3.6042
	0.7	0.5	0.0186	0.0324	1.4814	1.9766	3.5090
	0.7	0.6	0.0319	0.6203	0.6807	1.7928	3.1257
	0.7	0.7	0.3896	0.0965	0.7839	1.1541	2.4241
	0.7	0.8	0.0812	0.5798	0.6942	0.7758	2.1310
	0.7	0.9	0.2034	0.5722	0.6132	0.6320	2.0208
	0.7	1.0	0.3824	0.3898	0.3956	0.3821	1.5499
P'	0.7	N/A	0.0051	0.0348	1.3387	2.8349	4.2135

P: the problem of computing P.
P': the problem of computing P'.

for the problem of computing P and P'. The curves for both problems have the same tendency. We assume that different classes have the same number of stations. For example, 2 in the x-axis represents $(n_1, n_2, n_3, n_4) = (2, 2, 2, 2)$. Fig. 5(a) shows that the time requirement is fairly stable when the number of stations varies. Although different numbers of stations result in different objective functions, they do not affect the convergence speed. Fig. 5(b) shows that the total throughput goes down when the number of stations exceeds 8. The reason is that when the number of stations goes beyond a threshold value (8 in the simulation), the channel utilization is saturated, and so the overheads caused by collision and backoff quickly increase.

Table IV shows the total throughputs for the two problems of computing P and P', respectively, that are obtained for 16 initial points $x_{(0)}$, where $\mu_0 = 1$, $\beta = 8$, $\varepsilon = 10^{-9}$, $a = 0.7$, and $b = 0.7$ are assumed. We set $max\_throughput = 2.5$ and $max\_throughput = 4$, respectively, for the problem of computing P and P'. They are all extreme points; namely, their coordinates are set to extreme values (32 or 1024 for W_1, W_2, W_3, and W_4, and 41 or 2304 for L_1, L_2, L_3, and L_4). There are $2^8 = 256$ extreme points in total, and we randomly choose 16 from them. It is observed that the selection of $x_{(0)}$ has a great effect on the total throughput.

Tables III and IV and Figs. 2–5 together show that the two penalty algorithms for solving the two problems of computing P and P' similarly perform. On the other hand, the purpose of introducing the problem of computing P is to maintain both time fairness and throughput fairness while maximizing the total throughput. The effectiveness is verified in Table V. Table V shows the aggregate throughputs of four traffic classes and the total throughput for the two problems of computing P and P', where $\mu_0 = 1$, $\beta = 8$, $\varepsilon = 10^{-9}$, and $a = 0.7$ are assumed. We set $max\_throughput = 2.5$ and $max\_throughput = 4$, respectively, for the problem of computing P and P'. Each total throughput is the maximum of the total throughputs induced by the 256 extreme points. As described in [14], the globally optimal point is usually close to some extreme point.

For the problem of computing P, the total throughput decreases when the value of b increases. When $b = 1$, the aggregate throughputs (0.3824, 0.3898, 0.3956, and 0.3821 Mb/s) of the four traffic classes are close, but the total throughput (1.5499 Mb/s) is rather small. For the problem of computing P', although the total throughput (4.2135 Mb/s) is higher, the aggregate throughputs (0.0051, 0.0348, 1.3387, and 2.8349 Mb/s) of the four traffic classes are uneven. The better values of b range from 0.5 to 0.7 for this example. It can be also observed from Table V that when the value of b decreases, the total throughput for the problem of computing P approaches the total throughput for the problem of computing P'. Since a small value of b means a small effect of the throughput fairness, the problem of computing P is more general than the problem of computing P'.

Finally, there are two remarks on the penalty algorithm. One is that the penalty algorithm may generate a fractional solution, which should be rounded to an integral solution. A fractional solution can be rounded to any of at most 2^{2r} possible integral solutions. As shown in [14], the total throughputs induced by the 2^{2r} integral solutions are close to the total throughput induced by the original fractional solution.

