잡음 환경을 고려한 IEEE 802.11 DCF의 필터기반 Collision Resolution 메카니즘

정회원 유 상 신*

Filter-Based Collision Resolution Mechanism of IEEE 802.11 DCF in Noisy Environments

SangShin Yoo *Regular Member*

요 약

본 논문은 IEEE 802.11 DCF가 잡음과 부하가 고려된 실제적인 환경에서 효과적으로 작동할 수 있도록 환경 변화에 따라 contention window를 동적으로 적응시키는 필터 기반의 메카니즘을 제안한다. 이는 이미지 프로세싱에서의 미디언 필터개념을 적용시킨 것으로 필터링의 결과에 따라 윈도우의 크기를 조정한다. 이를 통하여 잡음과 충돌에 의해 필연적으로 패킷 전달에 문제가 발생할 수밖에 없는 무선 환경에서 불필요한 윈도우의 조정을 제거하고, 성능을 최적화 할 수 있게 된다. 또한 제안하는 방법은 기존에 전송된 결과를 반영하는 히스토리 비트-패턴을 이용함으로써 기존 연구결과들을 거의 포함하는 일반적인 방법으로 쉽게 확장할 수 있는 장점을 가지고 있다. 제안된 방법은 시뮬레이션을 통해 성능과 지연 측면에서 기존의 연구결과보다 확실한 향상을 가져옴을 확인할 수 있었다.

Key Words : IEEE 802.11 DCF, Contention Window, History-Buffer, NS-2

ABSTRACT

This paper proposes a filter-based algorithm to adaptively adjust the contention window in IEEE 802.11 DCF. The proposed mechanism is focused on the general and realistic environments that have various conditions regarding to noise, media types and network load. For this flexible adaptation, Filter-based DCF(FDCF) takes a more realistic policy such as median filter concept in the image processing technologies. We can handle these various environments by adjusting the contention window size according to the result of filtering based on history-buffer. We can ignore temporarily and randomly occurred transmission failures due to noise errors and collisions in noisy environments. In addition, by changing the reference number and history-buffer size, FDCF can be extended as a general solution including previous proposed mechanism. We have confirmed that the proposed mechanism can achieve the better performance than those of previous researches in aspects of the throughput and the delay in the realistic environments.

Ⅰ. Introduction

Most of the contention based MAC protocols follow the operational model of carrier sense multiple access (CSMA), incorporating handshaking signals and a back-off mechanism to reduce the

* 성공회대학교 소프트웨어공학과 (ssyoo@skhu.ac.kr)

논문번호 : KICS2007-01-019, 접수일자 : 2007년 1월 19일, 최종논문접수일자 : 2007년 8월 8일

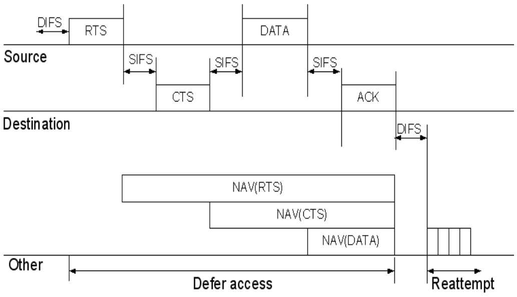


Fig. 1. 802.11 DCF RTS/CTS access mode

probability of collisions. The collision is inevitable due to the fundamental problems such as propagation delay, hidden terminal problem and exposed terminal problem in WLAN [1].

IEEE 802.11 MAC(Medium Access Control) designed for wireless LAN defines two access mechanisms: Point Coordination Function (PCF) and Distributed Coordination Function (DCF).

PCF is a centralized algorithm for the contention-free service, while DCF has a contention-based algorithm to provide access to all traffic.

In this paper, we only concern about the DCF. The IEEE 802.11 DCF protocol can operate in one of two different modes: infrastructure mode and ad hoc mode. In infrastructure mode communication between nodes must go through a central node, while in ad hoc mode nodes communicate directly with each other. Infrastructure mode fits the sensor networks in which sensors within a cluster communicate directly with their cluster head. On the other hand, ad hoc mode fits the sensor networks in which sensors communicate directly each other without any cluster head. In the IEEE 802.11 DCF, each node can start a transmission if the medium is sensed to be idle for an interval larger than the Distributed Inter Frame Space (DIFS). If the medium is busy, the node will start the deferring process as shown in Figure 1. The deferring process defers the transmission until a DIFS is passed and then generate a random back-off timer before transmitting the packet. The back-off timer will be decreased while the channel is sensed to be idle, and frozen while the channel is sensed to be busy, and resumed to decrease while the channel is sensed to be idle again. A node can initiate a transmission when the back-off timer reaches zero. The back-off timer is uniformly chosen in the range [0, CW]. CW is known as Contention Window, which is an integer with the range determined by the PHY characteristics $CW_{\min}$ and $CW_{\max}$. After each unsuccessful transmission, CW will be doubled till to the maximum value $CW_{\max}=2^{m-1}W_0-1$, where W0 equals to $CW_{\min}$. After each successful transmission, CW will be reset to $CW_{\min}$.

