

# Achieving proportional delay differentiation in wireless LAN *via* cross-layer scheduling

Yuan Xue<sup>\*†</sup>, Kai Chen and Klara Nahrstedt

*Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin, Urbana, IL 61801, U.S.A.*

## Summary

Providing service differentiation in wireless networks has attracted much attention in recent research. Existing studies so far have focused on the design of differentiated media access algorithms. Some quality of service (QoS) metrics, such as queueing delay cannot be completely addressed by these approaches. Moreover, without a formalized service differentiation goal that quantifies the outcome of differentiation, the performance of most of the approaches fluctuates, especially in short time-scales.

This paper addresses above problems by introducing the concept of proportional service differentiation, to the domain of wireless network and focuses on providing proportional delay differentiation in wireless LANs. Due to the unique characteristic of distributed medium sharing, the scheduling algorithm employed in wireline networks can not be applied directly to the context of wireless LANs. We argue that delay differentiation in wireless LAN can only be achieved through a joint packet scheduling at the network layer and distributed coordination at the media access control (MAC) layer. Therefore, we present a cross-layer waiting time priority (CWTP) scheduling algorithm. CWTP consists of two tiers: an intra-node waiting time priority (WTP) scheduler at the network layer and an inter-node distributed coordination function at the MAC layer. These two tiers coordinate *via* a mapping function, which maps the normalized waiting time at the network layer to the backoff time at the MAC layer. Two mapping schemes, namely linear mapping and piecewise linear mapping, are presented and evaluated in this paper. Extensive simulation results show that the CWTP algorithm can effectively achieve proportional delay differentiation in wireless LANs. Copyright © 2004 John Wiley & Sons, Ltd.

---

KEY WORDS: service differentiation; wireless LAN

---

## 1. Introduction

Wireless LAN has been a popular wireless networking solution due to its low cost and high bandwidth. With the emergence of diversified multimedia applications such as voice over IP, there is a growing need to support

service differentiation and quality of service (QoS) for multimedia real-time traffic in wireless LANs.

Towards this goal, existing studies have focused on the design of differentiated media access protocols in wireless LANs to achieve distributed priority scheduling (e.g. [5,23,10,13,15,14,20]) or fair scheduling

\*Correspondence to: Yuan Xue, Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin, Urbana, IL. 61801, U.S.A.

E-mail: xue@cs.uiuc.edu

Contract/grant sponsor: Motorola Partnerships.

Contract/grant sponsor: ONR MURI NAVY CU; contract/grant number: 37515-6281.

Contract/grant sponsor: NSF EIA; contract/grant number: 99-72884EQ.

(e.g. [19,24]). These approaches can only achieve performance differentiation of media access. Some QoS metrics, such as queueing delay, cannot be completely addressed by these approaches. Thus these approaches are insufficient to meet the diversified performance requirements of multimedia applications. Moreover, without a formalized service differentiation goal that quantifies the outcome of differentiation, the performance of most approaches (e.g. [10,5,14,20]) may be inconsistent and fluctuating, especially for packet delay over a short time-scale. Though some approaches (e.g. [19,24,13,15]) are based on the analytical model of the throughput, they still do not consider delay as a service differentiation metric.

Therefore, a quantitative delay differentiation study is still missing from the existing works on service differentiation in wireless LANs. This paper conducts this study in two steps. First, it introduces a service differentiation model, proportional delay differentiation model [12], to the domain of wireless LAN. In the proportional delay differentiation model, the delay of a service class is proportional to another according to the ratio of their differentiation parameters. Such a service differentiation model specifies a predictive and consistent delay differentiation behavior. The main advantages of this service differentiation model are: (1) it greatly facilitates service class selection for applications and (2) it provides flexible class provisioning and management through the tuning of its differentiation parameters. As a result, an application that requires an absolute delay bound can select an appropriate service class to meet its QoS requirement.

Second, this paper studies the problem of providing proportional delay differentiation in wireless LANs, which is a much more difficult problem than that in wireline networks. In a wireless LAN, wireless nodes share the same wireless channel, and packets waiting to be transmitted may originate from different nodes. At any given time, only one node may access the channel to transmit a packet. Such packet transmission is usually coordinated in a distributed manner at the media access control (MAC) layer (e.g. *via* IEEE 802.11 DCF). As a result of the distributed medium sharing, packet scheduling needs the cooperation among all the nodes. This is in contrast to wireline networks where packets that need to be scheduled originate from the same router, and hence the packet scheduling decision can be made by the route itself only considering its own packets. In this paper, we argue that delay differentiation in wireless LANs can only be achieved through a joint packet scheduling at

the network layer and distributed coordination at the MAC layer. Therefore, we present a cross-layer waiting time priority (CWTP) scheduling algorithm that is able to achieve proportional delay differentiation in wireless LANs. CWTP consists of two tiers: an intra-node waiting time priority scheduling algorithm at the network layer and an inter-node differentiated distributed coordination at the MAC layer. These two tiers coordinate *via* a mapping function, which maps the normalized waiting time at the network layer to the backoff time at the MAC layer. Two mapping schemes, namely linear mapping and piecewise linear mapping, are presented and evaluated in this paper. Extensive simulation results show that our CWTP algorithm can effectively achieve proportional delay differentiation in wireless LANs.

The main contributions of this paper are as follows. First, to the best of our knowledge, this paper is among the first works to study delay characteristics and its differentiation in wireless LANs. Second, compared with the existing works, our work is not limited to the study of MAC layer design, instead it explores the broader problem of how differentiated MAC can affect the performance of the queueing systems in wireless LANs. Further it studies how queueing delay can be controlled *via* coordination between the network and the MAC layer. The novelty of our work is a CWTP scheduling algorithm, which effectively achieves proportional delay differentiation in wireless LANs.

The rest of the paper is organized as follows. Section 2 gives the background of proportional delay differentiation model and IEEE 802.11 MAC layer protocol. Section 3 presents the CWTP scheduling algorithm with two mapping schemes. Section 4 shows the simulation results. Section 5 summarizes the related works. Section 6 concludes the paper.

## 2. Preliminaries

In this section, we briefly review the background knowledge. First, we discuss the proportional delay differentiation model and present the waiting time priority (WTP) scheduling algorithm. Second, we review the architecture of wireless LANs and the IEEE 802.11 MAC layer protocol.

### 2.1. Proportional Delay Differentiation and Waiting Time Priority Scheduling

Proportional service differentiation was first introduced as a per-hop-behavior (PHB) for DiffServ in

wireline networks [11]. It states that certain class performance metrics should be proportional to the differentiation parameters. In particular, if we consider the case of delay differentiation, in a network with  $C$  service classes, the proportional delay differentiation model imposes the following constraints for all pairs of classes.

$$\frac{\bar{d}_i(t, t + \tau)}{\bar{d}_j(t, t + \tau)} = \frac{\delta_j}{\delta_i}, \text{ for all } i \neq j \text{ and } i, j \in \{1, 2, \dots, C\} \quad (1)$$

where,  $\delta_i$  is the service differentiation parameter for class  $i$  and  $\bar{d}_i(t, t + \tau)$  is the average delay for class  $i, (i = 1, 2, \dots, C)$  in the time interval  $(t, t + \tau)$ , where  $\tau$  is the monitoring time scale.

The basic idea of proportional differentiation is that, even though the actual quality level of each class may vary with traffic loads, the quality ratio between classes should remain constant in various time-scales. In addition, such a quality ratio can be controlled by setting the service differentiation parameters, which provides flexible class provisioning and management. Under certain conditions (*i.e.* the network is well provisioned), applications with absolute delay requirements can select appropriate service classes to meet their requirements [6], even though the network offers only relative differentiation.

One of the packet scheduling algorithms that can realize the proportional delay differentiation model in a short time-scale is the WTP scheduler [12]. In this algorithm, a packet is assigned with a weight, which increases proportionally to the packet's waiting time. Service classes with higher differentiation parameters have larger weight-increase factors. The packet with the largest weight is served first in non-preemptive order. Formally, let  $w_p(t)$  be the waiting time of a packet  $p$  of class  $i$  at time  $t$ , we define its normalized waiting time  $\hat{w}_p(t, i)$  at time  $t$  as follows:

$$\hat{w}_p(t, i) = w_p(t) \times \delta_i \quad (2)$$

The normalized waiting time is then used as the weight for scheduling. The packet with the largest weight is then selected by the WTP scheduler for transmission. Formally, at time  $t$  it will transmit the packet  $p$ , which satisfies

$$p = \arg \max_{p \in \mathcal{P}} \hat{w}_p(t, i) \quad (3)$$

where,  $\mathcal{P}$  is the set of backlogged packets. It is shown that WTP scheduler is able to approximate the proportional delay differentiation model in wireline networks under heavy traffic condition [12].

### 2.2. Wireless LAN and IEEE 802.11

A wireless LAN consists of mobile nodes that are within the transmission range of each other. It can operate in a centralized or distributed manner, as shown in Figure 1. In the centralized model, a base station coordinates packet transmission and performs network management tasks. In the distributed model, every one plays the same role and carries the same responsibilities in arbitrating channel access and network management.

IEEE 802.11 MAC layer protocol [2] has been a popular standard for wireless LANs. It has two different access modes, namely distributed coordination function (DCF) and point coordination function (PCF).

