

# A Token-Based Scheduling Scheme for WLANs Supporting Voice/Data Traffic and its Performance Analysis

Ping Wang, *Student Member, IEEE*, and Weihua Zhuang, *Fellow, IEEE*

**Abstract**—Most of the existing medium access control (MAC) protocols for wireless local area networks (WLANs) provide prioritized access by adjusting the contention window sizes or inter-frame spaces for different traffic classes. Those MAC protocols can only provide statistical priority access and limited service differentiation. In this paper, a novel token-based scheduling scheme is proposed for a fully-connected WLAN that supports both voice and data traffic. The proposed scheme can provide guaranteed priority access to voice traffic and, at the same time, provide more precise and quantitative service differentiation for data traffic, which provides great flexibility and facility to the network service provider for service class management. Simulation results demonstrate that the proposed scheme can guarantee a small delay for voice traffic. For data traffic, it can effectively achieve proportional differentiation among different classes, while achieving fair resource sharing within the same class. In addition, compared with a contention based scheme and a centralized polling scheme, the proposed scheme significantly improves the channel utilization by avoiding collisions (in the contention based scheme) and the polling overhead (in the polling scheme). The performance analysis of the proposed scheme is also presented. The accuracy of the analytical results is verified by computer simulations.

**Index Terms**—WLAN, token, priority access, class differentiation, Metropolis-Hasting.

## I. INTRODUCTION

IN recent years, with the growth of multimedia applications and the advances of wireless communications technology, quality-of-service (QoS) provisioning over wireless networks has attracted extensive attention in both academia and industry. One most promising wireless network is the wireless local area networks (WLANs), which have been widely deployed to provide high-rate data services at low cost over local area coverage. Most of the existing medium access control (MAC) schemes for WLANs are contention window based schemes [1]– [4]. One common feature of those schemes is that they provide priority channel access to different traffic classes by assigning different inter-frame space (IFS) intervals

and/or contention window (CW) sizes to different classes. High priority traffic (e.g., real-time voice) is assigned smaller IFS,  $CW_{\min}$  and  $CW_{\max}$  values, thus has a larger chance than low priority traffic to access the channel. However, those schemes provide only statistical rather than guaranteed priority access to high priority traffic. High priority traffic may suffer performance degradation due to a heavy load of low priority traffic [5]. Such statistical priority access is difficult to satisfy the delay requirement of real-time traffic. On the other hand, although these schemes can provide a certain degree of service differentiation, it is difficult to quantify the degree of service differentiation, and even more difficult to adjust the degree flexibly among different classes based on some specific requirements of customers or network service providers. For example, when customers are charged differently for different services, it is desired that the received services (or resources) are proportional to what they are charged. Such kind of service model is referred to as proportional differentiation model [6], which assures that the performance of a class is proportional to that of another class according to a ratio preset by the network service provider. Although the actual service performance (e.g., throughput or delay) of each class may vary with the traffic load, the performance ratios among classes remain constant. Such a feature provides great flexibility and facility to network service providers for service management. Most of the existing MAC schemes for WLANs are contention window based schemes without support for the proportional service differentiation.

In this paper, we propose a novel token-based scheme to achieve guaranteed priority access to real-time traffic and, at the same time, achieve proportional differentiation (in terms of channel capacity) to data traffic in a fully-connected WLAN. The token approach has been adopted in many research work to address different problems in different systems. Ripple [7] is proposed for wireless mesh networks, protecting nodes from unintentional packet collisions and maximizing the spatial frequency reuse. A dynamic token ring based MAC protocol (DRP) [8] for mobile ad hoc networks is proposed to solve the intra-flow and inter-flow contention problems at the MAC layer. WTRP (Wireless Token Ring Protocol) [9] and ATP\_MAP (Adaptive Token Passing Multiple Access Protocol) [10] are proposed for WLANs, while [9] is to provide guaranteed QoS to real-time traffic and [10] is to guarantee fairness. The token approach has also been adopted for developing routing protocols [11], [12]. In most of the

Manuscript received November 2, 2006; revised February 21, and April 19, 2007; accepted June 1, 2007. The associate editor coordinating the review of this paper and approving it for publication was V. Leung. This paper was presented in part at IEEE ICC 2007. This work was supported by research grants from the Bell University Laboratories (BUL) and Natural Science and Engineering Research Council (NSERC) of Canada.

The authors are with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1 (e-mail: {p5wang, wzhuang}@bbcr.uwaterloo.ca).

Digital Object Identifier 10.1109/TWC.2007.060889.

previous token-based schemes, the token passing is controlled either by a central controller or by a deterministic logical ring. On the contrary, in our scheme, instead of using a fixed ring, each node passes the token to others stochastically so that proportional differentiation can be achieved. Simulation results demonstrate that the proposed scheme can guarantee a small delay for voice traffic, and that for data traffic it can effectively achieve proportional differentiation among different classes, while achieving fair sharing within the same class. In addition, compared with IEEE 802.11 DCF and the polling scheme [13], the proposed scheme significantly improves the channel utilization, especially when the traffic load is heavy.

The rest of this paper is organized as follows. Section II describes the system model. The proposed scheduling scheme is presented in Section III, and its performance is analyzed in Section IV. Section V is devoted to validating the accuracy of the analysis and evaluating the performance of the proposed scheme, followed by concluding remarks in Section VI.

## II. THE SYSTEM MODEL

Consider a fully-connected WLAN without a central controller, where all the nodes can hear each other. There is a single information channel in the network, through which all the nodes send their packets. Both voice and data services are supported. Voice traffic is represented by an *on/off* model: active voice users (at the *on* state) transmit at a constant rate and inactive users (at the *off* state) do not transmit. The durations of the states are independent and exponentially distributed. As voice traffic is sensitive to delay, guaranteed access priority should be provided to voice traffic over data traffic. Data traffic are categorized into different classes (with different throughput requirements). Specifically, for data traffic, we consider that the WLAN supports  $k$  classes and class  $s$  ( $s = 1, 2, \dots, k$ ) has  $N_s$  source nodes. The class differentiation ratio of any two classes,  $i$  and  $j$ , is denoted by  $C_i/C_j$ , i.e., when a node in class  $i$  gets  $C_i$  fraction of the channel time, a node in class  $j$  should get  $C_j$  fraction of the channel time. Normalized with the capacity share of class 1, the service differentiation parameter of class  $s$  is defined as  $r_s (= C_s/C_1)$ . We assume that the service differentiation parameter of each class is pre-defined by the network service provider and is known to all the data nodes in the system.

## III. THE PROPOSED TOKEN-BASED SCHEDULING SCHEME

There are two tokens in the system. One is circulated among voice nodes (referred to as voice token) and the other is circulated among data nodes (referred to as data token). When a node holds the token, it will transmit its packet(s) when the channel is available. For a voice node, after obtaining the voice token, it transmits all its backlogged packets. For a data node, after obtaining the data token, it is assigned a maximum channel occupancy time (which is the same for all the data nodes and is preset as a system parameter), during which the data node can transmit one or multiple packets depending on its packet size and transmission rate. The proposed scheme works in a distributed manner. There is no central controller passing the tokens to others. The current token holder decides which the next token holder is. When a backlogged voice/data

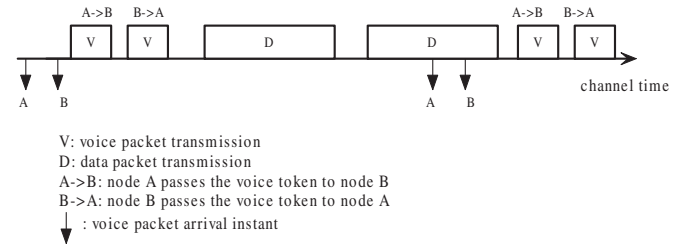


Fig. 1. An example of voice and data packet transmissions in the system.

node holds the token, it piggybacks the token in its voice/data packet transmission and passes it to the next node. Note that the destination of the voice/data packet and the next token holder may be different. When a data token holder has no packet to transmit or a voice token holder changes from the *on* state to the *off* state, the node passes the token directly to the next holder.

