A New Collision Resolution Mechanism to Enhance the Performance of IEEE 802.11 DCF

Chonggang Wang, Bo Li, *Senior Member, IEEE*, and Lemin Li

***Abstract*—The medium-access control (MAC) protocol is one of the key components in wireless local area networks (WLANs). The main features of a MAC protocol are high throughput, good fairness, energy efficiency, and support priority guarantees, especially under distributed contention-based environment. Based on the current standardized IEEE 802.11 distributed coordination function (DCF) protocol, this paper proposes a new efficient collision resolution mechanism, called GDCF. Our main motivation is based on the observation that 802.11 DCF decreases the contention window to the initial value after each success transmission, which essentially assumes that each successful transmission is an indication that the system is under low traffic loading. GDCF takes a more conservative measure by halving the contention window size after c consecutive successful transmissions. This "gentle" decrease can reduce the collision probability, especially when the number of competing nodes is large. We compute the optimal value for c and the numerical results from both analysis and simulation demonstrate that GDCF significantly improve the performance of 802.11 DCF, including throughput, fairness, and energy efficiency. In addition, GDCF is flexible for supporting priority access by selecting different values of c for different traffic types and is very easy to implement it, as it does not requires any changes in control message structure and access procedures in DCF.**

***Index Terms*—IEEE 802.11 DCF, wireless local area network (WLAN).**

I. INTRODUCTION

RECENTLY, we have witnessed a rapid development and deployment of wireless local area networks (WLANs), which in return has fueled the development in the standardization organization, such as the IEEE 802.11 working group, to improve its performance. One of the key components in WLAN is a medium-access control (MAC) protocol that primarily determines its performance. MAC protocols are commonly used in multiple-access environments, where multiple nodes compete for certain shared resources. The main functionality of MAC protocols is to arbitrate access for the shared transmission medium [1]. The performance metrics of interest include throughput, fairness, packet transmission delay, stability, and also priority in an environment supporting multiservices. In addition, in a WLAN, the energy efficiency is also a major performance index of interest.

In IEEE 802.11 standard [2], channel access is controlled by the use of interframe space (IFS) time between the frame transmissions. Three IFS intervals that have been specified by 802.11 standards include short IFS (SIFS), point coordination function IFS (PIFS), and distributed coordination function (DCF)-IFS (DIFS). The SIFS is the smallest and the DIFS is the largest.

There are two access mechanisms including point coordination function (PCF) and DCF. PCF is a centralized MAC algorithm used to provide contention-free service, while DCF uses a contention-based algorithm to provide access to all traffic. PCF is built on top of DCF and regulates transmission through a centralized decision maker or point coordinator, which makes use of PIFS when issuing polls. Because PIFS is smaller than DIFS, the point coordinator can seize the medium and lock out all asynchronous traffic (which uses DIFS to access channel) while it issues polls and receives responses. This paper focuses on DCF and we will give a brief introduction later.

In 802.11 DCF, a node starts its transmission if the medium is sensed to be idle for an interval larger than the distributed interframe space (DIFS). If the medium is busy, the node will defer its transmission until a DIFS is detected and then generate a random backoff period (backoff timer) before retransmission. The backoff timer will be decreased as long as the channel is sensed idle, frozen when the channel is sensed busy, and resumed when the channel is sensed idle again for more than a DIFS. A node can initiate a transmission when the backoff timer reaches zero. The backoff timer is uniformly chosen in the range $[0,\mathrm{CW})$. CW is known as *contention window*, which is an integer with the range determined by the PHY characteristics $\mathrm{CW}_{\min}$ and $\mathrm{CW}_{\max}$. After each unsuccessful transmission, CW will be doubled until reaching the maximum value $\mathrm{CW}_{\max} = 2^m W - 1$, where W equals to $(\mathrm{CW}_{\min} + 1)$. After each successful transmission, CW will reset to the minimum value $\mathrm{CW}_{\min}$. In 802.11 DCF for the DSSS physical channel, $\mathrm{CW}_{\min} = 31$, $\mathrm{CW}_{\max} = 1023$, and $m' = 5$.

802.11 DCF defines two channel-access modes: basic and request to send/clear to send (RTS/CTS) base access. In basic access mode [Fig. 1(a)], the destination node will wait for a SIFS interval immediately following the successful reception of the data frame and transmit a positive ACK back to the source node to indicate that the data packet has been received correctly. If the source node does not receive an ACK, the data frame is assumed to be lost and the source node will schedule the retransmission

Manuscript received November 9, 2003; revised January 9, 2004. B. Li's research was support in part by grants from the Research Grant Council under contracts HKUST6196/02E and HKUST6402/03E, an National Science Funds Council/RGC joint grant under contract N_HKUST605/02, and a grant from Microsoft Research under contract MCCL02/03.EG01.

C. Wang is with the the Special Research Centre for Optical Internet & Wireless Information Networks (ICOIWIN), ChongQing University of Posts and Telecommunications (CQUPT), Chongqing 400065, P. R. China, on leave from the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701 USA (e-mail: cgwang@cs.ust.hk).

B. Li is with the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, P.R. China (e-mail: bli@cs.ust.hk).

L. Li is with the University of Electronic Science and Technology of China, Chengdu 400065, P. R. China (e-mail: lml@uestc.edu.cn).

Digital Object Identifier 10.1109/TVT.2004.830951

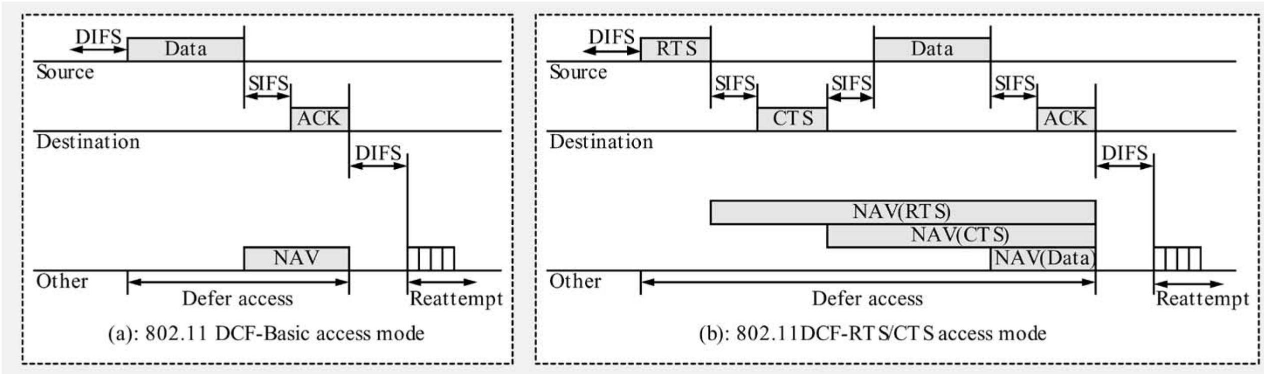


Fig. 1. IEEE 802.11 MAC mechanism.

with the doubled CW for backoff timer. When the data frame is being transmitted, all the other nodes hearing the data frame adjust their network-allocation vector (NAV), which is used for virtual carrier sense at the MAC layer, correctly based on the duration field value in the data frame received. This includes the SIFS and ACK frame transmission time following the data frame.

In RTS/CTS-based access mode, nodes transmit data utilizing special short RTS and CTS frames prior to the transmission of an actual data frame in order to shorten the collided time interval. As shown in Fig. 1(b), the node that needs to transmit a packet issues a RTS frame. When the destination receives the RTS frame, it will transmit a CTS frame after the SIFS interval, immediately following the reception of the RTS frame. The source node is allowed to transmit its packet if and only if it receives the CTS correctly. At the same time, all the other nodes will update the NAVs based on the RTS from the source node and the CTS from the destination node, which helps to overcome the hidden terminal problem. In fact, the node that is able to receive the CTS frames correctly can avoid collisions even when it is unable to sense the data transmissions from the source node. If a collision occurs with two or more RTS frames, less bandwidth is wasted as compared to the situation where larger data frames can collide in the basic access mode.

The remainder of this paper is organized as follows. Section II reviews the existing work and discusses the main features in our proposed GDCF. Section III introduces the new collision-resolution mechanism called GDCF. Theoretical analysis of GDCF, including normalized throughput and some other metrics, will be given in Section IV. In Section V, we present numerical results of GDCF and compare with that those of the IEEE 802.11 DCF protocol. Section VI concludes this paper.