IEEE 802.11 DCF is the basic access mechanism. It uses a carrier sense multiple access with collision avoidance (CSMA/CA) algorithm to coordinate the medium access. Before a packet is sent, the node senses the medium. If it is idle for at least a distributed interframe space (DIFS) period of time, the packet is transmitted. Otherwise, a backoff time  $b$  (in time slots) is uniformly chosen from the range  $[0, CW)$ , where  $CW$  is called the contention window. Initially,  $CW$  is set to the value of minimum contention window  $CW_{\min}$ . After the medium has been detected idle for at

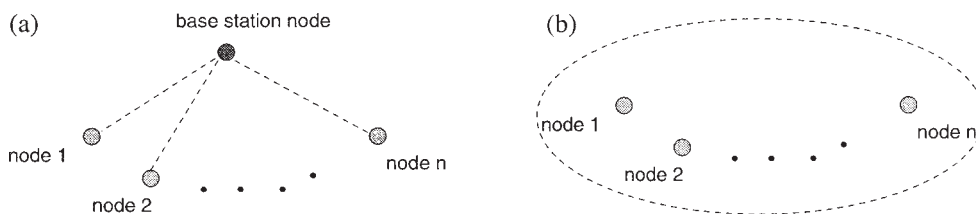


Fig. 1. Models of wireless LANs: (a) Centralized wireless LAN; (b) Distributed wireless LAN.

least a DIFS, the backoff timer is decremented by one for each time slot during which the medium remains idle. If the medium becomes busy during the backoff process, the backoff timer is paused, and is resumed when the medium has been sensed idle for a DIFS again. When the backoff timer reaches zero, the packet is transmitted. Upon detection of a collision, the CW is doubled, up to a maximum value  $CW_{\max} = 2^B CW_{\min}$  where,  $B$  is the maximum number of backoff stage. A new backoff time is then chosen and the backoff procedure starts again. After a successful transmission, the CW is reset to  $CW_{\min}$ . The backoff mechanism is also used after a successful transmission before sending the next packet. IEEE 802.11 DCF can be deployed on both centralized and distributed wireless LANs.

IEEE 802.11 PCF is designed for the centralized wireless LAN model. It provides a contention-free polling-based medium access method and supports time-bounded applications. In this paper, we only consider the problem of providing service differentiation in an IEEE 802.11 DCF-based wireless LAN due to its wide deployments.

### 3. Model and Algorithm

#### 3.1. Proportional Delay Differentiation in Wireless LANs

First, we introduce the proportional service differentiation model into the domain of wireless LAN. As we have discussed in Section 2, proportional service differentiation model was first introduced in the context of wireline networks as a per-hop-behavior. In this scenario, flows from the same router contend for the same wireline link as shown in Figure 2(a). In

wireless LANs, not only do the flows originating from the same node contend with each other, the flows from different nodes also contend for the same wireless channel, as shown in Figure 2(b). To extend the concept of proportional service differentiation to the wireless LAN, the flows among different pairs of nodes are considered. Specifically, our proportional delay differentiation model for wireless LANs states that the relation Equation (1) holds for all flows within the wireless LAN no matter whether they originate from the same node or not.

While proportional delay differentiation has been well studied in wireline networks, it still remains a challenge in wireless LANs. This is because WTP is a centralized scheduling scheme. It needs to know the waiting times of all packets before deciding which one to transmit. This is trivial in a wireline network, where all packets waiting to be scheduled originate from the same router. However, in the wireless LANs, packets that need to be scheduled originate from different source nodes. To acquire information of all packets inside the network, a node has to constantly receive reports from all other nodes, which is prohibitively expensive. Our effort towards this problem is one of the highlights of this work.

#### 3.2. Cross-Layer WTP Scheduling Algorithm

To schedule the packets from different nodes in a uniform and efficient way, we propose to perform cross-layer packet scheduling, where the network layer packet scheduler coordinates with the underlying MAC.

The CWTP scheduling algorithm divides the scheduling task into two parts, which are performed at two layers in the network stack. At the network layer, intra-node scheduling at node  $n$  selects a packet  $p_n^*$

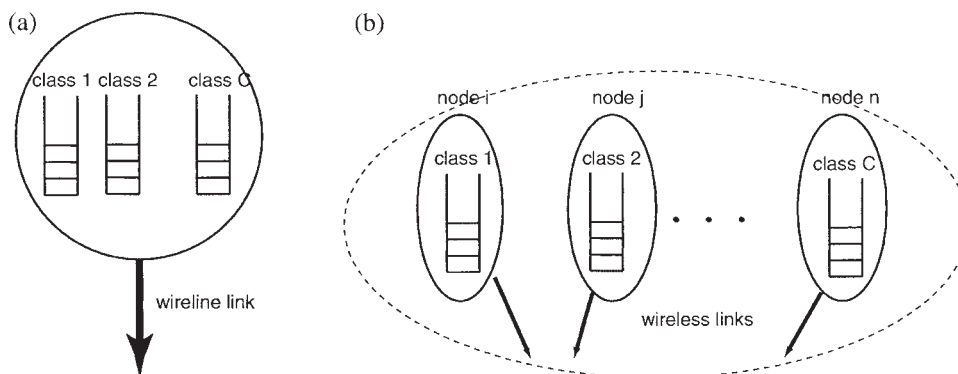


Fig. 2. Service differentiation in different types of networks: (a) Wireline networks; (b) Wireless LANs.

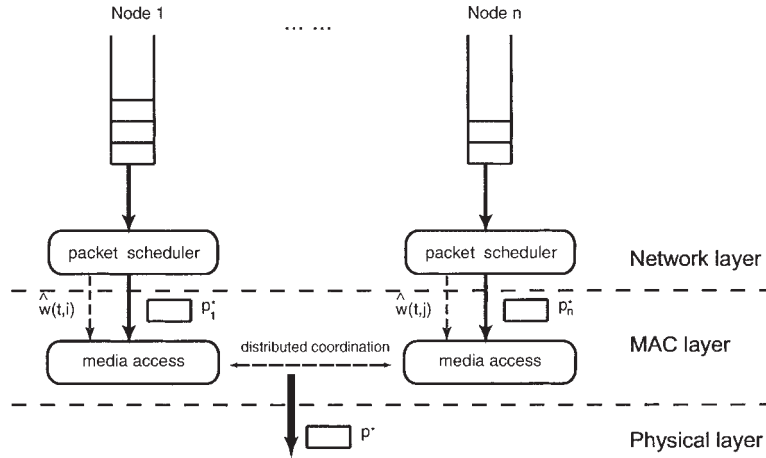


Fig. 3. Cross-layer architecture.

with the longest normalized waiting time, i.e. a packet  $p_n^*$ , which satisfies

$$p_n^* = \arg \max_{p \in \mathcal{P}_n} \hat{w}_p(t, i) \quad (4)$$

where,  $\mathcal{P}_n$  is the set of all backlogged packets at node  $n$ . At the MAC layer, inter-node scheduling selects packet  $p^*$  among  $p_n^*$ , which satisfies

$$p^* = \arg \max_{p_n^* n \in \mathcal{N}} \hat{w}_{p_n^*}(t, i) \quad (5)$$

where,  $\mathcal{N}$  is the set of wireless nodes.

Such an intra- and inter-node scheduling algorithm can fit well in the environment of wireless LANs. In particular, the intra-node scheduling can be implemented *via* network layer packet scheduling at each individual node and the inter-node scheduling can be implemented *via* MAC, which coordinates packet transmissions among nodes. Figure 3 illustrates such a cross-layer scheduling architecture. In this architecture, the packet scheduler at the network layer and the distributed coordination function at the MAC layer are coordinated using normalized packet waiting time  $\hat{w}$  as a cross-layer signal.

At MAC layer, in order to transmit the packet with larger normalized waiting time before the ones with smaller normalized waiting time, we map the normalized waiting time  $\hat{w}$  to the backoff time  $b$  *via* function  $b = \Phi(\hat{w})$ . The cross-layer scheduling algorithm is summarized in Table I. In the following two subsections, we will present two mapping schemes, namely linear mapping and piecewise linear mapping schemes.

### 3.3. Linear Mapping Scheme

In the linear mapping scheme, the normalized waiting time of a packet is mapped to its MAC layer backoff time *via* a linear function. Formally, let us consider a linear function  $\phi(x) : \mathfrak{R}^+ \rightarrow \mathfrak{R}$ ,

$$\phi(x) = \beta - \alpha \times x \quad (6)$$

where,  $\alpha, \beta > 0$  are parameters of this linear function. To ensure it to be a non-negative integer, the backoff time  $b$  (in number of time slot) of a packet with normalized waiting time  $\hat{w}$  is chosen as follows:

$$b = \Phi(\hat{w}) = \lceil [\phi(\hat{w})]^+ \rceil \quad (7)$$

where,  $[x]^+ = \max(0, x)$  and  $\lceil \cdot \rceil$  is the ceiling operation. These two operations round up the value of  $\phi(\hat{w})$  to a non-negative integer.

Table I. Cross-layer waiting time priority scheduling algorithm.

**At network layer** of node  $n$

- 1 Select a packet  $p_n^*$  among all backlogged packets  $\mathcal{P}_n$ , which satisfies  $p_n^* = \arg \max_{p \in \mathcal{P}_n} \hat{w}_p$  where,  $\hat{w}_p = w_p \times \delta_i$  is the normalized waiting time of packet  $p$  from class  $i$ .
- 2 Send the packet  $p_n^*$  to MAC layer with its normalized waiting time  $\hat{w}_p$ .

**At MAC layer** of node  $n$

- 1 Receive the packet  $p_n^*$  and its normalized waiting time  $\hat{w}_p$ .
- 2 Map the normalized waiting time  $\hat{w}_p$  to the backoff time  $b$  for packet  $p_n^* b = \Phi(\hat{w}_p)$ .
- 3 Start the backoff procedure of the distributed coordination function with value  $b$ .

It is obvious that  $\alpha$  and  $\beta$  determines the effectiveness of the mapping function, and thus the performance of the cross-layer scheduling algorithm. To have a better understanding of the impact of these two parameters, we conduct a simulation study in ns-2 [1]. In the experiments, two service classes with differentiation parameters  $\delta_1:\delta_2 = 1:2$  are provided in the wireless LAN. In the first experiment, two flows—each from one service class—are transmitted. In the second experiment, six flows—each service class with three flows—are transmitted. Here, we are interested in two performance metrics: differentiation index and channel utilization. The differentiation index ( $I$ ) is defined as the ratio of the average delay of the two service classes. Formally,

$$I = \frac{\bar{d}_1}{\bar{d}_2} \tag{8}$$

where,  $\bar{d}_i$  is the expected packet delay of service class  $i$ . This metric shows the effectiveness of the service

differentiation—how close the differentiation result matches the differentiation goal. Ideally, in these experiments  $I = 2$ . The channel utilization ( $\rho$ ) is defined as the percentage of the packet transmission time. Formally,

$$\rho = \frac{t_{tx}}{\tau} \tag{9}$$

where,  $\tau$  is the measurement time interval and  $t_{tx}$  is the packet transmission time during this interval. This metric shows the efficiency of the scheduling algorithm—how much channel time is used for packet transmission. Figure 4 plots the results of these two performance metrics. To scale the results, we introduce an index  $I_\alpha$  for  $\alpha$ , which is given as follows.

$$I_\alpha = \frac{10 \times \alpha}{\beta} \tag{10}$$

$I_\alpha$  normalizes the value of  $\alpha$  based on  $\beta$ . It takes a range of  $[0, 10]$  in the Figure 4, which makes the value of  $\alpha$  to be within the range of  $[0, \beta]$ .