### A. Access priority and dynamic token passing for voice traffic

Voice traffic is given a higher access priority than data traffic. Before a voice (or data) token holder transmits its packet(s), it must first wait for the channel to be idle for  $T_2$  (or  $T_1$ , where  $T_1 > T_2$ ). If the channel remains idle during  $T_2$  (or  $T_1$ ), the voice (or data) token holder transmits; otherwise, it waits for the channel to be idle for  $T_2$  (or  $T_1$ ) again. The shorter  $T_2$  ensures a higher access priority to a voice token holder. When the voice token holder starts to transmit, the data token holder will sense a busy channel during  $T_1$ , and defer its transmission. Note that when a voice node receives the token, it is possible that its packet buffer is empty<sup>1</sup>. In this case, the voice node holds the token till its next packet arrives, then when the channel is idle for  $T_2$ , it transmits the packet and passes the token to another node. Before the packet arrival of the voice token holder, the data token holder will sense an idle channel during  $T_1$ , and start its transmission. Fig. 1 illustrates an example of voice/data transmissions.

Considering the *on/off* characteristic of voice traffic, a voice node has no packet to transmit during an *off* period. In order to utilize the channel efficiently, it is desired that the voice token is not passed to the nodes which are at the *off* state. To do so, we let each voice node be aware of its transition point from the *on* to the *off* state. Since the voice traffic has a constant arrival rate during an *on* period, the voice node is able to determine when the next expected packet arrives during the period. If the expected packet does not arrive, the node is considered to be at an *off* state. When a voice node (say node A) holds the token and its state changes from *on* to *off*, the node sends a message to announce that it is in the *off* state and, at the same time, passes the token to the next one (say node B). Upon hearing this message, the previous voice token holder replaces node A with node B as the node to which it will pass the token, so that node A will not receive the token any more during the *off* period. On the

<sup>1</sup>This case may happen when the voice traffic load is low. Since the voice packets arrive periodically at a constant rate, it is possible that a node receives the token before the next expected packet arrives.

other hand, when a node (say node C) switches from the `off` state to the `on` state, it should be able to receive the token to transmit its packets. After waiting for the channel to be idle for  $T_3$  ( $< T_2$ ), node C transmits its packet immediately. In this case, the voice and data token holders will sense a busy channel during  $T_2$  and  $T_1$ , respectively, and defer their transmissions. The previous voice token holder monitors the channel after finishing its transmission. Upon hearing node C's transmission, it puts node C as the node to which it will pass the token next time. When node C finishes its transmission, the current voice token holder (say node D) will transmit its packet(s). Upon hearing node D's transmission, node C puts node D as the node to which it will pass the token.

Note that it is possible that a collision may happen when two or more voice nodes (which change from the `off` state to the `on` state) transmit simultaneously after the channel being idle for  $T_3$ . However, it is shown from both analytical and simulation results that the collision probability is very small and can be neglected. To deal with the rare collisions,  $p$ -persistent CSMA [14] can be used. In our scheme, the voice token passing follows a deterministic sequence, which is related to the packet arrival times of voice nodes. Thus the access delay variance is small.

### B. Proportional class differentiation among data traffic

The portion of channel time unused by voice nodes is shared by all the data nodes. The data token passing process can be modeled by a stationary Markov chain. Each data node in the WLAN is represented by a state in the Markov chain. The transition probability  $P_{ij}$  is the probability that node  $i$  passes the data token to node  $j$ , and the steady-state probability  $\pi_i$  of the Markov chain represents the frequency that node  $i$  holds the data token compared with others (i.e., on average, if node  $i$  holds the data token  $\pi_i$  times, then node  $j$  will hold the data token  $\pi_j$  times). If the following equation is hold for all the data nodes in the network

$$\pi_i = \frac{r_s}{\sum_{s=1}^k N_s \cdot r_s}, \quad \text{if node } i \in \text{class } s, s = 1, 2, \dots, k, \quad (1)$$

the channel occupancy ratios achieved by the data nodes in different classes are exactly the same as the required class differentiation ratios ( $r_s, s = 1, 2, \dots, k$ ), and the channel occupancy time of the data nodes within the same class will be the same.

Given the steady-state distribution, the goal is to find proper values of the transition probability  $P_{ij}$  of the Markov chain so that the chain's steady-state distribution is exactly the one given in (1). According to the Metropolis-Hasting algorithm [15], [16], when we assign a transition probability  $P_{ij}$  as follows

$$P_{ij} = \begin{cases} \frac{1}{N_d - 1} \cdot \min\{1, \frac{\pi_j}{\pi_i}\}, & \text{if } i \neq j \\ 1 - \sum_{k=1, k \neq i}^{N_d} \frac{1}{N_d - 1} \cdot \min\{1, \frac{\pi_k}{\pi_i}\}, & \text{if } i = j \end{cases} \quad (2)$$

where  $N_d$  is the total number of data source nodes, the corresponding steady-state probability of state  $i$  is exactly equal to  $\pi_i$  given in (1).

It should be pointed out that each data node may have a different transmission rate, depending on the location of the

source-destination pair, etc. In this case, even within the same class, a data node with a higher transmission rate will get a higher throughput than those with lower transmission rates. Note that, although the actual throughput may vary, the ratios of the channel time occupied by the data nodes in different classes remain the same as the required ratios. As a result, the system still achieves proportional class differentiation because the resources (i.e., channel time) are proportionally allocated to different classes. With the same amount of assigned resources, a data node may benefit more in term of throughput by increasing its own transmission rate.

Every data node calculates the transition probability  $P_{ij}$  individually according to (2). When a node (say node  $i$ ) holds the data token, it passes the token to the next node (say node  $j$ ) with probability  $P_{ij}$ . Eventually, the capacity share received by each data node satisfies (1). In order to let the proposed scheme works properly, the key point is to let all the data nodes have the same and accurate information of the steady-state distribution  $\pi_i$  at any time. In the WLAN, where all nodes can hear each other, to maintain such up-to-date information is not a difficult task. When a new data node wants to join the WLAN, it first broadcasts a JOIN message, announcing its class  $s$ . To avoid the potential collision between the JOIN message transmission and the voice/data packet transmission, after the channel becomes idle, we let the new data node wait for a short period  $T_4$  ( $< T_3$ )<sup>2</sup>. Upon receiving this JOIN message, all the existing data nodes update their  $\pi_i$  information, and re-calculate  $P_{ij}$  accordingly. One of the existing data node (e.g., the node currently holds the data token) sends a JOIN-ACK message, including the updated  $\pi_i$  information, to the new data node. The new data node then calculates its  $P_{ij}$  accordingly. Similarly, when a data node leaves the WLAN, it also broadcasts a LEAVE message<sup>3</sup>. All the data nodes update their  $\pi_i$  information accordingly.

It is possible that the proposed scheme consumes more power than contention based schemes (e.g., IEEE 802.11) when the traffic load is low since the nodes still consume some power for token transmissions even if they have no packet to deliver. However, the proposed scheme becomes more and more power efficient than contention based schemes when the traffic load becomes high because the power waste due to collisions in contention based schemes will not occur in the proposed scheme.

### C. Recovery of Lost Tokens

The tokens may be lost due to the unreliable wireless channel. For voice nodes, two scenarios can happen. The first scenario is that a node (say node A) still receives voice token although it is in the `off` state. This occurs if the announcement message sent by node A (when it changed from `on` to `off`) was not correctly received by the node which sends the voice token to node A. In this case, node A resends

<sup>2</sup>In our system, JOIN/JOIN-ACK/LEAVE messages have the highest priority to be sent, so a node can send these messages shortly when needed. Otherwise, a node may wait for a long time before it is allowed to send those messages, which is not a good strategy especially when the node leaves the WLAN.

