A Distributed Channel Access Scheme with Guaranteed Priority and Enhanced Fairness

Hai Jiang, *Member, IEEE,* Ping Wang, and Weihua Zhuang, *Senior Member, IEEE*

***Abstract*— Although the IEEE 802.11e enhanced distributed channel access (EDCA) can differentiate high priority traffic such as real-time voice from low priority traffic such as delay-tolerant data, it can only provide statistical priority, and is characterized by inherent short-term unfairness. In this paper, we propose a new distributed channel access scheme through minor modifications to EDCA. Guaranteed priority is provided to real-time voice traffic over data traffic, while a certain service time and short-term fairness enhancement are provided to data traffic. We also present analytical models to calculate the percentage of time to serve voice traffic and the achieved data throughput. Both analysis and simulation demonstrate the effectiveness of our proposed scheme.**

***Index Terms*— IEEE 802.11e EDCA, distributed channel access, quality of service, priority, short-term fairness.**

I. Introduction

THE mandatory distributed coordination function (DCF) in the IEEE 802.11 [1] is the most popular medium access control (MAC) mechanism for distributed wireless access. It is based on the carrier sense multiple access with collision avoidance (CSMA/CA). In DCF, after sensing the channel being idle for a duration of distributed interframe space (DIFS), a node chooses a random backoff timer uniformly distributed in its contention window (CW). The backoff timer counts down when the channel is continuously idle for more than DIFS, and freezes when a transmission from other nodes is sensed. The node transmits its frame when the backoff timer reaches zero. The CW is set to the initial value $W_{\min}$ for the first transmission attempt for the target frame, doubled upon a collision until the maximum value $W_{\max}$ is reached, and reset to $W_{\min}$ upon a successful delivery. Each node can also use the optional request-to-send (RTS)/clear-to-send (CTS) dialogue before DATA frame (i.e., the information frame from the sender) transmission to alleviate the hidden terminal problem in a multi-hop environment.

Due to the lack of a centralized controller, it is challenging to achieve quality of service (QoS) in terms of delay, delay jitter, and fairness in distributed channel access. To enhance the IEEE 802.11 MAC, IEEE 802.11e [2] proposes new features with QoS provisioning to real-time applications. As an extension of DCF, the enhanced distributed channel access (EDCA) in IEEE 802.11e has provided a priority feature among access categories (ACs) by classifying the arbitration interframe space (AIFS), and the initial ($W_{\min}$) and maximum ($W_{\max}$) contention window sizes. High priority traffic (e.g., real-time voice) has smaller AIFS, $W_{\min}$ and $W_{\max}$ values, and thus is more likely to get access to the channel than low priority traffic. However, EDCA provides only statistically rather than guaranteed prioritized access to high priority traffic [3]. Such statistically prioritized access is difficult to satisfy the delay requirement of each high priority packet[1]. Furthermore, high priority traffic can suffer performance degradation due to low priority traffic offering heavy loads [4]. In addition, there is no QoS guarantee for low-priority traffic in EDCA.

On the other hand, DCF and EDCA are characterized by inherent short-term unfairness [5]. A successful transmission from a node will set its CW to $W_{\min}$, giving its following packets a large chance to be served before packets in other nodes with a larger CW. It can be seen that the short-term unfairness results from the channel access policy of DCF and EDCA.

The objective of this paper is to address these challenges. We base our work on IEEE 802.11e EDCA. With minor modifications to EDCA, we propose a scheme to provide guaranteed priority to voice traffic over data traffic and, at the same time, provide data traffic with a certain amount of service share and short-term fairness improvement.

The rest of this paper is organized as follows. The related work is reviewed in Section II. Section III presents our proposed channel access scheme. In Section IV, we provide the performance analysis for voice and data traffic. Numerical results and discussion are given in Section V, followed by conclusion remarks in Section VI. As many symbols are used in this paper, Table I summarizes the important ones.

II. Related work

A. Priority Access

To enhance the priority provisioning of IEEE 802.11 DCF, the approaches in [6], [7] scale the CW or the IFS according to the priority of traffic in the node. In [8], [9], after waiting for the channel to be idle for an IFS time, the node jams the channel by pulses of energy, called black burst, the length of which is determined by how long the node has been waiting or by the traffic priority. The node with the longest black burst

Manuscript received September 20, 2005; revised March 10, 2006; accepted March 16, 2006. The associate editor coordinating the review of this paper and approving it for publication was Y.-B. Lin. This work was supported by a research grant from the Natural Science and Engineering Research Council (NSERC) of Canada.

H. Jiang is with the Electrical Engineering Department, Princeton University, Princeton, New Jersey 08544, USA (e-mail: haijiang@princeton.edu).

P. Wang and W. Zhuang are with the 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.2007.05746.

[1]In this paper, a network layer packet is transmitted by a link layer frame. Hence the terms "packet" and "frame" are used interchangeably.

TABLE I
SUMMARY OF IMPORTANT SYMBOLS USED.

Symbol	Definition
k_{v1} (k_{v2})	Number of transmissions from voice nodes with W_{v1} (W_{v2}) in an event
k_{d1} (k_{d2}, k_{d3})	Number of transmissions from data nodes with W_{d1} (W_{d2}, W_{d3}) in an event
N_d	Number of data nodes in the system
N_v	Number of voice nodes in the system
n_{d1} (n_{d2}, n_{d3})	Number of data nodes with contention window W_{d1} (W_{d2}, W_{d3})
n_v	Number of voice nodes at on state
n_v^c	Number of on voice nodes having backlogged packets for transmission
n_{v1}^c (n_{v2}^c)	Number of backlogged on voice nodes with contention window W_{v1} (W_{v2})
$\overline{p_d^c}$	Average collision probability of transmissions from data nodes
$p_d^s(n_{d1}, n_{d2})$/$p_d^c(n_{d1}, n_{d2})$	Success/collision probability of an event from state (n_{d1}, n_{d2})
$p_d^t(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3})$	Transition probability if the current state is (n_{d1}, n_{d2}) and the numbers of transmissions from data nodes with W_{d1}, W_{d2}, and W_{d3} are k_{d1}, k_{d2} and k_{d3}, respectively
$p_v^{ct}(i, j)$	Transition probability for n_v^c from state i to state j
$p_v^t(n_{v1}^c, n_{v2}^c;\ k_{v1}, k_{v2})$	Transition probability if the current state is (n_{v1}^c, n_{v2}^c) and the numbers of transmissions from voice nodes with W_{v1} and W_{v2} are k_{v1} and k_{v2}, respectively
$s_d(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3})$	The next state if the current state is (n_{d1}, n_{d2}) and the numbers of transmissions from data nodes with W_{d1}, W_{d2}, and W_{d3} are k_{d1}, k_{d2} and k_{d3}, respectively
$s_v(n_{v1}^c, n_{v2}^c;\ k_{v1}, k_{v2})$	The next state if the current state is (n_{v1}^c, n_{v2}^c) and the numbers of transmissions from voice nodes with W_{v1} and W_{v2} are k_{v1} and k_{v2}, respectively
$T_v(n_{v1}^c, n_{v2}^c)$	Average time needed for transitions from state (n_{v1}^c, n_{v2}^c) to the absorbing state (0, 0)
t_a	Voice packet inter-arrival time
$t_d(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3})$	Average time for transition if the current state is (n_{d1}, n_{d2}) and the numbers of transmissions from data nodes with W_{d1}, W_{d2}, and W_{d3} are k_{d1}, k_{d2} and k_{d3}, respectively
t_d^{BO}	Average backoff time in an event of data nodes
$t_v(n_{v1}^c, n_{v2}^c;\ k_{v1}, k_{v2})$	Average time for transition if the current state is (n_{v1}^c, n_{v2}^c) and the numbers of transmissions from voice nodes with W_{v1} and W_{v2} are k_{v1} and k_{v2}, respectively
$t_v^c(i)$	Average residence time in state $n_v^c = i$ before it transits to the new state (which may be the same as state i)
$\{W_{d1},\ W_{d2},\ W_{d3}\}$	The set of possible contention window sizes in the analysis for data traffic
$W_{\min}$ ($W_{\max}$)	Initial (maximum) contention window size
$\{W_{v1},\ W_{v2}\}$	The set of possible contention window sizes in the analysis for voice traffic
$1/\alpha$ ($1/\beta$)	Mean on (off) duration of traffic from a voice node
ϵ	Upper bound of the outage probability that data nodes cannot get required service
$\rho(n_v)$	Average portion of time used by voice packets when there are n_v on voice nodes
τ	Slot time duration
ϕ	Lower bound of the portion of service time that data nodes should receive

signal wins the channel. The scheme in [10] uses a binary count down mechanism to provide access priority. Each node has a binary number that is determined by its traffic class and its waiting time. The highest channel access priority is then obtained by the node with the highest binary number. In [3], dual busy tones are used to provide priority. The high priority source node sends a busy tone during carrier sensing period (DIFS + backoff time) before transmission. Any other node hearing the busy tone will send a busy tone at another busy tone channel. Upon hearing either busy tone, low priority source nodes will defer their transmissions.

