IEEE802.11-based Wireless LAN Achieving Small Delay Fluctuation

Hiroyuki YAMADA†, Hiroyuki MORIKAWA†, and Tomonori AOYAMA††

† Graduate School of Engineering, The University of Tokyo Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8656 Japan
†† Graduate School of Information Science and Technology, The University of Tokyo Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8656 Japan
E-mail: †{yamada,mori,aoyama}@mlab.t.u-tokyo.ac.jp

Abstract Real-time applications such as phone, videoconference, multimedia streaming, are typical applications which require relatively small delay and small delay fluctuation. To meet the QoS requirements of these applications, network infrastructures should have the ability to accommodate real-time traffic. Recently, IEEE802.11 wireless LANs, fundamentally based on CSMA/CA, have been widely used as parts of network infrastructures, so QoS support in wireless LANs becomes important. Several existing PCF-based centralized control mechanisms can support constant small delay of real-time traffic in a wireless LAN. However, these mechanisms require a centralized controlled coordinator, and cannot be utilized in any case. So, a decentralized control mechanism supporting real-time traffic is an important alternative solution to support real-time traffic. EDCF is a typical example of decentralized control mechanisms which provide relatively small delay for real-time traffic, but cannot achieve small delay fluctuation because of the burst feature of its backoff mechanism. We propose a decentralized control mechanism suppressing delay fluctuation in CSMA/CA networks, called DDFC (Decentralized Delay Fluctuation Control), and examine the performance of DDFC by simulation. The results of simulation confirm that we can achieve not only small delay but also small delay fluctuation in EDCF-based wireless LANs by controlling real-time traffic according to DDFC.
Key words wireless LAN, IEEE802.11, CSMA/CA, MAC, decentralized, QoS, delay fluctuation, real-time

IEEE802.11型無線LANにおける遅延揺らぎの抑制

山田 浩之† 森川 博之† 青山 友紀††

† 東京大学 大学院 工学系研究科 〒113-8656 東京都 文京区 本郷 7-3-1
†† 東京大学 大学院 情報理工学系研究科 〒113-8656 東京都 文京区 本郷 7-3-1
E-mail: †{yamada,mori,aoyama}@mlab.t.u-tokyo.ac.jp

あらまし 音声や動画像等リアルタイム性の高いアプリケーションでは、その品質を維持するためには遅延及びその揺らぎを小さく抑える必要がある。IEEE802.11 無線 LAN では低遅延で安定したフレーム転送のために PCF が規定されており、この PCF を用いてリアルタイムトラフィックをサポートしようとする研究が多数ある。しかし、これらの研究で提案されている方法はいずれも集中制御方式であり、集中制御端末を必要とする。そのため、集中制御端末を必要とせずあらゆる状況で適用出来る分散制御方式によりリアルタイムトラフィックをサポートする手法が検討されている。その中で代表的なものが DCF を拡張した EDCF であるが、これは差別化による優先制御でリアルタイムトラフィックの遅延を低く抑えられるものの、そのバックオフ方式が持つフレーム転送のバースト性が遅延の揺らぎを引き起こしてしまう。本稿では、遅延の揺らぎを抑制する分散制御手法である DDFC (Decentralized Delay Fluctuation Control) を提案し、シミュレーションにより評価する。シミュレーション結果からは、EDCF により優先的に転送されるリアルタイムトラフィックを DDFC で制御することで、平均遅延だけでなく遅延の揺らぎも低く抑えられることが確認出来た。
キーワード 無線 LAN, IEEE802.11, CSMA/CA, MAC, 分散, QoS, 遅延の揺らぎ, リアルタイム

1. Introduction

With the Internet spreading widely and the users of the Internet increasing explosively, various Internet applications have been developed and employed for the last decade. These applications have their own characteristics and QoS (Quality of Service) requirements. These applications can be categorized into 2 types. File transfer, electronic mail and World Wide Web are the typical applications which provide reliable data transfer, and require lossless transmission. On the other hand, phone, videoconference and multimedia streaming are the typical applications which provide sound and/or moving picture, and require relatively small delay and small delay fluctuation instead of lossless transmission. In this paper, we call the former non-real-time applications and the latter real-time applications, respectively.

To realize sophisticated Internet services and achieve their high performance, network infrastructures and end terminals must endeavor to meet the QoS requirements of these applications. While the requirement of non-real-time applications—lossless transmission—can be met by an end-to-end approach such as TCP (Transmission Control Protocol) [1], the requirements of real-time applications—relatively small delay and small delay fluctuation—cannot be met necessarily unless the network infrastructures transfer real-time traffic prior to non-real-time traffic. So, network infrastructures should have the ability to accommodate real-time traffic.

Recently, IEEE802.11 wireless LANs [2], fundamentally based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) [3], have been widely used as parts of network infrastructures. The IEEE802.11 specification [2] defines DCF (Distributed Coordination Function) as a fundamental access control method based on CSMA/CA. All

IEEE802.11 stations[1] implement DCF, and sense carrier and transmit frames complying with DCF. However, DCF in itself can hardly accommodate real-time traffic, so any other mechanism which can support QoS is required. Additionally, because wireless LANs have higher latency and lower bandwidth than other major wireline LANs, QoS support in wireless LANs is much more important.