II. Related Work

This paper focuses on the contention-based MAC protocols used in WLAN, specifically IEEE 802.11 DCF [2]. The analysis in [3] demonstrated that the throughput and fairness of 802.11 DCF could significantly deteriorate when the number of nodes increases. Several recent proposals have addressed this issue [4]–[7]. Furthermore, given the need to support multimedia applications and to consider the energy efficiency in mobile devices, there also are protocols to address a priority scheme in [8] and [9] and the energy efficiency issue in [10]. Cali *et al.* [4] proposed a dynamic and distributed algorithm, IEEE 802.11^{+}, which allows each node to estimate the number of competing nodes and to tune its contention window to the optimal value at run time. Results from simulations showed that the throughput of IEEE 802.11^{+} is very close to the theoretical upper bound. DCF+, proposed in [5], is a new ACK-integrated mechanism that combines the TCP ACK with MAC level ACK and obtains the improved throughput. One of the limitations is its ineffectiveness for other flows, such as UDP. It also violates the layering principle that leads to the complication in MAC ACK message structure. Peng [6] proposed a new measurement-based algorithm to adaptively configure the optimal value of the initial CW value to improve the throughput and fairness. However, it also needs to compute current channel status at run time and adjusts the RTS/CTS message structure. The fast collision resolution (FCR) is another MAC protocol proposed in [7], which actively redistributes the backoff timer for all competing nodes, thus allowing the more recent successful nodes to use smaller contention window and allowing other nodes to reduce backoff timer exponentially when they continuously meets some idle time slots, instead of reducing backoff timer by 1 after each idle time slots, as in the original IEEE 802.11 DCF. FCR can resolve collisions more quickly than 802.11 DCF and obtains higher throughput, but FCR itself can inversely affect the fairness unless it is combined with additional fair scheduling mechanism, as shown in [7]. Residual-energy-based tree splitting (REBS) [10] is an energy-efficient collision-resolution algorithm that can be used in the wireless *ad hoc* networks. REBS differentiates and splits all the competing nodes according to their residual energy and assigns the node with the least residual energy to seize the channel with the highest priority.

We propose a new collision-resolution mechanism called GDCF, which is a simple variation of 802.11 DCF, yet can significantly improve throughput and fairness. GDCF enables the priority support for multimedia application and obtains better energy efficiency than DCF itself. There are several unique advantages in the proposed GDCF. Comparingit to IEEE 802.11^{+} in [4] and self-adapt DCF in [6], GDCF is simpler in that it does not need to estimate network parameters such as competing node number in [4] and channel status in [6], although there is a Kalman filter-based algorithm to measure the number of competing nodes in [11]. Comparing it to DCF+ in [5], GDCF can support any upper protocols (TCP

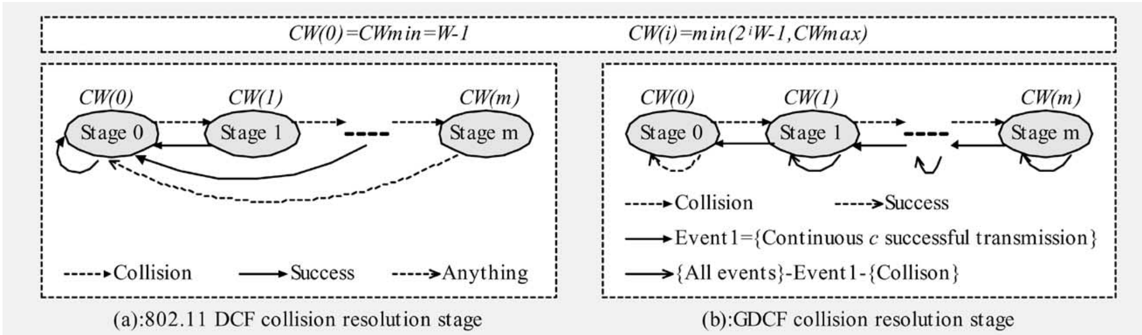


Fig. 2. Collision-resolution stage evolution in GDCF and 802.11 DCF.

or UDP) and does not need to change the RTS/CTS message structure. Comparing it to the FCR algorithm in [7], GDCF achieves better fairness and simplicity and can easily support priority or quality-of-service (QoS) differentiation effectively. GDCF maintains excellent compatibility with the original IEEE 802.11 DCF. In summary, the proposed GDCF achieves better throughput, fairness, and energy efficiency and enables priority support. In addition, it does not need to estimate the competing node and channel status; thus, it is simple for implementation.

III. Proposed GDCF Algorithm

From the discussions in the Section II, we can see that 802.11 DCF resolves collision through CW and backoff stage [Fig. 2(a)]. In the initial backoff stage (stage 0), the value of CW has the minimal value $\mathrm{CW}_{\min}$. After each transmission collision, the backoff stage will be increased by 1 and the CW will be doubled until it reaches the maximum, $\mathrm{CW}_{\max}$. After each successful transmission, the backoff stage will resume to initial stage 0 and the CW will be reset to $\mathrm{CW}_{\min}$, regardless of network conditions such as the number of competing nodes. In this method, we refer to heavy decrease, which tends to work well when there are only a few competing nodes. When the number of competing nodes increases, it will be shown to be ineffective, since new collisions can potentially occur and cause significant performance degradation.

For example, assuming that the current backoff stage is i with contention window $\mathrm{CW}(i) = 2^i W - 1$ and that there is a successful transmission, the next backoff stage will be stage 0 with $\mathrm{CW}(0) = 31$, according to 802.11 DCF specifications for DSSS PHY [1]. But if the number of current competing nodes is large enough, $(\gg 32)$, the new collision will likely occur at the backoff stage 0. The main argument is that since the current backoff stage is i, some collisions must have occurred recently. Now if the number of current competing nodes is larger than or close to $\mathrm{CW}(i)$ and if the backoff stage is reset to 0 after a successful transmission, there is a high probability that some new collision(s) will happen. Certainly, the number of current competing nodes may be smaller than $\mathrm{CW}(i)$ if there are several consecutive successful transmissions at the backoff stage i. Under this case, we can effectively begin to decrease the CW. This is the primary principle used in GDCF.

GDCF attempts to avoid useless collisions through the "gentle" decrease of contention window, referred as gentle DCF or GDCF. The collision-resolution stage evolution in GDCF is presented in Fig. 2(b). The difference between GDCF and DCF is that GDCF will halve CW value if there are c consecutive successful transmissions. On the contrary, DCF will reset CW once there is a successful transmission or the retry count overruns the threshold (m), so GDCF needs to maintain a counter for recording the number of continuous successful transmissions up to now. This counter will reset to zero after each collision, because what it records is the number of continuous successful transmissions, not the number of total successful transmissions. According to the channel status, the detailed collision resolution process in GDCF is as follows.

- **Collision**: Similar to the operations in DCF, GDCF will double the contention window and select a backoff timer value uniformly from [0,CW]. But GDCF also needs to reset the counter for recording the number of consecutive successful transmissions.
- **Successful transmission**: If there are c consecutive successful transmissions, GDCF will halve the CW and select a backoff timer value uniformly from [0,CW]. Then, the counter for recording the number of continuous successful transmissions is reset to zero. Otherwise, GDCF increases counter for the number of consecutive successful transmission and keeps the contention window unchanged.
- **Idle**: If the channel is idle, GDCF also reduces the backoff timer by 1, the same as in DCF.

In another word, CW in GDCF is gently and gradually decreased after consecutive successful transmissions. If there are a few competing nodes, many consecutive successful transmissions can appear and the backoff stage gradually goes down to the initial stage 0. If there are many competing nodes, the probability that the backoff stage will be down to initial stage 0 are very small and the backoff stage will oscillate between two large stages i and $i + 1$ with a high probability. This behavior brings two advantages. First, it will decrease the collision probability and improve the system throughput. Second, it will obtain better fairness because GDCF maintains all the nodes in the same stage (with the same CW) even if after several consecutive successful transmissions $(< c)$, especially under large node number. However, nodes in DCF will stay in a different stage (with different CW) after successful transmission, since it is reset to initial stage 0 after each successful transmission. GDCF can easily be extended to support priority applications or QoS differentiation through configuring te c value for different type of applications. A simple method is to let high-priority applications choose smaller c, while low-priority applications with larger c. Evidently, the nodes with smaller c can seize the channel more quickly and result in lower access delay. This is particularly important for some real-time multimedia applications. We will exploit and evaluate this capability of GDCF in Sections IV and V through simulations.

One issue in GDCF is how to set the parameter c. The intuition is that in the environment with many (or a few) competing nodes, it requires large (or small) value of c. If the number of current competing node number can be obtained, we can intelligently adjust the parameter c, but it usually is expensive to precisely obtain the number of competing nodes in a distributed dynamic environment, where nodes consistently move or/and frequently switches on or off. However, we will show that the parameter c is not too sensitive to the number of competing nodes in Section IV. In addition, we prove that the short range 4 c 8 is the optimal value for c if the number of competing nodes is larger than 10.

IV. GDCF Performance Analysis

In this section, we will analyze the performance of GDCF, discuss how to choose parameter c, and investigate the performance of GDCF when supporting priority traffics.