B. Fairness

For distributed random channel access, the access to the common medium by each node can be controlled by the evolution of the backoff timer, which is bounded by the CW. Distributed fair scheduling can be achieved by adjusting the CW or IFS according to the difference between expected and actually obtained services [11], [12]. In the distributed weighted fair queueing (DWFQ) mechanism [11], each contending node adjusts its CW dynamically either according to the difference between the experienced throughput and the desirable throughput, or according to the comparison of the ratio of its throughput to its assigned weight with those of other nodes. In the distributed deficit round-robin (DDRR) mechanism [12], each node has a deficit counter that is accumulated by the required service rate and is reduced by the service actually received. Hence, the deficit counter reflects the difference between the received service and the required service. The IFS in the backoff procedure is determined such that a node with a larger deficit counter has a smaller IFS, making lagging traffic flows transmit earlier and achieving a certain level of fairness. The distributed self-clocked fair queuing (SCFQ) is adopted in [13], [14] to achieve fairness.

In [15], a collision-resolution mechanism called GDCF is proposed to enhance the throughput performance of IEEE 802.11 DCF. DCF resets CW to the initial value upon a successful transmission. On the contrary, GDCF halves the CW when the target source node has c consecutive successful transmissions. Hence, the CW in GDCF is gradually decreased during consecutive frame deliveries, to achieve a better contention resolution. Enhanced long-term fairness is also observed.

One drawback of the above schemes is that most of them focus only on priority access or on fairness. Few of them provide a solution to both issues. Further, the schemes require non-trivial modifications to the current MAC, e.g., the IEEE 802.11 MAC series.

In this research, we consider a single-hop fully-connected wireless network. Real-time voice and elastic data applications are supported in the network. Each admitted voice node and data node will issue a one-way voice and data transmission, respectively. We propose a new distributed channel access scheme with minor modifications to the IEEE 802.11e EDCA, to achieve guaranteed priority for voice and improved short-term fairness for data.

III. The Proposed Distributed Channel Access

In this research, voice traffic is represented by an `on/off` model: active voice users (at the `on` state) transmit with a constant rate and inactive users (at the `off` state) do not transmit. The durations of the states are independent and exponentially distributed. Data traffic flows are long-lived file transmissions.

Voice traffic is assigned a higher priority over the data traffic. Transmission delay and jitter are the main QoS parameters of voice. Throughput and fairness are the QoS indication for data. As voice traffic is sensitive to delay, it should be guaranteed that voice nodes can access the channel successfully when needed, i.e., guaranteed access priority should be provided to voice over data.

Inspired by the idea of black-burst contention [8], here we propose an efficient distributed scheme to provide guaranteed access priority to voice, by minor modifications to EDCA. In our scheme, the AIFSs for voice traffic and data traffic remain the same as those in EDCA. The contention behaviors of voice and data nodes are modified as follows. For a contending voice (or data) node, after waiting for the channel to be idle for an AIFS[AC_voice] (or AIFS[AC_data]), instead of further waiting for the channel to be idle for a duration of the backoff time, the node will send a black burst[2] to jam the channel, and the length of the black burst (in the unit of slot time) is equal to its backoff timer. After the completion of its own black burst, the node monitors the channel for the duration of a slot time. If the channel is still busy (which means at least one other node is sending a black burst), the node will quit the current contention, keep its contention window, choose another backoff timer randomly from its contention window, and wait for the channel to be idle for AIFS[AC_voice] (or AIFS[AC_data]) again; otherwise, the node (which sends the longest black burst) will transmit its packet. The transmission may be successful (thus an ACK is received), or collided (thus the expected ACK is not received). In the same way as that in EDCA, collided nodes[3] in our scheme double their CW until the maximum CW value is reached, and a successful transmission will reset a node's CW to the initial value. Each data transmission follows the optional RTS-CTS-DATA-ACK as in EDCA. Due to the small payload size of a voice packet, voice nodes do not use the RTS/CTS dialogue.

As long as at least one voice contender exists, all the data nodes will sense the black burst by voice node(s) during the AIFS[AC_data] (>AIFS[AC_voice]), and defer their transmissions. When data nodes are contending for the channel, there should be no voice contender. Hence, the channel access consists of two modes: voice contending with voice, and data contending with data. The two modes alternate with time. For the voice (or data) contention, a node with the longest black burst (i.e., the largest backoff timer) transmits its packet. The transmission is successful if there is only one node with the largest backoff timer, or collided if there are at least two nodes with the (same) largest backoff timer. It can be seen that guaranteed priority for voice is provided by the difference between AIFS[AC_data] and AIFS[AC_voice]. Therefore, the initial and maximum CW values for data can be the same as those for voice.

By using our scheme, it seems that the waiting time (before getting the channel) of a node is longer than that in EDCA if the same CW settings are applied, because the node with the largest backoff timer instead of the smallest backoff timer (as in EDCA) wins the channel. However, as shown in Section V, the initial and maximum CW sizes for voice or data AC in our scheme can be set to much smaller values than those in EDCA. For example, we can set $W_{\min} = 3$ and $W_{\max} = 15$ in our scheme for up to 500 contending nodes, compared with $W_{\min} = 31$ and $W_{\max} = 1023$ in EDCA. Hence, the negative effect of the above largest-backoff-timer-winner is compensated by much smaller contention windows (and thus much smaller backoff timers).

In our scheme (or EDCA), when the packet from a node is collided, the node doubles its CW, making it more (or less) likely to choose the largest (or smallest) backoff timer, i.e., more (or less) likely to win the channel in the next contention; when a node transmits successfully, its CW will be reset to $W_{\min}$, and its chance to win the channel again will be smaller (or larger). Thus, our scheme distributes the channel access time more fairly in short term to the contending nodes than EDCA.

[2] A busy tone can also be used instead of a black burst.

[3] By the term "collided nodes", we mean the nodes whose transmitted packets are collided.

The difference of our scheme from those in [8], [9] which also use black burst is as follows. In [8], black burst is only used for real-time traffic, and the length of a black burst is proportional to the experienced delay. In [9], all traffic classes adopt a black burst. Different classes are differentiated by the length of black bursts. In our scheme, different classes are differentiated by different AIFSs and the black burst length is determined by the backoff timer value.

IV. Performance Analysis

Consider N_v voice nodes and N_d data nodes in the system. Voice traffic has guaranteed priority over data traffic. The unused time by the voice nodes is shared by all the data nodes. In the following, we first derive the average time to serve the voice nodes, then calculate the achieved data throughput.

A. Average Service Time for Voice Nodes

Based on the voice `on/off` model, at any time, there are a number, n_v ($\leq N_v$) [4], of voice nodes at the `on` state. In the following, we first derive the average percentage of time needed to serve the n_v `on` voice nodes.

1) Service Time for n_v `on` Voice Nodes: Given the arrival time of the first packet of the `on` burst, the packet arrivals of an `on` voice node are deterministic (one arrival after each t_a, the packet inter-arrival duration). However, in the `on/off` voice traffic model, the arrival time of the first packet of an `on` burst is random. Hence, we model the packet arrivals of an `on` voice node as follows: for every duration T less than t_a, the probability of a voice packet arrival is given by T/t_a.

The voice packet departure process of the n_v `on` voice nodes is a random process. The duration from a successful packet delivery to the next is a variable, which may include DATA and ACK transmission time, collision time, and backoff time.