802.11 DCF defines two channel access modes: basic access mode and RTS/CTS-based access mode. In case of RTS/CTS-based access mode, nodes transmit data by using special short RTS and CTS frames prior to the transmission of actual data frame in order to shorten the collided time interval. 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 immediately after SIFS interval. 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 are capable of updating the NAVs based on the RTS from the source node and the CTS from the destination node, which helps to circumvent the hidden terminal problems. In fact the nodes that is able to receive the CTS frames correctly, can avoid collisions even when it cannot sense the data transmissions from the source node. If a collision occurs with two or more RTS frames, much less bandwidth is wasted when compared with situations that larger data frames collide each other in basic access mechanism.

Through much effort for the solution of collision resolution mechanism, considerable improvement has been achieved. However, most of the existing mechanisms consider the environments that the channel is error-free.

Therefore, transmission failure occurs only due to the collision through the contention between nodes. Those may fit well in ideal environments, but they aren't appropriate for the realistic environments that have the time-varying network load and cause the errors randomly due to the noise and movement. So we need to have other approaches. In this paper, we try seeking for a general solution that can be adaptable in realistic environments.

The remainder of this paper is as follows. Section 2 reviews the solutions of the previous researches, and discusses the main features in our proposed filter-based DCF(FDCF).

Section 3 explain the concept and the analysis model. And then performs the theoretical analysis of FDCF on the throughput and delay. In Section 4, we evaluate the FDCF through the simulation. And we compare the performance of FDCF with the GDCF and the standard DCF. In Section 5, we conclude the paper.

Ⅱ. Related Works

This paper focuses on the contention-based MAC protocol, specifically IEEE 802.11 DCF [1]. Several recent researches on this topic have been worked [4, 13, 14, 8, 3, 10, 5]. The analysis in [2] shows that the throughput and fairness of 802.11 DCF can be significantly deteriorated when the large number of nodes participates in communication. But it was not considered the recently emerging applications such as real-time media and mobile ad-hoc networks. Some of these works are related with priority supporting for real-time multimedia applications [9]. Others are energy-saving MAC protocols that can be used in mobile ad hoc networks where each node assumes to be energy-limited [11].

As a important approach, there is an attempt to avoid useless collisions through the "gentle" decrease of contention window, referred as gentle DCF or GDCF [12]. GDCF is a major algorithm competed with our proposed protocol. GDCF halves CW value if there are consecutive successful transmissions. For this purpose, GDCF needs to maintain a counter for counting the number of continuous successful transmissions. This counter will reset to zero after each collision, because what it count is the number of continuous successful transmissions, not the number of total successful transmissions. In the case of collision, similar to the operations in DCF, GDCF will double the contention window and select a back-off timer value uniformly from $[0, CW]$. If there are Consecutive successful transmissions, GDCF will halve the CW and select a back-off 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. If the channel is idle, GDCF also reduces the back-off timer by 1, the same as in DCF.

GDCF can get a better performance result by avoiding useless collisions in the saturated state. It is very reasonable because only one successful transmission does not mean that network traffic is light. However, it requires an important assumption that unsuccessful transmission is occurred due to not noise error but only collision. It is not the realistic environments.

Table 1: State Transition of the Standard DCF

R_t	SDCF
0	Reset(W_0)
1	Increment

Table 2: State Transition of the GDCF

$R_{\_4}$	$R_{\_3}$	$R_{\_2}$	$R_{\_1}$	R_t	GDCF
0	0	0	0	0	Decrement
X	X	X	1	0	Hold
X	X	1	X	0	Hold
X	1	X	X	0	Hold
1	X	X	X	0	Hold
X	X	X	X	1	Increment

Based on this observation and insight, we propose a new filter-based collision resolution mechanism called FDCF. We can handle these various environments by adjusting the contention window size according to the result of filtering based on the history-buffer information. The filtering is being started counting the cumulative number of failed transmissions among the recently tried transmissions. If the failed transmission counter is greater than the reference number n, even though last transmission was succeeded, FDCF will stop halving the contention window size and keep the current size unchangeable. If the failed transmission counter is less than the reference number n, even though last transmission was failed, FDCF will stop doubling the contention window size and keep the current size. As a result, we can ignore temporarily and randomly occurred transmission failures due to varying noisy and collision environments. In addition, by changing the reference number and history-buffer size, FDCF can be extended as a general solution including previous proposed mechanism.