<sup>3</sup>When the current data token holder leaves the WLAN, it passes the token to another node before departure.

the announcement message and passes the token to the next voice node. The second scenario is that a node in the `on` state does not correctly receive the voice token. We let the node (say node B) which passes the token to the next token holder (say node C) monitor the activity of node C. If the activity of node C is not observed (i.e., the token passed to node C is lost), node B resends the token. After several consecutive failed retransmissions, node B passes the token to the next voice node. When the number of backlogged packets of node C is more than one, node C will re-contend for the channel. Similarly, for data nodes, if the current token holder (say node D) cannot pass the token to the next token holder (say node E) after several consecutive transmissions, node D will pass the data token to another node (say node F). Next time when node D chooses node F as the next data token holder, it will replace node F with node E and pass the data token to node E so that the channel access opportunity of each data node remains unchanged.

#### D. Advantages of the proposed scheme

The proposed scheme has the following advantages:

- 1) Most of the conventional contention based MAC schemes (e.g., IEEE 802.11e) provide statistical priority to voice traffic, in which the performance of voice service degrades with an increase of the data traffic load. On the contrary, the proposed scheme provides guaranteed priority to voice traffic, thus the voice performance is not affected by the data traffic load.
- 2) Compared with the conventional contention based MAC schemes, the service class differentiation can be achieved quantitatively in the proposed scheme. Each class can get exactly the desired portion of the channel capacity. Such a class differentiation is difficult to achieve by adjusting the contention windows (or inter-frame spaces) in contention based schemes.
- 3) The contention based MAC schemes are subject to collisions. This nature renders those schemes inefficient channel utilization. By passing the token among the nodes in the WLAN, the proposed scheme eliminates collisions which occur in contention based schemes; therefore, it achieves higher resource utilization, especially when the traffic load is high.
- 4) Compared with a centralized polling scheme where a central controller polls each node to grant the transmission opportunities, the proposed scheme utilizes the resources more efficiently. As pointed out in [17], the polling frames incur a considerable overhead. In the proposed scheme, the token is mostly piggybacked over the voice/data packet transmission. Thus, the overhead is significantly reduced.

## IV. PERFORMANCE ANALYSIS

To make the analysis tractable, we make the following assumptions: a) The voice traffic follows the `on/off` model. The packet arrivals at each data node follow a Poisson process; b) There is no packet loss in radio transmission and no node failure; c) Each node has the same transmission rate. All the voice (or data) packets have the same size. When a data node

gets the token, it transmits one data packet and then passes the token to the next node.

#### A. The channel time occupancy fraction of voice traffic

Given  $N_v$  voice source nodes, the fraction of channel time used by voice traffic, denoted by  $\phi$ , can be derived as follows. The traffic from each voice node follows the `on/off` model, and the durations of the `on` and `off` states are exponentially distributed with mean values  $1/\alpha$  and  $1/\beta$ , respectively. Hence, at any time instant, each voice node is at the `on` state with probability  $\beta/(\alpha + \beta)$ . Denote the voice packet inter-arrival duration as  $T_v$ . During  $T_v$ , each voice node which is at the `on` state generates one voice packet. Thus, the average channel time used by voice traffic during  $T_v$  is given by

$$\bar{T} = \sum_{i=1}^{N_v} \binom{N_v}{i} \left(\frac{\beta}{\alpha + \beta}\right)^i \left(\frac{\alpha}{\alpha + \beta}\right)^{N_v-i} \cdot i \cdot T_{voice}, \quad 0 \leq i \leq N_v \quad (3)$$

where  $T_{voice}$  is the voice packet transmission time. Then the fraction  $\phi$  is given by  $\phi = \frac{\bar{T}}{T_v}$ . On the other hand, given  $\phi$ , we can determine the maximum number of voice nodes  $N_v$  that can be admitted to the network. This result facilitates call admission control of voice traffic when we need to guarantee data traffic a fraction of the channel time.

#### B. Voice delay

The delay is defined as the time period from the moment that a packet arrives at a node to the moment that the packet is successfully transmitted from the node. The voice node at the `on` state can be modeled by a D/G/1 queue model, where the packets arrive at a constant rate, and the service time is the voice token recurrence time, which is defined as the time duration between two consecutive token passing time instants of a node. As discussed in Section III, the token passing sequence keeps track of the packet arrival orders of different nodes, and voice packets arrive periodically, so the variance of the voice token recurrence time is expected to be small (which is verified by simulations). Thus, the queue can be approximated by a D/D/1 queue, where the queueing delay is small, and is bounded by the packet inter-arrival time.

#### C. Collision probability of voice nodes from the `off` state to the `on` state

Here, we are interested in the collision probability in the worst case, where the collisions of voice nodes from the `off` state to the `on` state are most likely to happen. Since a data packet needs a long time duration to transmit (compared with a voice packet and a token frame), it is more likely that collisions of voice nodes occur after data packet transmissions. Thus, the worst case occurs when data traffic are saturated (i.e., the data source nodes always have a packet to transmit). At any time instant, a voice node is at the `off` state with probability  $\alpha/(\alpha + \beta)$ . Given that a voice node is at the `off` state, the conditional probability that a transition to the `on` state happens within duration  $t$  is given by  $1 - e^{-\beta t}$ . In our case,  $t$  can be a data packet transmission time  $T_{data}$ , or a voice

$$P_c^{data} = \sum_{i=1}^{N_v} \binom{N_v}{i} \left(\frac{\alpha}{\alpha+\beta}\right)^i \left(\frac{\beta}{\alpha+\beta}\right)^{N_v-i} \cdot [1 - (e^{-\beta T_{data}})^i - i \cdot (e^{-\beta T_{data}})^{i-1} \cdot (1 - e^{-\beta T_{data}})], \quad 0 \leq i \leq N_v. \quad (4)$$

$$R_i = \phi \cdot R_i + T_{data} \cdot \rho_i + T_{token} \cdot (1 - \rho_i) + \sum_{j=1, j \neq i}^{N_d} \frac{\pi_j}{\pi_i} \cdot [T_{data} \cdot \rho_j + T_{token} \cdot (1 - \rho_j)], \quad \rho_i \leq 1, \rho_j \leq 1, i = 1, 2, \dots, N_d. \quad (5)$$

packet transmission time  $T_{voice}^4$ . Given  $N_v$  voice nodes in the network, the probability that a collision happens after a data packet transmission, denoted by  $P_c^{data}$ , is given by (4) at the top of this page.

Similarly, by replacing  $T_{data}$  with  $T_{voice}$  in (4), we get  $P_c^{voice}$ , the probability that a collision happens after a voice packet transmission. For any packet transmission, the probability that it is a voice packet transmission is

$$\frac{\phi/T_{voice}}{\phi/T_{voice} + (1-\phi)/T_{data}}.$$

Then the collision probability in the worst case,  $P_c$ , is given by

$$P_c = \frac{\phi/T_{voice}}{\phi/T_{voice} + (1-\phi)/T_{data}} \cdot P_c^{voice} + \frac{(1-\phi)/T_{data}}{\phi/T_{voice} + (1-\phi)/T_{data}} \cdot P_c^{data}.$$