To accommodate real-time traffic in IEEE802.11 wireless LANs, the IEEE802.11 specification defines PCF (Point Coordination Function) as an optional access control method based on access points' polling. PCF is a centralized control protocol having an access point supervise the rights of medium access, and can support real-time traffic by providing contention free medium access. Several previous researches [4] [5] [6] [7] [8] proposed and examined applications of PCF to support constant small delay of real-time traffic in a wireless LAN. However, the applications can be utilized only in the case where every station is supported by a centralized controlled coordinator such as an access point. Therefore, decentralized control mechanisms which do not require any centralized controlled coordinators are researched as alternative solutions to support QoS. Decentralized control mechanisms can be used in any case, including the case where stations are connected with one another ad hoc.

Several previous researches [9] [10] proposed and examined DCF-based decentralized control mechanisms supporting real-time traffic. These mechanisms accommodate real-time traffic by differentiating real-time traffic from non-real-time traffic; hence real-time traffic can achieve relatively small delay. EDCF (Enhanced DCF) [11] [12] can be also used to accommodate real-time traffic. EDCF is a decentralized control protocol which is an extension of DCF and provides differentiated service for categorized traffic. Although these mechanisms achieve relatively small delay of real-time traffic by discriminating in favor of real-time traffic, delay fluctuation is yet so large owing to the burst feature of backoff in DCF or EDCF.

We propose a decentralized control mechanism suppressing delay fluctuation in CSMA/CA networks, called DDFC (Decentralized Delay Fluctuation Control). For the purpose of suppressing delay fluctuation, the backoff algorithm of DDFC considers waiting time of a frame, the time elapsing since the frame was enqueued into the interface queue, and sees that frames which have larger waiting time tend to have smaller backoff time. Thereby, longer waiting frames can be transmitted earlier, consequently delay fluctuation is suppressed. Additionally, DDFC can be easily used in EDCF-based wireless LANs. An EDCF-based wireless LAN using DDFC to control real-time traffic is realized by replacing the backoff algorithm used for real-time traffic in an EDCF-based wireless LAN by DDFC. We call this DDFC-aware EDCF-based wireless LAN the combination of EDCF with DDFC. While vanilla EDCF achieves small delay of real-time traffic, the combination of EDCF with DDFC achieves not only small delay of real-time traffic but also small delay fluctuation of real-time traffic. Moreover, DDFC does not interfere with existing IEEE802.11-compliant protocols such as PCF and HCF (Hybrid Coordination Function) [13]. Later, we examine the performance of DDFC by simulation.

The structure of the paper is as follows. In Sect. 2., we describe the motivation of our research to suppress delay fluctuation. In Sect. 3., we present DDFC (the decentralized delay fluctuation control mechanism). Afterward, we evaluate the performance of DDFC by simulation in Sect. 4.. Finally, in Sect. 5., we conclude this paper.

2. Motivation

Our research is motivated by delay characteristics of the current DCF-based wireless LANs inappropriate to support real-time traffic—delay fluctuation. Delay fluctuation is one of the factors which deteriorate the performance of many real-time applications. Hence, it is desirable to suppress delay fluctuation so far as possible. It is because DCF adopts BEB (Binary Exponential Backoff) as its backoff algorithm that DCF causes large delay fluctuation. EDCF also adopts BEB to control the medium access of categorized traffic, so EDCF cannot achieve small delay fluctuation of real-time traffic though it can achieve small delay.

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(1, CW)(slots)
```

where RC is retransmission count, which is set to be 0 when a frame attempts to be transmitted for the first time and incremented by 1 every time a 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; and B(slots) is backoff time. The pseudocode presents that the size of contention window is doubled when a frame is not transmitted successfully and minimized when a frame is transmitted successfully. In this algorithm, 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 contention window transmits no frame. Several researches [14] [15] pointed out this burst feature of BEB. The burst feature encourages delay fluctuation.

MILD (Multiplicative Increase Linear Decrease) [15] is a backoff algorithm proposed to mitigate the burst feature of transmission. Though MILD is not supposed to be used in IEEE802.11 wireless LANs, MILD can be used in IEEE802.11 wireless LANs by replacing the backoff algorithm of DCF by 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(1, CW)
```

where $max(a,b)$ is the function returning the larger number of a and b. The pseudocode presents that the size of

(1) : The word station means a terminal or an access point equipped with a wireless LAN interface.

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; as a result, it suppresses delay fluctuation.

However, MILD is unsuitable as a solution to suppress delay fluctuation because of the following reasons.