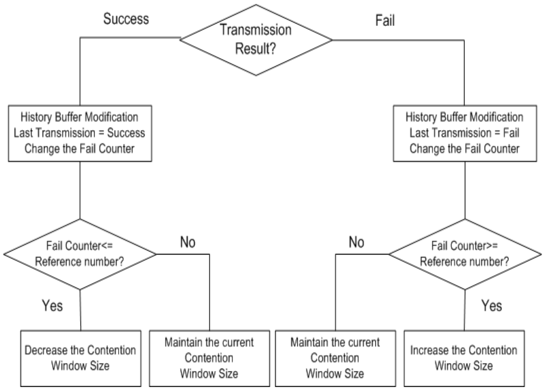

Success
Transmission Result?
Fail
History Buffer Modification
Last Transmission = Success
Change the Fail Counter
History Buffer Modification
Last Transmission = Fail
Change the Fail Counter
Fail Counter<= Reference number?
No
No
Fail Counter>= Reference number?
Yes
Yes
Decrease the Contention Window Size
Maintain the current Contention Window Size
Maintain the current Contention Window Size
Increase the Contention Window Size

Fig. 2. The flow chart of the FDCF

Ⅲ. Filter-Based DCF

To overcome the rigidness of the previous contention window scheme, we propose a filter-based decision making scheme. The new mechanism provide a gentle treatment on the contention window under a noise environment avoiding to increase the window unnecessarily. FDCF takes a more flexible policy by adjusting the contention window size according to the result of history-buffer based filtering. This mechanism operates effectively by preventing the useless fluctuation of contention window size, especially when the competing node number is large and noisy error rate cannot be ignored.

For the purpose of clear understanding of the differentiation among various approaches, we tried to show the transition of contention window size using simple tabular form. At first, SDCF(Standard DCF) can be explained, as shown in table 1. If R_t means the result of the last received packet(0:success, 1:fail), the change of the contention window size will be entirely determined according to that value. If R_t is 0, contention window size will be initialized as W0. If R_t is 1, contention window size will be doubled.

GDCF actually counts the number of consecutively successful transmissions. If the counter value is greater than or equal to the reference value c, the contention window size will be halved. If not, even though last packet transmission is succeeded, contention window size will not be changed. If last transmission is failure, window size always will be doubled. This mechanism can be expressed by history buffer as shown in table 2(in case of c=4). Rt means the result of current time and R_{-1}, R_{-2}, R_{-3}, and R_{-4} mean the results of the previous time respectively.

FDCF uses the concept of history-buffer. Figure 3 shows the overall flow of the FDCF. The history-buffer based filtering of FDCF is started by checking the pattern of failed transmissions among the recently tried transmissions. It determines the success or failure based on the two informations. First is the result of current transmission(R_{-1}). The other is related with history-buffer contents that maintains the success or failure records of recently received packets.

This mechanism can be expressed by history buffer as shown in table 3. In other words, when the current transmission is failed or succeeded,

Table 3: State transition of the FDCF : in case of c=4 and n=1

R_{-4}	R_{-3}	R_{-2}	R_{-1}	R_t	FDCF
0	0	0	0	0	Decrement
0	0	0	1	0	Decrement
0	0	1	0	0	Decrement
0	0	1	1	0	Hold
0	1	0	0	0	Decrement
0	1	0	1	0	Hold
0	1	1	0	0	Hold
0	1	1	1	0	Hold
0	0	0	0	1	Hold
0	0	0	1	1	Increment
0	0	1	0	1	Increment
0	0	1	1	1	Increment
0	1	0	0	1	Increment
0	1	0	1	1	Increment
0	1	1	0	1	Increment
0	1	1	1	1	Increment
1	0	0	0	0	Decrement
1	0	0	1	0	Hold
1	0	1	0	0	Hold
1	0	1	1	0	Hold
1	1	0	0	0	Hold
1	1	0	1	0	Hold
1	1	1	0	0	Hold
1	1	1	1	0	Hold
1	0	0	0	1	Increment
1	0	0	1	1	Increment
1	0	1	0	1	Increment
1	0	1	1	1	Increment
1	1	0	0	1	Increment
1	1	0	1	1	Increment
1	1	1	0	1	Increment
1	1	1	1	1	Increment

Table 4: Various mechanisms expressed by History-buffer method in case of c=3

History Pattern				Action			
R_{-3}	R_{-2}	R_{-1}	R_t	FDCF(n=1)	FDCF(n=2)	GDCF	SDCF
0	0	0	0	Dec	Dec	Dec	Dec
0	0	1	0	Dec	Dec	Hold	Dec
0	1	0	0	Dec	Dec	Hold	Dec
0	1	1	0	Hold	Dec	Hold	Dec
1	0	0	0	Dec	Dec	Hold	Dec
1	0	1	0	Hold	Dec	Hold	Dec
1	1	0	0	Hold	Dec	Hold	Dec
1	1	1	0	Hold	Hold	Hold	Dec
0	0	0	1	Hold	Hold	Inc	Inc
0	0	1	1	Inc	Hold	Inc	Inc
0	1	0	1	Inc	Hold	Inc	Inc
0	1	1	1	Inc	Inc	Inc	Inc
1	0	0	1	Inc	Hold	Inc	Inc
1	0	1	1	Inc	Inc	Inc	Inc
1	1	0	1	Inc	Inc	Inc	Inc
1	1	1	1	Inc	Inc	Inc	Inc

this mechanism decides whether current contention window size will be changed or not, based on transmission history-buffer information.