#### D. Data throughput

The system throughput is defined as the ratio of the time used for data packet transmission to the total channel time. Consider the WLAN with  $N_d$  data source nodes and  $N_v$  voice source nodes. For the  $i$ th data node, let  $\lambda_i$  denote the average packet arrival rate. To calculate the system throughput, we first derive the average token recurrence time of node  $i$ , denoted by  $R_i$ . As discussed, for  $N_v$  voice nodes, the voice traffic occupy a constant fraction  $\phi$  of the channel time. Since the arrival time of the first packet of a talk burst at each voice node is random, we assume that the channel time occupied by all the voice traffic is uniformly distributed over the total channel time. Thus, during  $R_i$ , the voice traffic occupies  $\phi \cdot R_i$  channel time on average. The average service rate of node  $i$ ,  $\mu_i$ , is simply the reciprocal of its token recurrence time. The queue utilization ratio at node  $i$  is denoted by  $\rho_i = \frac{\lambda_i}{\mu_i} = \lambda_i \cdot R_i$ . Node  $i$  is empty (with no packet to send) with probability  $1 - \rho_i$ . When a data token holder has a packet to transmit, it takes  $T_{data}$  to transmit; otherwise, it takes  $T_{token}$  to pass the token to the next node. The steady-state probability  $\pi_i$  given in (1) reflects the frequency that node  $i$  holds the data token. In a long term, in the time interval of  $R_i$  (during which node  $i$  holds the data token once), on average, node  $j$  holds the data token  $\frac{\pi_j}{\pi_i}$  times. Thus, we have (5) at the top of this page. Solving (5), we get the token recurrence time  $R_i$  ( $i =$

$1, 2, \dots, N_d$ ). Then the system data throughput is given by

$$S = (1-\phi) \cdot \frac{\sum_{i=1}^{N_d} \pi_i \cdot T_{data\_payload} \cdot \rho_i}{\sum_{i=1}^{N_d} \pi_i \cdot (T_{data} \cdot \rho_i + T_{token} \cdot (1 - \rho_i))}, \quad \rho_i \leq 1$$

where  $T_{data\_payload}$  is the time to transmit the payload of a data packet. When  $\rho_i = 1$  for all the data nodes, the system is in an overload condition.

#### E. Data packet delay

With the Poisson arrival assumption, the packet arrival and departure at each data node can be modeled by an M/G/1 queue. To determine the average delay, we first need to obtain the average queue length at each data node. According to [18] (p. 175), for an M/G/1 queue, arrivals, departures, and random observers all see the same distribution of the number of customers in the system. Thus, we conclude that the average queue length at an arbitrary time is equal to the average queue length at any packet departure instant. We consider an imbedded Markov chain in which the state transitions occur at the packet departure instants of a tagged node. We define the state of this imbedded Markov chain to be the number of packets left behind by the departing packet. For simplicity of analysis, we assume that the probability of a node having more than  $M$  ( $M$  is chosen to be a large number) packets in the queue is negligible. The state transition diagram of the imbedded Markov chain is shown in Fig. 2. The average queue length at the tagged node is given by  $L = \sum_{k=0}^M k p(k)$ , where  $p(k)$  is the steady-state probability of state  $k$ . To find the steady-state probability vector of this Markov chain, we should first obtain the state transition probability from any state  $i$  to  $j$ , denoted by  $pr(i, j)$ .

The derivation of  $pr(i, j)$  for the case  $i = 0$  is more complex than for the case  $i > 0$ . We first consider the case  $i > 0$ , i.e., the tagged node is backlogged (when a packet leaves the node, there is at least one packet left in the queue).  $pr(i, j)$  ( $i > 0$ ) is the probability that, during a token recurrence time of a backlogged data node,  $j - i + 1$  packets arrive at that node. Obviously, for all  $j \leq i - 2$ ,  $pr(i, j) = 0$ . Denote the average traffic arrival rate at the tagged node as  $\lambda$ , we have

$$pr(i, j) = \int_0^\infty \frac{(\lambda x)^{j-i+1}}{(j-i+1)!} e^{-\lambda x} b(x) dx, \quad i > 0 \quad (6)$$

where  $b(x)$  is the pdf (probability density function) of the token recurrence time at the tagged node, which is backlogged. In order to obtain  $pr(i, j)$ , we need to know the distribution of this token recurrence time. Since this distribution is difficult to obtain directly, we use the Laplace transform  $B^*(s)$  for  $b(x)$  defined as  $B^*(s) = \int_0^\infty e^{-sx} b(x) dx$ . The derivation of  $B^*(s)$  is presented in Appendix. We define random variable  $v$  as

<sup>4</sup>Here, we do not take token transmission time into account. In the worst case, the token frame transmissions rarely happen since all the data tokens are piggybacked over data packets, and the voice token is transmitted only when a voice source node is from the on state to the off state. Besides, since the token transmission time is very short, the fraction of channel time used for token frame transmissions is negligible.

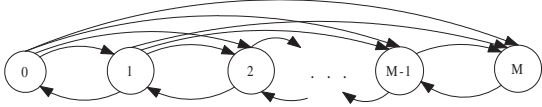


Fig. 2. The state transition diagram of the imbedded Markov chain.

the number of packet arrivals during a token recurrence time of a backlogged data node, and the  $z$ -transform of the pmf (probability mass function) of this random variable is given by  $V(z) = \sum_{k=0}^{\infty} P[v = k]z^k$ . The relationship between  $V(z)$  and  $B^*(s)$  is as follows [18] (p. 184),

$$V(z) = B^*(\lambda - \lambda z). \quad (7)$$

Then  $pr(i, j) (i > 0)$  is given by

$$pr(i, j) = \frac{1}{(j - i + 1)!} \cdot \frac{d^{j-i+1}}{dz^{j-i+1}} V(z)|_{z=0}.$$

For the derivation of  $pr(i, j)$  with  $i = 0$  (i.e., when a packet leaves the tagged node, the node has an empty queue), we consider two scenarios. The first scenario is that, when the tagged node gets a new chance to transmit (we denote this time instant as  $t_s$ , which is the time instant that the node gets the data token again and the channel is available for the data node to transmit), the queue is still empty. In this case, it will pass the token immediately to another node. The other scenario is that when the tagged node gets a new chance to transmit, it has at least one packet in the queue. In this case, it will transmit the data packet and piggyback the token over the data packet. Once state 0 in Fig. 2 occurs, we sample the number of packets at the tagged node, denoted by  $k$ , at the time instants  $t_s$ , and stop sampling when  $k > 0$ . Fig. 3 shows the state transition diagram of the sampled process. There is one starting state (the same as state 0 in Fig. 2), one transient state ( $k = 0$ ), and  $M$  absorbing states ( $k = 1, 2, \dots, M$ ), with state transition probabilities  $Pt(\cdot, \cdot)$ . The derivation of these state transition probabilities is similar to that of  $pr(i, j) (i > 0)$ . Note that different time intervals are involved. The time interval from the starting state to state  $k$  ( $k = 0, 1, \dots, M$ ), denoted by  $t_1$ , is the time period from the moment that the last packet in the queue leaves the tagged node to the moment that the node gets a new chance to transmit. The time interval from the transient state to an absorbing state, denoted by  $t_2$ , is token recurrence time of an empty node. Both  $t_1$  and  $t_2$  are random variables and the Laplace transforms of their pdfs are represented by  $H_1^*(s)$  and  $H_2^*(s)$ , respectively. The derivations of  $H_1^*(s)$  and  $H_2^*(s)$  are given in Appendix. Let  $v_1$  and  $v_2$  denote the number of packet arrivals during  $t_1$  and  $t_2$ , respectively, and  $V_1(z)$  and  $V_2(z)$  the  $z$ -transform of the pmfs of  $v_1$  and  $v_2$ , respectively. Similar to (7), we have  $V_1(z) = H_1^*(\lambda - \lambda z)$  and  $V_2(z) = H_2^*(\lambda - \lambda z)$ .

