Vol. 46 No. 2 IPSJ Journal Feb. 2005

Regular Paper

DDFC: Decentralized Delay Fluctuation Control Algorithm for IEEE802.11-based Wireless LANs

Hiroyuki Yamada,[†] Hiroyuki Morikawa[††] and Tomonori Aoyama[†††]

Our target is to support both small delay and small delay fluctuation of real-time traffic in IEEE802.11-based wireless LANs by using a decentralized manner. Several previous researches which aimed at supporting real-time traffic in IEEE802.11 wireless LANs developed decentralized control mechanisms achieving small delay of real-time traffic by differentiating real-time traffic from non-real-time traffic, but they cannot achieve small delay fluctuation because of the burst feature of the IEEE802.11 backoff mechanism. We propose a decentralized control mechanism for suppressing delay fluctuation in IEEE802.11-based wireless LANs. Our main proposal is a new backoff algorithm, called decentralized delay fluctuation control (DDFC), which can suppress delay fluctuation in a fully decentralized manner. DDFC can be easily used in IEEE802.11-based wireless LANs by replacing the current backoff algorithm of IEEE802.11 with DDFC. We examine the performance of DDFC, which is assumed to be used for real-time traffic in an IEEE802.11-based wireless LAN, by simulation. The results of computer simulation confirm that we can achieve not only small delay but also small delay fluctuation of real-time traffic in IEEE802.11-based wireless LANs by controlling real-time traffic according to DDFC.

1. Introduction

Real-time applications such as phone, videoconference and multimedia streaming are typical applications which require small delay and small delay fluctuation[☆]. Now that these applications use not only wired networks but also wireless LANs, it is desirable to support real-time traffic even in wireless LANs in order that these applications perform well. Since IEEE802.11[1)], fundamentally based on carrier sense multiple access with collision avoidance (CSMA/CA)[2)], is one of the most familiar standards of wireless LANs, how to support quality of service (QoS) in IEEE802.11-based wireless LANs is an important research topic.

The mechanisms which support real-time traffic in IEEE802.11-based wireless LANs can be categorized into 2 types — centralized control mechanisms and decentralized control mechanisms. Centralized control mechanisms[3)~7)] have a centralized coordinator, which is identical with an access point in wireless LANs, supervise accesses to the medium and provide contention free medium access only for the stations polled by the coordinator. In these mechanisms, the coordinator polls only the stations intending to transmit real-time flows and allows them to access medium free from contention. Thereby, centralized control mechanisms achieve constant small delay of real-time traffic. Here, the word "delay" means the time it takes a data frame from to be enqueued into the interface queue in a sending station until to be received successfully in a receiving station[☆☆]. Hereafter, we will use the word "delay" similarly. On the other hand, decentralized control mechanisms[8)~11)] have every station sharing the medium differentiate real-time traffic from non-real-time traffic distributed-autonomously. In wireless LANs controlled by these mechanisms, real-time traffic is given more transmission opportunities than non-real-time traffic in every station, so that real-time traffic can achieve relatively small delay.

The essential difference between centralized control mechanisms and decentralized control

† School of Engineering, The University of Tokyo
†† Graduate School of Frontier Sciences, The University of Tokyo
††† Graduate School of Information Science and Technology, The University of Tokyo

☆ To some readers, the word "delay jitter", which is a synonym of delay fluctuation, may be more familiar than "delay fluctuation". We use the word "delay fluctuation" instead of "delay jitter" because most of the words "delay fluctuation" in this paper mean fluctuation of delay generated and raised in an individual wireless LAN, not meaning delay jitter experienced by applications.

☆☆ Delay consists of queueing delay, medium access delay, propagation delay, processing delay and so on.

mechanisms is whether a centralized coordinator is required or not. Although centralized control mechanisms can support constant small delay of real-time traffic, they can be utilized only in the case where every station is supported by a centralized coordinator. On the other hand, decentralized control mechanisms do not require any centralized coordinators, so they can be used in any case, including the case where stations are connected with one another ad hoc. There are possible situations where supporting real-time traffic in wireless ad hoc networks is required, including the situation where mobile terminals use real-time applications in wireless ad hoc networks connected to the Internet [12]. Therefore, decentralized control mechanisms are important alternative solutions to support QoS. Hereafter, this paper focuses on decentralized control mechanisms.

Although existing decentralized control mechanisms can realize small delay of real-time traffic by discriminate in favor of real-time traffic, delay fluctuation is yet so large owing to the burst feature of frame transmission ascribable to current IEEE802.11-based wireless LAN's backoff mechanism. Delay fluctuation is one of the factors which deteriorate the performance of real-time applications, so it is desirable to suppress delay fluctuation so far as possible.

We propose a decentralized control mechanism for suppressing delay fluctuation in IEEE802.11-based wireless LANs. Our main proposal is a new backoff algorithm, called decentralized delay fluctuation control (DDFC), which can suppress delay fluctuation in a fully decentralized manner. DDFC can be easily used in IEEE802.11-based wireless LANs by replacing the current backoff algorithm of IEEE802.11 with DDFC.

The structure of the paper is as follows. In Section 2, we give an outline of IEEE802.11. In Section 3, we describe the motivation and goal of our research. In Section 4, we describe decentralized delay fluctuation control (DDFC) algorithm, mentioning existing backoff algorithms. Afterward, we evaluate the performance of DDFC by simulation in Section 5. Finally, in Section 6, we conclude this paper.

2. IEEE802.11

IEEE802.11 standard describes MAC layer and physical layer specifications for IEEE802.11 wireless LANs [1]. The MAC layer specifications of IEEE802.11 define two access control protocols, Distributed Coordination Function (DCF) and Point Coordination Function (PCF).

Besides, IEEE802.11 task group e is standardizing the specifications for IEEE802.11e wireless LANs [13]. The access control scheme specified in IEEE802.11e is called Hybrid Coordination Function (HCF), and can support QoS in a more sophisticated manner than current IEEE802.11. HCF consists of 2 access control protocols, HCF Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA).

In this section, we give an outline of DCF and EDCA, omitting an outline of PCF and HCCA because they are centralized control protocols.