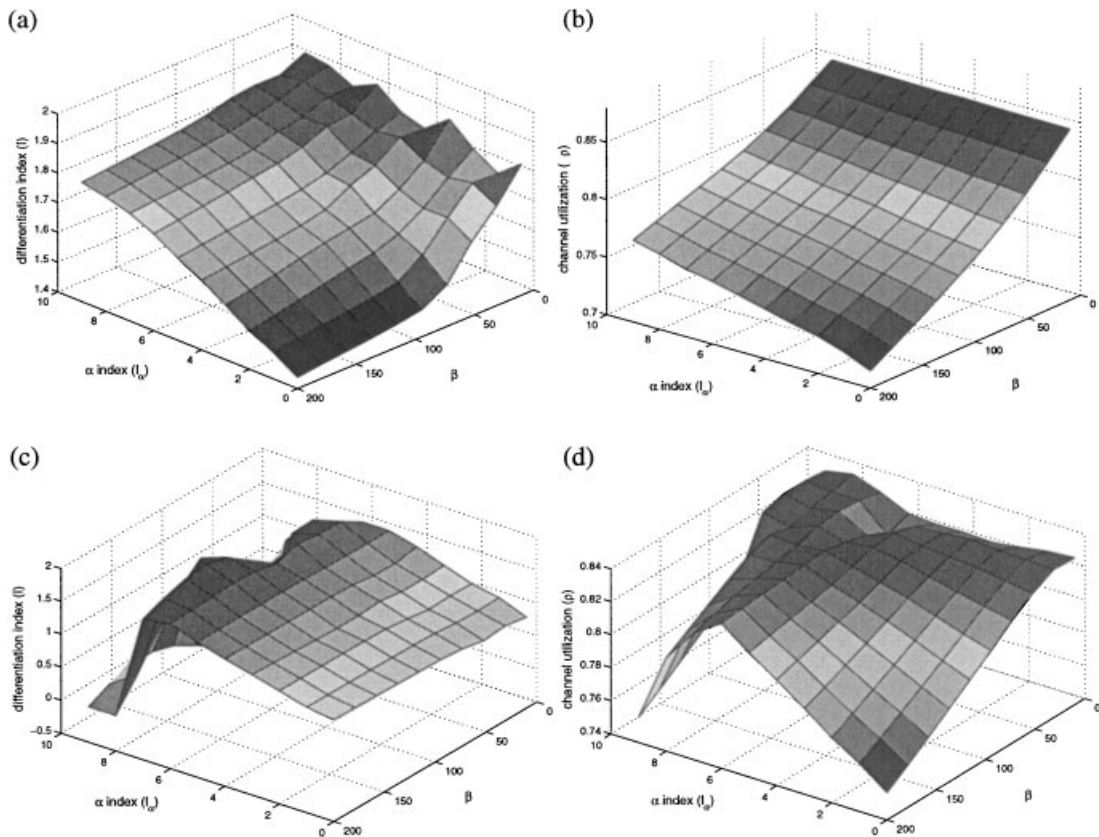


Fig. 4. Impact of  $\alpha$  and  $\beta$  on  $I$  and  $\rho$ : (a) Differentiation index ( $I$ ): 2 flows; (b) Channel utilization ( $\rho$ ): 2 flows; (c) Differentiation index ( $I$ ): 6 flows; (d) Channel utilization ( $\rho$ ): 6 flows.

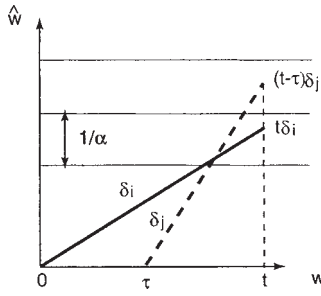


Fig. 5. Impact of  $\alpha$  in linear mapping.

From the results, we have the following observations. First,  $I_\alpha$  greatly affects the result of differentiation index, i.e. the effectiveness of the scheduling algorithm. In the case of two flows, large  $I_\alpha$  leads to higher differentiation ratio. In the case of six flows, large  $I_\alpha$  has higher differentiation ratio when  $I_\alpha < 6$ . When  $I_\alpha$  is increased beyond 6, the differentiation ratio decreases dramatically. The impact of  $\alpha$  is further illustrated in Figure 5 to explain these observations. Function  $\Phi(\cdot)$  is not strictly decreasing due to the ceiling operation. Two packets with different normalized waiting times may be mapped into the same backoff time slot. When such mapping occurs, these two packets are not differentiated and the differentiation index is decreased.  $1/\alpha$  gives the upper bound to the differences between which the two normalized waiting times are mapped into the same backoff time slot. Obviously, a large  $\alpha$  lowers the possibility of such mapping.

On the other hand, when the collision is taken into consideration as in the case of six flows, too large value of  $I_\alpha$  (i.e.  $\alpha/\beta$ ) will result in very small  $b$  for all packets, and thus causing severe collision conditions. This explains the drop of  $I$  when  $I_\alpha > 6$ . Second,  $\beta$  mainly affects the result of channel utilization. In the case of two flows, small value of  $\beta$  increases the channel utilization. In the case of six flows, too small values of  $\beta$  will cause collision and decrease the channel utilization. These results show that the values of  $\alpha$  and  $\beta$  must be appropriately selected to balance

the efficiency and the effectiveness of the scheduling algorithm.

Now the question is, how we can determine the appropriate values of  $\alpha$  and  $\beta$  so that the linear mapping with these two parameters can lead the cross-layer scheduling algorithm to the goal of proportional delay differentiation. Here we present a dynamic tuning algorithm of  $\alpha$  and  $\beta$ . Let  $\overline{cw}$  be the expected value of contention window under IEEE 802.11 DCF without differentiation. The backoff time  $b$  is uniformly chosen from  $[0, \overline{cw})$ . Let  $\hat{w}_{max}$  and  $\hat{w}_{min}$  be the maximum and minimum normalized waiting time respectively. Preferably, the maximum normalized waiting time  $\hat{w}_{max}$  can be mapped to the smallest backoff time (0) for efficient channel utilization; and  $\hat{w}_{min}$  can be mapped to  $\overline{cw}$  for similar contention behavior as IEEE 802.11 without differentiation as shown in Figure 6.

$$\overline{cw} = \beta - \alpha \times \hat{w}_{min} \tag{11}$$

$$0 = \beta - \alpha \times \hat{w}_{max} \tag{12}$$

Solving  $\alpha, \beta$  from Equations (11,12), we have

$$\alpha = \frac{\overline{cw}}{(\hat{w}_{max} - \hat{w}_{min})} \tag{13}$$

$$\beta = \overline{cw} + \alpha \times \hat{w}_{min}$$

Via Equation (13),  $\alpha$  and  $\beta$  can be determined dynamically based on the knowledge of  $\hat{w}_{max}$  and  $\hat{w}_{min}$ . Table II summarizes the dynamic linear mapping scheme.

### 3.4. Piecewise Linear Mapping Scheme

Linear mapping scheme neglects the fact that the distribution of the normalized waiting time can be non-uniform. If there is a higher density over a certain interval of time, then it will increase the possibility of packets with different normalized waiting time being mapped into the same backoff time. It can also

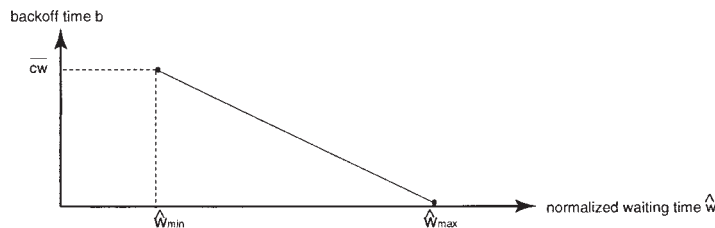


Fig. 6. Linear mapping.

Table II. Linear mapping algorithm.

---

**Adjust  $\alpha$  and  $\beta$ :** every  $T$  seconds  
 $\alpha = \overline{cW} / (\hat{w}_{\max} - \hat{w}_{\min})$   
 $\beta = \overline{cW} + \alpha \times \hat{w}_{\min}$

**Calculate backoff time  $b$ :** on receiving a packet with normalized waiting time  $\hat{w}$ :  
 $b = \Phi(\hat{w}) = \lceil [\beta - \alpha \times \hat{w}]^+ \rceil$

---

increase the possibility of packet collision at the MAC layer. Figure 7(a) illustrates these problems of the linear mapping algorithm.