The transition probability from the starting state to state  $k$  ( $k = 0, 1, \dots, M$ ), denoted by  $Pt(s, k)$ , is given by

$$Pt(s, k) = \frac{1}{k!} \cdot \frac{d^k}{dz^k} V_1(z)|_{z=0}.$$

Also, the transition probability from the transient state ( $k = 0$ ) to state  $k$  ( $k = 0, 1, \dots, M$ ), denoted by  $Pt(0, k)$ , is given

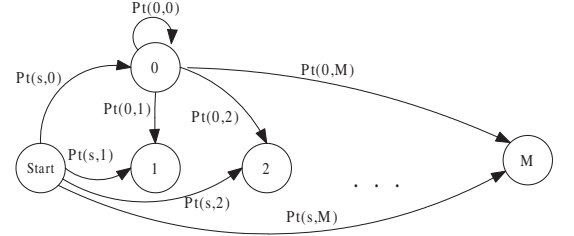


Fig. 3. The state transition diagram of the sampled process.

by

$$Pt(0, k) = \frac{1}{k!} \cdot \frac{d^k}{dz^k} V_2(z)|_{z=0}.$$

Denote  $Pa(k)$  the probability that the sampled process enters absorbing state  $k$  ( $k = 1, 2, \dots, M$ ). Given the state transition probabilities, it is straightforward to get  $Pa(k)$  as follows

$$Pa(k) = Pt(s, k) + \frac{Pt(s, 0)Pt(0, k)}{1 - Pt(0, 0)}, \quad k = 1, 2, \dots, M.$$

$Pa(k)$  is actually the conditional probability that the tagged node (which finds itself empty when the latest packet departs) finds  $k (> 0)$  packets in the queue when it gets a new chance to transmit under the condition that, when it gets the new chance to transmit, it finds at least one packet present. Then we have

$$pr(0, j) = \sum_{k=0}^j Pa(k+1) \frac{e^{-\lambda T_{data}} (\lambda T_{data})^{j-k}}{(j-k)!}, \quad j = 0, 1, \dots, M. \quad (8)$$

We take  $pr(0, 0)$  and  $pr(0, 1)$  as examples to explain (8).  $pr(0, 0)$  represents the possibility that, when the tagged node gets a chance to transmit, it finds one packet in the queue (with probability  $Pa(1)$ ), and there is no packet arrival (with probability  $e^{-\lambda T_{data}}$ ) during the packet transmission time. For  $pr(0, 1)$ , there are two cases: One case is that, when the tagged node gets a chance to transmit, it finds one packet in the queue (with probability  $Pa(1)$ ) and, during the packet transmission time, one new packet arrives (with probability  $e^{-\lambda T_{data}} \lambda T_{data}$ ); The other case is that it finds two packets in the queue (with probability  $Pa(2)$ ), and no packet arrives (with probability  $e^{-\lambda T_{data}}$ ) during the packet transmission time.

With  $pr(i, j)$ , we can obtain the steady-state probabilities  $p(k)$  ( $k = 0, 1, \dots, M$ ) of the chain shown in Fig. 2. Then we have the average queue length  $L$ . According to the Little's law, the average delay at the tagged node is given by  $D = \frac{L}{\lambda}$ .

## V. NUMERICAL RESULTS AND PERFORMANCE EVALUATION

In this section, we validate our analysis and evaluate the performance of the proposed scheme by extensive simulations. For voice traffic, we compare delay performance with IEEE 802.11e. We choose the GSM 6.10 codec as the voice source as an example. The voice packet size is 107 bytes with 33-byte payload and 74-byte RTP/UDP/IP headers. The voice packet inter-arrival period is 20 ms. For data traffic, we compare the

TABLE I  
SYSTEM PARAMETERS USED IN SIMULATION AND ANALYSIS.

Parameter	Value
Slot time	20 $\mu$ s
$T_1$ /AIFS[AC_data]	60 $\mu$ s
$T_2$ /AIFS[AC_voice]	40 $\mu$ s
$T_3$	20 $\mu$ s
$T_4$ /SIFS	10 $\mu$ s
$CW_{min}$ [voice]	15
$CW_{max}$ [voice]	63
$CW_{min}$ [data]	31
$CW_{max}$ [data]	1023
PHY preamble	192 $\mu$ s
RTS frame size	20 bytes
CTS frame size	14 bytes
polling/token frame size	36 bytes
data packet size	1000 bytes
voice packet size	107 bytes
channel rate	11 Mbps
basic rate	2 Mbps
$1/\alpha$	352 ms
$1/\beta$	650 ms
$T_v$	20 ms

performance of channel utilization with IEEE 802.11 DCF (which is a contention based scheme) and the centralized polling scheme<sup>5</sup> [13]. The simulation parameters are given in Table I, where the channel rate is to transmit voice/data packets, and the basic rate is to transmit RTS, CTS, polling frames and the token. The simulation is done in Matlab. Each data node uses a random generator to randomly choose the next token holder based on the transition probabilities. In each run, we simulate 50 seconds of the channel time (except those scenarios which have a specific simulation end time). Each of the simulation results represents an average of 10 independent runs.

*Voice traffic analysis accuracy* – Table II shows the fraction of the channel time occupied by voice traffic with different number of voice source nodes ( $N_v$ ). It can be seen that the analytical results match closely with the simulation results. For the collision probability ( $P_c$ ), the simulation results demonstrate that it is small, even when  $N_v$  is large. For  $N_v$  equals to 110, 120 and 130, the analytical results of  $P_c$  are 0.14%, 0.15% and 0.15%, respectively, while the simulations results are 0.14%, 0.15%, and 0.16%, respectively. For delay performance, we consider an integrated voice/data scenario with  $N_v = 50$ . Fig. 4 compares the delays of the proposed scheme and IEEE 802.11e with different number of data source nodes. It can be seen that the average voice delay increases greatly with the increase of data source nodes when using IEEE 802.11e,

<sup>5</sup>For the contention based scheme, since the RTS/CTS mechanism can improve the performance compared with the basic access scheme [19], we adopt the RTS/CTS mechanism in our simulation. In the polling scheme, a central controller polls each node (based on its scheduling policy) by broadcasting a polling frame. Upon being polled, a node is granted a transmission opportunity to transmit its packets. If the polled node has no packet to send, the central controller polls next node immediately. For fair comparison, the proposed scheme and the polling scheme have the same scheduling policy, each polled node (or data token holder) is granted the same channel time, and the data token size is chosen to be the same as the polling frame size.

TABLE II  
THE CHANNEL TIME FRACTION OCCUPIED BY VOICE TRAFFIC

The number of voice source nodes	20	30	40	50	60
Simulation results	11%	17%	23%	28%	35%
Analytical results	11%	16%	21%	26%	32%

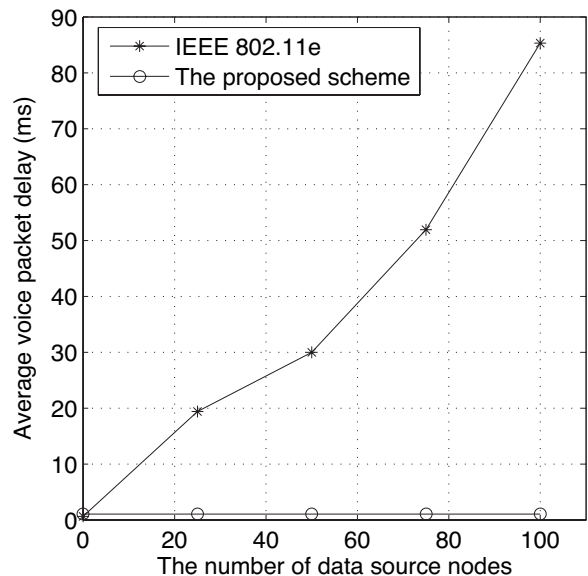


Fig. 4. The voice packet delay versus the number of data source nodes in the simulations with  $N_v = 50$ .

but remains unchanged in the proposed scheme. The reason is that our scheme provides guaranteed priority to voice nodes, thus the voice performance is not affected by the data traffic. We also observe that the voice delay in our scheme is very low (around 1 ms), which verifies the conclusion drawn in our analysis.