2.1 Distributed Coordination Function

Distributed Coordination Function (DCF) is a decentralized control protocol based on CSMA/CA. DCF must be implemented in all IEEE802.11 stations. In DCF, if a station has a data frame intending to be transmitted, the station decides backoff time of the frame. Algorithm used to decide backoff time is called backoff algorithm. DCF adopts Binary Exponential Backoff (BEB) as its backoff algorithm.

The algorithm of BEB is specified by the following pseudocode:

```
if(RC = 0){
    // first transmission
    CW := CW_min
}
else{
    // retransmission
    CW := 2CW + 1
    CW := min(CW, CW_max)
}
B := rand(0, CW) × SlotTime
```

where RC is retransmission count, which is set to be 0 when a data frame attempts to be transmitted for the first time and incremented by 1 every time a data frame attempts to be retransmitted; CW is contention window; CW_{min} is the minimum value of CW; CW_{max} is the maximum value of CW; $min(a, b)$ is the function returning the smaller number of a and b; $rand(a, b)$ is the function returning an integer chosen randomly from the interval from a to b; B is backoff time; and $SlotTime$ is the duration of a backoff slot.

After backoff time is decided, it is decremented by a station only while the medium is determined to be idle during a term longer

Table 1 Interframe spaces (IFSs) defined in IEEE802.11 standard. IFS is the time interval between frames. Every frame uses an IFS as the time interval during which the frame must wait before transmitted. $ACKTime$ is the duration of an ACK frame. $PreambleLength$ and $PLCPHeaderLength$ are the duration of a physical layer convergence protocol (PLCP) preamble and the duration of a PLCP header, respectively.

IFS	frames using the IFS / duration of the IFS
short interframe space (SIFS)	ACK frames, CTS frames, the second or subsequent frames of a fragment burst, frames responding to a polling frame during CFP, frames transmitted by an access point during CFP
	—
PCF interframe space (PIFS)	frames transmitted by an access point during CFP
	$PIFS = SIFS + SlotTime$
DCF interframe space (DIFS)	data frames during CP
	$DIFS = SIFS + 2SlotTime$
extended interframe space (EIFS)	frames transmitted by a station which has just detected an erroneous frame
	$EIFS = SIFS + 8ACKTime + PreambleLength + PLCPHeaderLength + DIFS$

than a DCF interframe space (DIFS; IFSs are detailed in **Table 1**). When the backoff time becomes 0, the station transmits the frame. After a station transmits a data frame, the station repeats the above procedures for the next data frame with resetting RC to be 0 if it receives the ACK frame; otherwise it repeats the above procedures for the same data frame with incrementing RC by 1.

2.2 HCF Enhanced Distributed Channel Access

HCF Enhanced Distributed Channel Access (EDCA) is a MAC layer protocol which has a scheme providing differentiated services. EDCA was originally developed as an extension of DCF called Enhanced Distributed Coordination Function (EDCF)[11], the recent version of the IEEE802.11e draft[13] redefines EDCF as a part of HCF, named HCF Enhanced Distributed Channel Access (EDCA).

In EDCA, all flows are classified into multiple access categories. In an EDCA station, every access category has its own priority, queue, and protocol parameters such as CW_{min}, CW_{max} and IFS. While IFS is equal to $DIFS$ necessarily in DCF, every access category can use different IFS value, which is equal to or greater than $DIFS$, in EDCA. In EDCA, if two or more access categories of an identical station attempt to transmit a data frame concurrently, the access category having the highest priority may transmit and the others should retransmit. Except this point, every station contends with other stations by using DCF. EDCA can differentiate real-time traffic from non-real-time traffic by classifying real-time traffic and non-real-time traffic into different access categories.

3. Motivation and Goal

Our research is motivated by that existing decentralized control mechanisms cannot suppress delay fluctuation in IEEE802.11-based wireless LANs. It is because these mechanisms adopt Binary Exponential Backoff (BEB) as their backoff algorithms that they cause large delay fluctuation. In BEB, as described in Section 2.1, the size of contention window is doubled when a frame is not transmitted successfully and minimized when a frame is transmitted successfully. In this case, a flow which has transmitted a frame successfully retains small contention window and transmits several frames during a short term, while another flow which has large backoff time transmits no frame. Several researches[14),15)] pointed out this burst feature of BEB. The burst feature of BEB encourages delay fluctuation.

Although existing decentralized control mechanisms cannot achieve small delay fluctuation of real-time traffic because they use BEB as their backoff algorithms, they support relatively small delay of real-time traffic by using their differentiation schemes. If there were a backoff algorithm suppressing delay fluctuation effectively, these mechanisms using the algorithm instead of BEB could achieve both small delay and small delay fluctuation of real-time traffic. Take into account that EDCA is being

Table 2 An example of vanilla EDCA and an example of EDCA with a new backoff algorithm.

(a) An example of vanilla EDCA.

category (priority)	protocol		parameters			service	for which traffic
		backoff algorithm	CW_{min}	CW_{max}	IFS		
1 (high)	EDCA	BEB	15	255	$50\,\mu$s	small delay	real-time
2 (low)		BEB	31	1023	$70\,\mu$s	—	non-real-time

(b) An example of EDCA with a new backoff algorithm.

category (priority)	protocol		parameters			service	for which traffic
		backoff algorithm	CW_{min}	CW_{max}	IFS		
1 (high)	EDCA	**new algorithm**	15	255	$50\,\mu$s	small delay, **small delay fluctuation**	real-time
2 (low)		BEB	31	1023	$70\,\mu$s	—	non-real-time

standardized as a part of IEEE802.11e unlike other decentralized control mechanisms, we believe that it is essential to develop a backoff algorithm which can suppress delay fluctuation effectively and can be used in the differentiation scheme of EDCA.

The goal of our research is to realize EDCA-based wireless LANs achieving both small delay and small delay fluctuation of real-time traffic. **Table 2** (a) shows an example of vanilla EDCA. In Table 2 (a), 2 access categories are used, and real-time traffic and non-real-time traffic are classified into category 1 and category 2 respectively. Both categories adopt BEB. Category 1 uses smaller CW_{min}, CW_{max} and IFS than category 2 for the purpose of differentiation. In this case, real-time traffic achieves small delay, but cannot achieve small delay fluctuation.