A. Saturation Throughput

First, we will deduce the normalized system throughput S of GDCF, which is equal to the ratio between "average payload duration in a slot time" and "average length of slot time," using some similar procedures and symbols in [3] and [5]. Let p be the probability that a transmitted packet collides, τ be the probability that a node transmits in a randomly chosen slot time, i be the backoff stage, m be the maximal backoff stage [Fig. 2(b)], k be the backoff time slot, (i,k) be the bidimensional state of each node, $b_{i,k}$ be the stable probability of state (i,k), and $P\{i,k|i-1,k-1\}$ be the one-step transition probability from state $(i-1,k-1)$ to state (i,k). For convenience, we use CW(i) and W_i interchangeably in the following discussions.

After every transmission collision, GDCF will back off (increase the stage i) and double the contention window, so $P\{i,k|i-1,0\} = p/W_i$ ($k \in [0, W_i - 1], i \in [1,m]$). The backoff timer will decrease by 1 if the channel is sensed idle, so $P\{i,k|i,k+1\} = 1$ ($k \in [0, W_i - 2], i \in [0,m]$). If there are c continuous successful transmissions, GDCF will decrease the backoff stage i and halve the contention window; otherwise, the node will stay at the current backoff stage i and keeps the contention window unchanged. We can approximate this transition probability as follows. $P\{i-1,k|i,0\} = p'/W_{i-1}$ ($k \in [0, W_{i-1} - 1], i \in [1,m]$) and $P\{i,k|i,0\} = (1-p-p')/W_i$ ($k \in [0, W_i - 1], i \in [1, m-1]$), where $p' = (1-p)^c$ and let $\rho = p/p'$. Then, we can easily construct corresponding transition equations of GDCFs Markov model (see Fig. 4) according to its collision-resolution process in Fig. 3.

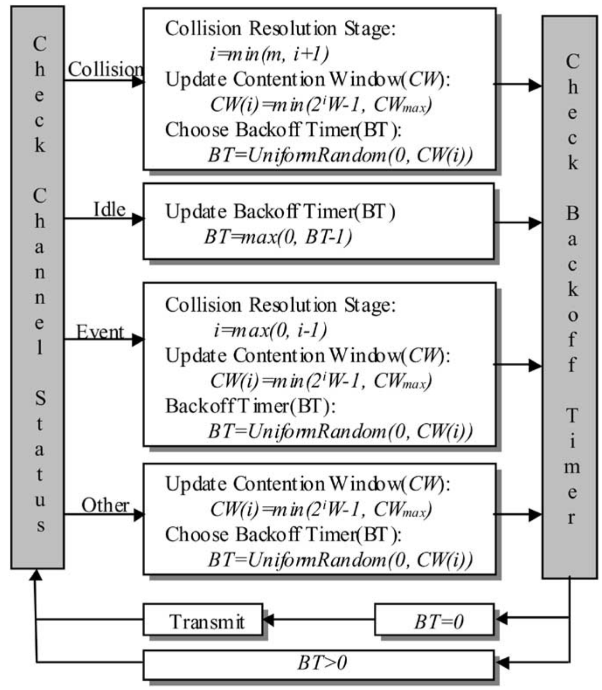


Fig. 3. Collision-resolution process in GDCF.

The nonnull one-step transition probabilities can be computed as

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

Let $\rho = (p/p')(p' = (1-p)^c)$ and we can aggregate the state (i,k) $k \in [0, W_i - 1]$ into a single state $(i,0)$, so it is easy to get that

$$b_{i,0} = \rho^i b_{0,0} \quad 0 \le i \le m. \tag{2}$$

For each $k \in (0, W_i - 1]$, $b_{i,k}$ also has the relationship shown in (3) at the bottom of the page.

With (2) and $\rho = (p/p')$, (3) can be simplified as

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0} \quad 0 \le i \le m. \tag{4}$$

Because the sum of stationary distribution for all states must be equal to 1, therefore

$$\sum_{i=0}^{m} \sum_{k=0}^{W_i - 1} b_{i,k} = 1 \overset{(4)}{\Rightarrow} \sum_{i=0}^{m} b_{i,0} \frac{W_i + 1}{2} = 1. \tag{5}$$

$$b_{i,k} = \frac{W_i - k}{W_i} \begin{cases} (1-p)b_{0,0} + p' b_{i+1,0} & i = 0 \\ p b_{i-1,0} + (1-p-p') b_{i,0} + p' b_{i+1,0} & 0 < i < m \\ p b_{i-1,0} + (1-p') b_{m,0} & i = m. \end{cases} \tag{3}$$

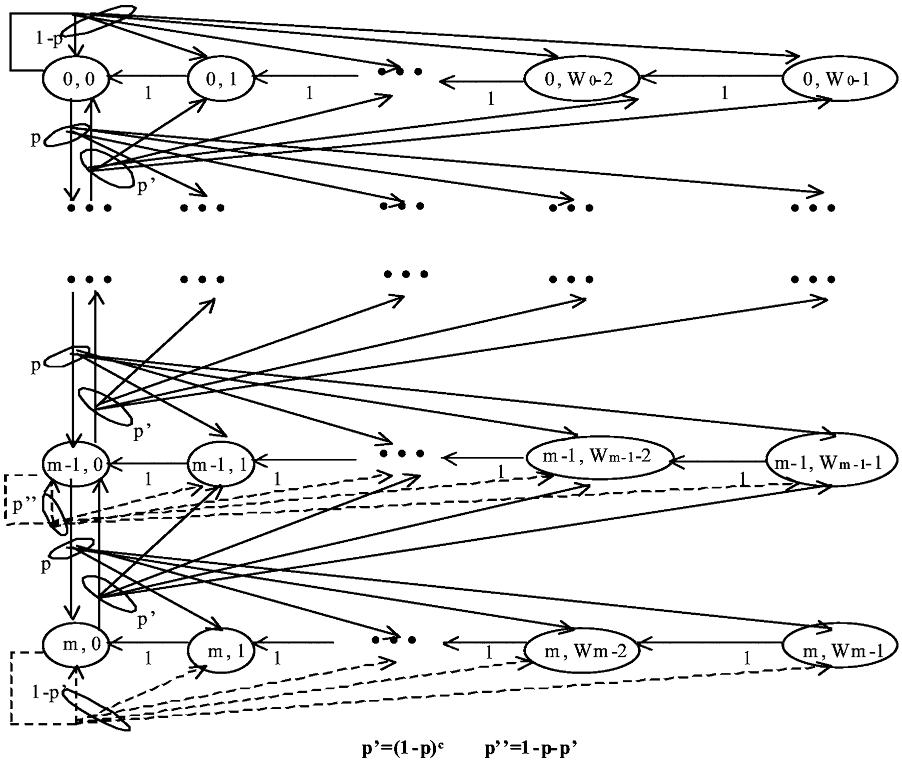


Fig. 4. Markov chain model of GDCF.

In (5), $b_{i,0}$ can be computed using (2) and W_i is standardized in 802.11b as follows (for DSSS PHY in 802.11, $m'=5$):

$$W_i = \begin{cases} 2^i W, & i \le m' \\ 2^{m'} W, & i > m'. \end{cases} \tag{6}$$

Replacing (5) with (2) and (6), we can get the value of $b_{0,0}$ in (7) as

$$\begin{aligned} \frac{1}{b_{0,0}} &= \begin{cases} T_1(\rho, m) & m \le m' \\ T_1(\rho, m') + T_2 & m > m' \end{cases} \\ T_1(\rho, m) &= \frac{1-\rho^{m+1}}{2(1-\rho)} + \frac{W(1-(2\rho)^{m+1})}{2(1-2\rho)} \\ T_2 &= \frac{(2^{m'}W+1)}{2} \times \frac{\rho^{m'+1}(1-\rho^{m-m'})}{(1-\rho)}. \end{aligned} \tag{7}$$

Then, the probability τ that a node transmits in a randomly chosen slot time can be expressed as

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{1-\rho^{m+1}}{1-\rho} b_{0,0}. \tag{8}$$

For convenience of the following discussions, we write the Markov modeling results of (8) into the following function f_M:

$$\tau = f_M(c, p). \tag{9}$$

In the stationary state, a node transmits a packet with probability τ, so the probability p that the transmission is collided conditioned that there are transmissions can be computed as follows, because collision must happen if there are at least two nodes to transmit simultaneously:

$$p = 1 - (1-\tau)^{N-1} \tag{10}$$

where N is number of competing nodes. Equations (9) and (10) can be solved by numerical computing methods to obtain the value of p or τ. Then, we can get the normalized system throughput S as