To address above problems, we present a piecewise linear mapping algorithm which considers the effect of normalized waiting time distribution. As illustrated in Figure 7(b), in the piecewise linear mapping algorithm the normalized waiting times  $\hat{w}$  are divided into  $L$  intervals of equal lengths defined by points  $\hat{w}_{\min} = \hat{w}_0, \hat{w}_1, \hat{w}_2, \dots, \hat{w}_L = \hat{w}_{\max}$ . During each interval, function  $\Phi_i(\hat{w}) = \lceil [\beta_i - \alpha_i \times \hat{w}]^+ \rceil$  will be used for the mapping. Let  $h_i, i = \{0, 1, \dots, L - 1\}$  be the number of packets whose normalized waiting time appeared in the interval  $[\hat{w}_i, \hat{w}_{i+1})$ . Then, we have the following relations

$$\beta_i - \alpha_i \times w_{i+1} = \beta_{i+1} - \alpha_{i+1} \times w_{i+1}, \quad i = 0, \dots, L - 1 \quad (14)$$

$$\frac{\alpha_i}{h_i} = \frac{\alpha_j}{h_j}, i, j = 0, \dots, L - 1 \quad (15)$$

$$\overline{cW} = \beta_0 - \alpha_0 \times \hat{w}_0 \quad (16)$$

$$0 = \beta_{L-1} - \alpha_{L-1} \times \hat{w}_L \quad (17)$$

$$\hat{w}_{\min} = \hat{w}_0 \quad (18)$$

$$\hat{w}_{\max} = \hat{w}_L \quad (19)$$

$$\hat{w}_i - \hat{w}_{i+1} = \hat{w}_{i-1} - \hat{w}_i \quad (20)$$

Among these equations, Equations (14,18,19,20) give the conditions on the interval partition. Equation (15) is the relation between the interval size and the density of normalized waiting times. Equations (16,17) is similar to Equations (11,12) in linear mapping:

Based on these relations, we can derive  $\alpha_i$  as follows:

$$\alpha_i = h_i \times L \times \frac{\overline{cW}}{(\hat{w}_{\max} - \hat{w}_{\min}) \times \sum_{i=0}^{L-1} h_i} \quad (21)$$

and  $\beta_i$  can be calculated recursively

$$\beta_i = \beta_{i+1} + (\alpha_i - \alpha_{i+1}) \times \hat{w}_{i+1} \quad (22)$$

with initial condition

$$\beta_{L-1} = \alpha_{L-1} \times \hat{w}_{\max} \quad (23)$$

The piecewise linear mapping algorithm is summarized in Table III.

To implement the cross-layering scheduling algorithm, the normalized waiting time can be piggy-backed onto the data packet header so that nodes can acquire their distribution information. The presented cross-layering scheduling algorithm can be

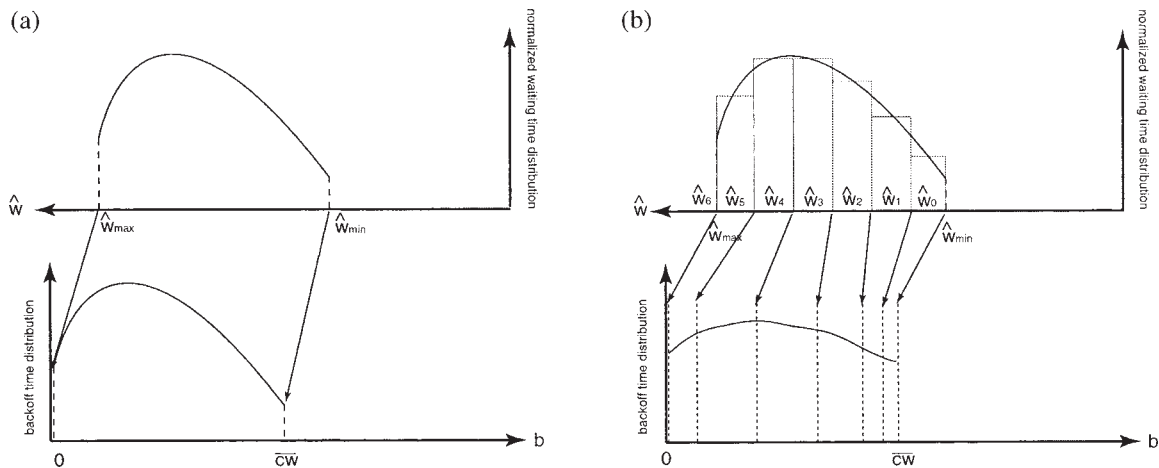


Fig. 7. Linear mapping and piecewise linear mapping: a comparison: (a) Linear mapping; (b) Piecewise linear mapping.

Table III. Piecewise linear mapping algorithm.

---

```

Adjust  $\alpha$  and  $\beta$ : every  $T$  seconds
 $l = (\hat{w}_{\max} - \hat{w}_{\min})/L$ 
for ( $i = 0; i <= L; i++$ )
     $\hat{w}_i = \hat{w}_{\min} + l \times i$ 
 $d = \overline{c\bar{w}} / ((\hat{w}_{\max} - \hat{w}_{\min}) \times \sum_{i=0}^{L-1} h_i)$ 
for ( $i = 0; i < L; i++$ )
     $\alpha_i = h_i \times L \times d$ 
 $\beta_{L-1} = \alpha_{L-1} \times \hat{w}_{\max}$ 
for ( $i = L - 2; i >= 0; i--$ )
     $\beta_i = \beta_{i+1} + (\alpha_i - \alpha_{i+1}) \times \hat{w}_{i+1}$ 
Calculate backoff time  $b$ : on receiving a packet with normalized
waiting time  $\hat{w}$ :
if ( $\hat{w} <= \hat{w}_0$ )
     $b = \lceil [\beta_0 - \alpha_0 \times \hat{w}]^+ \rceil$ 
for ( $i = 0; i < L; i++$ )
    if ( $(\hat{w} >= \hat{w}_i)$  and ( $\hat{w} < \hat{w}_{i+1}$ )) {
         $b = \lceil [\beta_i - \alpha_i \times \hat{w}]^+ \rceil$ 
        break
    }
if ( $\hat{w} >= \hat{w}_L$ )
     $b = \lceil [\beta_{L-1} - \alpha_{L-1} \times \hat{w}]^+ \rceil$ 

```

---

deployed on either a centralized wireless LAN with a base station or a distributed wireless LAN.

- In a centralized wireless LAN, the base station collects the normalized waiting time and calculates its distribution. Periodically (with interval  $T$ ), it computes the values of  $\alpha$  and  $\beta$ , and broadcasts them to other nodes.
- In a distributed wireless LAN, each node records the normalized waiting time of its own packets and listens to the channel in promiscuous mode for the normalized waiting time of packets from other nodes. It then computes the values of  $\alpha$  and  $\beta$  independently with interval  $T$ .

#### 4. Simulation Study

To evaluate the performance of CWTP, we simulate the algorithm under both linear mapping and piecewise linear mapping schemes on a variety of network settings in ns-2 [1].

In the simulation, the number of nodes ( $N$ ) is a parameter that is used to show the impact of network size on CWTP. Each node in the wireless LAN sets up a connection. The transmission rate of each flow is configured to give the network an aggregated load of 1500 kbps as shown in Table IV. Each packet has a size of 512 bytes. Each simulation runs for 100 s. The

Table IV. Default simulation parameter.

---

| Number of nodes | Rate (kbps) | $\overline{c\bar{w}}$ |
|-----------------|-------------|-----------------------|
| 2               | 750         | 34                    |
| 6               | 250         | 45                    |
| 10              | 150         | 50                    |
| 20              | 75          | 60                    |
| 50              | 30          | 105                   |

---

default algorithm parameters are given as follows:  $L = 2$ ,  $T = 1$  s,  $\overline{c\bar{w}}$  is configured according to the number of nodes in the network based on the results from Reference [7], and is also given in Table IV.

In the simulation study, packet delay is measured as follows. Let  $t_1$  be the time instance at which a packet arrives at the queue in the network layer. Further, let  $t_2$  be the time instance at which the IEEE 802.11 acknowledgement of this packet is received at MAC layer, i.e. the packet departs the MAC layer of its sending node and is confirmed to be received. Then the packet delay is  $d = t_1 - t_2$ , which includes both queueing delay and the service delay at the MAC layer. Here packet delay  $d$  is also called the instantaneous packet delay at time instance  $t_2$ .

##### 4.1. Impact of Network Size

We first show the impact of network size on CWTP algorithm. In this experiment, two service classes with  $\delta_2/\delta_1 = 2$  and 4 are supported in the network. Further, these two service classes are configured to have the same number of flows. Figure 8 shows the differentiation index  $I$  with different number of nodes in the network. The target differentiation parameter ratios are 2 and 4 respectively. We observe that both linear and piecewise linear mapping schemes can provide delay differentiation, which closely matches the target differentiation index, when the network size is relatively small ( $N < 20$ ). When the network size is large (e.g.  $N = 50$ ), the piecewise linear mapping scheme performs much better than the linear mapping scheme.

Then, we show the instantaneous delay behaviors with  $N = 10$  in Figure 9. From these results, we observe that the piecewise linear mapping scheme gives much more consistent and smooth delay behavior than the linear mapping scheme. This is because, with the consideration of normalized waiting time distribution, the piecewise linear mapping significantly reduces the possibility of packet collision at the MAC layer.

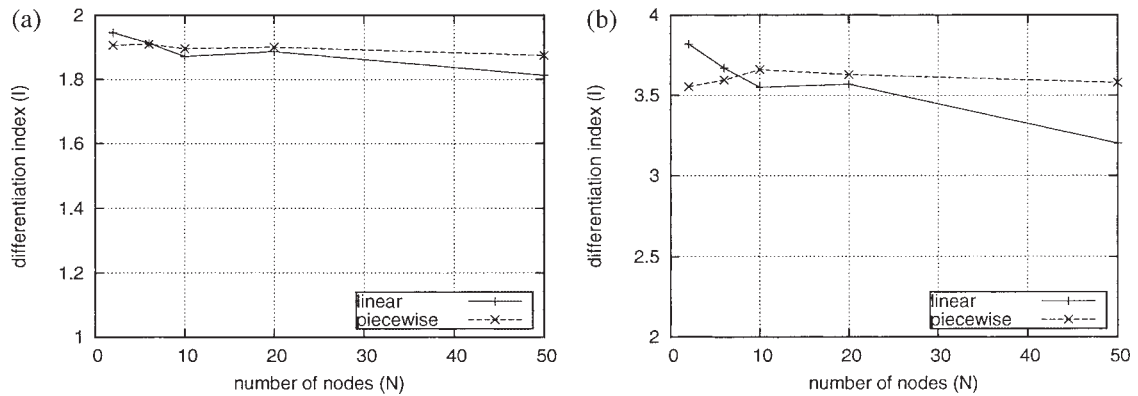


Fig. 8. Impact of number of nodes: (a)  $\delta_2/\delta_1 = 2$ ; (b)  $\delta_2/\delta_1 = 4$ .