An ideal wireless LAN we aim to realize is like as described in Table 2 (b). This LAN uses 2 access categories and differentiates them by using the differentiation scheme of EDCA as well as the vanilla EDCA shown in Table 2 (a). Unlike vanilla EDCA, this LAN adopts another algorithm as the backoff algorithm of category 1 for the purpose of suppressing delay fluctuation. Thereby, this LAN achieves both small delay and small delay fluctuation of real-time traffic. To realize such an EDCA-based LAN, a backoff algorithm which can suppress delay fluctuation effectively is required.

4. Backoff Algorithm for Suppressing Delay Fluctuation

4.1 Existing Backoff Algorithms

Several researches [15)~18)] proposed MAC layer mechanisms including their original backoff algorithms. While almost all of them targeted on throughput fairness, e.g., frame length-aware fairness, weighted fairness, MACAW research [15)] proposed a backoff algorithm which mitigates the burst feature of transmission, named Multiplicative Increase Linear Decrease (MILD).

The algorithm of MILD is specified by the following pseudocode:

```
if(RC = 0){
    // first transmission
    CW := CW − 1
    CW := max(CW, CW_min)
}
else{
    // retransmission
    CW := 1.5CW
    CW := min(CW, CW_max)
}
B := rand(0, CW) × SlotTime
```

where $max(a, b)$ is the function returning the larger number of a and b. The pseudocode presents that the size of contention window is multiplied by 1.5 when a frame is not transmitted successfully and decremented by 1 when a frame is transmitted successfully. MILD mitigates the burst feature of transmission by preventing the contention window of a flow having transmitted a frame successfully from being minimized and preventing the contention window of a flow not having transmitted a frame successfully from exploding. As a result, MILD suppresses delay fluctuation to some extent.

However, MILD is unsuitable as a solution to suppress delay fluctuation in IEEE802.11-based wireless LANs because of the following reasons:

- MILD cannot achieve higher performance than BEB in almost all cases: MILD increases average delay in return for the fluctuation of contention window's size. This is due to MILD's feature of keeping contention window large. The feature encourages large backoff time, and consequently average delay tends to be large.

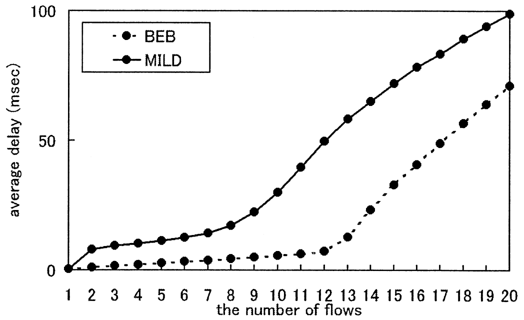
Figure 1 shows a typical example of average delay and standard deviation of delay in the case where an IEEE802.11-based wireless LAN adopts BEB or MILD as backoff algorithm. In Fig. 1 (a), we can find that the average delay in MILD is larger than that in BEB.

Additionally, the standard deviation of delay in MILD is larger than that in BEB when the number of flows is smaller than 18 as shown in Fig. 1 (b). This illustrates MILD is unsuitable for suppressing delay fluctuation except in the case where traffic is heavy.

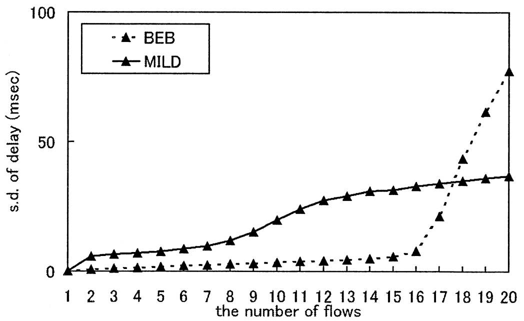
- MILD is unsuitable for controlling real-time traffic in the differentiation framework of EDCA: When MILD is used for the purpose of suppressing delay fluctuation of real-time traffic in the environments where EDCA categorizes traffic into real-time and non-real-time, non-real-time traffic controlled by BEB achieves higher performance than real-time traffic controlled by MILD; in other words, the differentiation of EDCA does not work well. This is because BEB, frequently resetting CW, allows more frames to be transmitted than MILD.

  Figure 2 shows the average delay of category 1 traffic and category 2 traffic in an EDCA-based wireless LAN configured as shown in **Table 3** and Table 4. In Table 3, p is the backoff algorithm used by category 1 and n is the number of category 1 flows. If p = BEB, the average delay of category 1 traffic remains small even when n grows larger; the differentiation of EDCA works well. However, if p = MILD, the average delay of category 1 traffic is larger than that of category 2 traffic; the differentiation of EDCA does not work well.

In the next subsection, we present a backoff algorithm suppressing delay fluctuation of real-time traffic in EDCA-based wireless LANs.



(a) Average delay.



(b) Standard deviation of delay.

Fig. 1 The delay characteristics of BEB and MILD. The parameter settings of the LAN are shown in Table 4, and CW_{min} and CW_{max} are 31 and 1023 respectively. The bit rate of a flow is 192 kbps, the packet size is 512 B, and the queue length is 4 frames.

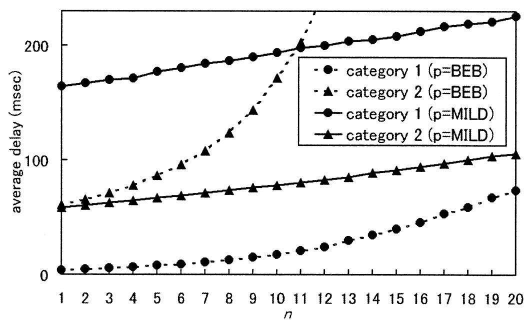


Fig. 2 The average delay characteristics in the differentiation framework of EDCA. The parameter settings of the LAN are shown in Table 4, and EDCA settings are shown in Table 3. p is the backoff algorithm used by category 1 and n is the number of category 1 flows.