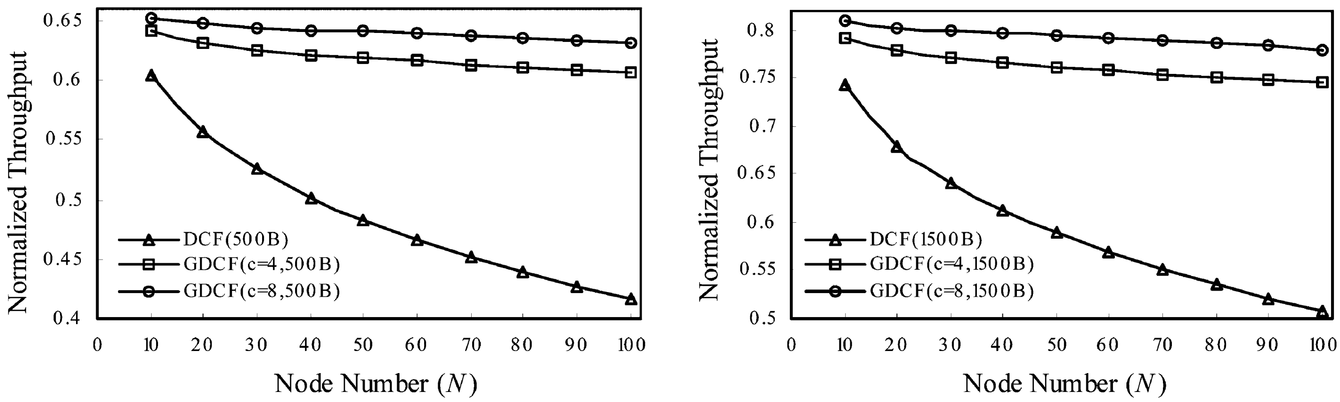
$$S = \frac{P_s P_{tr} E[L]}{(1-P_{tr})\sigma + P_s P_{tr} T_s + (1-P_s) P_{tr} T_c} \tag{11}$$

where P_{tr} is the probability that there is at least one transmission in the considered slot time. P_s is the probability that a transmission is successful. T_s and T_c are the average time the channel is sensed busy because of a successful transmission or collision, respectively. The $E[L]$ represents the average packet length and σ is the duration of an empty slot time. P_{tr} and P_s can be computed as

$$\begin{aligned} P_{tr} &= 1 - (1-\tau)^N \\ P_s P_{tr} &= N\tau(1-\tau)^{N-1}. \end{aligned} \tag{12}$$

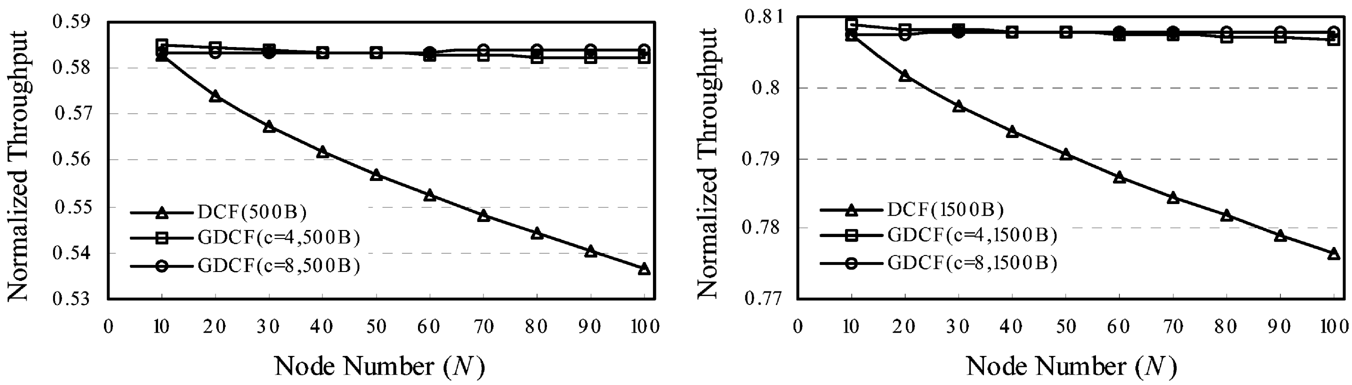
T_s and T_c can be computed for basic access mode and RTS/CTS access mode, respectively, as shown in (13) and (14) at the bottom of the next page where $H = PHY_{hdr} + MAC_{hdr}$ is the packet header, δ is the propagation delay, and $E[L^*]$ is the average length of the longest packet payload involved in a collision. In this paper, all the packets have the same fixed size, so $E[L] = E[L^*] = L$.

We have calculated the normalized system throughput of GDCF and DCF according to (11). The results are presented in Figs. 5 and 6, respectively for basic access and RTS/CTS access modes. It can be easily observed that the influences on throughput resulted from factors such as access mode, packet length, and c value in GDCF. First, for both DCF and GDCF, RTS/CTS access mode and/or large packet size will bring



(a) (b)

Fig. 5. Normalized throughput of basic access mode. (a) Packet length 500 B and (b) packet length 1500 B.



(a) (b)

Fig. 6. Normalized throughput of RTS/CTS access mode. (a) Packet length 500 B and (b) packet length 1500 B.

higher throughput. GDCF can obtain improved performance for both access modes, but the improved performance under the basic access mode is much larger. Under the basic access mode, the improved performance will increase with the increasing of the packet length, because the effect resulted from lowered collision probability will be more apparent under long packet length. On the contrary, under RTS/CTS access mode, when the packet length is small, the collided slot is more comparable, so the improved performance will decrease with the increasing of packet length. As the results in Fig. 5, GDCF with $c = 8$ obtains higher throughput than $c = 4$ under the basic access mode. However, both $c = 4$ and 8 obtained nearly the same throughput under the RTS/CTS mode, so the problem is how to choose the optimal c for different competing node numbers. This problem will be further analyzed in Section IV-B.

B. Optimal Value for c

It can be seen that the c value will heavily influence the throughput performance. The problem is of which value of c is the most optimal for throughput conditioned that the system parameters, such as in Table I, are given. We can use the following method to determine the optimal value of c. Let us write (11) to the following form:

$$S = \frac{E[L]}{(T_s + f(\tau))}$$
$$f(\tau) = \frac{(1-\tau)^N\sigma + [1-(1-\tau)^N - N\tau(1-\tau)^{N-1}]T_c}{N\tau(1-\tau)^{N-1}}. \tag{15}$$

In order to maximize S, $f(\tau)$ must be minimal, so let $f'(\tau) = 0$ and we can get the optimal value of τ as

$$(1-\tau)^N\sigma + [1 - N\tau - (1-\tau)^N]T_c = 0.$$

When N is too large, we can use the approximation

$$(1-\tau)^N \approx 1 - N\tau + \frac{N(N-1)}{2}\tau^2.$$

$$\begin{cases} T_s^{bas} = DIFS + H + E[L] + \delta + SIFS + ACK + \delta \\ T_c^{bas} = DIFS + H + E[L^*] + SIFS + ACK \end{cases} \tag{13}$$

$$\begin{cases} T_s^{rts/cts} = DIFS + RTS + SIFS + \delta + CTS + SIFS + \delta + H + E[L] + SIFS + \delta + ACK + \delta \\ T_c^{rts/cts} = DIFS + RTS + SIFS + CTS \end{cases} \tag{14}$$

TABLE I
SYSTEM PARAMETERS (802.11 DSSS)

Packet payload	11680bits
MAC overhead	224bits
PHY header	192µs
ACK	112bits+PHY header
RTS	160bits+PHY header
CTS	112bits+PHY header
Channel bit rate	2Mbps
Propagation delay	1us
Slot time	20us
SIFS	10us
DIFS	50us
W=32, m'=5.0, m=7.0	

Let $T_c' = T_c/\sigma$ [T_c can be computed according to (13) and (14) for basic access and RTS/CTS access mode, respectively] be the normalized average collision length in the number of slot times; then, we can finally obtain the optimal value of τ as

$$\tau_0 = \frac{\sqrt{\frac{[1+2(N-1)(T_c'-1)]}{(N-1)}}}{(N-1)(T_c'-1)}. \tag{16}$$

After obtaining the optimal value of τ, we can easily obtain the optimal value of c according to (9) and (10). Because the optimal value of τ is dependent on node number N and the normalized average collision length T_c' [see (16)], so the optimal value of c is also dependent on N and T_c'. However, we can see from (13) that T_c' is related to packet length under basic access mode, so the optimal value of c under the basic access mode is also dependent on packet length. When the packet length is large, the collision will cause heavy influence on throughput, so it is required to choose large value of c. Under the RTS/CTS access mode, because the normalized average collision length T_c' is independent of packet length, the optimal value of c is also irrelevant to it, so we only present the results of RTS/CTS access mode in Fig. 7, which also presents the minimal and maximal value of c to make the GDCF throughput higher than DCF, where the case of $c = 0$ means that DCF obtains higher throughput at this time (for example, when node number is equal to 2 and 4). It can be observed that: 1) When $N \geq 6$, GDCF can improve throughput performance (see the dashed curve labeled with "minimal c"), even through the original purpose of GDCF is to improve performance for large node number and 2) when $N \geq 10$, the optimal value of c for any node number (≥ 100) is smaller than the minimum of maximal value of c, so when $N \geq 10$, GDCF with any value of c in the range of 1 c 8 will obtain higher throughput than DCF. Fig. 6 conclusively demonstrates that the optimal c can be obtained in the narrow range 4 c 8 and nearly independent of node number N when the node number is larger than 10. This implies that even though one of the original arguments in GDCF is to improve performance for large node number, to our pleasant surprise, this essentially shows that GDCF obtains better performance even if the node number is small when c is chosen properly. We will further verify this result through simulations in Section V.