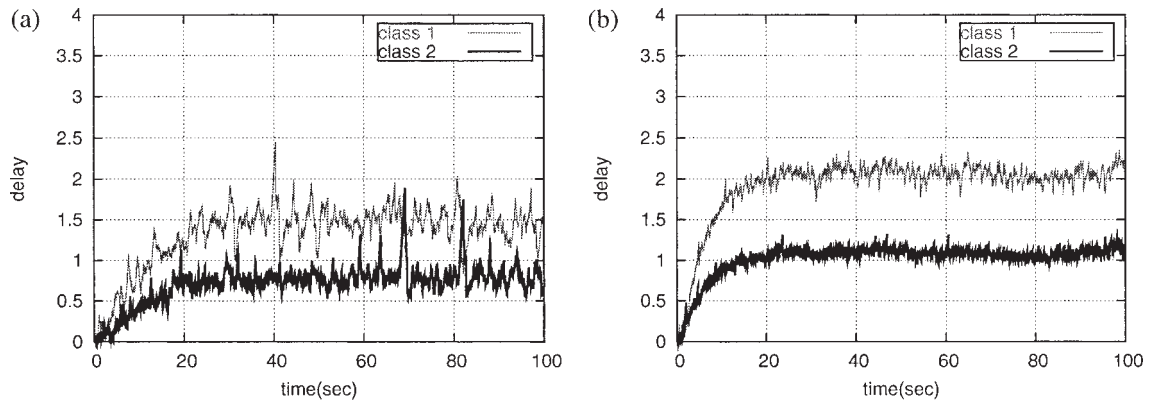


Fig. 9. Instantaneous delay behavior ( $N = 10$ ): (a) Linear mapping; (b) Piecewise linear mapping.

## 4.2. Impact of Traffic

Now we proceed to show the impact of traffic on the performance of CWTP algorithm. We conduct two sets of experiments. In both the sets, the network has 10 nodes. In the first experiment, flows from different service classes have different transmission rates. Figure 10(a) shows the differentiation index under different rate configurations. The  $x$ -axis in the figure gives  $r_1:r_2$ , where  $r_i$  ( $i = 1, 2$ ) is the rate of each flow in service class  $i$ . The instantaneous packet delay for the cases  $r_1:r_2 = 700 \text{ kbps}:140 \text{ kbps}$  and  $r_1:r_2 = 140 \text{ kbps}:700 \text{ kbps}$  are given in Figure 11 and Figure 12 for different mapping schemes.

From the results, we observe that the piecewise linear mapping is less sensitive to the load variance compared with the linear mapping. Further when the rate of service class 1 increases and is higher than the rate of

service class 2, the differentiation index is closer to the target value 2. When the rate of service class 2 increases and is higher than the rate of service class 1, the differentiation index deviates more from the target value 2, especially for the linear mapping scheme. Moreover, linear mapping shows less consistent delay behavior with larger delay jitter when the rate of service class 2 increases. This is because higher sending rate in high service class leads to higher density of large normalized waiting time and aggravates the collision.

In the second experiment, different service classes have different number of flows. Each flow has the same transmission rate of 140 kbps. Figure 10(b) shows the differentiation index under different flow number configurations. The  $x$ -axis in the figure gives  $n_1:n_2$ , where  $n_i$  ( $i = 1, 2$ ) is the number of flows in service class  $i$ . From the results, we observe that the CWTP algorithm maintains a value of differentiation

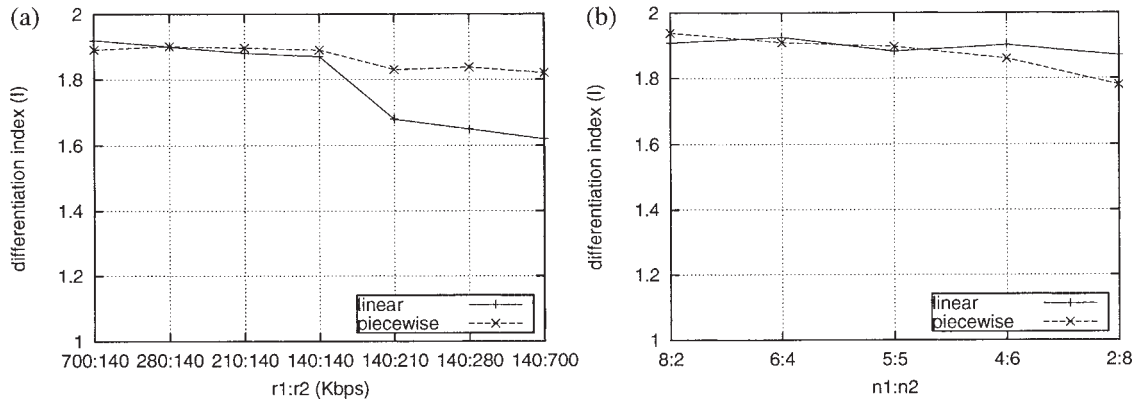


Fig. 10. Impact of traffic: (a) Different rates for different classes; (b) Different numbers of flows in different classes.

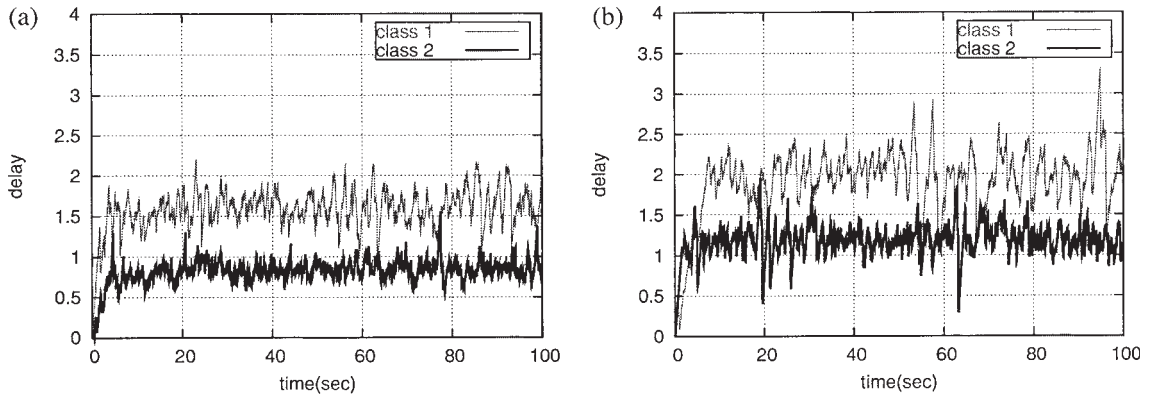


Fig. 11. Instantaneous delay behavior with heterogenous rates under linear mapping: (a) 700 kbps:140 kbps; (b) 140 kbps:700 kbps.

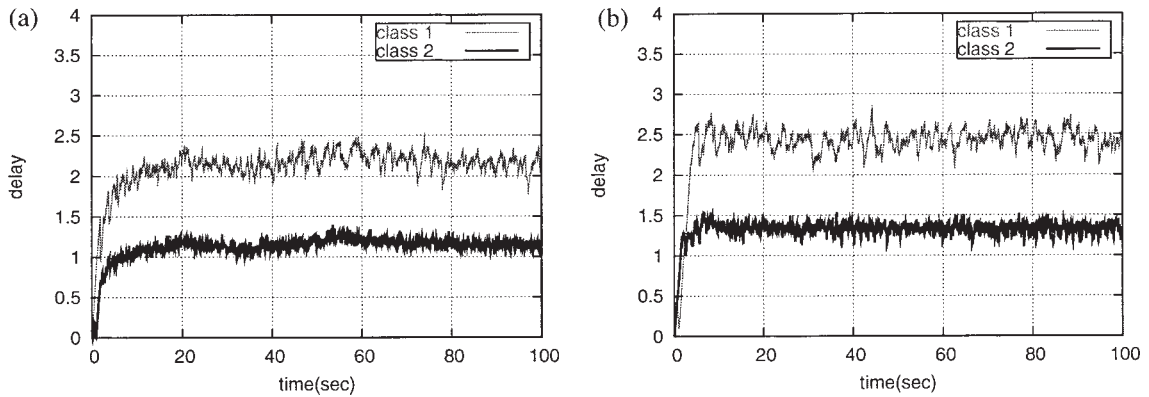


Fig. 12. Instantaneous delay behavior with heterogenous rates under piecewise linear mapping: (a) 700 kbps:140 kbps; (b) 140 kbps:700 kbps.

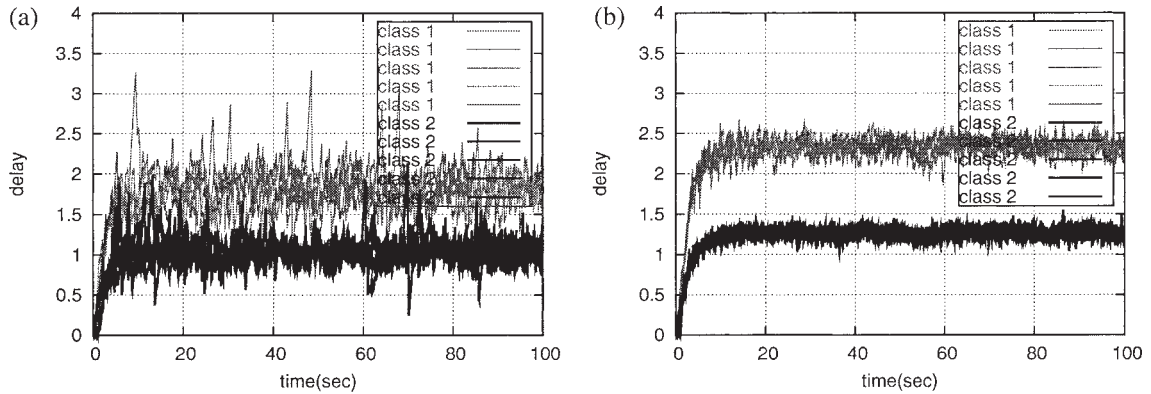


Fig. 13. Instantaneous delay behavior with heterogeneous intra-class traffic: (a) Linear mapping; (b) Piecewise linear mapping.

index very close to the target value under both mapping schemes, while the piecewise linear mapping scheme is less sensitive to the flow number variance compared with the linear mapping scheme.