Table 3 EDCA settings. p is BEB or MILD.

category (priority)	protocol		parameters			bit rate of a flow	frame size	queue length	flows
		backoff algorithm	CW_{min}	CW_{max}	IFS				
1 (high)	EDCA	p	15	255	50 μs	192 kbps	512 B	4 frames	n flows
2 (low)		BEB	31	1023	70 μs	—	1,500 B	8 frames	4 flows

4.2 Decentralized Delay Fluctuation Control Algorithm

We design a new backoff algorithm to perform as well as BEB when traffic is light and to prompt more longly waiting frames to be transmitted sooner only when traffic is heavy. This is because BEB achieves high performance constitutionally when traffic is light. When traffic is light, retransmissions rarely come about. In this condition, BEB frequently resets CW to be CW_{min} and allows almost all frames to be transmitted with short backoff time so that BEB can achieve high performance when traffic is light. Therefore, small delay and small delay fluctuation are achieved without any complicated techniques. As described in Section 4.1, MILD, which was devised to mitigate the fluctuation of contention window, degrades the wireless LAN performance ironically when traffic is light. On balance, we conclude that BEB's performance in the case where traffic is light is adequate.

With the above design policy in mind, we specify a new backoff algorithm, called Decentralized Delay Fluctuation Control (DDFC), as the following pseudocode:

if($RC = 0$){
  // first transmission
  $CW := CW_{min}$
}
else{
  // retransmission
  if($t > t_s$){
    $$CW := \frac{(CW_{min}+1)2^{RC}t_0}{t-(t_s-t_0)}$$
  }
  else{
    $CW := (CW_{min}+1)2^{RC} - 1$
  }
  $CW := min(CW, CW_{max})$
}
$B := rand(0, CW) \times SlotTime$

where t is waiting time of a frame, the time elapsing since the frame was enqueued into the interface queue; and t_s and t_0 are protocol parameters.

A time t_s is a threshold which DDFC uses to guess whether traffic is light or heavy. DDFC is expected to perform as well as BEB when traffic is light and to prompt more longly waiting frames to be transmitted sooner when traffic is heavy. So, we need to give DDFC a way to easily guess whether traffic is light or heavy. We design DDFC to determine that traffic is light if waiting time of a frame, t, is not over t_s. Taking into account that almost all frames ought to be transmitted not experiencing retransmissions or long backoffs in the case of light traffic, we regard waiting time as a useful barometer to determine traffic density. In the case where $t \le t_s$, DDFC sets CW to be $(CW_{min}+1)2^{RC} - 1$, not caring waiting time t. This CW is the same to the CW decided by BEB on condition that DDFC and BEB use the same CW_{min} and RC.

On the other hand, DDFC determines that traffic is heavy if t is over t_s. In this case, DDFC decides CW caring not only RC but also waiting time t. **Figure 3** shows the size of contention window controlled by DDFC, comparing it with the size of contention window controlled by BEB. As shown in Fig. 3, when $t > t_s$, DDFC sets CW to be smaller than BEB and monotonically decreased with increasing t. The expression $CW := \frac{(CW_{min}+1)2^{RC}t_0}{t-(t_s-t_0)}$ is derived from the 3 following straightforward conditions: 1) CW draws the curve inversely proportional to t to suppress delay fluctuation easily and effectively when $t > t_s$; 2) CW converges to 0 when $t \to \infty$; and 3) CW is continuous at the point where $t = t_s$ ☆. The protocol parameter t_0 can adjust the influence of t upon CW. t_0 is the constant of proportionality of the adjusting factor $\frac{t_0}{t-(t_s-t_0)}$, so using smaller t_0 means more wildly decreasing CW with increasing t as shown in Fig. 3 (b). The adjuster t_0 can be used to configure the persistence in suppressing delay fluctuation.

DDFC, however, exceptionally sets CW not caring t as well as BEB only in the case where $RC = 0$, even though t is over t_s. This is because the condition that t is large and $RC = 0$ may be caused by malicious traffic regardless of polite traffic density. In fact, if a malicious station is equipped with a huge interface queue and fills the queue with frames while other stations offer to transmit few frames, waiting time

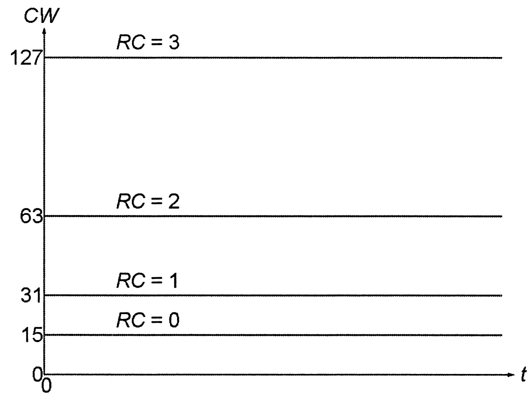
☆ Assume $CW := (CW_{min}+1)2^{RC} \times (\frac{a}{t-x_0} + y_0)$, where we use an inversely proportional curve $y = \frac{a}{x-x_0} + y_0$. We can derive the expression $CW := \frac{(CW_{min}+1)2^{RC}a}{t-(t_s-a)}$ using the conditions: $(t, CW) \to (\infty, 0)$ and $(t, CW) = (t_s, (CW_{min}+1)2^{RC})$. We refer the constant of proportionality a as t_0.

of a frame in the queue is so large when the frame attempts to be transmitted for the first time. If DDFC were configured to control CW caring t in the case where $RC = 0$, frames of a malicious station would be prompted to be transmitted earlier and polite stations would suffer unfair share of transmission opportunities. DDFC avoids such an undesirable situation. Consequently, DDFC prompts frames to be transmitted earlier only when $RC > 0$ and $t > t_s$.