C. Performance Under Priority Traffics

Assume that total competing nodes (N) can be divided into M priorities or groups. Each group i has N_i competing nodes and is configured with parameter c_i. It is obvious that the group with small value of c_i will has the large probability τ_i and that the node belonging to this group will transmit in a slot time and have large probability p_i that the transmissions from this group is collided. Let the normalized throughput obtained by group i be S_i. Thus, we can describe each group i using the five-tuple $(N_i, c_i, \tau_i, p_i, S_i)$. In the following, we will calculate the throughput ratio between any two groups i and j.

We can write the results from Markov-modeling into the following function f_M:

$$\tau_i = f_M(c_i, p_i). \tag{17}$$

When the transmission is issued by only a node of group i and all other nodes in this group and other group keep idle, this transmission will succeed. Otherwise, the transmission will be collided. So the probability p_i can be computed as

$$p_i = 1 - (1-\tau_i)^{N_i - 1} \prod_{j \neq i, j=1}^{M} (1-\tau_j)^{N_j}. \tag{18}$$

For given parameters N_i and c_i, the numerical results of p_i and τ_i can be obtained through solving (17) and (18). Then, the throughput ratio between any two groups i and j can be calculated as

$$\frac{S_i}{S_j} = \frac{N_i \tau_i (1-p_i)}{N_j \tau_j (1-p_j)}. \tag{19}$$

Without loss of generality, it is assumed that $c_1 \leq c_i \leq \cdots \leq c_M$. According to the principle of GDCF, the node with small c has the large probability that it will transmit in a slot time, so $\tau_1 \geq \tau_i \geq \cdots \geq \tau_M$. From (18) we can easily get the following conclusion:

$$\begin{aligned} &\tau_i \geq \tau_j \overset{(18)}{\Rightarrow} p_i \leq p_j \\ \text{If } N_i = N_j, \quad &p_i \leq p_j \overset{(19)}{\Rightarrow} S_i \geq S_j. \end{aligned} \tag{20}$$

Through selecting different combination of c for competing nodes, GDCF can make the nodes with smaller c obtain lower MAC access delay (because of lower probability τ_i) and larger throughput (and corresponding total queuing delay if the traffic arrival rate is limited). This interesting property in GDCF can be directly utilized to support differentiated QoS in the MAC layer. For example, if some node needs supporting real-time applications , has better channel quality, or has lower energy, we can let it choose a small value of c to obtain a lower delay of real-time application, optimal throughput, or higher energy efficiency and longer system alive time. Although there are some other QoS-supporting approaches, such as fair queuing in the MAC layer, GDCF is very simple and flexible. We will further exploit this property through simulations in Section V.

Although c is assumed to be an integral up to now, c can be also configured to be a real number. Then, GDCF needs only a minor change, as presented in Fig. 8. However, this change will not influence the simplicity of GDCF.

In Fig. 8, *conSuccNum* is the continuous successful transmissions and *deficit* is the additional parameters used to count the accumulated deficit quantum in each node. Through using parameter *deficit*, GDCF can guarantee that node will halve CW

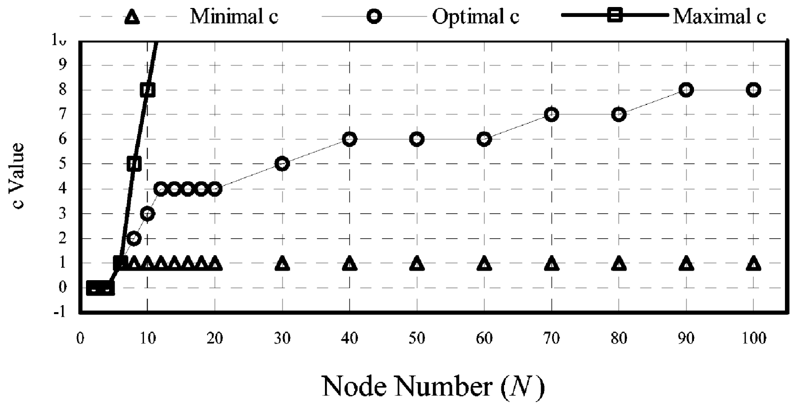


Fig. 7. Optimal c value under the RTS/CTS access mode (when $c = 0$, which means that DCF is the best choice at this time).

```
Initialize: deficit=0;
When there is a successful transmission
  If (conSuccNum>(c-deficit))
        Halve CW;
        deficit=deficit+(conSuccNum-c);
```

Fig. 8. GDCF algorithm when c is a real number.

once every c continuous successful time, on average, even if c is a real number.

V. Simulation Results

This section presents the extensive simulation results of GDCF. Since the collision probability has more influence on the basic access mechanism than on the RTS/CTS-based access mechanism, GDCF is sure to obtain better performance improvement under the nasic access mechanism than under the RTS/CTS-based access mechanism as that shown in Section IV. Therefore, here we only present results using the RTS/CTS-base access mechanism to observe the performance improvement in GDCF. The same parameters in Table I are also used in simulations. The nodes are uniformly distributed in the 100 m $\times$ 100 m two-dimensional (2-D) square spaces. We use the shadowing propagation model in order to match the real environment and assume that the 90% packet can be correctly received within the distance of 150 m. The main performance metrics of interest are system throughput, fairness index, RTS failure ratio, and QoS-supporting capability. The throughput is used to quantify the throughput gain obtained by GDCF. We adopt the use of fairness index defined in [12], as it is a commonly accepted metric. The RTS failure ratio (RFR) can be used to evaluate the energy cost to transmit packets. If RFR is large, then the energy cost will be high because more RTS messages are collided and more energy will be wasted. The simulation time is selected to be 100 s and all the following results are the average values obtained in ten simulations. In all the figures of this section, we use $c = 0$ to represent the 802.11 DCF algorithm, for convenience of comparison with the GDCF algorithm.

A. Saturation Traffic

In this section, the traffic is configured to saturate the system, so there are always some competing nodes attempting to transmit packets. We will investigate the throughput, fairness, and RTS failure ration in GDCF and DCF, respectively. The optimal value of c is obtained and compared to the analyzed results in Section IV.

Experiment A.1—TCP Traffic: In this case, all traffic sources are TCP (NewReno) flows, whose packet size and window size are 1460 B and 40 packets, respectively. We collect the RTS failure ratio, saturation throughput, and fairness for a different number of competing node.

As analyzed in Section IV, GDCF obtains higher throughput than DCF and keeps good fairness simultaneously. The results are presented in Table II and Fig. 9, respectively, for a smaller node number and large node number. Observing from Fig. 9 and Table II, GDCF obtains higher throughput and better fairness than DCF. When c is between 4 c 8, GDCF improves throughput by about 15%–20% for large node number [Fig. 9(a)] and a bit for small node number (Table II). GDCF also maintains good fairness property while obtaining higher throughput. Under large node number, GDCF has much better fairness than DCF, especially when $c \geq 4$ [Fig. 9(b)]. The larger value of c (for example, $c = 15$), the better fairness will be obtained because large value of c will make all competing nodes stay in the same backoff stage with high probability. Although the fairness of GDCF can be worse than DCF when the node number is small (Table II), GDCF still has good fairness (>0.9) and this deterioration is only slight. If the fairness index is smaller than 1, the bandwidth obtained in some nodes will be smaller than the average bandwidth by $\varepsilon(0 < \varepsilon < \overline{BW})$. According to the definition of fairness index in [11], if the fairness index is the same, the decreased bandwidth ε will be decreased when the total node number is small. So the fairness deterioration in GDCF under the small node number will cause slighter influence on bandwidth sharing than the fairness deterioration in GDCF under a large node number. Moreover, the total system throughput and the average bandwidth of each node under small node is large than that under large node, which will make the influence resulted from fairness deterioration in GDCF under small node number more slight.