From the random packet arrival and departure processes, the number of total backlogged voice packets at all the n_v nodes varies with time, denoted by n_v^c. For simplicity of analysis, we assume that the probability of a voice node having two or more backlogged voice packets is negligible, and each `on` voice node has up to one backlogged voice packet. This assumption is to be validated in Section V. Hence, the number of voice nodes contending for the channel is also n_v^c, which should be bounded by n_v. Define an "event" as a successful transmission or a collision. We sample the value of n_v^c after each event except the collision event of voice nodes. As voice traffic is provided guaranteed priority over data traffic, data successful transmission or collision events only happen when there is no contending voice node. The sampled n_v^c value changes with time, and can be characterized by a process with state space $\{0, 1, ..., n_v\}$. Fig. 1 shows an example of the sampled n_v^c process with 5 `on` voice nodes (i.e., $n_v = 5$) and 2 data nodes. At time t_1, assume all backlogged voice packets have been transmitted successfully, thus $n_v^c = 0$, and the data nodes begin to contend for the channel. At time t_2, the two data node transmissions are collided. As there is no voice packet

[4]The value of n_v varies with time t. For the simplicity of presentation, here we omit the index t for n_v and the n_v^c discussed later.

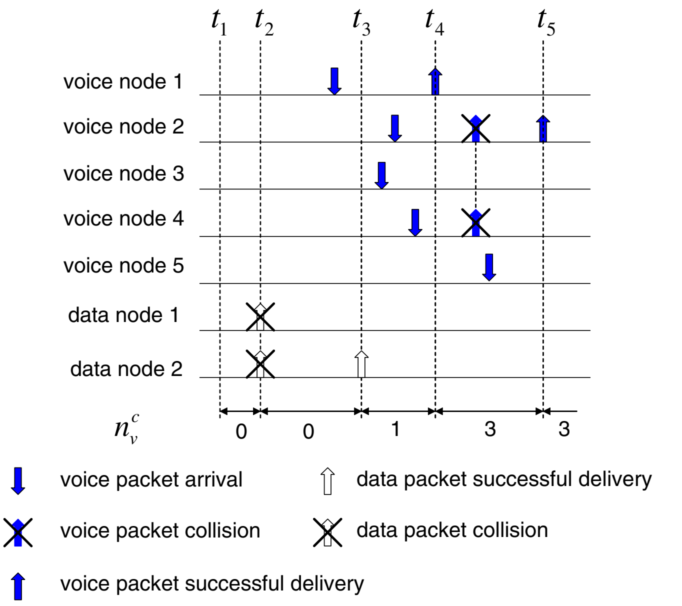

Fig. 1. The state process of sampled n_v^c.

arrival in $(t_1, t_2]$, n_v^c remains 0, and data nodes continue to contend. At time t_3, a packet from data node 2 is transmitted successfully. From this point, n_v^c changes to 1 as one packet of voice node 1 arrives during $(t_2, t_3]$. At time t_4, the packet of voice node 1 is transmitted successfully, and n_v^c changes to 3 as three packets arrive from voice nodes 2, 3, and 4 during $(t_3, t_4]$. At time t_5, the packet of voice node 2 is delivered successfully, and n_v^c remains 3 as a voice packet has arrived from voice node 5. Note that although there is a collision event of voice packets from nodes 2 and 4 during $(t_4, t_5]$, we do not sample the n_v^c value at a collision event of transmissions from voice traffic.

Let $t_v^c(i)$ denote the average residence time in state $n_v^c = i$ before it transits to the new state (which may be the same as state i) upon the next event (except the collision event of voice nodes). It is actually the duration between the previous and the next events (except the collision event of voice nodes) at state $n_v^c = i$. When $i > 0$, $t_v^c(i)$ is the average time to serve one (backlogged `on`) voice node at state $n_v^c = i$. Let $p_v^{ct}(i, j)$ denote the state transition probability of n_v^c from state i to j.

For state $n_v^c = 0$, as no voice nodes contend for the channel, the data nodes will occupy the channel. During the transmissions of data packets, any newly arrived voice packet will have to wait (before contending for the channel) until the completion of a successful transmission or collision event of data nodes. If the average backoff time in an event of data nodes is t_d^{BO}, and the average collision probability of data transmissions is $\overline{p_d^c}$, we have

$$t_v^c(0) = t_d^{BO} + \tau + \overline{p_d^c} \cdot T_d^c + (1 - \overline{p_d^c}) \cdot T_d^s \tag{1}$$

where τ is the slot time duration and in the above equation it means the black burst detection time after a node finishes its own black burst, T_d^s and T_d^c are the durations (excluding the backoff time) for a successful transmission and for a collision of data nodes, respectively, given by

$$\begin{cases} T_d^s = \text{AIFS[AC\_data]} + t_{\text{RTS}} + \text{SIFS} + t_{\text{CTS}} \\ \qquad + \text{SIFS} + t_{d\_\text{DATA}} + \text{SIFS} + t_{\text{ACK}} \\ T_d^c = \text{AIFS[AC\_data]} + t_{\text{RTS}} + t_{\text{CTS\_TIMEOUT}} \end{cases} \quad (2)$$

SIFS is the short interframe space, t_{RTS}, t_{CTS}, t_{d_DATA}, t_{ACK} are the time to transmit an RTS frame, a CTS frame, a DATA frame from a data node, and an ACK frame, respectively, and $t_{\text{CTS\_TIMEOUT}}$ is the CTS timeout value. The values of $\overline{t_d^{BO}}$ and $\overline{p_d^c}$ are to be derived in Section IV-B. Further, we have

$$p_v^{ct}(0,j) = \binom{n_v}{j}(\frac{t_v^c(0)}{t_a})^j(1-\frac{t_v^c(0)}{t_a})^{n_v-j}, \quad 0 \le j \le n_v. \quad (3)$$

For state $n_v^c = i > 0$, i voice nodes are backlogged, each having one packet. The average state residence time $t_v^c(i)$ is derived in the Appendix. During $t_v^c(i)$, the packet from a voice node is transmitted successfully, and the other $n_v - i$ unbacklogged on voice nodes may have packet arrivals. Hence,

$$p_v^{ct}(i,i+j) = \binom{n_v-i}{j+1}(\frac{t_v^c(i)}{t_a})^{(j+1)}(1-\frac{t_v^c(i)}{t_a})^{n_v-i-(j+1)}, \quad -1 \le j \le n_v - i - 1. \quad (4)$$

For the process of the sampled n_v^c, based on the above state transition probabilities, we can derive the steady-state probability $\pi(i)$, $0 \le i \le n_v$. The average portion of time used by voice packets when there are n_v on voice nodes is given by

$$\rho(n_v) = \frac{\sum_{i=1}^{n_v}\pi(i)\cdot t_v^c(i)}{\sum_{i=0}^{n_v}\pi(i)\cdot t_v^c(i)}. \quad (5)$$

2) Service Time for N_v Voice Nodes in the System: For the N_v voice nodes, the traffic from each node follows an on/off model, and the durations of the on and off states are exponentially distributed with mean value $1/\alpha$ and $1/\beta$, respectively. Hence, at any time instant, each voice node is on with probability $\beta/(\alpha+\beta)$. Then the probability mass function of n_v is given by

$$\text{Prob}\{n_v = i\} = \binom{N_v}{i}(\frac{\beta}{\alpha+\beta})^i(1-\frac{\beta}{\alpha+\beta})^{N_v-i} \quad (6)$$

and the average portion of time to serve N_v on/off voice nodes is $\sum_{i=1}^{N_v}\text{Prob}\{n_v = i\}\cdot\rho(i)$.

For a given N_v, the n_v varies with time. We refer the duration over which n_v does not change as n_v *residence time*. The n_v residence time varies depending on the arrival and completion of voice on bursts. In order not to starve data nodes, we need to limit the number of voice node N_v admitted into the system. We require that, during an n_v residence time, the data nodes should have a portion of service time no less than ϕ $(0 < \phi < 1)$. The outage probability that data nodes cannot get the required service share needs to be bounded by a threshold ϵ. The outage probability with N_v voice nodes is given by

$$\sum_{i=1}^{N_v}\text{Prob}\{n_v = i\}\cdot I\{\rho(i) > 1-\phi\}$$

where the indication function

$$I\{x\} = \begin{cases} 1, & \text{if } x \text{ is true} \\ 0, & \text{if } x \text{ is false.} \end{cases} \quad (7)$$

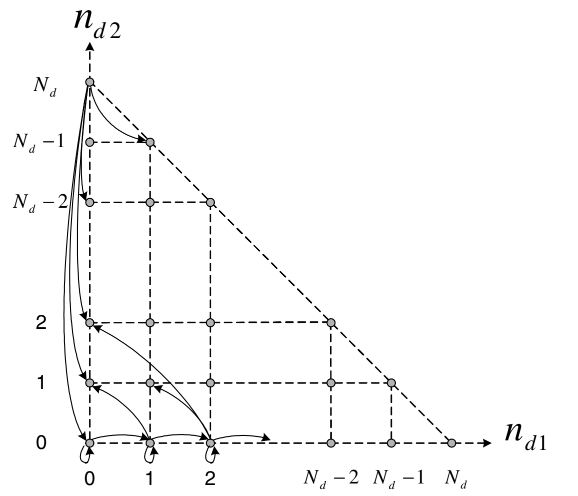

Fig. 2. The transition diagram for state (n_{d1}, n_{d2}) of data nodes.