*Proportional class differentiation of data traffic* — Since the performance of proportional class differentiation to data traffic is not affected by the voice traffic, for simplicity, we consider a WLAN with 20 data source nodes in the absence of voice nodes. First, we vary the class differentiation requirements to see if the proposed scheme can provide quantitative class differentiation based on an arbitrary requirement. We consider three scenarios with different classes and required class differentiation ratios. In the first scenario, the nodes are classified into two classes, each having 10 nodes. The required differentiation ratio is 1 : 2. In the second scenario, there are three classes with a desired differentiation ratio 1 : 1.5 : 3. The number of nodes in class 1, 2 and 3 are 5, 5 and 10, respectively. In the third scenario, the nodes are grouped into four classes, each having 5 nodes. The differentiation ratio is 1 : 0.5 : 2 : 3. The throughput of each node in the three scenarios are shown in Table III. It is clear that the nodes of different classes achieve different throughputs. The ratios are very close to the requirements. The nodes in the same class achieve almost the same throughput as expected. The simulation results demonstrate that the proposed scheme can effectively provide proportional class differentiation based on a specific differentiation requirement.

Note that in all the above three scenarios, the number of

TABLE III  
THE THROUGHPUT ACHIEVED BY EACH DATA NODE IN THE THREE SCENARIOS.

Node No.	Scenario 1		Scenario 2		Scenario 3	
	Class No.	throughput (Mbps)	Class No.	throughput (Mbps)	Class No.	throughput (Mbps)
1	1	0.2819	1	0.2027	1	0.2626
2	1	0.2826	1	0.2047	1	0.2666
3	1	0.2750	1	0.2043	1	0.2620
4	1	0.2826	1	0.1993	1	0.2602
5	1	0.2822	1	0.2020	1	0.2634
6	1	0.2859	2	0.2985	2	0.1318
7	1	0.2852	2	0.3032	2	0.1315
8	1	0.2836	2	0.2996	2	0.1315
9	1	0.2837	2	0.2963	2	0.1358
10	1	0.2861	2	0.2980	2	0.1305
11	2	0.5643	3	0.6066	3	0.5234
12	2	0.5776	3	0.5973	3	0.5278
13	2	0.5685	3	0.5897	3	0.5235
14	2	0.5782	3	0.6095	3	0.5274
15	2	0.5710	3	0.6114	3	0.5248
16	2	0.5727	3	0.6081	4	0.7900
17	2	0.5752	3	0.5945	4	0.8001
18	2	0.5724	3	0.6048	4	0.7910
19	2	0.5679	3	0.6052	4	0.7857
20	2	0.5754	3	0.6063	4	0.8023

nodes in the WLAN remains unchanged during the whole simulation time. Next, we vary the number of nodes (i.e., some nodes join or leave the WLAN) to verify that the proposed scheme is adaptive to the network dynamic and provides consistent class differentiation. Consider three classes, with required class differentiation ratio 1 : 2 : 3. At the beginning of the simulation (i.e.,  $t = 0$ ), there are 5 nodes in each class. At  $t = 5$ s, a new node of class 1 joins the WLAN and, at  $t = 10$ s, a node in class 3 leaves the WLAN and, at  $t = 15$ s, a new node of class 2 joins the WLAN. We obtain the throughput of each node, shown in Fig. 5. The throughput of each node is reduced from  $t = 5$ s because of the new node arrival, and increases from  $t = 10$ s because of the node departure, and reduced again from  $t = 15$ s because a new node joins the WLAN. Note that the throughput ratios among the three classes remain very close to the constant (as 1 : 2 : 3) during the whole simulation time. Although the actual throughput of each node changes due to the network dynamics, the proposed scheme maintains a consistent class differentiation ratio among different classes.

*Data throughput and delay analysis accuracy* — Consider the WLAN with two classes, each having 15 data source nodes. The number of voice source nodes is 20. The required class differentiation ratio is 1 : 2. Fig. 6 shows the aggregate throughput and the throughput achieved by each class, with different system traffic loads. For delay performance, consider three classes with a required class differentiation ratio 1 : 2 : 3. Fig. 7 shows the average delay of the three classes versus the system traffic load. Obviously, when the traffic load becomes high, the delay suffered by the node in each class increases. From Figs. 6 and 7, it is clearly that the analytical and

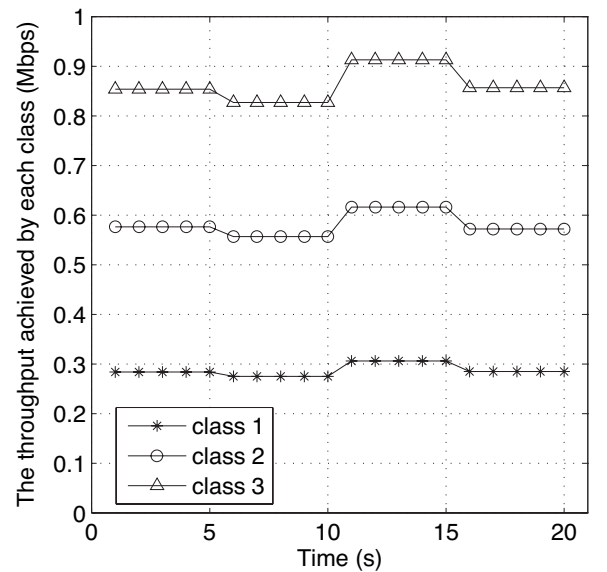


Fig. 5. The throughput achieved by a node in each class.

simulation results of the throughput and delay agree with each other very well.

*Channel utilization* — Channel utilization is represented by the ratio of the achieved system throughput to the channel rate. From the simulations, we find that the channel utilization performance is not sensitive to class differentiation. For simplicity, here we consider a homogeneous WLAN with a single class. We fix the number of the data source nodes to be 20 in the WLAN, and vary the traffic arrival rate. Fig. 8 compares the channel utilization of IEEE 802.11 DCF, the polling

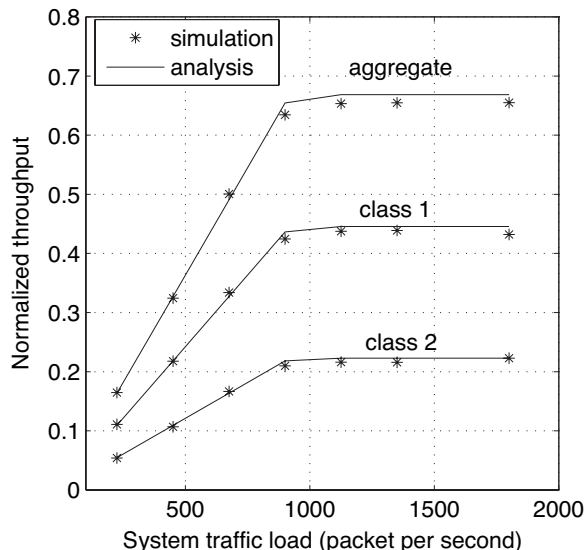


Fig. 6. The throughput versus the system traffic load with  $N_v = 20$ .

scheme, and the proposed scheme over a perfect channel. It is clear that the proposed scheme achieves much higher channel utilization than IEEE 802.11 DCF, when the traffic load becomes high. With an increase of traffic load, collisions occur more frequently with the contention based scheme. By avoiding those collisions, the proposed scheme utilizes the channel more efficiently. Compared with the centralized polling scheme, the proposed scheme also achieves higher utilization, because it reduces the overhead incurred by the polling frames. The channel utilization of the proposed scheme over an unreliable channel is also shown in Fig. 8, where the packet error probability  $P_e = 5\%$ . The impact of the channel is negligible when the traffic load is low, but results in an approximate 5% reduction in the channel utilization. It is observed that the proposed scheme still outperforms the other two schemes, taking into account of the possible token loss.