Fig. 10 presents the results of RTS failure ratio, from which it can be seen that: 1) GDCF has smaller RTS failure ratio than DCF for any c value; 2) the RTS failure ratio of GDCF will decrease when c value increases, but it will keep at a certain level (for example, the two curves for $c = 8$ and $c = 15$ are nearly

TABLE II
THROUGHPUT AND FAIRNESS: SMALL NODE NUMBER ($c = 0$ IS FOR DCF)

Node	Throughput (Kbps)					Fairness				
	c=0	*c*=1	*c*=4	*c*=8	*c*=15	*c=0*	*c*=1	*c*=4	*c*=8	*c*=15
2	*1292.23*	1291.51	1280.62	1263.71	1235.35	*1*	1	1	1	1
4	*1286.64*	1287.5	1278.11	1257.84	1224.6	*0.99976*	0.99664	0.97739	0.9681	0.96359
6	*1275.04*	1278.26	1277.28	1256.61	1220.69	*0.99693*	0.98678	0.95947	0.93705	0.9462
8	*1252.65*	1265.8	1276.81	1256.25	1223.2	*0.9889*	0.98469	0.95275	0.94354	0.95341
10	*1241.09*	1260.17	1273.26	1258.39	1225.43	*0.96775*	0.97058	0.93946	0.9285	0.9736

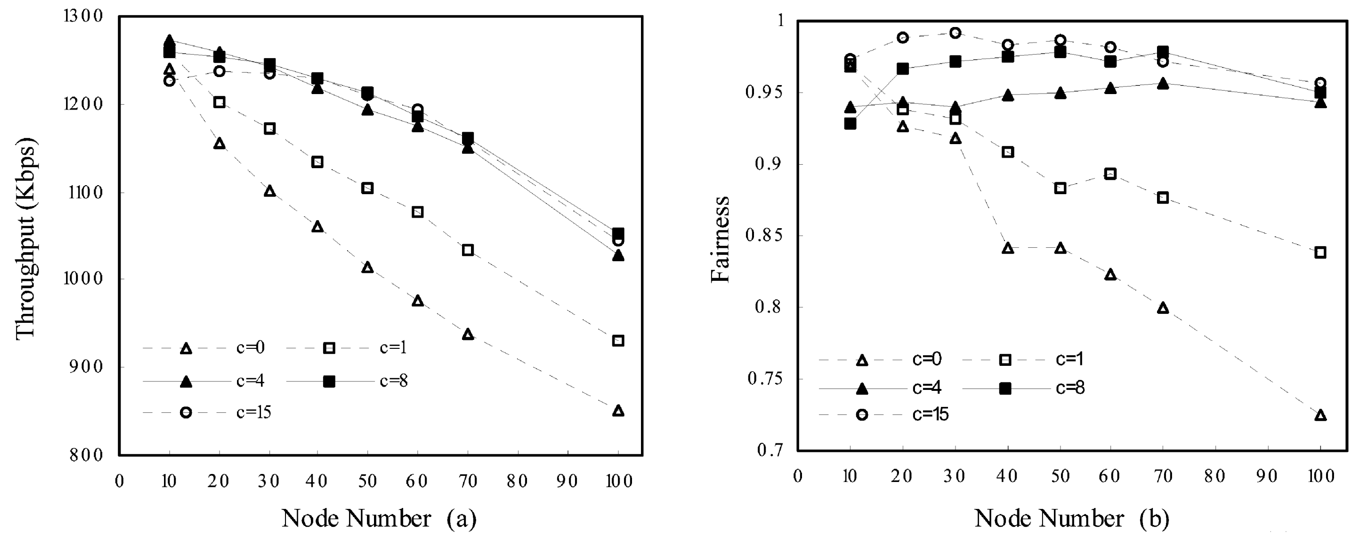


Fig. 9. Throughput and fairness-large node number ($c = 0$ is for DCF).

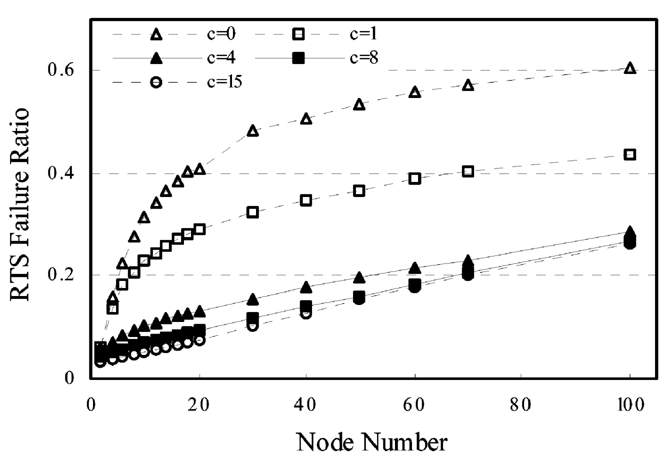


Fig. 10. RTS failure ratio-saturation traffic.

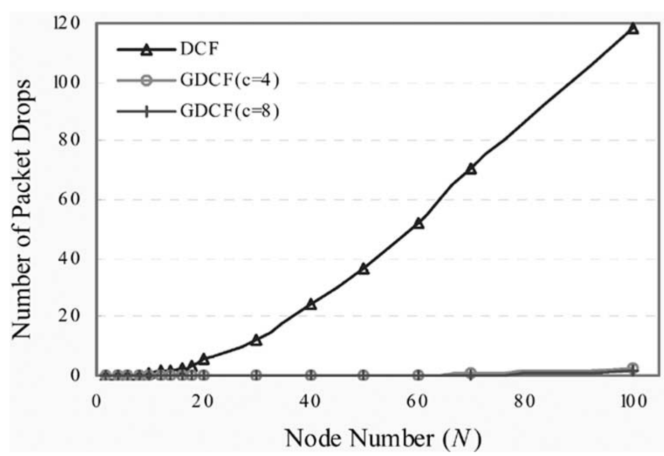


Fig. 11. Packet dropping in the MAC layer.

overlapped); and 3) the gain of the RTS failure ratio obtained by GDCF is more apparent when the node number is large; 4) the RTS failure ratio will increase when the node number increases and GDCF decreases the backoff stage by only 1 if and only if there are c continuous transmissions, so it will keep larger contention window with high probability than DCF and has larger RTS success ratio independent of node number, as in Fig. 10. Also, it is reasonable that RTS failure ratio will increase when c decreases or node number increases. The results do show than GDCF has lower RTS failure ratio than DCF. This means that nodes in GDCF issue fewer RTS message than DCF for transmitting the same information volume.

In simulations, we also collect the packet drops in the MAC layer throughout the whole simulation time and Fig. 11 presents the results for different number of competing node. It can be observed that there are few packet drops in the MAC level under GDCF. On the contrary, DCF cause many packet drops in MAC level, especially when there are many competing nodes.

Experiment A.2—UDP Traffic: We also change the input traffic from TCP into UDP and vary the total traffic density from 0.1 to 1.0. GDCF also obtains the improved throughput. Because the fairness under each case is close to 0.99 and has little difference under GDCF and DCF, we only present the results of system throughput (see Fig. 12). When UDP traffic

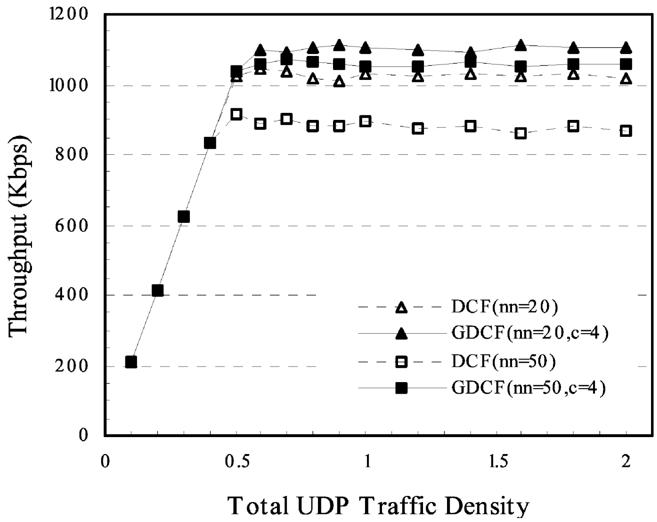


Fig. 12. System throughput—UDP traffic.