Now we study the behavior of CWTP algorithm when flows within the same service class have different sending rates. In this experiment, 10 flows are set up in the network, among which 3 flows from service class 1 have sending rate of 280 kbps and 2 flows from service class 1 have sending rate of 140 kbps. On the other hand, 2 flows from service class 2 have sending rate of 280 kbps and 3 flows from service class 2 have sending rate of 140 kbps. The instantaneous packet delay behavior of each flow is plotted in Figure 13. From the result, we observe that the instantaneous packet delay behavior under the piecewise linear mapping scheme is highly consistent with very small jitter. Though the flows within the same service class have different rates, their delay behaviors are the

same. Similar results can also be observed under the linear mapping scheme. Yet the flows under the linear mapping scheme experience larger delay jitter.

We also experiment on traffic with different packet sizes. The results show that the packet size does not affect the performance of delay differentiation.

#### 4.3. More Service Classes

Now we show the delay behavior of our algorithm when more service classes are provided in the network. In this experiment, the network has three service classes with differentiation parameters  $\delta_1: \delta_2: \delta_3 = 1:2:4$ . The number of flows in service class 1, 2 and 3 is 4, 4 and 2 respectively. The instantaneous packet delay behavior of each flow is shown in Figure 14. The results validate that the CWTP algorithm can provide proportional delay differentiation when more service classes are required.

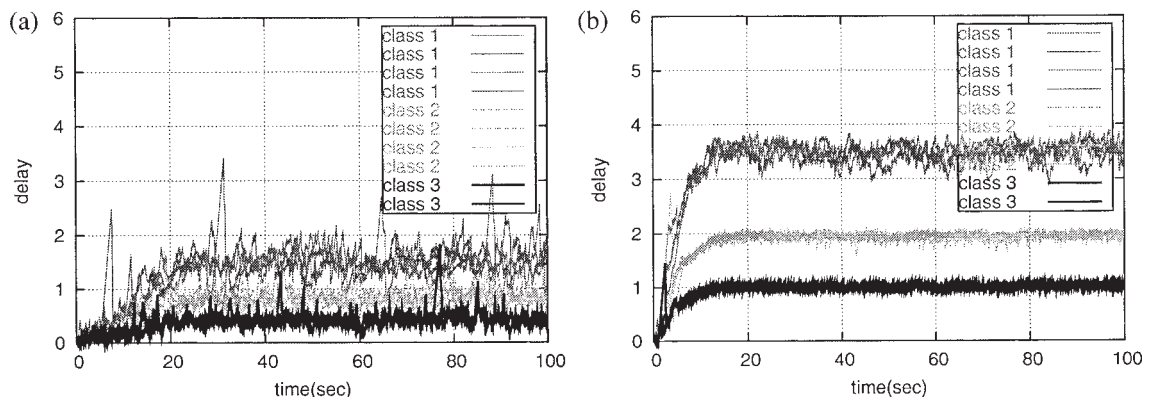


Fig. 14. Instantaneous delay behavior with three service classes: (a) Linear mapping; (b) Piecewise linear mapping.

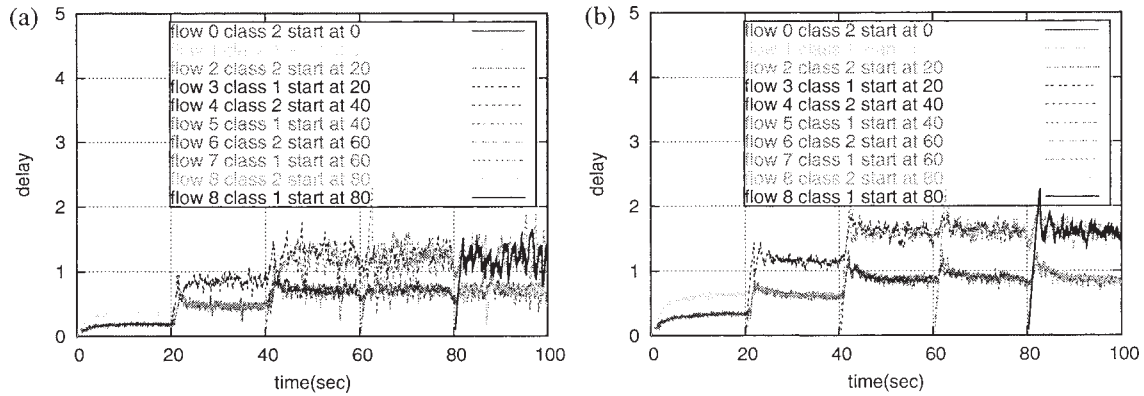


Fig. 15. Instantaneous delay behavior when flows start at different times: (a) Linear mapping; (b) Piecewise linear mapping.

4.4. Traffic Dynamics

We first study the performance of CWTP when traffic dynamically joins the network at different times. Figure 15 shows that both linear and piecewise linear mapping schemes can adapt to dynamic traffic joining and converge to the desired delay differentiation quickly.

To further illustrate how CWTP adapts to dynamic traffic joining, we study the instantaneous values of  $\alpha$  and  $\beta$  under the linear mapping scheme. Figure 16(a) plots the scenario, where eight flows, all start at the same time (0 s). While in Figure 16(b), four of them start at time 60 s. From the results, we observe that  $\alpha$  and  $\beta$  can quickly adapt to the newly joined traffic, thus support appropriate CW tuning operation.

Then, we study the performance of CWTP under bursty traffic. In the simulation, 6 out of 20 flows have exponential on/off traffic with burst time and idle time both as 0.5 s. Each exponential traffic flow has a rate of 100 kbps. Among these exponential traffic, three flows belong to service class 1, and the other three belong to service class 2. We show the instantaneous delay behavior under the piecewise linear mapping scheme in Figure 17(a), and the instantaneous values of  $\alpha_i$  and  $\beta_i$  ( $i = 1, 2$ ) in Figure 17(b)

and (c) respectively. From the results we observe that the bursty traffic causes large delay jitter, yet CWTP can still provide consistent delay differentiation.

4.5. Centralized Implementation Versus Distributed Implementation

CWTP can be deployed on both centralized wireless LANs with base stations and distributed wireless LANs. In the previous experiments, we have shown the performance of the centralized implementation. Now we show the performance of CWTP under the distributed implementation in Figure 18. From the results, we conclude that the distributed implementation can achieve approximately the same performance as the centralized implementation in terms of service differentiation. Figure 19 further illustrates the instantaneous delay behaviors. Similar to the centralized implementation, the piecewise linear mapping scheme gives much more consistent and smooth delay behavior than the linear mapping scheme.

4.6. Choice of Parameter  $L$

Now we show the impact of parameter  $L$ , which is the number of intervals used in the piecewise linear

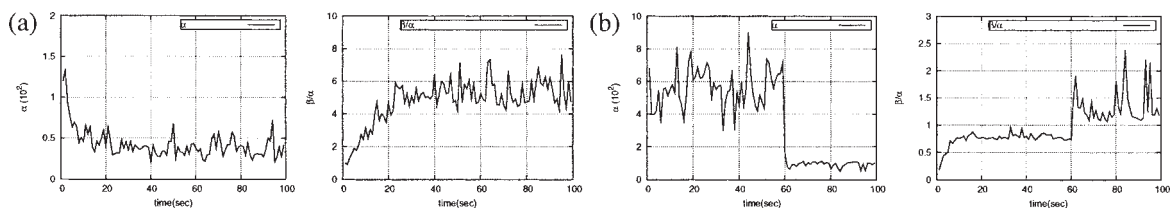


Fig. 16. Instantaneous values of  $\alpha$  and  $\beta$  (linear mapping): (a) All traffic starts at time 0 s; (b) New traffic joins at time 60 s.

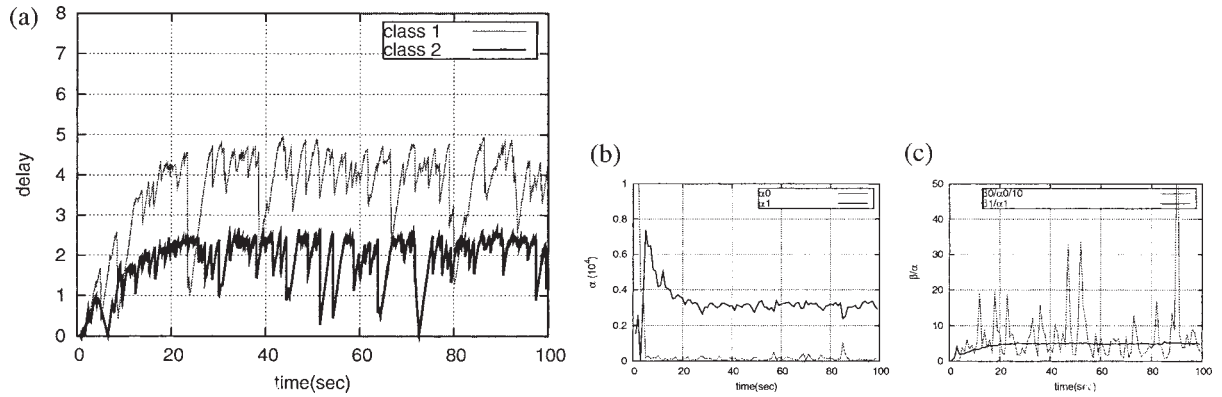


Fig. 17. Exponential on/off traffic under the piecewise linear mapping scheme: (a) Instantaneous delay; (b) Value of  $\alpha$ ; (c) Value of  $\beta$  normalized by  $\alpha$ .