At first, if the result of current transmission is failure, it is able to choose one of two possibilities. If the failed transmission counter is greater than or equal to the reference number n, we think that the transmission is failed by doubling the contention window size. That possibility is expressed as p^{f}_{down}.

However, if the failed transmission counter is less than the reference number n, we consider that this situation can be filtered or ignored. So the contention window size is not changed. That possibility is expressed as p^{f}_{hold}. Next, if the result of current transmission is success, it is also able to choose one of two possibilities. If the failed transmission counter is less than or equal to the reference number n, we think that the transmission is succeeded by halving the contention window size. That possibility is expressed as p^{s}_{up}. However, if the failed transmission counter is greater than the reference number n, we consider that this situation can be ignored or filtered. So the contention window size is not changed. That possibility is expressed as p^{s}_{hold}.

Table 3 shows an example of history-buffer size c=4 and reference number n=1.

As we can see through above description, various mechanisms including SDCF and GDCF can be expressed by the filter-based mechanism,

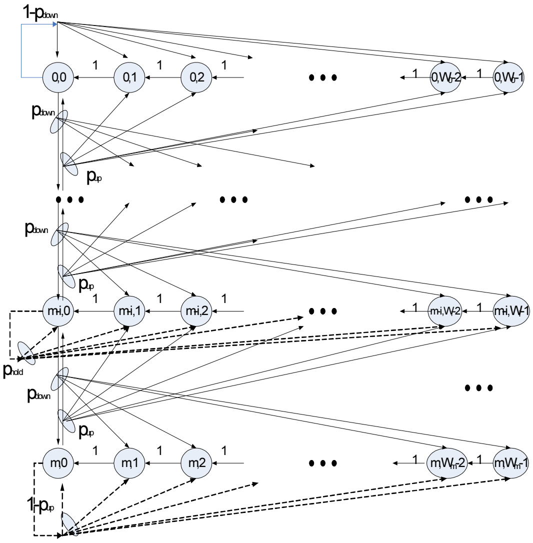


Fig. 3. The analysis Model of The FDCF

as shown in table 4. This is the reason whyFDCF can be extended as a general approach by the adjustment of the contention window size in DCF.

Let c be the history buffer size. Let x be the total failure counter within history-buffer and let n be the reference number. Based on these numbers, we can get p^f_{up}, p^f_{hold}, p^s_{down}, p^s_{hold}.

$$p^f_{up} = p_d \sum_{k=\eta}^{c} \binom{c}{k} (p_d)^k (1-p_d)^{c-k}, \text{if } x \geq \eta \quad (1)$$

$$p^f_{hold} = p_d \sum_{k=0}^{\eta-1} \binom{c}{k} (p_d)^k (1-p_d)^{c-k}, \text{if } x < \eta \quad (2)$$

$$p^s_{hold} = (1-p_d) \sum_{k=\eta+1}^{c} \binom{c}{k} (p_d)^k (1-p_d)^{c-k}, \text{if } x > \eta \quad (3)$$

$$p^s_{down} = (1-p_d) \sum_{k=0}^{\eta} \binom{c}{k} (p_d)^k (1-p_d)^{c-k}, \text{if } x \leq \eta \quad (4)$$

where p_d is the probability that a transmitted packet fails due to collision or error.

If a node in FDCF stays in a certain state, it can be moved to upper state, lower state, or current state according to the filter-based decision. In this paper, we let them p_{up}, p_{hold}, p_{down}, respectively. And it can be expressed as follows.

$$p_{up} = p^s_{up} \quad (5)$$

$$p_{hold} = p^f_{hold} + p^s_{hold} \quad (6)$$

$$p_{down} = p^f_{down} \quad (7)$$

This approach can adapt to various environments. It can be used as a general solution by adjusting c and n, without much modification of standard DCF. In addition, by considering the timing factor, this mechanism can accept the bursty data traffic that have severe fluctuations in times. So this approaches can include the merits of the previous DCF such as DCF and GDCF. And it can be used effectively for various traffic patterns in various environments.