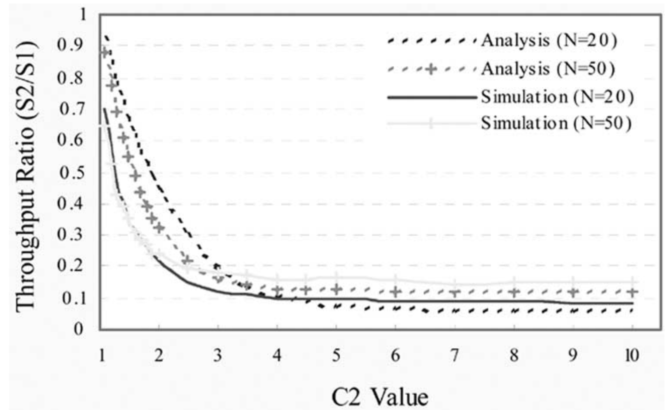


Fig. 13. Experiment B.1—throughput ratio.

density is smaller than the system saturation throughput, the system throughput of DCF and GDCF will increase with the UDP density. After the UPD density overruns the saturation throughput, the system of DCF and GDCF will nearly keep constant. Similar to the results in Fig. 10, GDCF also obtains a bit higher throughput than DCF under this case. It can also be observed that the throughput under UDP is somewhat smaller than that under TCP, especially when the node number is equal to 50. The reason is may be that the factually competing node number may smaller than the total node number because of TCP ACK-based flow and congestion-control mechanism.

In summary, GDCF obtains higher throughput and better fairness at the same time. On the selection of the c value, it can be seen from the above discussions (Table II and Fig. 9) that GDCF with 1 c 4 has better performance when the node number is very small (between 2 and 6) and 4 c 8 is more suitable for cases with a large node number. Considering the tradeoff between throughput decrease (under very small node number) and throughput and fairness improvement (under large node number), 4 c 8 is the better choice if the number of competing node cannot be known. Recall that the optimal c value (4 c 8) from theoretical in Section IV; we can conclude that the simulation results are very consistent with the theoretical results in Fig. 7, so we can choose c value from 4 c 8 in practical deployment.

B. QOS Supporting

Recently, it was shown in 802.11e and enhanced DCF (EDCF) [8] that it can provide QoS differentiation by configuring small (or large) $CW_{\min}$ and $CW_{\max}$ and DIFS for high- (or low-) priority traffic. In a DCF environment, however, it is hard to deploy admission control for the high-priority traffic, so EDCF will cause performance deterioration if most of applications are high-priority traffics with small $CW_{\min}$ and $CW_{\max}$ value. One of the properties in GDCF is that the node with smaller c will get the access chances more quickly. We can use this property directly to support QoS differentiation. In this section, we investigate the priority-supporting ability in GDCF.

Experiment B.1: In this experiment, we divided the total nodes into two groups with the same node number. The c value in the first group is fixed at $c1 = 1$, and the $c2$ for the second group is varied from 1.1 to 10. UDP traffic is used in simulations. According to the analysis in Section V-A, the first group will obtain higher throughput than the second group, which has larger value of c. We calculate the throughput ratio between the two groups using (19) and compare it with the simulation results (see Fig. 13). When $c2 = 2$, there is a large throughput decrease in the second group. When $c2 > 4$, the throughput ratio curve is gradually flat and the increase of $c2$ has no apparent effect. Although there is some mismatch between analysis and simulation results, the first group with smaller c does obtain higher throughput. On the contrary, the second group obtains lower throughput. The throughput ratio between the two groups will decrease with the increasing of $c2$, so we can adjust the throughput ratio between different types of traffic in deployment through configuring c.

As for node number $N = 20$, we also collected instantaneous throughput and delay for each group. Fig. 14(a) presents the instantaneous throughput of group 1 (G1) and group 2 (G2) and Fig. 14(b) illustrates the instantaneous delay of randomly chosen nodes from these two groups, respectively. It can be seen that group 1 with a smaller value of c obtains higher throughput and lower delay, so GDCF provides a simple and flexible approach to supporting differentiated QoS.

Experiment B.2: In this experiment, two types of traffic are configured: 1) high-priority traffic $(c = c(h) = 1)$-UDP source and 2) low-priority traffic $(c = c(l) = 4)$-TCP source. The parameters used in TCP are the same as in Section V-A. The packet size of UDP is equal to 1500 B. The traffic density of the UDP source is varied from 0.05 to 0.6. In simulations, the node number is set to 20 and 40, respectively, and one half of nodes support UDP and the other half support TCP. We collect the UDP delay, UDP delay jitter, TCP delay, and total system throughput. Fig. 15 shows that: 1) the average delay and delay jitter of UDP source in DCF quickly increase with its traffic density; 2) the average delay and delay jitter of UDP source in GDCF is not sensitive to the competing node number and traffic density of UDP source; and 3) UDP source has very lower delay (0.02–0.035 s) and delay jitter under GDCF than that under DCF. At the same time GDCF achieves higher system

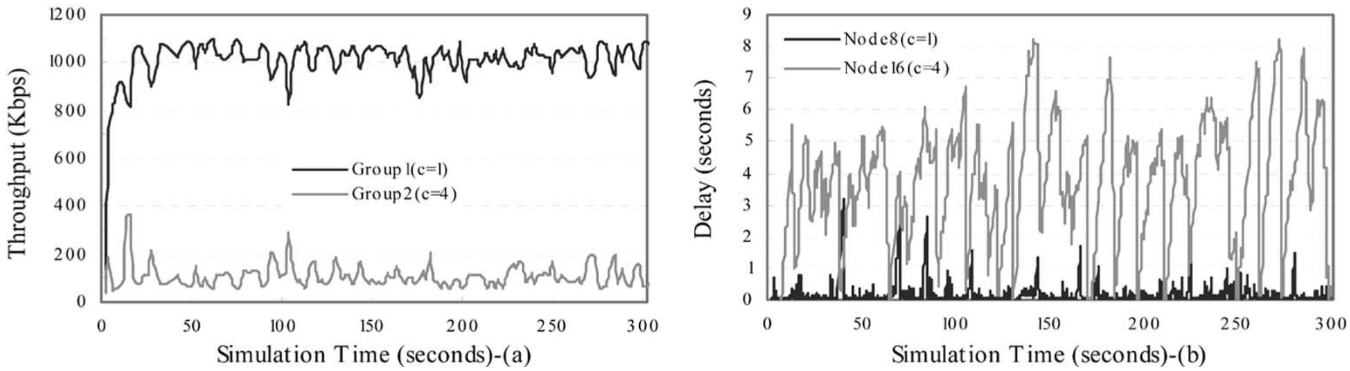


Fig. 14. Experiment B.1—instantaneous throughput and delay.

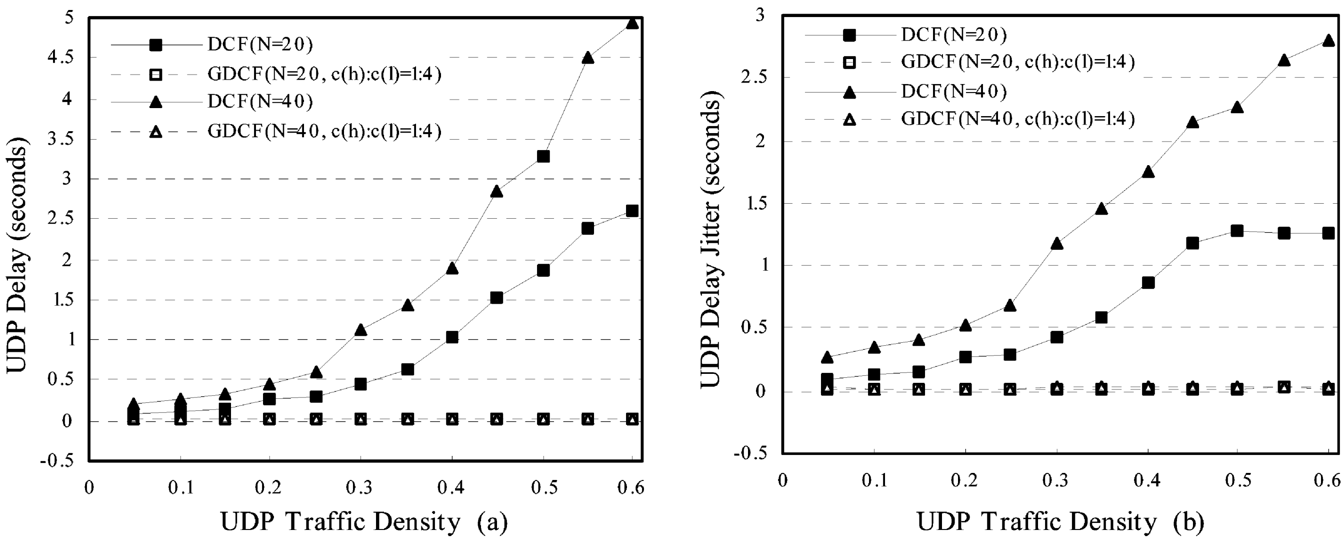


Fig. 15. Experiment B.2—UDP delay and delay jitter.