## VI. CONCLUSION

In this paper we have proposed a novel token-based scheduling scheme for a fully-connected WLAN that supports both voice and data traffic. The proposed scheme can provide guaranteed priority access to voice traffic and, at the same time, provide precise and quantitative service differentiation for data traffic, which provides great flexibility and facility to the network service provider for service class management. Our current work focuses on providing quantitative service differentiation and maintaining fairness among different classes in a fully-connected WLAN. To guarantee QoS for each class, we may need to incorporate call admission control (CAC) on both voice and data traffic. However, introducing CAC may cause some loss in the radio resource utilization. How to balance QoS and channel efficiency needs to be further investigated. Also, the system model of a fully-connected network under consideration may limit the application of the proposed scheme. Extending to a partially-connected network is technically very challenging and requires further investigation.

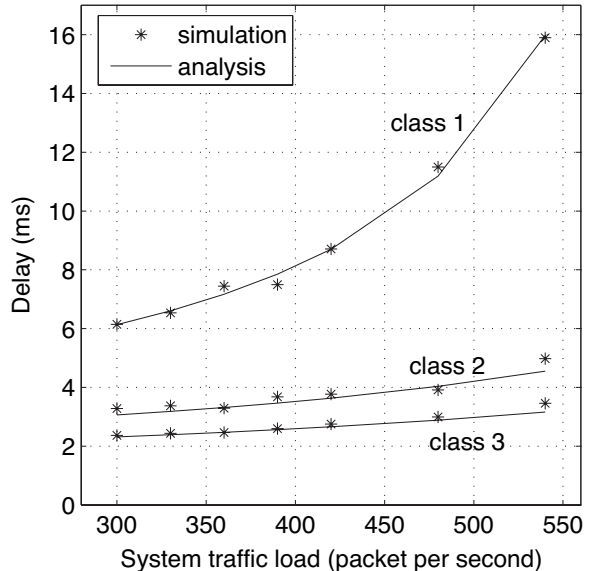


Fig. 7. The average packet delay versus the system traffic load with  $N_v = 20$ .

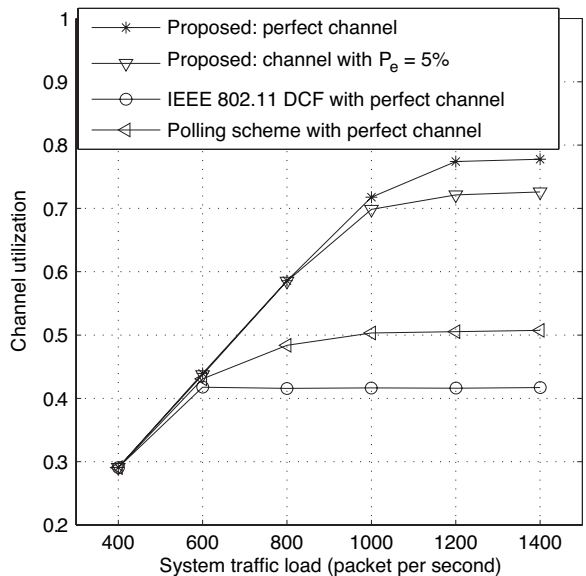


Fig. 8. The channel utilization versus the system traffic load.

## APPENDIX: THE DERIVATION OF $B^*(s)$ , $H_1^*(s)$ , AND $H_2^*(s)$

### A. Derivation of $B^*(s)$

$B^*(s)$  is the Laplace transform of the pdf of a backlogged data node's token recurrence time. As discussed, the data token passing process in the proposed scheme can be modeled by a Markov chain. Each transition from state  $i$  to  $j$  represents that the token is passed from node  $i$  to node  $j$ . In order to get  $B^*(s)$ , we re-draw this Markov chain as follows.

For simplicity of presentation, we take a simple example with three data nodes in the WLAN and take node 1 as the tagged node. We split state 1 into two states:  $1_a$  and  $1_b$ , as shown in Fig. 9. State  $1_a$  is the starting point, representing that node 1 is passing the data token to others; and state  $1_b$  is the ending point, representing that node 1 is receiving the

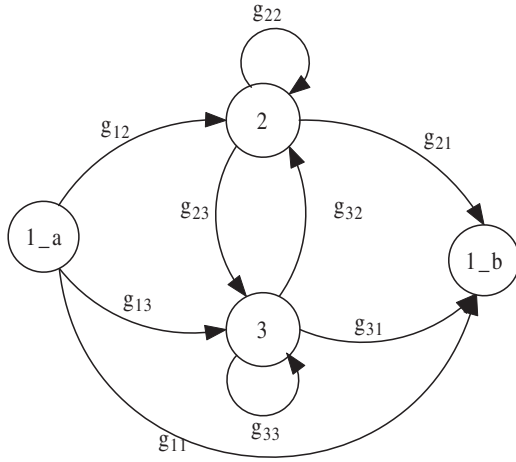


Fig. 9. An example of state transition of the data token passing process.

data token from others. Assuming node 1 is backlogged, the time duration taken from state 1\_a to state 1\_b is actually the token recurrence time of a backlogged data node, whose pdf has Laplace transform  $B^*(s)$ .

To get  $B^*(s)$ , a method similar to that used in [20] is applied here. The Markov chain shown in Fig. 9 can be treated as a signal flow graph [21]. State 1\_a is the source of the signal and state 1\_b is the sink. All the other nodes are signal repeaters. Each branch (state transition) is associated with a branch transmittance. The signals travel along the branches and are modified by the corresponding branch transmittances. A repeater combines all the incoming signals and sends the outgoing signal along all the branches diverging from that repeater. The signal transfer function from the source to the sink can be obtained by application of the Mason's rule or other flow reduction methods [21]. It has been found out that, if the branch transmittances are properly defined, the probability generating function of total transition time from the source to the sink can be obtained from the signal transfer function [20]. The probability generating function is actually the  $z$ -transform of the pmf of the total transition time from the source to the sink. Since Laplace transform has many of the same properties as  $z$ -transform, by properly defining the branch transmittances, we can get the Laplace transform of the pdf of the total transition time from the source to the sink.

Next, we discuss how to define the branch transmittances. In the following, instead of using the transition probability, we use branch transmittance, denoted by  $g_{ij}$ , to associate with the state transition from  $i$  to  $j$ . The transmittance is defined as  $g_{ij} = P_{ij} \cdot e^{-st}$ , where  $P_{ij}$  is the transition probability from state  $i$  to  $j$  (i.e., data token passing probability from node  $i$  to  $j$ ),  $t$  is the transition time from state  $i$  to  $j$  (i.e., the time duration that the process remains in state  $i$  before transiting to state  $j$ ), and  $s$  is a dummy variable. For a specific state transition, the transition time may not be a constant. For example, considering the transition from state 2 to 3, when node 2 has packets to send, it takes  $T_{data}$  to transit to state 3; otherwise, it takes  $T_{token}$  to transit to state 3. Considering the presence of voice traffic, the data token holder has to wait for

the voice traffic to be transmitted first. During the voice packet inter-arrival time  $T_v$ , on average, the  $i$ th data node holds the token  $T_v/R_i$  times, where  $R_i$  is given in (5). So for each data transmission, the time delayed by voice traffic is given by  $T_{delay} = \frac{\bar{T}}{T_v \sum_{i=1}^{N_d} \frac{1}{R_i}}$ , where  $\bar{T}$  is given in (3). So the term  $e^{-st}$  is re-written as  $po_2 \cdot e^{-s(T_{token}+T_{delay})} + (1-po_2) \cdot e^{-s(T_{data}+T_{delay})}$ , where  $po_2$  is the probability that node 2 has no packet to send. As discussed in Section IV, we have  $po_2 = 1 - \rho_2$ . Notice that node 1 is assumed to be backlogged, so the transition time from state 1\_a to other states is a constant (i.e.,  $T_{data} + T_{delay}$ ). From the above discussion, we have

$$g_{ij} = \begin{cases} P_{ij} e^{-s(T_{data}+T_{delay})}, & i = 1 \\ P_{ij} \cdot [po_j \cdot e^{-s(T_{token}+T_{delay})} \\ + (1-po_j) \cdot e^{-s(T_{data}+T_{delay})}], & i \neq 1. \end{cases}$$

According to [21], the signal transfer function from the source 1\_a to the sink 1\_b, denoted by  $STF(1_a, 1_b)$ , is given by

$$STF(1_a, 1_b) = g_{11} + \frac{g_{13} \cdot g_{31}}{1 - g_{33}} + \frac{(g_{12} + \frac{g_{13} \cdot g_{32}}{1 - g_{33}}) \cdot (g_{21} + \frac{g_{23} \cdot g_{31}}{1 - g_{33}})}{1 - g_{22} - \frac{g_{32} \cdot g_{23}}{1 - g_{33}}}. \quad (9)$$

Based on the conclusion drawn in [20],  $B^*(s)$  is simply equal to  $STF(1_a, 1_b)$ .