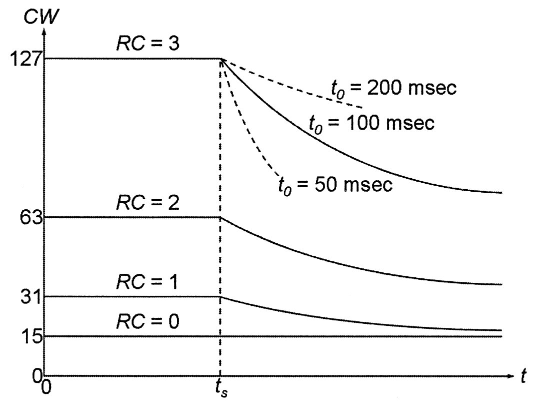
4.3 Features of DDFC

DDFC has the following features:

- DDFC is a fully decentralized algorithm: DDFC controls backoff time in a fully decentralized manner. In fact, IEEE802.11-based wireless LANs using DDFC instead of BEB work as decentralized control mechanisms, not requiring any centralized coordinators.
- DDFC achieves as high performance as BEB when traffic is light: When wireless LAN traffic is light, almost all frames do not need retransmissions and are transmitted not waiting so long. Therefore, DDFC performs as well as BEB, so that DDFC achieves as high performance as BEB.
- DDFC can be used to control real-time traffic in the differentiation scheme of EDCA: One of our targets is to adapt a new backoff algorithm suppressing delay fluctuation to EDCA-based wireless LANs. To attain the target, the differentiation scheme of EDCA is required to work well when access categories for real-time traffic use DDFC instead of BEB. In fact, as shown in Fig. 3, contention window controlled by DDFC is smaller than or equal to that controlled by BEB in the case of the same CW_{min}, CW_{max} and RC. So, whenever CW_{min}, CW_{max} and IFS of DDFC are smaller than or equal to those of BEB respectively, DDFC necessarily gives frames more transmission opportunities than BEB. Therefore, so long as the priority of categories using DDFC is higher than that of categories using BEB, the differentiation scheme of EDCA works not causing such corruption of differentiation as is found in MILD.



(a) Contention window controlled by BEB.



(b) Contention window controlled by DDFC. t_s and t_0 are DDFC's protocol parameters.

Fig. 3 Contention window controlled by BEB and DDFC, where $CW_{min} = 15$ and t is waiting time of a frame.

5. Performance

In this section, we examine the performance of DDFC by simulation. **Figure 4** illustrates the outline of our simulation. We use the ns-2 Network Simulator[19] specially equipped with EDCA-based traffic categorization function and DDFC backoff algorithm. We use the settings

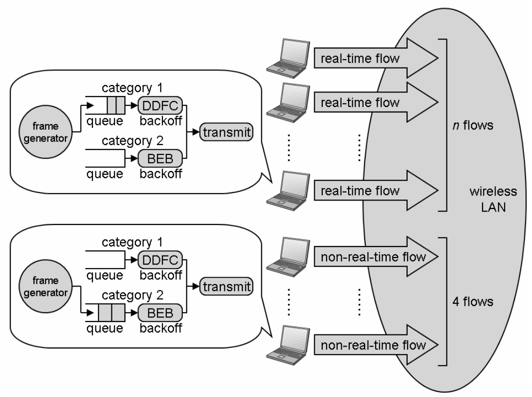


Fig. 4 Outline of the simulation.

Table 5 EDCA settings. p is BEB or DDFC. t_s and t_0 are DDFC's protocol parameters, used only if p = DDFC.

category (priority)	protocol		parameters					bit rate of a flow	frame size	queue length	flows
		backoff algorithm	CW_{min}	CW_{max}	IFS	t_s	t_0				
1 (high)	EDCA	p	15	255	50 μs	t_s	t_0	192 kbps	512 B	4 frames	n flows
2 (low)		BEB	31	1023	70 μs	—	—	—	1,500 B	8 frames	4 flows

Table 4 Parameter settings. We use the settings for all the simulations in this paper. *DATATime* and *ACKTime* are the duration of a data frame and the duration of an ACK frame, respectively.

SlotTime	20 μs
SIFS	10 μs
DIFS	50 μs
basic rate	1 Mbps
data rate	11 Mbps
PreambleLength	144 μs (144 bits)
PLCPHeaderLength	48 μs (48 bits)
RetryLimit	7
ACKTimeout	*DATATime* + *ACKTime* + 14 μs

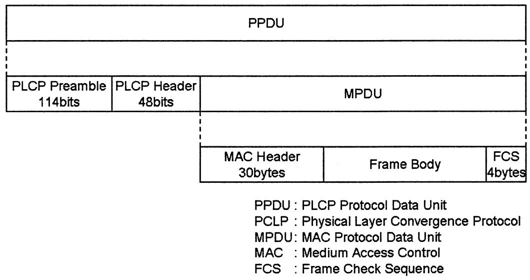


Fig. 5 Frame format.

shown in **Table 4** and **Table 5** for the simulation in this section. The parameter settings shown in Table 4 are based on the direct sequence spread spectrum (DSSS) physical specification of IEEE802.11, and similar to the settings used in similar researches [3),4)] or the ns-2 default settings. The data rate setting of 11 Mbps is based on the complementary code keying (CCK) physical specification of IEEE802.11b [20)], which is an extension of the IEEE802.11 DSSS specification. Frame format is shown in **Fig. 5**. In our simulation environments, frames intending to be inserted into an interface queue are dropped when the queue overflows. Category 1 flows are constant bit rate (CBR) flows, which we assume to be real-time flows. On the other hand, we assume category 2 flows to be non-real-time flows, and the bit rate of category 2 flows is as high as possible on condition that they obey IEEE802.11 protocol. Additionally, we assume that every station is not a hidden terminal to any other station [☆]. Delay measured in the simulation means how long it took a frame from to be enqueued into the interface queue in a sending station until to be received successfully in a receiving station. All the results in this section are produced by the computer simulation configured as shown in Table 4 and Table 5.

5.1 Basic Characteristics

In this subsection, we present the basic characteristics of DDFC, which are obtained by the simulation in the settings shown in Table 4 and Table 5. **Figure 6** shows a typical example of average delay, standard deviation of delay and throughput in an EDCA-based wireless LAN. p is the backoff algorithm used by category 1 and n is the number of category 1 flows.