The other is that an offline execution of the penalty algorithm can be considered if the computation time is concerned. Given an instance of (n_1, n_2, n_3, n_4) numbers of stations with different data rates, an AP can run the proposed algorithms to obtain the result values of $(W_1, \ldots, W_r, L_1, \ldots, L_r)$ and then save them in a table. It is not necessary for the AP to precompute (and save) the values for all possible instances of (n_1, n_2, n_3, n_4) numbers of stations with different data rates. Instead, the AP only needs to precompute the values for those instances of (n_1, n_2, n_3, n_4) that have higher occurrence probabilities, which can statistically be predicted. Some new protocols such as message exchange and network status feedback are necessary for offline execution.

V. Conclusion

In this paper, we have addressed the performance anomaly when a multirate traffic was presented in IEEE 802.11 WLANs. Stations with different data rates were categorized into different traffic classes. To avoid the performance anomaly, two fairness indices were introduced to quantify the time fairness and the throughput fairness among the different traffic classes. With the two fairness indices, we regulated the channel occupancy time and the aggregate throughput among the different traffic classes by adjusting the minimum contention window sizes and MAC frame sizes.

The problem of maximizing the total throughput subject to time fairness and throughput fairness was formulated as a mixed-integer nonlinear programming problem. A penalty algorithm that is accompanied with a gradient-based algorithm was proposed to solve the optimization problem. Their effectiveness was verified by computational experiments. Some useful guidelines for choosing the parameters are suggested below.

First, choose the escalation factor β from the range [8, 32] to obtain a large objective value while maintaining a moderate convergence speed. Second, choose the initial penalty multiplier μ_0 from the range (0, 10] to avoid a small total throughput. Third, choose the feasibility tolerance ε from the range $(-\infty, 10^{-9}]$ for the gradient-based algorithm to successfully reach a stationary point. Finally, choose as the initial point $x_{(0)}$ the extreme point that induces the maximal total throughput among the 2^{2r} extreme points.

Computational experiments were also made for comparing the problem of maximizing the total throughput subject to time fairness and throughput fairness with the problem of maximizing the total throughput subject to only time fairness. Although the latter induces a higher total throughput than the former, it induces uneven aggregate throughputs for the different traffic classes. In addition, the problem of computing P is more general than the problem of computing P'.

For typical radio-based indoor WLAN service environments, since the portable computing devices (stations) usually leave or join with less mobility (less than the walking speed of WLAN users in most cases), the numbers of stations with different data rates are not acutely changed during a short period of time. Hence, the network can stay comparably static enough to determine a feasible solution.[1]

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11, 1999.

[2] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2003, pp. 836–843.

[3] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE Std. 802.11g/D3.0, 2002.

[4] *The Network Simulator ns-2*. [Online]. Available: http://www.isi.edu/nsnam/ns/

[5] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE Std. 802.11b, 1999.

[6] R. Bruno, M. Conti, E. Gregori, and R. Fantacci, "Throughput vs. temporal fair MAC protocols in multi-rate WLANs: Analysis and performance evaluation," in *Proc. IEEE VTC*, May 2004, vol. 4, pp. 2017–2021.

[7] J. Dunn, M. Neufeld, A. Sheth, D. Grunwald, and J. Bennet, "A practical cross-layer mechanism for fairness in 802.11 networks," *Mobile Netw. Appl.*, vol. 11, no. 1, pp. 37–45, Feb. 2006.

[8] S. H. Yoo, J. H. Choi, J. H. Hwang, and C. Yoo, "Eliminating the performance anomaly of 802.11b," in *Proc. ICN*, Apr. 2005, vol. 3421, pp. 1055–1062.

[9] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs," in *Proc. Appl., Technol., Architectures, Protocols Comput. Commun., SIGCOMM*, 2005, pp. 121–132.

[10] H. Kim, S. Yun, I. Kang, and S. Bahk, "Resolving 802.11 performance anomalies through QoS differentiation," *IEEE Commun. Lett.*, vol. 9, no. 7, pp. 655–657, Jul. 2005.

[11] A. Munaretto, M. Fonseca, K. A. Agha, and G. Pujolle, "Fair time sharing protocol: A solution for IEEE 802.11b hot spots," in *Proc. IFIP/IEEE ICT*, Jul. 2004, vol. 3124, pp. 1261–1266.