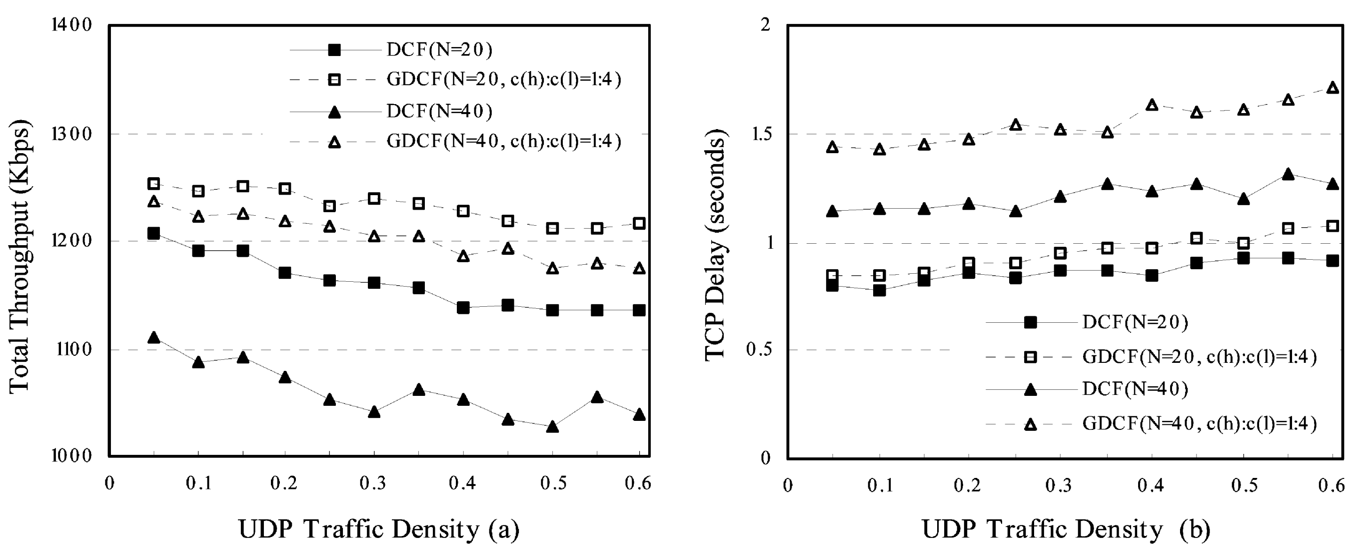


Fig. 16. Experiment B.2—Total throughput and TCP delay.

throughput (Fig. 16), but slightly bigger TCP delay than DCF (Fig. 16).

We have evaluated GDCF through such thee types of simulations: *TCP traffic, UDP traffic,* and *combined traffic*. The conclusions that GDCF improve better performance than DCF can be drawn from the extensive simulations. GDCF first acquires about 15%–20% higher saturation throughput than DCF. Second, GDCF simultaneously maintains good fairness. When the number of competing node is some large, GDCF has better fairness than DCF. Third, GDCF has lower RTS failure ratio, which means that GDCF will issue smaller RTS message and consumes fewer energy than DCF in order to transmit the same number of packets. GDCF also drops fewer packets at the MAC level, while DCF drops many packets in the MAC level, so GDCF realize better integration of the high-protocol layer and the low-MAC layer. Finally, GDCF can effectively support priority traffic. Through simple parameter configuration, GDCF can make high-priority traffics get much lower delay and delay jitter and provide differentiated QoS, and keep total higher throughput at the same time. GDCF needs neither changes of RTS/CTS structure nor measures of node number; thus, it is very easy to deploy. The GDCF proposed in this paper is deterministic approach. It is certain that the behavior of GDCF can be realized through some probability-based approaches. For example, we can decrease backoff stage by 1 and halve the contention window with probability f after each successful transmission. We also investigate the performance of this approach and find out that it realizes similar performance with GDCF and that the optimal value of f is about 0.2. But the fairness of this approach is a bit worse than GDCF because of its probabilistic behavior in contention resolutions. In future work, we will plan to evaluate the priority–guarantee mechanism in GDCF and its performance under multihop environments.

VI. Conclusion

This paper investigates the MAC protocol for WLAN and the corresponding collision-resolution algorithm and proposes an effective algorithm, GDCF, based on 802.11 DCF protocols. Theoretical analysis and simulations are carried out, which show that the proposed GDCF brings several benefits: 1) it obtains higher throughput than traditional DCF, especially with a large number of competing nodes; 2) it maintains a good fairness property; 3) GDCF has a lower RTS failure ratio and issues less RTS messages than DCF in order to transmit the same information volume, so it is more energy efficient; 4) GDCF drops fewer packets in the MAC level and can easily extend to support priority application with the flexibility of selecting different c values; and 5) GDCF is very easy to be deployed, as it does not need to estimate a competing node number or change the control message structure and access procedures in DCF.

In our future work, we will investigate the performance of GDCF when the number of nodes vary frequently and further study comparison between the performance of GDCF when supporting QoS and 802.11e [8].

References

[1] A. Chandra, V. Gummalla, and J. O. Limb, "Wireless medium access control protocols," *IEEE Commun. Surveys Tutorials*, vol. 3, no. 2, pp. 2–15, Apr. 2000.

[2] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, ISO/IEC 8802-11:1999(E), Aug. 1999.

[3] G. Bianchi, "Analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 535–547, Mar. 2000.

[4] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoritical throughput limit," *IEEE/ACM Trans. Networking*, vol. 8, pp. 785–799, Dec. 2000.

[5] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement," in *Proc. IEEE INFOCOM'02*, vol. 2, June 2002, pp. 599–607.

[6] P. Yong, H. Wu, S. Cheng, and K. Long, "A new self-adapt DCF algorithm," in *Proc. IEEE GLOBECOM'02*, vol. 1, Nov. 2002, pp. 87–91.

[7] Y. Kwon, Y. Fang, and H. Latchman, "A novel MAC protocol with fast collision resolution for wireless LANs," in *IEEE INFOCOM'03*, Apr. 2003.

[8] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e wireless LAN for quality of service," in *Proc. Eur. Wireless (EW'02)*, vol. 1, Feb. 2002, pp. 32–39.

[9] Y. Xiao, "A simple and effective priority scheme for IEEE 802.11," *IEEE Commun. Lett.*, vol. 7, pp. 70–72, Feb. 2003.

[10] Y. E. Sagduyu and A. Ephremides, "Energy-efficient collision resolution in wireless Ad-Hoc networks," presented at the IEEE INFOCOM'03, Apr. 2003.

[11] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," presented at the IEEE INFOCOM'03, Apr. 2003.

[12] R. Jain, *The Art of Computer Systems Performance Analysis*. New York: Wiley, 1991.

Chonggang Wang received the B.Sc. (Hons.) degree from Northwestern Polytechnic University, Xi'an, China, in 1996 and the M.S. and Ph.D. degrees in communication and information systems from the University of Electrical Science and Technology, Chengdu, China and Beijing University of Posts and Telecommunications, Beijing, China, in 1999 and 2002, respectively.

From September 2002 to Novemeber 2003, he was an Associate Researcher with the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, P. R. China. He currently is a Visiting Professor with the Special Research Centre for Optical Internet & Wireless Information Networks (ICOIWIN), ChongQing University of Posts and Telecommunications (CQUPT), Chongqing, P. R. China, and a Postdoctoral Research Fellow with the University of Arkansas, Fayetteville.

Bo Li (S'93–M'95–SM'99) received the B.S. (*summa cum laude*) and M.S. degrees in computer science from Tsinghua University, Beijing, P. R. China, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 1993.

From 1994 to 1996, he worked on high-performance routers and asynchronous transfer mode (ATM) switches with the IBM Networking System Division, Research Triangle Park, NC. Since January 1996, he has been with Computer Science Department, the Hong Kong University of Science and Technology, where he is an Associated Professor and Codirector for the ATM/Internet protocol (IP) cooperate research center, a government-sponsored research center. Since 1999, he has also been an Adjunct Researcher with Microsoft Research Asia (MSRA), Beijing, China. He has been an Editor or Guest Editor for 16 journals and is involved in the organization of about 40 conferences. His current research interests include wireless mobile networking supporting multimedia, video multicast, and all optical networks using WDM, in which he has published over 150 technical papers in referred journals and conference proceedings.

Dr. Li was the Co-TPC Chair for IEEE INFOCOM'04 and is a Member of the Association for Computing Machinery (ACM).

Lemin Li graduated from Jiaotong University, Shanghai, China, in 1952, majoring in electrical engineering.

From 1952 to 1956, he was with the Department of Electrical Communications, Jiaotong University, Shanghai, P. R. China. Since 1956, he has been with Chengdu Institute of Radio Engineering (currently the University of Electronic Science and Technology of China), Chengdu, P. R. China. From August 1980 to August 1982, he was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, University of California, San Diego, where he did research on digital and spread-spectrum communications. He currently is a Professor of Communication and Information Engineering. His research work is in the area of communication networks, including broad-band and wireless networks.

Mr. Li is a Member of the Chinese Academy of Engineering.