5.1.1 Delay Characteristics

Figure 6 (a) shows that the average delay characteristics of category 1 traffic in DDFC are almost as same as those in BEB even when n grows larger; thus, the differentiation of EDCA using DDFC for category 1 traffic works as well as vanilla EDCA. On the other hand, Fig. 6 (b) shows the standard deviation of category 1 traffic in DDFC is much smaller than that in BEB. This illustrates DDFC's ability to suppress delay fluctuation.

5.1.2 Throughput Characteristics

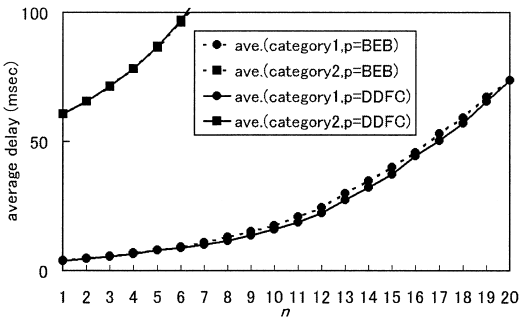
Figure 6 (c) shows that the throughput characteristics when p = DDFC and those when p = BEB. The throughput plotted in Fig. 6 (c) means wireless LAN throughput, namely the sum of the throughputs of all the flows in a wireless LAN. Hereafter, we will refer to wireless LAN throughput as simply throughput.

Figure 6 (c) shows that the throughput characteristics when p = DDFC are almost as same as those when p = BEB. Whether p is BEB or

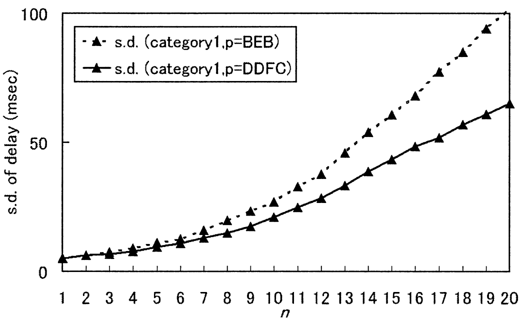
☆ Our research does not address the hidden terminal problem for efficiency and intelligibility of performance evaluation, as well as several similar researches [3),4),8)∼10),16)]. In fact, not only networks using DDFC but also all CSMA/CA networks suffer from the hidden terminal problem, so the problem does not essentially affect the comparison between DDFC and BEB.

DDFC, the larger n is, the smaller throughput is. This is because of the feature of CSMA/CA networks which increases collisions with increasing the number of flows. The feature is irrelevant to whether increasing flows are real-time or non-real-time although we show only the graph presenting the relation of throughput to the number of real-time flows.

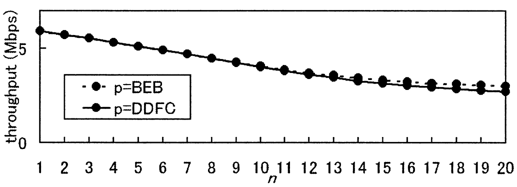
On the other hand, we find in Fig. 6 (c) when n is large, namely when traffic is heavy, the throughput characteristics when $p =$ DDFC are a little bit worse than those when $p =$ BEB. This is because the feature of DDFC making backoff time smaller causes more collisions. As our solution is using DDFC only for real-time traffic, the throughput degradation ascribable



(a) Average delay.



(b) Standard deviation of delay.



(c) Wireless LAN throughput.

Fig. 6 The characteristics of EDCA. The parameter settings of the LAN are shown in Table 4, and EDCA settings are shown in Table 5. p is the backoff algorithm used by category 1 and n is the number of category 1 flows. $t_s = 20$ ms and $t_0 = 100$ ms.

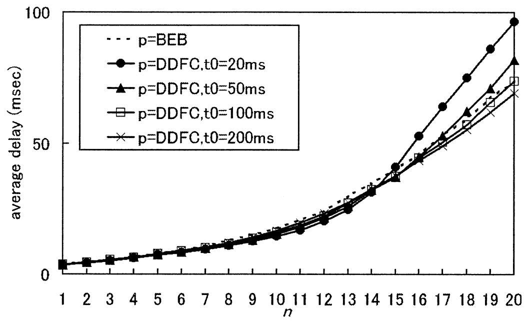
this feature is caused only by real-time traffic. This throughput degradation is a compensation for suppression of delay fluctuation, but the compensation is hardly matter because too heavy real-time traffic hardly happens in practical wireless LANs. In Fig. 6 (c), the LAN experiences explicit throughput degradation only when $n > 12$. Real-time traffic occupies more than half throughput when $n = 12$, and non-real-time traffic is hardly transmitted when $n > 12$. Almost all traffic is non-real-time in practical wireless LANs, so such a case as heavy real-time traffic hardly happens practically. Additionally, heavy real-time traffic can be avoided absolutely by a distributed admission control mechanism [8].

5.2 Impacts of Parameters

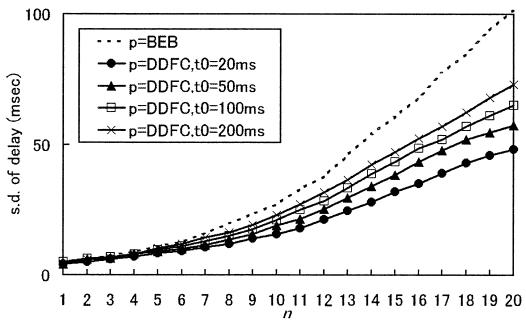
In this subsection, we present the impacts of DDFC's parameters, t_s and t_0.

5.2.1 Impacts of t_0

Figure 7 shows the impacts of t_0 on the delay characteristics of DDFC. In Fig. 7 (a), we can find that the characteristics when traffic is



(a) Average delay.



(b) Standard deviation of delay.

Fig. 7 The delay characteristics of category 1 traffic. The parameter settings of the LAN are shown in Table 4, and EDCA settings are shown in Table 5. p is the backoff algorithm used by category 1 and n is the number of category 1 flows. $t_s = 20$ ms.

heavy and those when traffic is light are a little bit different.