Based on the description on FDCF, we can make the analysis model shown in Figure 3 [12,10,7]. And we will get the performance of FDCF using same procedures and symbols in [2] and [13]. Let p_d be the probability that a transmitted packet fails due to collision or error, and τ be the probability that a node transmits in a randomly chosen slot time, i be the back-off stage, m be the maximal back-off stage, k be the back-off time slot, (i,k) be the bi-dimensional 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). If the result of filtering function is failure, FDCF will back-off (increase the stage i and double the contention window). The back-off timer will decrease by 1 if the channel is sensed idle. If the result of filtering function is success, FDCF will decrease the back-off stage i and halve the contention window. Otherwise the node will stay at the current back-off stage i and keeps the contention window size unchanged. Then we can easily construct corresponding transition equations of the Markov model. The non-null one-step transition probabilities can be computed as follows:

$$P\{i,k|i,k+1\} = 1 \text{ for } k \in [0,\ W_i - 2] \text{ and } i \in [0,m]$$

$$P\{0,k|0,0\} = \frac{1-p_{down}}{W_0} \text{ for } k \in [0,\ W_0 - 1]$$

$$P\{i,k|i-1,0\} = \frac{p_{down}}{W_i} \text{ for } k \in [0,\ W_0 - 1] \text{ and } i \in [1,m]$$

$$P\{m,k|m,0\} = \frac{1-p_{up}}{W_m} \text{ for } k \in [0,\ W_m - 1]$$

$$P\{i-1,k|i,0\} = \frac{p_{up}}{W_{i-1}} \text{ for } k \in [0,\ W_i - 1] \text{ and } i \in [1,m]$$

$$P\{i,k|i,0\} = \frac{p_{hold}}{W_i} \text{ for } k \in [0,\ W_i - 1] \text{ and } i \in [1,m-1]$$

where p_{up}, p_{down}, and p_{hold} is the probability that its state has to be changed to the state of upper, lower, and current state respectively, according to the filtering result.

Let $\rho = \frac{p_{down}}{p_{up}}$ 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

$$b_{i,0} = p^i b_{0,0} \tag{8}$$

For each $k \in [0, W_i - 1]$, $b_{i,k}$ also has relationship shown in equation (9).

$$b_{i,k} = \frac{W_i - k}{W_i} \left\{ \begin{array}{c} (1-p_{down})b_{0,0} + p_{up} b_{i+1,0} \text{ if } i = 0 \\ p_{down} b_{i-1,0} + (1 - p_{down} - p_{up}) b_{i,0} \\ + p_{up} b_{i+1,0} \text{ if } 0 < i < m \\ p_{down} b_{i-1,0} + (1 - p_{up}) b_{m,0} \text{ if } i = m \end{array} \right\} \tag{9}$$

With equation (8) and $\rho = \frac{p_{down}}{p_{up}}$, equation (9) can be simplified as

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0} \text{ for } 0 \le i \le m \tag{10}$$

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

$$\sum_{i=0}^{m} \sum_{k=0}^{W_i - 1} b_{i,k} = 1 \; implies \; \sum_{i=0}^{m} b_{i,0} \frac{W_i + 1}{2} = 1 \tag{11}$$

In equation (11), $b_{i,0}$ can be computed using equation (8) and W_ii is standardized in 802.11b as follows (for DSSS PHY in 802.11, m ′ = 5):

$$W_i = \begin{pmatrix} 2^i W_0, \text{for } i \le m' \\ 2^{m'} W_0, \text{for } i > m' \end{pmatrix} \tag{12}$$

Replacing Equation (11) with equation (8) and equation (12), we can get the value of $b_{i,0}$ in equation (13) as

$$\frac{1}{b_{0,0}} = \begin{pmatrix} T_1(\rho, m), \text{for } m \le m' \\ T_1(\rho, m') + T_2, \text{for } m > m' \end{pmatrix} \tag{13}$$

$$T_1(\rho, m) = \frac{1 - \rho^{m+1}}{2(1-\rho)} + \frac{W_0(1 - (2\rho)^{m+1})}{2(1-2\rho)} \tag{14}$$

$$T_2(\rho, m') = \frac{(2^{m'} W_0 + 1)}{2} \times \frac{\rho^{m'+1}(1 - \rho^{m-m'})}{(1-\rho)} \tag{15}$$

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

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{1 - \rho^{m+1}}{1-p} b_{0,0} \tag{16}$$

Because we have to consider the various environments such as noisy environments, p_d is expressed as:

$$p_d = 1 - (1 - p_e)(1 - \tau)^{n-1} \tag{17}$$

Equations (16) and (17) represent a nonlinear system in two unknowns τ and p_d which can be solved using numerical techniques.