[12] G. Tan and J. Guttag, "Time-based fairness improves performance in multi-rate WLANs," in *Proc. USENIX Annu. Tech. Conf.*, Jun. 2004, pp. 269–282.

[13] Y. Wu and S. Fahmy, "A credit-based distributed protocol for long-term fairness in IEEE 802.11 single-hop networks," in *Proc. IEEE Wireless Mobile Comput., Netw. Commun. (WiMob)*, Aug. 2005, vol. 2, pp. 98–105.

[14] Y. L. Kuo, K. W. Lai, F. Y. S. Lin, Y. F. Wen, E. H. K. Wu, and G. H. Chen, "Multi-rate throughput optimization for wireless local area network anomaly problem," in *Proc. IEEE Broadband Netw.*, Oct. 2005, vol. 1, pp. 591–601.

[15] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming*, 2nd ed. Hoboken, NJ: Wiley, 1993.

[16] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Digital Equipment Corp., Hudson, MA, DEC Res. Rep. TR-301, Sep. 1984.

[17] S. M. Ross, *Introduction to Probability Models*, 8th ed. New York: Academic, 2003.

[1] High-mobility cases are not our major concern since they may involve other issues such as fading or handoff.

Yu-Liang Kuo received the B.S. degree in computer science from the National Cheng Chi University, Taipei, Taiwan, in 1999 and the M.S. and Ph.D. degrees in computer science and information engineering from the National Taiwan University (NTU), Taipei, in 2002 and 2007, respectively.

Since 2007, he has been with Ruckus Wireless Inc., Taipei, where he has been responsible for quality-of-service (QoS) assurance for multimedia services in wireless networks. His primary research interests include performance evaluation, QoS assurance of wireless networks, and design and analysis of algorithms.

Kun-Wei Lai received the B.S. degree in management information systems from the National Cheng Chi University, Taipei, Taiwan, in 2002 and the M.S. degree in information management from the National Taiwan University, Taipei, in June 2004.

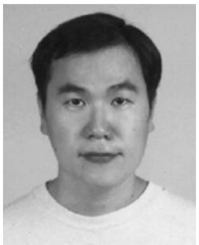
He is currently a Software Engineer with Foxconn International Holdings, Taipei. His current research interests include embedded systems and wireless communication.

Frank Yeong-Sung Lin received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1983 and the Ph.D. degree in electrical engineering from the University of Southern California (USC), Los Angeles, in 1991.

After graduating from USC, he was with Telcordia Technologies (formerly Bell Communications Research, abbreviated as Bellcore) in New Jersey, where he was responsible for developing network planning and capacity management algorithms. In 1994, he was with the faculty of the Electronic Engineering Department, National Taiwan University of Science and Technology, Taipei. Since 1996, he has been with the faculty of the Department of Information Management, National Taiwan University. His research interests include network optimization, network planning, network survivability, performance evaluation, high-speed networks, distributed algorithms, content-based information retrieval/filtering, biometrics, and network/information security.

Yean-Fu Wen (A'07) received the M.S. degree from the National Taiwan University of Science Technology, Taipei, Taiwan, in 1998 and the Ph.D. and Doctoral degrees from the National Taiwan University, Taipei, in July 2007.

Since August 2008, he has been an Assistant Professor with the Department of Management Information Systems, National Chiayi University, Chiayi, Taiwan. His research interests include network planning, resource allocation, performance optimization, and cross-layer technology in next-generation wireless networks.



Eric Hsiao-Kuang Wu (M'98) received the B.S. degree in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, in 1989 and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles, in 1993 and 1997, respectively.

He is currently a Professor of computer science and information engineering with the National Central University, Taoyuan, Taiwan. His primary research interests include wireless networks, mobile computing, and broadband networks. Dr. Wu is a member of the Institute of Information and Computing Machinery.

Gen-Huey Chen received the Ph.D. degree in computer science from the National Tsing Hua University, Hsinchu, Taiwan, in January 1987.

He joined the faculty of the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, in February 1987, where he has been a Professor since August 1992. His current research interests include wireless communication and mobile computing, graph theory and combinatorial optimization, and the design and analysis of algorithms.