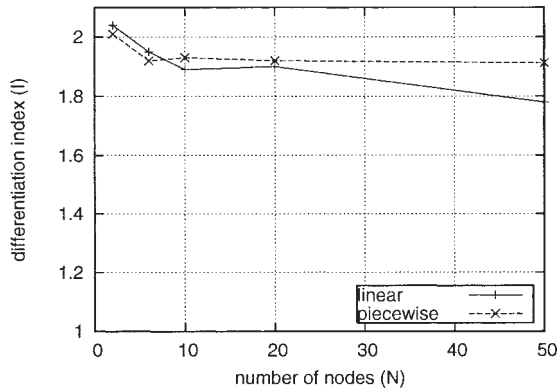


Fig. 18. Differentiation index under distributed implementation.

mapping scheme. We conduct the experiments with different values of  $L$ . Figure 20 shows two cases with  $L = 5$  and  $L = 8$ . In the experiments, we find that a

small value of  $L$  ( $L \leq 5$ ) is sufficient for effective mapping. Moreover, large  $L$  ( $L > 6$ ) will introduce large delay jitter. This is because the statistics of the normalized waiting time may vary in different renew periods. Noise is then introduced when the normalized waiting time is categorized into different intervals. The noise will lead to different values of  $\alpha$  and  $\beta$  during each renew period, and thus cause inconsistent delay differentiation behaviors. A large number of intervals will increase such noise and lead to large delay jitter.

#### 4.7. Comparison with IEEE 802.11

Finally, we compare the performance of CWTP with the IEEE 802.11 standard. Figure 21(a) compares the average delay of CWTP with IEEE 802.11 DCF. Here, the average delay is defined as the delay average of all

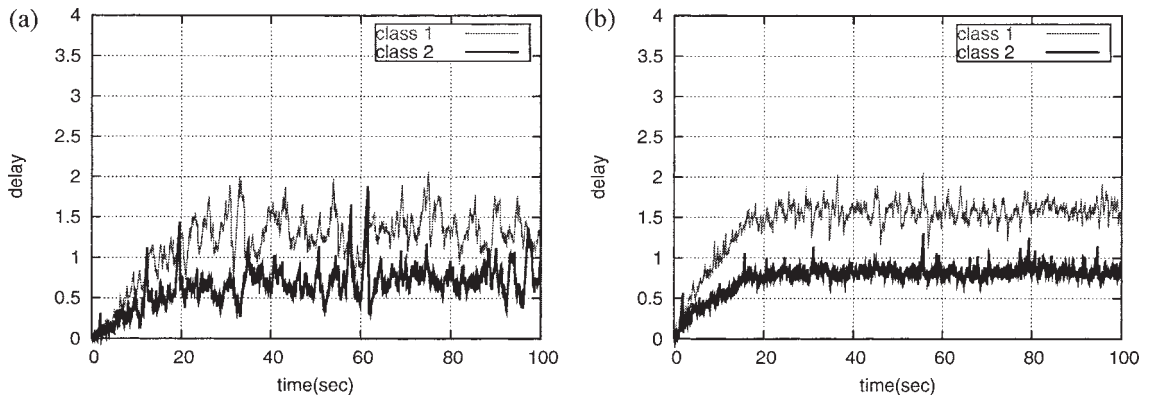


Fig. 19. Instantaneous delay behavior under distributed implementation ( $N = 10$ ): (a) Linear mapping; (b) Piecewise linear mapping.

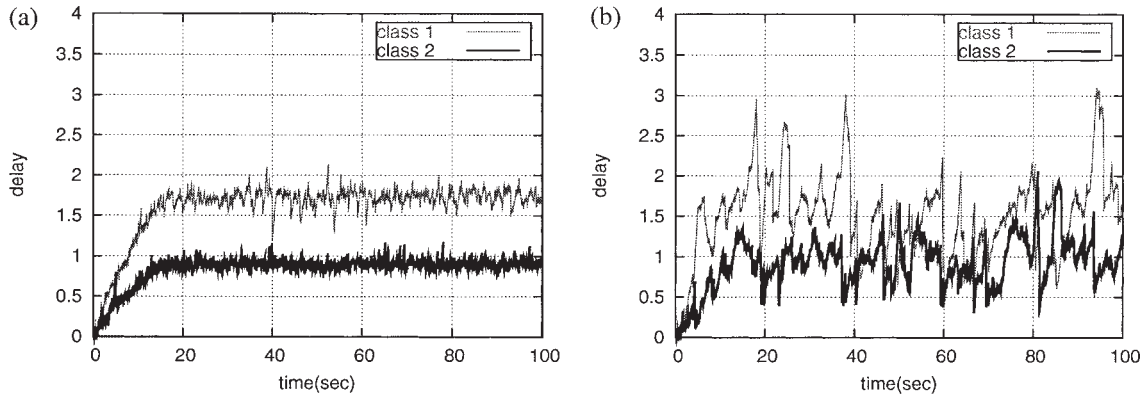


Fig. 20. Different choices of  $L$ : (a)  $L = 5$ ; (b)  $L = 8$ .

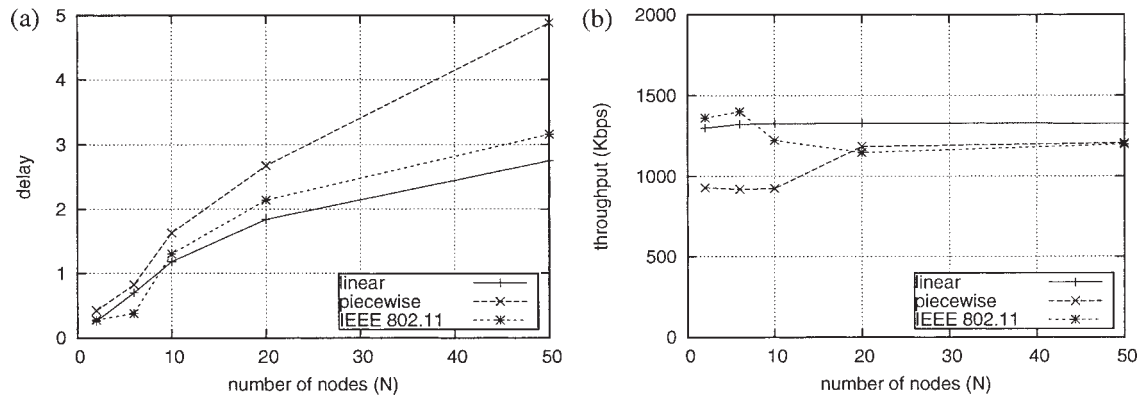


Fig. 21. Comparison with IEEE 802.11: (a) Average delay; (b) Average throughput.

packets from all service classes. From the results, we observe that the linear mapping scheme has the best average delay performance. The piecewise linear mapping scheme can achieve a comparable average delay as IEEE 802.11, when the network size is small ( $N \leq 10$ ). When  $N > 10$ , the piecewise linear scheme has a slightly larger delay. We further compare the average throughput performance under these three schemes in Figure 21(b). The results show that when the network size is small ( $N < 10$ ), IEEE 802.11 has the largest throughput, but only slightly higher than the linear mapping scheme. When  $N > 10$ , the piecewise linear mapping scheme has a comparable throughput as IEEE 802.11, while the linear mapping scheme has the largest throughput.

Above observations can be explained from two aspects. First, both mapping schemes try to achieve the same expected CW size as IEEE 802.11. Thus they

have similar throughput performance as IEEE 802.11. Second, CWTP gives priority to traffic with higher service differentiation parameter. Thus such traffic experiences smaller delay and larger throughput. In the linear mapping scheme, such traffic plays a dominant role in the average operation. On the other hand, to achieves more consistent and stable differentiation, the piecewise linear mapping scheme leads to a more conservative CW size in comparison to the linear mapping scheme. Thus overall, the linear mapping scheme has a better average delay and throughput performance. These observations show the tradeoff between the effectiveness (i.e. proportional delay differentiation property) and the efficiency (i.e. average delay and throughput).

To summarize, the CWTP algorithm can achieve the goal of proportional delay differentiation in wireless LANs with different network sizes and under

different traffic patterns. In addition, the piecewise linear mapping function is able to achieve more consistent instantaneous delay and smaller delay jitter with a small penalty in channel resource utilization.

## 5. Related Work

In this section, we review the existing literatures and compare our work with the existing works.

A large number of works have focused on differentiated MAC layer scheduling under IEEE 802.11 DCF. In these works, flows with different QoS requirements are assigned with different priorities. The packets with different priorities are treated differently at the MAC layer such that high priority packets are transmitted *in priori* to low priority ones. In particular, the work of Reference [5] proposes three service differentiation schemes for IEEE 802.11. The first scheme scales the CW according to the priority of each packet. The second scheme assigns different inter-frame spacings to packets with different priorities. The third one uses different maximum frame lengths for packets with different priorities. The work of Reference [10] proposes that the high priority packets randomly choose their backoff intervals from  $[0, 2^{i+1} - 1]$ , while low priority packets choose from  $[2^{i+1}, 2^{i+2} - 1]$ , where  $i$  is the number of consecutive times that a node attempts to send a packet. The work of Reference [13] analyzes the throughput of a  $p$ -persistent version of IEEE 802.11 DCF with multiple QoS traffic classes. Based on the analytical results, each node tunes its transmission probabilities for different traffic classes to achieve the targeted throughput ratio, while at the same time maximizing the total system capacity. The work of Reference [15] also presents an analytical model for saturation throughput of IEEE 802.11 DCF with multiple service classes and proposes to achieve relative throughput differentiation by varying the initial contention window. Moreover, IEEE 802.11 DCF is extended to incorporate service differentiation. The work of Reference [14] discusses and evaluates enhanced distribution coordination function (EDCF), which is a distributed medium access scheme adopted in IEEE 802.11e to allow prioritized medium access. Reference [20] presents an adaptive enhanced distributed coordination function (AEDCF) based on this new EDCF in IEEE 802.11e standard, which aims to share the transmission channel efficiently. Furthermore, the work of Reference [28] uses a Markov-chain-based model to analyze the performance of IEEE 802.11e. In

these schemes, packets with different priorities may have different queueing delays. However, the relation between a packet's priority and its delay has not quantitatively given. Hence it would be difficult for an application to choose a service class to satisfy its delay requirement. In light of this observation, our prior work [26] provides end-to-end delay differentiation *via* priority mapping based on the support of existing MAC priority scheduling. And the work of Reference [25] achieves end-to-end delay assurances in multihop wireless local area networks *via* dynamic service class selection.