### B. Derivation of $H_1^*(s)$ and $H_2^*(s)$

Recall that  $H_1^*(s)$  is the Laplace transform of the pdf of the random variable  $t_1$ , which is the time period from the moment that the last packet in the queue leaves a data node to the moment that the node gets a new chance to transmit. Let random variable  $x$  denote the token recurrence time of a backlogged data node, whose pdf has Laplace transform  $B^*(s)$ . By the definition of  $t_1$  and  $x$ , we have  $t_1 = x - T_{data}$ . According to the property of Laplace transform, we have  $H_1^*(s) = B^*(s)e^{-sT_{data}}$ .

$H_2^*(s)$  is the Laplace transform of the pdf of the random variable  $t_2$ , which is the token recurrence time of an empty data node. The derivation of  $H_2^*(s)$  is similar to that of  $B^*(s)$ . The only difference is in the calculation of  $g_{ij}$ . By the definition of  $t_2$ , node 1 (corresponding to state 1\_a in Fig. 9) is empty now. So the transition time taken from state 1\_a to any other state is  $T_{token} + T_{delay}$ . Similarly, the term  $e^{-st}$  is re-written as  $po_j \cdot e^{-s(T_{token}+T_{delay})} + (1-po_j) \cdot e^{-s(T_{data}+T_{delay})}$  ( $j = 2, 3$ ) for the transitions from node 2 and node 3. Thus, for  $H_2^*(s)$ , we have

$$g_{ij} = \begin{cases} P_{ij} e^{-s(T_{token}+T_{delay})}, & i = 1 \\ P_{ij} \cdot [po_j \cdot e^{-s(T_{token}+T_{delay})} \\ + (1-po_j) \cdot e^{-s(T_{data}+T_{delay})}], & i \neq 1. \end{cases}$$

Applying  $g_{ij}$  to (9), we get  $H_2^*(s)$ .

### ACKNOWLEDGEMENT

The authors wish to thank the anonymous reviewers for their helpful reviews and suggestions which improved the quality and presentation of this paper.

## REFERENCES

- [1] IEEE 802.11 WG, IEEE 802.11e/D11, "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Medium Access Control(MAC) Quality of Service (QoS) Enhancements," Oct. 2004.
- [2] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1353-1368, 1999.
- [3] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *Proc. IEEE INFOCOM'01* pp. 209-218, 2001.
- [4] M. Barry, A. T. Campbell, and A. Veres, "Distributed control algorithms for service differentiation in wireless packet networks," in *Proc. IEEE INFOCOM'01*, pp. 582-590, 2001.
- [5] J. W. Robinson and T. S. Randhawa, "Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 22, no. 5, pp. 917-928, June 2004.
- [6] C. Dovrolis and P. Ramanathan, "A case for relative differentiated services and the proportional differentiation model," *IEEE Network*, vol. 13, no. 5, pp. 26-34, 1999.
- [7] R.-G. Cheng, C.-Y. Wang, L.-H. Liao, and J.-S. Yang, "Ripple: a wireless token-passing protocol for multi-hop wireless mesh networks," *IEEE Commun. Lett.*, vol. 10, no. 2, pp. 123-125, Feb. 2006.
- [8] X. Lu, G. Fan, and R. Hao, "A dynamic token passing MAC protocol for mobile ad hoc networks," in *Proc. ACM IWCMC'06*, pp. 743-748, July 2006.
- [9] M. Ergen, D. Lee, R. Sengupta, and P. Varaiya, "WTRP-wireless token ring protocol," *IEEE Trans. Veh. Technol.*, vol. 53, no. 6, pp. 1863-1881, 2004.
- [10] D. Chen, J. Li, and J. Ma, "Multiple access protocol for WLAN based on adaptive token passing with fairness guarantee," in *Proc. IEEE AINA'06*, vol. 2, Apr. 2006.
- [11] D. Lin, T.-S. Moh, and M. Moh, "A delay-bounded multi-channel routing protocol for wireless mesh networks using multiple token rings: extended summary," in *Proc. IEEE ICLCN'06*, pp. 845-847, Nov. 2006.
- [12] L. Wang, Y. Lin, and Z. Chen, "A distributed tokens-based distance-vector routing algorithm for mobile ad-hoc networks," in *Proc. IEEE ICCNMC'03*, pp. 474-477, Oct. 2003.
- [13] IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, Standard, IEEE, Aug. 1999.
- [14] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: part I-carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1400-1416, Dec. 1975.
- [15] N. Metropolis *et al.*, "Equations of state calculations by fast computing machines," *J. Gem. Phys.*, vol. 21, pp. 1087-1092, 1953.
- [16] W. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97-109, 1970.
- [17] P. Wang, H. Jiang, and W. Zhuang, "Capacity improvement and analysis for voice/data traffic over WLAN," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1530-1541, Apr. 2007.
- [18] L. Kleinrock, *Queueing Systems*, vol. 1. New York: Wiley, 1975.
- [19] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.
- [20] H. Zhai, Y. Kwon, and Y. Fang, "Performance analysis of IEEE 802.11 MAC protocols in wireless LANs," *Wireless Commun. Mobile Comput.*, vol. 4, no. 8, pp. 917-931, Dec. 2004.
- [21] L. P. A. Robichaud, M. Boisvert, and J. Robert, *Signal Flow Graphs and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1962.



**Ping Wang** (S'08) received the B.E. and M.E. degrees in 1994 and 1997, respectively, both in electrical engineering, from Huazhong University of Science and Technology, Wuhan, China. She is currently working toward a Ph.D. degree at the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Her current research interests include QoS provisioning and resource allocation in multimedia wireless communications. She is a co-recipient of a Best Paper Award from IEEE ICC 2007.



**Weihua Zhuang** (M'93-SM'01-F'08) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, China, and the Ph.D. degree from the University of New Brunswick, Canada, all in electrical engineering. Since October 1993, she has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, where she is a Professor. She is a co-author of the textbook *Wireless Communications and Networking* (Prentice Hall, 2003). Her current research interests include wireless communications and networks, and radio positioning. Dr. Zhuang is a co-recipient of a Best Paper Award from IEEE ICC 2007, a Best Student Paper Award from IEEE WCNC 2007, and the Best Paper Award from QShine 2007. She received the Outstanding Performance Award in 2005 and 2006 from the University of Waterloo for outstanding achievements in teaching, research, and service, and the Premier's Research Excellence Award (PREA) in 2001 from the Ontario Government for demonstrated excellence of scientific and academic contributions. She is the Editor-in-Chief of *IEEE Transactions on Vehicular Technology*, and an Editor of *IEEE Transactions on Wireless Communications*, *EURASIP Journal on Wireless Communications and Networking*, and *International Journal of Sensor Networks*.