When n is small, namely when traffic is light, the average delay in DDFC is a little bit smaller than that in BEB. This is because of DDFC's feature to prompt frames waiting longer to be transmitted earlier. Additionally, we can see in Fig. 7 (a), the smaller t_0 is, the smaller the average delay is, when traffic is light.

On the other hand, when n is large, namely when traffic is heavy, we can see in Fig. 7 (a), the smaller t_0 is, the larger the average delay is. Especially, if $t_0 \leq 50$ ms, the average delay in DDFC is larger than that in BEB when n is large. Smaller t_0 encourages frames waiting longer to be transmitted earlier, but when traffic is heavy, too small t_0 causes frequent collisions, and the performance is deteriorated consequently.

In Fig. 7 (b), we can see the smaller t_0 is, the smaller the standard deviation of delay is. This is because DDFC with smaller t_0 assigns delayed frames to smaller CW, and prompts the frames to be transmitted earlier. However, using too small t_0 is undesirable because it increases average delay in the case where traffic is heavy as described above. Totally, when $100\,\text{ms} \leq t_0 \leq 200\,\text{ms}$, DDFC achieves desirable performance in our simulation environments.

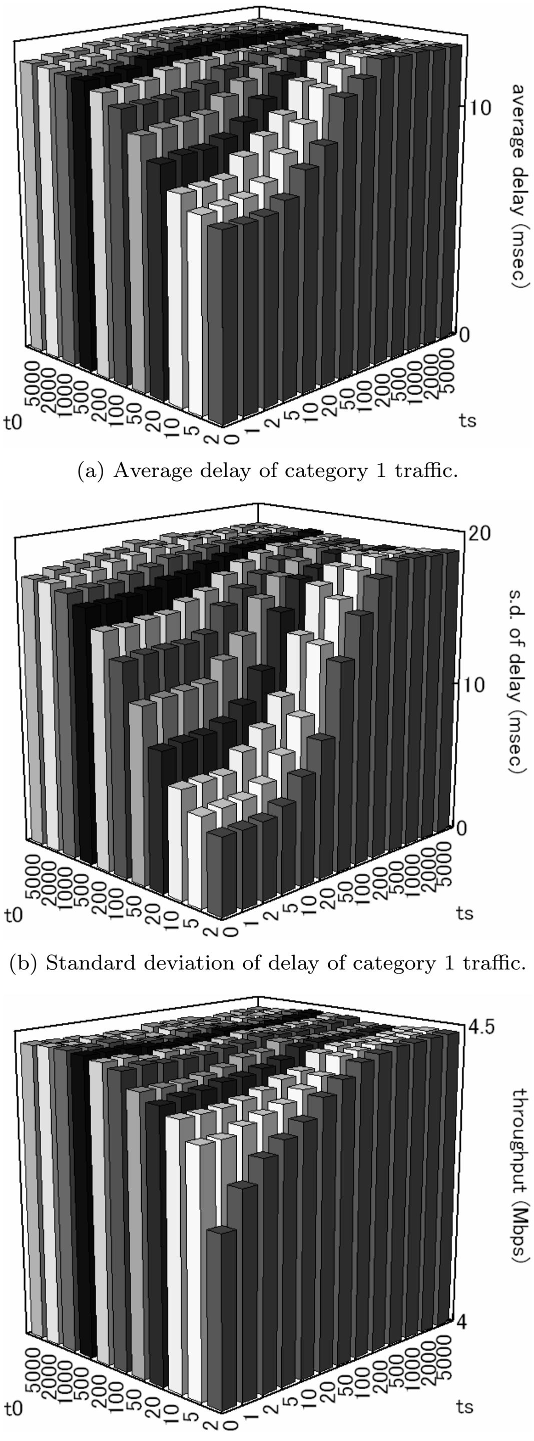
5.2.2 Impacts of t_s

Figure 8 shows the impacts of t_s and t_0 on the performance of DDFC. In the case where t_0 is large, t_s has little influence on the performance. Except in the case, DDFC with smaller t_s achieves smaller average delay and smaller standard deviation of delay. However, too small t_s is not desirable because throughput is decreased as shown in Fig. 8 (c). In fact, Fig. 8 (a) and Fig. 8 (b) show delay and delay fluctuation of category 1 traffic is hardly decreased if t_s is set to be value which is smaller than 10 ms, so it is not beneficial to set t_s to be too small. Totally, when $20\,\text{ms} \leq t_s \leq 100\,\text{ms}$, DDFC works well in our simulation environments.

6. Conclusion

To realize IEEE802.11-based wireless LANs which achieve both small delay and small delay fluctuation of real-time traffic in a decentralized manner, we proposed decentralized delay fluctuation control algorithm, called DDFC, which can be used in EDCA-based wireless LANs.

We examined the performance of DDFC by simulation, configuring the environments where



(a) Average delay of category 1 traffic.

(b) Standard deviation of delay of category 1 traffic.

(c) Wireless LAN throughput.

Fig. 8 Impacts of t_s and t_0. The parameter settings of the LAN are shown in Table 4, and EDCA settings are shown in Table 5. $n = 8$.

DDFC is used for real-time traffic in an EDCA-based wireless LAN. The results of computer simulation confirmed that we can achieve not only small delay but also small delay fluctuation in EDCA-based wireless LANs by controlling real-time traffic according to DDFC. As a part of future work, we intend to investigate how improved the performance of real-time applications is owing to DDFC.