The work of Reference [23] makes use of the carrier sense capabilities of network interfaces. It jams the channel with pulse of energy of specific duration for the transmission of high priority packets. The resulting scheme guarantees the strict priority and provides bounded access delay to high priority packets. This scheme is further extended in References [22,27] for multi-hop ad hoc network environments. In these schemes, the low priority packets can be starved and thus suffer long, unbounded and unpredictable delays.

Besides priority scheduling, fair scheduling in wireless LANs has been studied in existing literatures. In the work of Reference [19], the CW size of each node is properly selected to reflect the relative weights among the flows as well as the number of nodes contending for the wireless channel. Thus it can achieve both weighted fairness and maximizes the aggregated throughput. In the work of Reference [24], a IEEE 802.11 DCF-based fair scheduling algorithm is proposed to schedule transmission such that the bandwidth allocated to different flows is proportional to their weights. Fair scheduling can achieve queueing delay bound with admission control and traffic regulation. In a wireless LAN, where traffic is highly dynamic with node mobility, using fair scheduling to achieve delay differentiation can be expensive. On the other hand, our CWTP algorithm can provide delay differentiation under different traffic patterns and dynamics.

The work of Reference [9] presents a service differentiation solution, which supports time bounded, services within the framework of IEEE 802.11 PCF. This solution requires the support of a base station, while our approach can be deployed on both wireless LANs with base stations and distributed wireless LANs where base stations are not available. The work of Reference [3] evaluates four mechanisms for providing service differentiation in IEEE 802.11 wireless LANs *via* simulation study. The evaluated schemes are the point coordination function (PCF) of IEEE 802.11 [2], the EDCF of the proposed IEEE

802.11e extension to IEEE 802.11 [17], distributed fair scheduling [24] and blackburst [23].

Service differentiation in wireless LAN has also been studied under MAC protocols other than IEEE 802.11. For example, the work of Reference [8] studies the problem of supporting both real-time and non-real-time services in a wireless LAN with dynamic time-division duplexed (D-TDD) transmission. Further the works of References [4,21,18] present QoS architectures that support service differentiation in wireless networks from a system point of view. There also exist works that achieve better QoS for wireless LAN without service differentiation. For example, the fast collision resolution (FCR) algorithm proposed in Reference [16] reduces the average number of idle slots and thus can significantly increase throughput and achieve low latency. The fair scheduling FCR (FS-FCR) algorithm could simultaneously achieve high throughput and fairness.

## 6. Conclusion

Service differentiation in wireless networks is a critical approach to support QoS as well as promote effective resource allocation. This paper studies the problem of proportional delay differentiation in the context of IEEE 802.11 DCF MAC layer protocol. It presents a CWTP scheduling algorithm, which is deployed between the network layer and the MAC layer. Extensive simulation study shows that CWTP algorithm can effectively and efficiently achieve proportional delay differentiation in wireless LANs.

## Acknowledgements

This research was supported by the ONR MURI NAVY CU 37515-6281 grant, and the NSF EIA 99-72884EQ grant. The Board of Trustees of the University of Illinois is the recipient of a Motorola Partnerships in Research Grant. Any opinions, findings and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

## References

1. The Network Simulator—ns-2.
2. IEEE standard 802.11, wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999.
3. Lindgren A, Almquist A, Schelén O. Quality of service schemes for IEEE 802.11 wireless LANs—An evaluation. *Mobile Networks and Applications* 2003; **8**(3): 223–235.
4. Veres A, Campbell AT, Barry M, Sun L. Supporting service differentiation in wireless packet networks using distributed control. *IEEE Journal of Selected Areas in Communications (JSAC), Special Issue on Mobility and Resource Management in Next-Generation Wireless Systems* 2001; **19**(10): 2094–2104.
5. Aad I, Castelluccia C. Differentiation mechanisms for IEEE 802.11. In *Proceedings of IEEE INFOCOM*, 2001.
6. Dovrolis C, Ramanathan P. Dynamic class selection: from relative differentiation to absolute QoS. In *IEEE International Conference on Network Protocols*, 2001.
7. Cali F, Conti M, Gregori E. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transactions Networks* 2000; **8**(6): 785–799.
8. Choi S, Shin KG. A unified wireless LAN architecture for real-time and non-real-time communication services. *IEEE/ACM Transactions Networks* 2000; **8**(1): 44–59.
9. Coutras C, Gupta S, Shroff NB. Scheduling of real-time traffic in IEEE 802.11 wireless LANs. *Wireless Network* 2000; **6**(6): 457–466.
10. Deng DJ, Chang RS. A priority scheme for IEEE 802.11 DCF access method. *IEICE Transactions on Communication* 1999; **E82-B**(1): 96–102.
11. Dovrolis C, Ramanathan P. A case for relative differentiated services and the proportional differentiation model. *IEEE Network* 1999; **13**(5): 26–34.
12. Dovrolis C, Ramanathan P, Stiliadis D. Proportional differentiated services: delay differentiation and packet scheduling. *IEEE/ACM Transactions on Networking* 2002; **10**: 12–26.
13. Ge Y, Hou J. An analytical model for service differentiation in IEEE 802.11. In *IEEE International Conference on Communications (ICC)*, 2003, pp. 1157–1162.
14. Gu D, Zhang J. QoS enhancement in IEEE 802.11 wireless local area networks. *IEEE Communications Magazine* 2003; **41**(6): 120–124.
15. He J, Zheng L, Yang Z, Chou CT. Performance analysis and service differentiation in IEEE 802.11 WLAN. In *Proceedings of 28th Annual IEEE International Conference on Local Computer Networks (LCN)*, 2003, pp. 691–697.
16. Kwon Y, Fang Y, Latchman H. A novel MAC protocol with fast collision resolution for wireless LANs. In *Proceedings of IEEE INFOCOM*, 2003.
17. Benveniste M, Chesson G, Hoeben M, Singla A, Teunissen H, Wentink M. *EDCF Proposed Draft Text, IEEE Working Document 802.11-01/131r1*, 2001.
18. Mahadevan I, Sivalingam KM. Architecture and experimental framework for supporting QoS in wireless networks using differentiated services. *Mobile Networks Applications* 2001; **6**(4): 385–395.
19. Qiao D, Shin KG. Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF. In *Proceedings of The Tenth International Workshop on Quality of Service (IWQoS)*, 2002.
20. Romdhani L, Qiang N, Turletti T. Adaptive EDCF: enhanced service differentiation for IEEE 802.11 wireless ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking (WCNC)*, 2003, pp. 1373–1378.
21. Park S, Kim D, Kim K, Choi S, Hong S. Collaborative QoS architecture between diffserv and 802.11e wireless LAN. In *Proceedings IEEE VTC'03-Spring*, 2003.
22. Sobrinho JL, Krishnakumar AS. Quality-of-service in ad hoc carrier sense multiple access wireless networks. *IEEE Journal of Selected Areas in Communications* 1999; **17**(8): 1353–1368.
23. Sobrinho JL, Krishnakumar AS. Real-time traffic over the IEEE 802.11 medium access control layer. *Bell Labs Technical Journal* 1996; 172–187.

24. Vaidya NH, Bahl P, Gupta S. Distributed fair scheduling in a wireless LAN. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, 2000, pp. 167–178.
25. Wang K, Ramanathan P. End-to-end delay assurances in multi-hop wireless local area networks. In *Proceedings of IEEE Globecom*, 2003.
26. Xue Y, Chen K, Nahrstedt K. Distributed end-to-end proportional delay differentiation in wireless LAN. In *IEEE International Conference on Communications (ICC)*, 2004.
27. Yang X, Vaidya NH. Priority scheduling in wireless ad hoc networks. In *Proceedings of ACM Mobihoc*, 2002, pp. 71–79.
28. Zhao J, Guo Z, Zhang Q, Zhu W. Performance study of MAC for service differentiation in IEEE 802.11. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2002, pp. 778–782.

## Authors' Biographies



**Yuan Xue** received her B.S. degree in 1998 from Department of Computer Science and Engineering, Harbin Institute of Technology, and her M.S. in 2002 from the Department of Computer Science, University of Illinois at Urbana-Champaign. Currently, she is a Ph.D. candidate in the Department of Computer Science at University of Illinois at Urbana-Champaign. Her research interests include wireless network, network-level and application-level quality of service provisioning and mobile computing.

Her research interests include wireless network, network-level and application-level quality of service provisioning and mobile computing.



**Kai Chen** is currently a Ph.D. candidate in the Department of Computer Science at the University of Illinois at Urbana-Champaign. His research interests include mobile ad hoc networks, wireless networking, transport layer protocols, quality of service and pervasive computing. He received his B.Eng. degree from Tsinghua University, Beijing, China in 1995, and M.S. degree from the University of Delaware, Newark, DE, U.S.A. in 1998. From 1998 to 2000, he worked as a research programmer at the National

Center for Supercomputing Applications (NCSA) at Urbana, Illinois, U.S.A.



**Klara Nahrstedt** (M'94) received her A.B., M.Sc degrees in mathematics from the Humboldt University, Berlin, Germany, and Ph.D. in computer science from the University of Pennsylvania. She is an associate professor at the University of Illinois at Urbana-Champaign, Computer Science Department, where she does research on quality of service (QoS)—aware systems with

emphasis on end-to-end resource management, routing and middleware issues for distributed multimedia systems. She is the coauthor of the widely used multimedia book *'Multimedia: Computing, Communications and Applications'* published by Prentice Hall, and the recipient of the Early NSF Career Award, the Junior Xerox Award and the IEEE Communication Society Leonard Abraham Award for research achievements, and the Ralph and Catherine Fisher Professorship Chair. Since June 2001, she serves as the editor-in-chief of the *ACM/Springer Multimedia System Journal*.