Then the maximum number of voice nodes that can be admitted is given by

$$N_v^{\max} = \max\{N_v : \sum_{i=1}^{N_v}\text{Prob}\{n_v = i\}\cdot I\{\rho(i) > 1-\phi\} < \epsilon\}. \quad (8)$$

B. Throughput of Data traffic

With N_v voice nodes, the portion of time unused by voice nodes, $1-\sum_{i=1}^{N_v}\text{Prob}\{n_v = i\}\cdot\rho(i)$, is shared by all the data nodes. As voice contention period and data contention period alternate with time, the achieved data throughput can be approximated by the product of the portion of data time and the data throughput when $N_v = 0$. Hence, it is essential to calculate the throughput of our system with only data nodes.

Consider the case when $N_v = 0$ and N_d data nodes contend for the channel. For simplicity of presentation, the CW of each data node takes values from the set $\{W_{d1}, W_{d2}, W_{d3}\}$ where $W_{d2} = 2\cdot(W_{d1}+1)-1$ and $W_{d3} = 2\cdot(W_{d2}+1)-1$. Our analysis can be easily extended to a case with more choices for the CW size. Define state (n_{d1}, n_{d2}), where n_{d1} and n_{d2} are the numbers of data nodes with contention windows of W_{d1} and W_{d2}, respectively. The number of data nodes with a contention window of W_{d3} is $n_{d3} = N_d - n_{d1} - n_{d2}$. The state transition diagram after each event (i.e., a successful transmission or a collision) is shown in Fig. 2. For example, state (2, 0) means 2 data nodes with W_{d1}, 0 node with W_{d2}, and N_d-2 nodes with W_{d3}. Its next state is (3, 0) if one node with W_{d3} transmits successfully, (0, 2) if the two nodes with W_{d1} collide, (1, 1) if only one node with W_{d1} and at least one node with W_{d3} collide, remains at (2, 0) if no node with W_{d1} and at least 2 nodes with W_{d3} collide, or if one node with W_{d1} transmits successfully.

For a state (n_{d1}, n_{d2}), an event (i.e., one or more transmissions from the nodes) will lead to the next state (which may be the same as (n_{d1}, n_{d2})). For an event, denote the numbers of transmissions from nodes with W_{d1}, W_{d2}, and W_{d3} as $k_{d1}(\le n_{d1})$, $k_{d2}(\le n_{d2})$, and $k_{d3}(\le n_{d3})$, respectively, where $k_{d1}+k_{d2}+k_{d3} > 0$. The event is a successful transmission if $k_{d1}+k_{d2}+k_{d3}=1$, or a collision otherwise. Denote the next state after the event as $s_d(n_{d1},n_{d2};\ k_{d1},k_{d2},k_{d3})$. Then we have

$$s_d(n_{d1},n_{d2};\ k_{d1},k_{d2},k_{d3}) = \begin{cases} (n_{d1}+k_{d2}+k_{d3}, n_{d2}-k_{d2}), \\ \quad \text{if } k_{d1}+k_{d2}+k_{d3}=1 \\ (n_{d1}-k_{d1}, n_{d2}+k_{d1}-k_{d2}), \\ \quad \text{if } k_{d1}+k_{d2}+k_{d3}>1. \end{cases} \tag{9}$$

Denote the probability of the above transition as $p_d^t(n_{d1},n_{d2};\ k_{d1},k_{d2},k_{d3})$, and the average time of the transition as $t_d(n_{d1},n_{d2};\ k_{d1},k_{d2},k_{d3})$. If $k_{d1}\neq 0$, the successful transmission or collision happens when the largest backoff timer among all the data nodes takes values from [0, W_{d1}]. For notation simplicity, we define

$$f(x,y,a,b)=\binom{x}{y}a^y\cdot b^{x-y}. \tag{10}$$

We have

$$\begin{aligned} p_d^t(n_{d1},n_{d2};\ k_{d1},k_{d2},k_{d3}) &= \sum_{i=0}^{W_{d1}} f(n_{d1},k_{d1},\frac{1}{W_{d1}+1},\frac{i}{W_{d1}+1}) \\ &\quad \cdot f(n_{d2},k_{d2},\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \end{aligned} \tag{11}$$

where the three terms in the summation mean the probabilities that k_{d1}, k_{d2}, and k_{d3} data nodes with contention window size W_{d1}, W_{d2}, W_{d3} choose a backoff timer value i while the other $n_{d1}-k_{d1}$, $n_{d2}-k_{d2}$, $n_{d3}-k_{d3}$ data nodes choose backoff timer values less than i, respectively.

With the condition of the above transition, the conditional probability that the largest backoff timer value in the successful transmission or collision is i can be given by

$$\begin{aligned} &\frac{1}{p_d^t(n_{d1},n_{d2};\ k_{d1},k_{d2},k_{d3})} \\ &\quad \cdot f(n_{d1},k_{d1},\frac{1}{W_{d1}+1},\frac{i}{W_{d1}+1}) \\ &\quad \cdot f(n_{d2},k_{d2},\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \end{aligned}$$

and we have

$$\begin{aligned} t_d(n_{d1},n_{d2};\ k_{d1},k_{d2},k_{d3}) &= \frac{1}{p_d^t(n_{d1},n_{d2};\ k_{d1},k_{d2},k_{d3})} \\ &\cdot \Bigg(\sum_{i=0}^{W_{d1}} f(n_{d1},k_{d1},\frac{1}{W_{d1}+1},\frac{i}{W_{d1}+1}) \\ &\quad \cdot f(n_{d2},k_{d2},\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1})\cdot i\cdot\tau\Bigg)+\tau+T_{xd}. \end{aligned} \tag{12}$$

On the right side of (12), the first term represents the time used by the black burst, the second term (i.e., τ) is the duration for black burst detection after a node finishes its own black burst, and the third term T_{xd} is the data collision or successful transmission time (excluding the backoff time), given by

$$T_{xd}=\begin{cases} T_d^s, & \text{if } k_{d1}+k_{d2}+k_{d3}=1 \\ T_d^c, & \text{if } k_{d1}+k_{d2}+k_{d3}>1. \end{cases} \tag{13}$$

If $k_{d1}=0$, the successful transmission or collision can happen when the largest backoff timer among all the data nodes takes values from [0, W_{d2}]. We have

$$\begin{aligned} p_d^t(n_{d1},n_{d2};\ 0,k_{d2},k_{d3}) &= \sum_{i=0}^{W_{d1}} f(n_{d1},0,\frac{1}{W_{d1}+1},\frac{i}{W_{d1}+1}) \\ &\quad \cdot f(n_{d2},k_{d2},\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \\ &+\sum_{i=W_{d1}+1}^{W_{d2}} f(n_{d2},k_{d2},\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \end{aligned} \tag{14}$$

$$\begin{aligned} t_d(n_{d1},n_{d2};\ 0,k_{d2},k_{d3}) &= \frac{1}{p_d^t(n_{d1},n_{d2};\ 0,k_{d2},k_{d3})} \\ &\cdot \Bigg(\sum_{i=0}^{W_{d1}} f(n_{d1},0,\frac{1}{W_{d1}+1},\frac{i}{W_{d1}+1}) \\ &\quad \cdot f(n_{d2},k_{d2},\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1})\cdot i\cdot\tau \\ &+\sum_{i=W_{d1}+1}^{W_{d2}} f(n_{d2},k_{d2},\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1})\cdot i\cdot\tau\Bigg)+\tau+T_{xd}. \end{aligned} \tag{15}$$