References

1) IEEE: *IEEE std 802.11, 1999 Edition, Information Technology — Telecommunications and Information Exchange between Systems — Local and Metropolitan Area Networks — Specific Requirements — Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (1999).
2) Kleinrock, L. and Tobagi, F.A.: Packet Switching in Radio Channels: Part I — Carrier Sense Multiple Access Modes and Their Throughput-Delay Characteristics, *IEEE Trans. Comm.*, Vol.COM-23, No.12, pp.1400–1416 (1975).
3) Crow, B.P., Widjaja, I., Kim, J.G. and Sakai, P.T.: Investigation of the IEEE 802.11 Medium Access Control (MAC) Sublayer Functions, *Proc. IEEE INFOCOM'97*, Kobe, Japan, pp.126–133 (1997).
4) Crow, B.P., Widjaja, I., Kim, J.G. and Sakai, P.T.: IEEE 802.11 Wireless Local Area Networks, *IEEE Communications Magazine*, Vol.35, No.9, pp.116–126 (1997).
5) Coutras, C., Gupta, S. and Shroff, N.B.: Scheduling of real-time traffic in IEEE 802.11 wireless LANs, *ACM Wireless Networks*, Vol.6, No.6, pp.457–466 (2000).
6) Veeraraghavan, M., Cocker, N. and Moors, T.: Support of voice services in IEEE 802.11 wireless LANs, *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, USA, pp.488–497 (2001).
7) Saitoh, K., Inoue, Y., Iizuka, M. and Morikura, M.: Communications Quality Control Schemes by Using Priority Control Mechanism for Ethernet Based Wireless LANs, *The Transactions of the Institute of Electronics, Information and Communication Engineers B*, Vol.J84-B, No.9, pp.1598–1612 (2001).
8) Barry, M., Campbell, A.T. and Veres, A.: Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks, *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, USA (2001).
9) Veres, A., Campbell, A.T., Barry, M. and Sun, L.-H.: Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control, *IEEE Journal of Selected Areas in Communications*, Vol.19, No.10, pp.2081–2093 (2001).
10) Kuppa, S. and Prakash, R.: Service differentiation mechanisms for IEEE 802.11-based wireless networks, *Proc. IEEE WCNC 2004*, Atlanta, Georgia, USA, IEEE (2004).
11) Chesson, G., Diepstraten, W., Hoeben, M., Singla, A., Teunissen, H. and Wentink, M.: IEEE P802.11 Wireless LANs EDCF Proposed Draft Text (2001). IEEE 802.11-01/131r1.
12) Grand, G.L., Meraihi, R., Tohme, S. and Riguidel, M.: Intelligent ambient ad hoc networking to support real-time services, *Proc. 58th IEEE Vehicular Technology Conference (VTC)*, The Hilton in the Walt Disney World Resort, Orlando, Florida, USA, IEEE, IEEE (2003).
13) IEEE: *LAN/MAN Specific Requirements — Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Wireless Medium Access Control (MAC) Quality of Service (QoS)* (2004). IEEE P802.11e/D8.0.
14) Takeshima, Y., Fukuda, K., Takayasu, H. and Takayasu, M.: An Analysis on the Critical Phenomena in CSMA/CD Network Traffic Model by Computer Simulations, *Transactions of the Institute of Electronics, Information and Communication Engineers B*, Vol.J84-B, No.5, pp.840–848 (2001).
15) Bharghavan, V., Demers, A., Shenker, S. and Zhang, L.: MACAW: A Media Access Protocol for Wireless LAN's, *Proc. ACM SIGCOMM'94*, London, UK, pp.212–225 (1994).
16) Vaidya, N.H., Bahl, P. and Gupta, S.: Distributed Fair Scheduling in a Wireless LAN, *Proc. ACM Mobicom 2000*, Boston, MA, USA, pp.167–178 (2000).
17) Nandagopal, T., Kim, T.-E., Gao, X. and Bharghavan, V.: Achieving MAC Layer Fairness in Wireless Packet Networks, *Proc. ACM Mobicom 2000*, Boston, MA, USA, pp.87–98 (2000).
18) Wang, Y. and Bensaou, B.: Achieving Fairness in IEEE802.11 DFWMAC with Variable Packet Lengths, *Proc. IEEE GLOBECOM 2001*, San Antonio, Texas, USA (2001).
19) Information Sciences Institute: ns-2 Network Simulator. http://www.isi.edu/nsnam/ns/.
20) IEEE: *IEEE Std 802.11b-1999, Supplement to IEEE Standards for Information Technology — Telecommunications and Information Exchange between Systems — Local and Metropolitan Area Networks — Specific Requirements — Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer*

(PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band (1999).

(Received May 19, 2004)
(Accepted November 1, 2004)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.1, pp.63–74.)

Hiroyuki Yamada was born in 1977. He received his B.E. degree from Kyoto University, Japan in 2000, and M.E. degree from The University of Tokyo, Japan in 2002. He is now a Ph.D. student of Engineering at The University of Tokyo. His current research interests are mobile/wireless communication technologies. He is a member of IEEE and IEICE.

Hiroyuki Morikawa was born in 1965. He received his B.E., M.E. and Dr.Eng. degrees in electrical engineering from The University of Tokyo, Japan, in 1987, 1989 and 1992, respectively. He is currently an associate professor of the Department of Frontier Informatics at The University of Tokyo. From 1997 to 1998, he stayed in Columbia University as a visiting research associate. His research interests are in the areas of ubiquitous network, wireless networks, mobile computing, distributed computing and network services. He is a member of IEEE, ACM, ISOC, IEICE, IPSJ and ITE.

Tomonori Aoyama was born in 1943. He received his B.E., M.E. and Dr.Eng. degrees from The University of Tokyo, Japan, in 1967, 1969 and 1991, respectively. Since he joined NTT Public Corporation in 1969, he has been engaged in research and development on communication networks and systems in the Electrical Communication Laboratories. From 1973 to 1974, he stayed in MIT as a visiting scientist to study digital signal processing technologies. In 1994, he was appointed to the director of NTT Opto-Electronics Laboratories, and in 1995 he became the director of NTT Optical Network Systems Laboratories. In 1997, he left NTT and joined The University of Tokyo. He is currently a professor in the Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo. His research activities cover the next generation networking technologies from layer 1 (e.g., photonic networks) to higher layers including middleware for network collaboration, P2P routing, mobile networking and ubiquitous networking. He is involved in several governmental projects such as Japan Gigabit Network (JGN) and the Ubiquitous Networking Forum, and in some non-profit organizations and consortia such as the Photonic Internet Forum (PIF) and Digital Cinema Consortium of Japan (DCCJ), in which he is serving as Chief Director. He is an IEEE Fellow and was a Members-at-Large of the IEEE ComSoc Board of Governors. He served as the president of IEICE Communications Society. He also served as a chair of IEEE ComSoc Japan Chapter. He is a member of IEEE, IEICE and IPSJ.