Let S be the normalized throughput, that is the fraction of channel time used for successful transmission. The throughput S is described as follows [2],[12]:

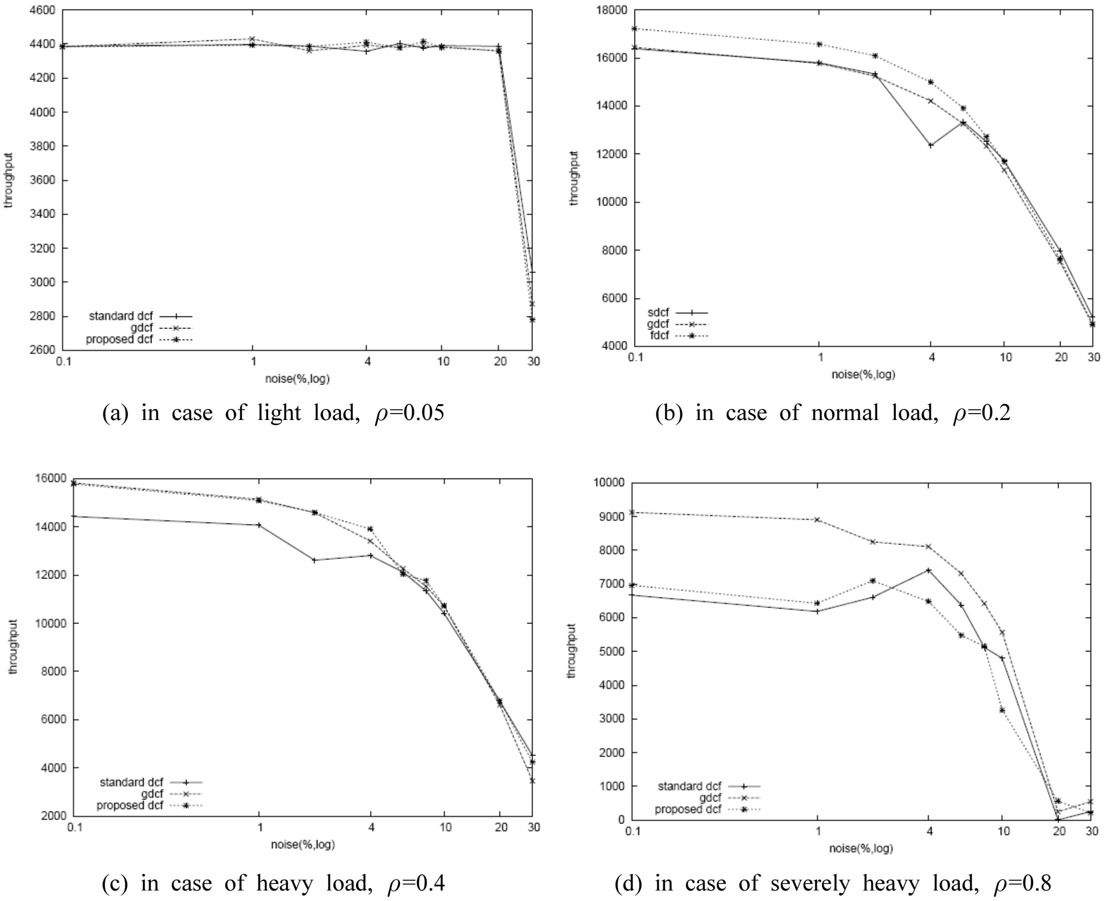
$$S = \frac{P_{tr} P_s E[L]}{(1 - P_{tr)} \sigma + P_{tr} P_s T_s + (1 - P_s) P_{tr} T_c} \tag{18}$$

where T_S and T_C are the average time the channel is sensed busy because of a successful transmission and collision, respectively. $E[L]$ denotes the mean payload size, and σ is the duration of empty slot time. P_{tr} is the probability that at least one transmission occurs in a given slot time, P_s is the conditional probability that exactly one transmission exists at the slot time under the condition of at least one transmission. They are expressed as

$$P_{tr} = 1 - (1 - \tau)^n \tag{19}$$

$$P_s = n\tau(1 - \tau)^{n-1} \tag{20}$$

For the basic access mechanism and the RTS/CTS access mechanism, T_S and T_C were given as



(a) in case of light load, ρ=0.05

(b) in case of normal load, ρ=0.2

(c) in case of heavy load, ρ=0.4

(d) in case of severely heavy load, ρ=0.8

Fig. 4. The simulation results on the total throughput

$$T_s^{bas} = L_H + E[L] + T_{DIFS+}T_{SIFS+}T_{ACK} + 2\delta$$
$$T_c^{bas} = L_H + E[L] + T_{DIFS} + \delta$$
$$T_s^{rts} = T_{RTS} + 3T_{SIFS} + 4\delta + T_{CTS} + L_H + E[L] + T_{ACK}$$
$$T_c^{rts} = T_{RTS} + T_{DIFS} + \delta$$