If $k_{d1}=0$ and $k_{d2}=0$, the successful transmission or collision can happen when the largest backoff timer among all the data nodes takes values from [0, W_{d3}]. We have

$$\begin{aligned} p_d^t(n_{d1}, n_{d2};\ 0,0,k_{d3}) &= \sum_{i=0}^{W_{d1}} f(n_{d1},0,\frac{1}{W_{d1}+1},\frac{i}{W_{d1}+1}) \\ &\quad \cdot f(n_{d2},0,\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \\ &+ \sum_{i=W_{d1}+1}^{W_{d2}} f(n_{d2},0,\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \\ &+ \sum_{i=W_{d2}+1}^{W_{d3}} f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \end{aligned} \quad (16)$$

$$\begin{aligned} t_d(n_{d1}, n_{d2};\ 0,0,k_{d3}) &= \frac{1}{p_d^t(n_{d1}, n_{d2};\ 0,0,k_{d3})} \\ &\cdot \Bigg(\sum_{i=0}^{W_{d1}} f(n_{d1},0,\frac{1}{W_{d1}+1},\frac{i}{W_{d1}+1}) \\ &\quad \cdot f(n_{d2},0,\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \cdot i \cdot \tau \\ &+ \sum_{i=W_{d1}+1}^{W_{d2}} f(n_{d2},0,\frac{1}{W_{d2}+1},\frac{i}{W_{d2}+1}) \\ &\quad \cdot f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \cdot i \cdot \tau \\ &+ \sum_{i=W_{d2}+1}^{W_{d3}} f(n_{d3},k_{d3},\frac{1}{W_{d3}+1},\frac{i}{W_{d3}+1}) \cdot i \cdot \tau \Bigg) \\ &+ \tau + T_{xd}. \end{aligned} \quad (17)$$

Denote the success and collision probabilities of an event from state (n_{d1}, n_{d2}) as $p_d^s(n_{d1}, n_{d2})$ and $p_d^c(n_{d1}, n_{d2})$, which are given by (18) and (19) at the top of next page, respectively.

Let $p_d(n_{d1}, n_{d2})$ denote the steady-state probability of state (n_{d1}, n_{d2}). From the state transition probabilities and $\sum_{(n_{d1},n_{d2})} p_d(n_{d1}, n_{d2}) = 1$, we can first obtain each $p_d(n_{d1}, n_{d2})$, and then obtain the system throughput (the average transmitted packet number per time unit) S_d and the average number of collisions per time unit C_d, given by (20) and (21) on next page, respectively. Further, the average collision probability of transmissions from data nodes is given by

$$\overline{p_d^c} = C_d/(C_d + S_d). \quad (22)$$

In each time unit, the durations for successful transmissions and for collisions (excluding the backoff time) plus black burst detection time are $S_d \cdot (T_d^s + \tau)$ and $C_d \cdot (T_d^c + \tau)$, respectively. Thus the average backoff time in an event of data nodes is given by

$$t_d^{BO} = \frac{1 - S_d \cdot (T_d^s + \tau) - C_d \cdot (T_d^c + \tau)}{S_d + C_d}. \quad (23)$$

The values of $\overline{p_d^c}$ and t_d^{BO} are used to estimate average residence time at state $n_v^c = 0$ in Section IV-A.

TABLE II

SIMULATION PARAMETERS.

Parameter	Value
Highest channel rate	11 Mbps
Basic rate	2 Mbps
Slot time τ	20 μs
SIFS	10 μs
AIFS[AC_voice]	40 μs
AIFS[AC_data]	60 μs
PHY preamble	192 μs
MAC header	34 bytes
Voice header	40 bytes
Voice packet payload size	33 bytes
Data packet payload size	1000 bytes
RTS	20 bytes
CTS	14 bytes
ACK	14 bytes
$1/\alpha$	352 ms
$1/\beta$	650 ms
t_a	20 ms
ϕ	20%
ϵ	1%

V. PERFORMANCE EVALUATION

To evaluate the priority and fairness performance of our proposed scheme supporting integrated voice/data traffic, we compare with IEEE 802.11e EDCA and GDCF introduced in [15] by simulations. Different from DCF where CW is reset to the initial value upon a successful transmission, GDCF halves the CW when the target (source) node has c consecutive successful transmissions. To provide priority, GDCF suggests that higher priority nodes choose a smaller c. Thus higher priority nodes can get access to the channel at a faster manner. In our scheme, both voice and data nodes choose the same ($W_{\min}$: $W_{\max}$) setting, and $W_{\max} \leq 63$. For EDCA, $W_{\min}$[AC_voice]=15, $W_{\max}$[AC_voice]=127, $W_{\min}$[AC_data]=31, $W_{\max}$[AC_data]=1023. For GDCF, as suggested in [15], $W_{\min} = 31$, $W_{\max} = 1023$, and the c value for voice and data is 1 and 4, respectively.

The simulation for each run takes 150-second channel time, and statistics are collected in the last 140 seconds. Simulation parameter values are listed in Table II, where the highest channel rate is to transmit DATA and ACK frames, while the basic rate is to transmit RTS and CTS frames.

A. Voice Performance

We first validate the accuracy of our analysis model for voice. Fig. 3 shows the long-term-averaged percentage of time used by voice nodes when $N_d = 10$, 50 or 100 where “sim” means “simulation”. To ensure that the service time share of voice nodes during an n_v residence time is no more than $1 - \phi = 80\%$, the maximum number of admitted voice nodes ($N_v^{\max}$) is calculated to be 36. Within the maximum number, we can see that the simulation results are close to

$$p_d^s(n_{d1}, n_{d2}) = \sum_{0 \le k_{d1} \le n_{d1},\ 0 \le k_{d2} \le n_{d2},\ 0 \le k_{d3} \le N_d - n_{d1} - n_{d2},\ k_{d1}+k_{d2}+k_{d3}=1} p_d^t(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3}) \tag{18}$$

$$p_d^c(n_{d1}, n_{d2}) = \sum_{0 \le k_{d1} \le n_{d1},\ 0 \le k_{d2} \le n_{d2},\ 0 \le k_{d3} \le N_d - n_{d1} - n_{d2},\ k_{d1}+k_{d2}+k_{d3}>1} p_d^t(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3}) \tag{19}$$

$$\begin{aligned} S_d &= \frac{E[\text{successfully transmitted packet number in an event}]}{E[\text{Average time of an event}]} \\ &= \frac{\sum_{(n_{d1},n_{d2})} p_d(n_{d1}, n_{d2}) p_d^s(n_{d1}, n_{d2})}{\sum_{(n_{d1},n_{d2})} p_d(n_{d1}, n_{d2}) \sum_{(k_{d1},k_{d2},k_{d3})} p_d^t(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3}) \cdot t_d(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3})} \end{aligned} \tag{20}$$

$$\begin{aligned} C_d &= \frac{E[\text{Collision number in an event}]}{E[\text{Average time of an event}]} \\ &= \frac{\sum_{(n_{d1},n_{d2})} p_d(n_{d1}, n_{d2}) p_d^c(n_{d1}, n_{d2})}{\sum_{(n_{d1},n_{d2})} p_d(n_{d1}, n_{d2}) \sum_{(k_{d1},k_{d2},k_{d3})} p_d^t(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3}) \cdot t_d(n_{d1}, n_{d2};\ k_{d1}, k_{d2}, k_{d3})}. \end{aligned} \tag{21}$$

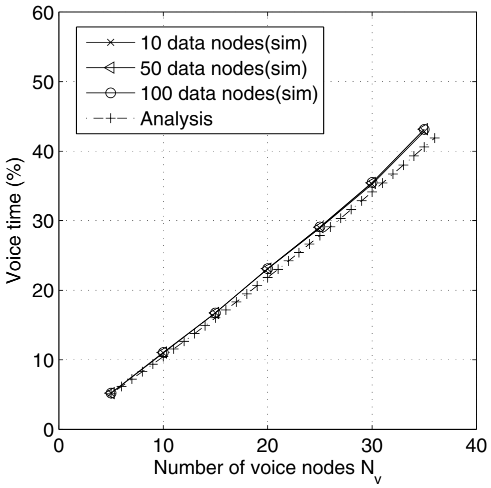

Fig. 3. The long-term-averaged percentage of time used by voice nodes in our scheme.

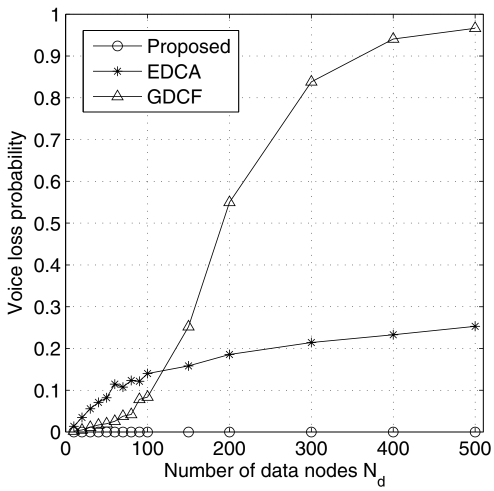

Fig. 4. The voice packet loss probability versus number of data nodes in the simulations when $N_v = 20$.

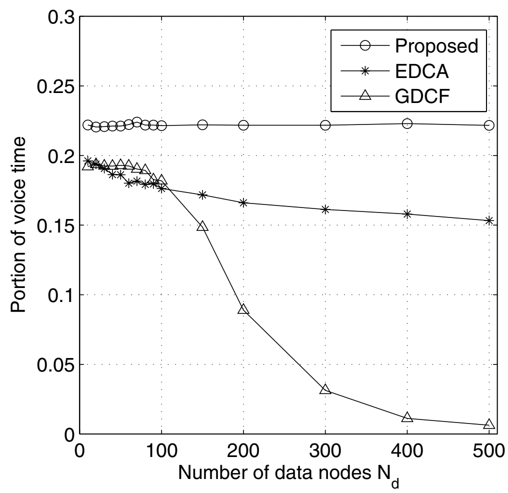

Fig. 5. The portion of voice time versus number of data nodes in the simulations when $N_v = 20$.

the analytical results. When the maximum number of voice nodes are admitted, we can see that the long-term-averaged percentage of time used by voice traffic is in the neighborhood of 40%, less that $1 - \phi$. This is because of the stringent requirement that service time share of voice nodes should be no more than $1 - \phi$ during each n_v residence time. From the simulations, we also observe that the percentage of time that a voice node has two or more backlogged packets is less than 0.05% for all the three scenarios of N_d values. This validates the assumption in Section IV-A that the probability of a voice node having two or more backlogged voice packets is negligible.

As voice traffic is delay-sensitive, packets with a large delay are considered useless and discarded. In the simulation, we set the voice packet delay bound as 40 ms. If a voice packet cannot be delivered successfully within the delay bound after its generation, it will be dropped by the voice sender. From the simulations, the voice packet loss probability due to the dropping is shown in Fig. 4 when $N_v = 20$, and N_d changes

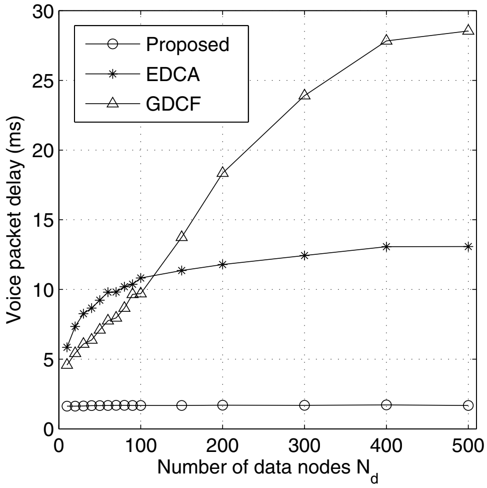

Fig. 6. The voice packet delay versus number of data nodes in the simulations when $N_v = 20$.

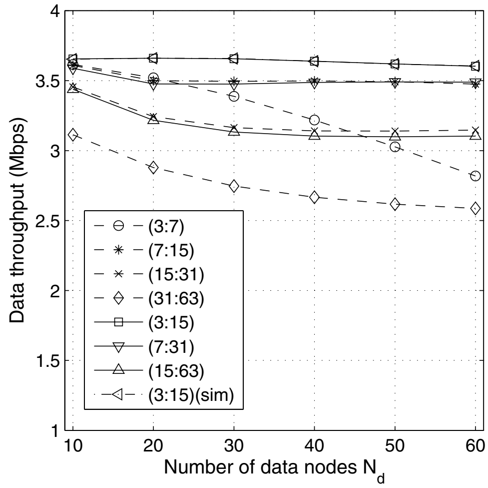

Fig. 8. Data throughput for the data-only scenario with different ($W_{\min}$: $W_{\max}$) settings in our scheme.

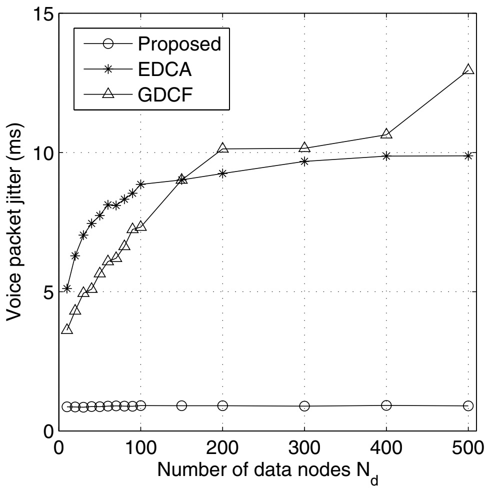

Fig. 7. The voice packet jitter versus number of data nodes in the simulations when $N_v = 20$.

from 10 to 500 [5]. It can be seen that there is no packet dropping for all the N_d cases in our proposed scheme. For EDCA, the voice packet loss probability is greater than 10% when $N_d > 70$. For GDCF, the packet loss probability is greater than 25% when $N_d > 150$, and almost all voice packets are dropped when N_d approaches 500. Accordingly, the time used by voice nodes decreases with N_d in EDCA and GDCF, but almost remains the same in our scheme, as shown in Fig. 5. When N_d increases, the delay and jitter of voice packets almost remain unchanged in our scheme, but increase in EDCA and GDCF, as shown in Figs. 6 and 7. The reason for all the above observations is that our scheme provides guaranteed priority to voice nodes, thus the voice performance is not affected by the traffic load of data nodes. EDCA and GDCF provide only statistical priority to voice nodes, thus the voice performance greatly depends on the traffic load of data nodes.

[5]It is an extreme case to have 500 nodes. We use this example here to demonstrate the performance of the schemes under a very large number of nodes.

B. Data Performance

For our scheme, as the data throughput in the integrated voice/data scenario is analyzed based on the throughput in the data-only scenario with $N_v = 0$, we plot the analyzed throughput (in the unit of Mbps) and collision number per second in Fig. 8 and Fig. 9, respectively, for the data-only scenario in different two-window (i.e., $W_{\max} = 2 \cdot (W_{\min} + 1) - 1$) and three-window (i.e., $W_{\max} = 4 \cdot (W_{\min} + 1) - 1$) cases. It can be seen that, the larger the ($W_{\min}$: $W_{\max}$) setting, the smaller the collision number per second. However, a larger ($W_{\min}$: $W_{\max}$) also leads to longer time in black burst transmission. There exists a tradeoff between the two conflicting effects, and the best (in terms of throughput) ($W_{\min}$: $W_{\max}$) setting is found to be (3:15) for all the two-window and three-window cases as shown in Fig. 8. Actually, for the best setting, when we increase the $W_{\max}$ from 15 (3-window case) to 31 (4-window case), and to 63 (5-window case), the throughput performance is almost the same. We choose ($W_{\min}$: $W_{\max}$) = (3:15) for simplicity. Simulations for different ($W_{\min}$: $W_{\max}$) settings are also carried out, and only the results for (3:15) are shown in Fig. 8 and Fig. 9. It can be seen that the simulations match well with the analysis. The throughput and collision number per second are quite stable with the different values of N_d when ($W_{\min}$: $W_{\max}$) = (3:15).

Next consider an integrated voice/data scenario with $N_v = 20$. From Fig. 3, we can see that the percentage of time used by 20 one-way `on/off` voice transmissions in our scheme is 22%. The unused 78% time is shared by all the data nodes. Fig. 10 shows the analyzed throughput of data nodes in 78% time for our scheme, and the simulated data throughput of our scheme, EDCA, and GDCF. It can be seen that, the simulation and analysis match well for our scheme. The data throughput in GDCF increases with N_d when $N_d < 300$. This gain is due to the sacrifice of the high priority voice traffic, i.e., the packet loss probability of voice packets increases with N_d in GDCF (as shown in Fig. 4). When N_d further increases beyond the value 300, the data throughput in GDCF decreases due to the increased collision probability. For EDCA, with the increase

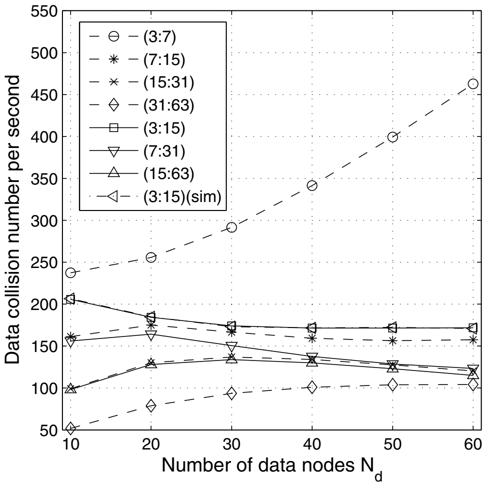

Fig. 9. Data collision number per second for the data-only scenario with different ($W_{\min} : W_{\max}$) settings in our scheme.