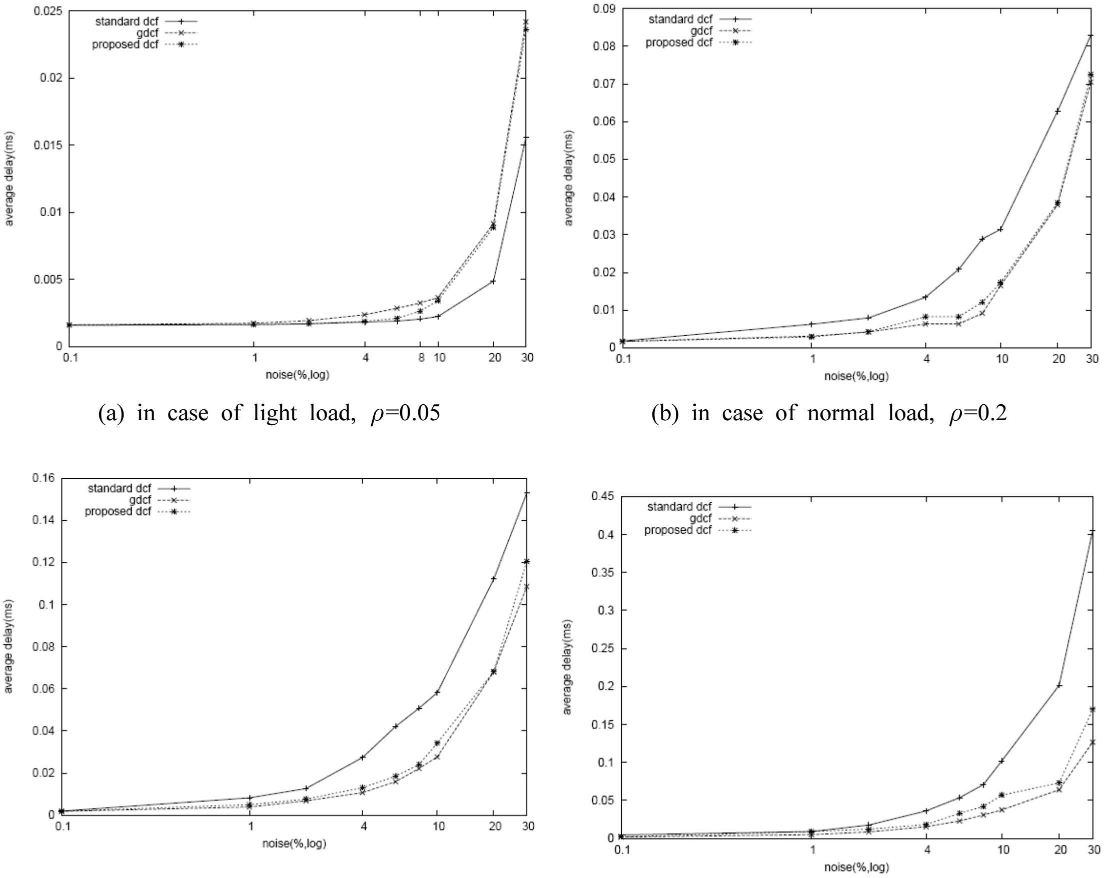
where L_H is the length of packet header, and δ is the propagation delay.

Ⅳ. FDCF Performance Analysis and Simulation

This section will present the simulation results about FDCF using NS2 [6]. We mainly tested in case of various network loads under different error possibilities. The nodes are uniformly distributed along the circle of diameter 300m in the 700m*500m 2-dimensional square space. The main performance metrics of interest are system throughput and delay. All the parameters used in analytical model and our simulations follow the

Table 5. System Parameters(802.11 DSSS)

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



(a) in case of light load, ρ=0.05

(b) in case of normal load, ρ=0.2

(c) in case of heavy load, ρ=0.4

(d) in case of severely heavy load, ρ=0.8

Fig. 5. The simulation results on the average delay

parameters of DSSS, and are summarized in Table 5. Especially history-buffer size c and reference number n were limited as c=4 and n=1 that was turned out to be optimal value in GDCF mechanism. respectively. If proposed FDCF uses more large history-buffer size c, it will show better characteristics. However, in this study, we constrained the history-buffer size c to 4 that optimized to GDCF.

System throughput is used to show the capability of FDCF on the throughput improvement in realistic environments. We repeatedly simulated with changing two factors, network loads and error rates.

Figure 4 shows simulation results on the throughput in noisy environments that have different error rates. We tried to measure the throughput in light, normal, heavy, and severely heavy loaded circumstances. In Figure 4(a), all DCF show similar performances in aspect of throughput. As the almost previous researches implies throughout no concern, good performance is not important issue in the light loaded circumstances. The only noticeable point is the feature that the overall throughput deteriorates rapidly from below the certain error rate.

In Figure 4(b), in the case of most important error rates, the proposed DCF outperforms GDCF and SDCF. In case of very high error rates, all DCF have similar performances. Generally, as we focus on the normal loaded networks with some considerable error rates, we think that it is very important result. In Figure 4(c) related with heavy load networks, proposed DCF and GDCF outperforms SDCF. In Figure 4(d) of severely heavy load networks, GDCF shows better performances than the proposed DCF and SDCF. As the severely heavy loaded networks makes lots of collisions, as we can predict, GDCF which cannot easily reduce the size of contention window showed good performances. However, it cannot be used as an important indicator because packet losses under this condition are more than 90%. This results show the proposed DCF can be effectively applied to the various environments with different error rates and collisions. Especially, in case of normal load networks and heavy load networks, we have confirmed that the proposed DCF could get considerable merits.