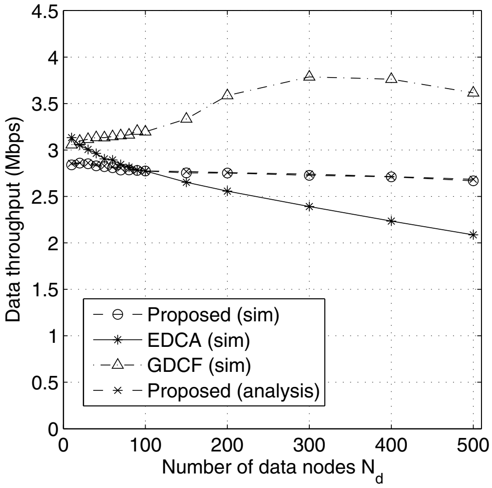

Fig. 10. Data throughput with $N_v = 20$.

of N_d, the time portion shared by the data nodes increases as implied by Fig. 5. However, the data throughput still decreases due to the increased collision probability when N_d increases.

For data traffic, the fairness is measured by the Fairness Index (FI) [16] defined as

$$\mathrm{FI} = \frac{\left(\sum_{i=1}^{N_d} S_i\right)^2}{N_d \cdot \sum_{i=1}^{N_d} S_i^2}. \tag{24}$$

where S_i is the throughput of the ith data node during an examined period. The higher the FI value, the better the fairness performance. The upper bound of the FI is 1, which is achieved when S_i is independent of i. To evaluate the short-term fairness, for each N_d value in the simulations, we calculate the FI for every $6N_d$ transmitted data packets (i.e., on average each data node transmits 6 packets), and demonstrate the average FI value in Fig. 11. It can be seen that, our scheme shows good fairness performance, with the FI being kept around 0.9. As expected, EDCA is not fair, with FI approximately in the range [0.5, 0.7]. For GDCF, when

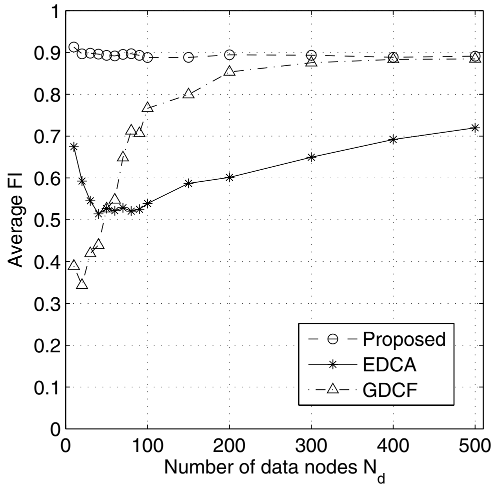

Fig. 11. Data short-term fairness performance with $N_v = 20$.

N_d is large (> 200), its FI approaches 0.9; but its fairness performance is quite poor when N_d is small (< 50).

Note that our scheme is intended to support integrated voice/data traffic, and to provide guaranteed priority for voice and achieve fairness improvement for data. If only one traffic type is considered (e.g., data only) and fairness is not the main concern, EDCA and GDCF can achieve a slightly more throughput. The network designer should choose a proper scheme based on the traffic types and QoS requirements.

VI. Conclusion

Although IEEE 802.11e EDCA has some features for QoS support, it is not effective in providing priority to real-time traffic such as delay-sensitive voice. The short-term fairness performance of EDCA is not good either due to its channel access policy. By sending a black burst instead of waiting for the expiration of the backoff timer in EDCA, our proposed distributed channel access scheme can provide guaranteed priority to delay-sensitive voice traffic, and also enhance the short-term fairness performance of data traffic. Our proposed analytical models are validated by extensive simulations. As only minor modifications are needed, our scheme can be easily incorporated in IEEE 802.11e implementation.

Appendix
The Derivation of $t_v^c(i)$ $(i > 0)$

In order to get $t_v^c(i)$ $(i > 0)$, we first calculate the time needed to deliver all the n_v^c backlogged voice packets from the n_v^c backlogged on voice nodes. When a voice node contends successfully (i.e., it is the only one with the largest backoff timer), it will leave the contention as its backlogged packet has been delivered. Consider the case where no new voice nodes join the contention. For simplicity of presentation, the CW of each voice node takes values from the set $\{W_{v1}, W_{v2}\}$ where $W_{v2} = 2 \cdot (W_{v1} + 1) - 1$. Our analysis can be easily extended to a case with 3 or more choices for the CW size. Define state (n_{v1}^c, n_{v2}^c), where n_{v1}^c and n_{v2}^c are the numbers of voice nodes with contention windows of W_{v1} and W_{v2},

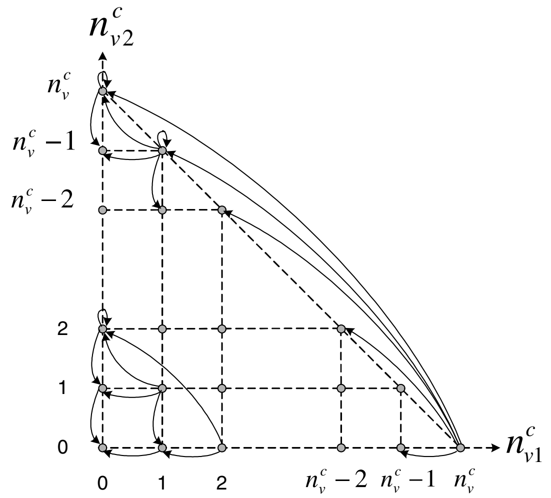

Fig. 12. The transition diagram for state (n^c_{v1}, n^c_{v2}) of voice nodes.

respectively[6]. After each event (i.e., a successful transmission or a collision), the state will evolve, remaining in the current state or moving to another. The state transition is shown in Fig. 12, where the state $(0, 0)$ is the absorbing state when all the voice nodes are served. There are totally $\binom{n^c_v+2}{2}$ states. To understand the state transition diagram, we use state $(1, n^c_v-1)$ as an example. Its next state is $(0, n^c_v - 1)$ if the voice node with W_{v1} transmits successfully, $(1, n^c_v - 2)$ if one voice node with W_{v2} transmits successfully, $(0, n^c_v)$ if the node with W_{v1} collides with one or more other nodes, or remains in $(1, n^c_v-1)$ if no node with W_{v1} and at least two nodes with W_{v2} collide.

Let $T_v(n^c_{v1}, n^c_{v2})$ denote the average time needed for transitions from state (n^c_{v1}, n^c_{v2}) to the absorbing state (0, 0). Obviously, we have $T_v(0, 0) = 0$, and $T_v(n^c_v, 0)$ is approximately the average time to serve all the n^c_v backlogged on voice nodes.

For a state (n^c_{v1}, n^c_{v2}), an event (i.e., one or more transmissions from the $n^c_{v1} + n^c_{v2}$ nodes) will lead to the next state (which may be same as (n^c_{v1}, n^c_{v2})). In an event, denote the number of transmissions from voice nodes with W_{v1} and W_{v2} as $k_{v1}(\leq n^c_{v1})$ and $k_{v2}(\leq n^c_{v2})$, respectively, where $k_{v1} + k_{v2} > 0$. It is clear that the event is a successful transmission if $k_{v1}+k_{v2} = 1$, or a collision otherwise. Denote the next state after the event as $s_v(n^c_{v1}, n^c_{v2};\ k_{v1}, k_{v2})$. Then we have

$$s_v(n^c_{v1}, n^c_{v2};\ k_{v1}, k_{v2}) = \begin{cases} (n^c_{v1} - k_{v1}, n^c_{v2} - k_{v2}), & \text{if } k_{v1} + k_{v2} = 1 \\ (n^c_{v1} - k_{v1}, n^c_{v2} + k_{v1}), & \text{if } k_{v1} + k_{v2} > 1. \end{cases} \quad (25)$$

Denote the probability of the above transition as $p^t_v(n^c_{v1}, n^c_{v2};\ k_{v1}, k_{v2})$, and the average time of the transition as $t_v(n^c_{v1}, n^c_{v2};\ k_{v1}, k_{v2})$. If $k_{v1} \neq 0$, i.e., the successful transmission or collision happens when the largest backoff timer among all the $n^c_{v1} + n^c_{v2}$ voice nodes takes values from [0, W_{v1}], we have

$$p^t_v(n^c_{v1}, n^c_{v2};\ k_{v1}, k_{v2}) = \sum_{i=0}^{W_{v1}} f(n^c_{v1}, k_{v1}, \frac{1}{W_{v1}+1}, \frac{i}{W_{v1}+1}) \cdot f(n^c_{v2}, k_{v2}, \frac{1}{W_{v2}+1}, \frac{i}{W_{v2}+1}) \quad (26)$$

where $f(\cdot)$ is given in (10), and

$$t_v(n^c_{v1}, n^c_{v2};\ k_{v1}, k_{v2}) = \frac{1}{p^t_v(n^c_{v1}, n^c_{v2};\ k_{v1}, k_{v2})} \cdot \left(\sum_{i=0}^{W_{v1}} f(n^c_{v1}, k_{v1}, \frac{1}{W_{v1}+1}, \frac{i}{W_{v1}+1}) \cdot f(n^c_{v2}, k_{v2}, \frac{1}{W_{v2}+1}, \frac{i}{W_{v2}+1}) \cdot i \cdot \tau \right) + \tau + T_{xv} \quad (27)$$

where T_{xv} is the successful transmission or collision time (excluding the backoff time) given by

$$T_{xv} = \begin{cases} \text{AIFS[AC\_voice]} + t_{v\text{\_DATA}} + \text{SIFS} + t_{\text{ACK}}, & \text{if } k_{v1} + k_{v2} = 1 \\ \text{AIFS[AC\_voice]} + t_{v\text{\_DATA}} + t_{\text{ACK\_TIMEOUT}}, & \text{if } k_{v1} + k_{v2} > 1 \end{cases} \quad (28)$$

$t_{v\text{\_DATA}}$ is the time to transmit a DATA frame from a voice node, and $t_{\text{ACK\_TIMEOUT}}$ is the ACK timeout value.

If $k_{v1} = 0$, the successful transmission or collision can happen when the largest backoff timer among all the $n^c_{v1}+n^c_{v2}$ voice nodes takes values from [0, W_{v2}]. We have

$$p^t_v(n^c_{v1}, n^c_{v2};\ 0, k_{v2}) = \sum_{i=0}^{W_{v1}} f(n^c_{v1}, 0, \frac{1}{W_{v1}+1}, \frac{i}{W_{v1}+1}) \cdot f(n^c_{v2}, k_{v2}, \frac{1}{W_{v2}+1}, \frac{i}{W_{v2}+1}) + \sum_{i=W_{v1}+1}^{W_{v2}} f(n^c_{v2}, k_{v2}, \frac{1}{W_{v2}+1}, \frac{i}{W_{v2}+1}) \quad (29)$$

and

$$t_v(n^c_{v1}, n^c_{v2};\ 0, k_{v2}) = \frac{1}{p^t_v(n^c_{v1}, n^c_{v2};\ 0, k_{v2})} \left(\sum_{i=0}^{W_{v1}} f(n^c_{v1}, 0, \frac{1}{W_{v1}+1}, \frac{i}{W_{v1}+1}) \cdot f(n^c_{v2}, k_{v2}, \frac{1}{W_{v2}+1}, \frac{i}{W_{v2}+1}) \cdot i \cdot \tau + \sum_{i=W_{v1}+1}^{W_{v2}} f(n^c_{v2}, k_{v2}, \frac{1}{W_{v2}+1}, \frac{i}{W_{v2}+1}) \cdot i \cdot \tau \right) + \tau + T_{xv}. \quad (30)$$

[6]We omit the index t of n^c_{v1} and n^c_{v2}.

Hence, for state (n_{v1}^c, n_{v2}^c), consider all possible transitions from it, we have

$$T_v(n_{v1}^c, n_{v2}^c) = \sum_{\substack{0 \le k_{v1} \le n_{v1}^c \\ 0 \le k_{v2} \le n_{v2}^c \\ k_{v1}+k_{v2}>0}} p_v^t(n_{v1}^c, n_{v2}^c;\ k_{v1}, k_{v2}) \cdot \Big(T_v\big(s_v(n_{v1}^c, n_{v2}^c;\ k_{v1}, k_{v2})\big) + t_v\big(n_{v1}^c, n_{v2}^c;\ k_{v1}, k_{v2}\big)\Big) \tag{31}$$

From (31) and $T_v(0,0) = 0$, we can compute the values of $T_v(n_{v1}^c, n_{v2}^c)$. As $T_v(i,0)$ is approximately the average time to serve i backlogged $\mathtt{on}$ voice nodes, we have the average time to serve one (backlogged $\mathtt{on}$) voice node $t_v^c(i) \approx T_v(i,0) - T_v(i-1,0)$.

REFERENCES

[1] IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, Standard, IEEE, Aug. 1999.

[2] 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.

[3] X. Yang and N. H. Vaidya, "Priority scheduling in wireless ad hoc networks," in *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing 2002*, pp. 71–79.

[4] J. W. Robinson and T. S. Randhawa, "Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 5, pp. 917–928, June 2004.

[5] C. E. Koksal, H. Kassab, and H. Balakrishnan, "An analysis of short-term fairness in wireless media access protocols," in *Proc. ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems 2000*, pp. 118–119, June 2000.

[6] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *Proc. IEEE INFOCOM'01* pp. 209–218.

[7] 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.

[8] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1353–1368, Aug. 1999.

[9] J.-P. Sheu, C.-H. Liu, S.-L. Wu, and Y.-C. Tseng, "A priority MAC protocol to support real-time traffic in ad hoc networks," *Wireless Networks,* vol. 10, no. 1, pp. 61–69, Jan. 2004.

[10] T. You, C.-H. Yeh, and H. Hassanein, "CSMA/IC: a new class of collision-free MAC protocols for ad hoc wireless networks," in *Proc. IEEE ISCC'03*, pp. 843–848.

[11] A. Banchs and X. Perez, "Distributed weighted fair queuing in 802.11 wireless LAN," in *Proc. IEEE ICC'02*, pp. 3121–3127.

[12] W. Pattara-Atikom, S. Banerjee, and P. Krishnamurthy, "Starvation prevention and quality of service in wireless LANs," in *Proc. 5th International Symposium on Wireless Personal Multimedia Communications 2002*, pp. 1078–1082.

[13] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Proc. ACM MOBICOM'00*, pp. 167–178.

[14] Y. Kwon, Y. Fang and H. Latchman, "Design of MAC protocols with fast collision resolution for wireless local area networks," *IEEE Trans. Wireless Commun.*, vol. 3, no. 3, pp. 793–807, May 2004.

[15] C. Wang, B. Li, and L. Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF," *IEEE Trans. Veh. Technol.*, vol. 53, no. 4, pp. 1235–1246, July 2004.

[16] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: an explanation," ATM Forum Document Number: ATM_Forum/99-0045, Feb. 1999.

Hai Jiang (M'07) received the B.S. degree in 1995 and the M.S. degree in 1998, both in electronics engineering, from Peking University, Beijing, China, and the Ph.D. degree in 2006 in electrical engineering from the University of Waterloo, Canada. He is currently a Postdoctoral Fellow at Electrical Engineering Department, Princeton University, USA. His research interests include radio resource management, cellular/WLAN interworking, and cross-layer design for wireless multimedia communications.

Ping Wang 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.

Weihua Zhuang (M'93-SM'01) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, Liaoning, China, and the Ph.D. degree from the University of New Brunswick, Fredericton, NB, Canada, all in electrical engineering. Since October 1993, she has been with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada, where she is a full professor. She is a co-author of the textbook *Wireless Communications and Networking* (Prentice Hall, 2003). Her current research interests include multimedia wireless communications, wireless networks, and radio positioning. Dr. Zhuang is a licensed Professional Engineer in the Province of Ontario, Canada. She received 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*.