Figure 5 shows simulation results on the end-to-end delay measured in realistic environments that have different error rates and network load. We also tried to measure the average delay in light, normal, heavy, and severely heavy loaded circumstances.

In Figure 5(a), all DCF show similar performances in aspect of delay. Even though standard DCF has a little better performance than GDCF and FDCF, the difference is so small(notice that the y-axis scale of each graph is different). In Figure 5(b), in the case of almost every error rates, the proposed DCF and GDCF outperform standard DCF. In Figure 5(c) of heavy load networks, proposed DCF and GDCF have better results than SDCF. In Figure 5(d) of severely heavy load networks, the proposed DCF and GDCF also outperform standard DCF. In aspect of delay, GDCF shows the best delay characteristics. But the proposed DCF has similar results. In other words, In case of light loaded environments, three approaches did not have some noticeable differences. However, if the network load increase more and more, standard DCF has lower performances. Based on the simulation results, we have confirmed that proposed FDCF has the actual merits in the aspects of various environments. This results show FDCF can be effectively applied to the various environments. Especially, in case of normal load and severe error rates, we could show that FDCF have considerable merits.

Ⅴ. Conclusions

We proposed a filter-based algorithm to adaptively adjust the contention window in IEEE 802.11 DCF. The proposed mechanism is focused on the general and realistic environments that have various conditions regarding to noise, media types and network load. We confirmed that FDCF can handle these various environments by adjusting the contention window size according to the result of filtering based on history-buffer.

In addition, FDCF can be extended as a general solution by adjusting the history-buffer size c and reference number n. Lots of variations including the previous mechanism such as standard DCF and GDCF can be easily implemented. Applying weight for each history-buffer information shall give more flexibility and possibility. Finding optimal pattern also will be an important further study theme.

We expect that FDCF can be easily extended to support priority applications or QoS differentiation through configuring reference value n for different type of applications. This is particularly important for some real time multimedia applications.

References

〔1〕 IEEE standard for Wireless LAN Medium Access Control and Physical Layer Specifications, ISO/IEC 8802-11:1999(E), Aug 1999.

〔2〕 G. Bianchi. Analysis of the IEEE 802.11 distributed coordination function. IEEE JSAC, 18(3):535–547, Mar 2000.

〔3〕 G. Bianchi and I. Tinnirello. Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network. In IEEE INFOCOM, Apr 2003.

〔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. Net., 8(6):785–799, Dec 2000.

〔5〕 Periklis Chatzimisios and Anthony C. Boucouvalas. Performance analysis of IEEE 802.11 DCF in presence of transmission errors. In Proceedings of the IEEE International Conference on Communications, Jun 2004.

〔6〕 ISI. NS2-Network Simulator. http://www-mash.cs.berkeley.edu/ns/.

〔7〕 R. Jain. The Art of Computer Systems Performance Analysis. John Wiley and Sons Inc, 1991.

〔8〕 Y. Kwon, Y. Fang, and H. Latchman. A novel MAC protocol with fast collision resolution for wireless LANs. In IEEE INFOCOM, Apr 2003.

〔9〕 S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor. IEEE 802.11e wireless LAN for quality of service. In European Wireless, pages 32–39, Feb 2002.

〔10〕 Tamer Nadeem and Ashok Agrawala. IEEE 802.11 DCF enhancements for noisy environments. In IEEE International Symposium on PRIMC, Sep 2004.

〔11〕 Y. E. Sagduyu and A. Ephremides. Energy-efficient collision resolution in wireless ad-hoc networks. In IEEE INFOCOM, Apr 2003.

〔12〕 C. Wang, B. Li, and L. Li. A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. In IEEE Trans. Vehicular Technology, volume 53, pages 1235–1246, Jul 2004.

〔13〕 H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma. Performance of reliable transfer protocol over IEEE 802.11 wireless LAN: Analysis and enhancement. In Proc. on the Joint Conference of the IEEE Computer and Communication Societies (INFOCOM), pages 599–607, 2002.

〔14〕 P. Yong, H.Wu, S. Cheng, and K. Long. A new self-adapt DCF algorithm., IEEE GLOBECOM, volume 1, pages 87–100, Nov 2002.

유 상 신 (SangShin Yoo) 정회원

1986년 2월 고려대학교 전자공학과 졸업

1988년 2월 고려대학교 전자공학과 석사

1997년 2월 고려대학교 전자공학과 박사

2000년 8월~현재 성공회대학교 소프트웨어공학과 부교수

<관심분야> 임베디드 시스템, 무선통신, 센서네트워크, 신재생에너지