- MILD cannot achieve higher performance than BEB in almost all cases: MILD increases average delay in return for suppressing delay fluctuation. 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 a DCF-based wireless LAN adopts BEB or MILD as backoff algorithm. In fact, in Fig. 1, we can see 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 16. This illustrates MILD cannot suppress delay fluctuation except in the case where traffic is heavy.
- MILD is unsuitable for controlling real-time traffic in the differentiation framework of EDCF: In EDCF, every traffic category has its own priority, interface queue and protocol parameters such as CW_{min}, CW_{max} and IFS. IFS stands for interframe space, within which flows must not transmit data frames just after the medium turned to be idle. If two or more flows of an identical station attempt to transmit frames concurrently, the flow having the highest priority may transmit and the other should retransmit. Except this point, all flows obey with DCF. When MILD is used for the purpose of suppressing delay fluctuation of real-time traffic in the environments where EDCF 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; thus, the differentiation of EDCF does not work well. This is because BEB, frequently resetting CW, allows frames to be transmitted after smaller backoff than MILD. Figure 2 shows the average delay of category 1 traffic and category 2 traffic in an EDCF-based wireless LAN configured as shown in Table 3 and Table 1, where real-time traffic and non-real-time traffic are categorized into category 1 and category 2, respectively. Now, as shown in Table 1, we use p as the protocol used for category 1 traffic and n as the number of category 1 flows, respectively. If p = BEB, the average delay of category 1 traffic remains small even when n grows larger; the differentiation of EDCF works well. However, if p = MILD, the average delay of category 1 traffic is larger than that of cateogry 2 traffic; the differentiation of EDCF does not work well.

In the next section, we challenge to decrease delay fluctuation without deteriorating throughput or increasing average delay.

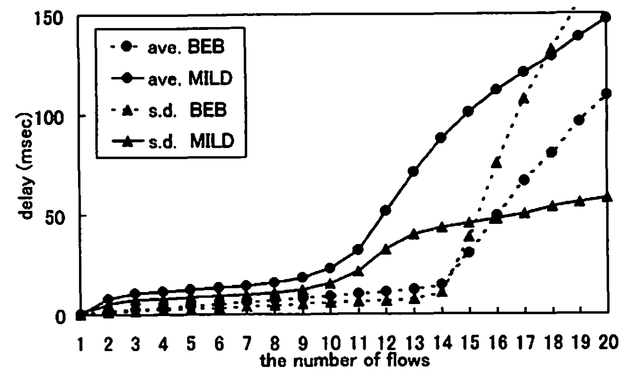


Figure 1 The delay characteristics of BEB and MILD. The parameter settings of the LAN are shown in Table 3, and CW_{min} and CW_{max} are 31 and 1023 respectively. The bit rate of every flow is 64kbps, the size of every packet is 256B, and the queue size is 4frames.

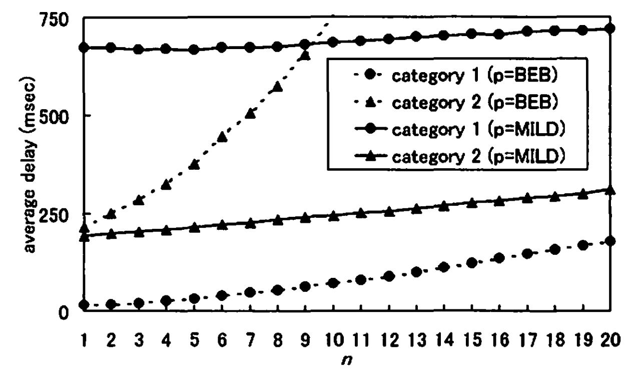


Figure 2 The average delay characteristics in the differentiation framework of EDCF. The parameter settings of the LAN are shown in Table 3, and EDCF settings are shown in Table 1. p and n are defined in Table 1.

3. Decentralized Delay Fluctuation Control Mechanism

3.1 Design Policy

We design DDFC (Decentralized Delay Fluctuation Control mechanism) to be based on the following policy.

- DDFC is a decentralized control mechanism: DDFC can be utilized without requiring any centralized controlled coordinators. Therefore, DDFC can be used in any case, including the case where stations are connected with one another ad hoc.
- DDFC can be used in EDCF-based wireless LANs: In practical wireless LANs, real-time traffic and non-real-time traffic coexist. The IEEE802.11 task group e is now standardizing EDCF as a decentralized controlled differentiation framework for IEEE802.11 wireless LANs. So, we assume that DDFC is utilized for real-time traffic in the differentiation framework of EDCF and that BEB is utilized for non-real-time traffic likely in vanilla IEEE802.11 wireless LANs.
- DDFC does not interfere with existing IEEE802.11-compliant protocols: EDCF is designed not to interfere with PCF or HCF. So, it is not desirable that the collaboration of EDCF with DDFC interferes with IEEE802.11-compliant protocols such as PCF and HCF.
- Delay fluctuation in DDFC is smaller than or equal to that in DCF in any case. Besides, throughput in DDFC is not much smaller than that in DCF and average delay in DDFC is not much larger than that in DCF.

To meet the above requirements, we design the backoff algorithm of DDFC to have the ability to suppress delay fluctuation, and the other parts of DDFC to be identical to those of DCF. In fact, DDFC is decentralized controlled because DDFC is identical to DCF except the difference in backoff algorithm. Additionally, the combination of EDCF with DDFC is easily realized by replacing the backoff algorithm for real-time traffic by the backoff algorithm of DDFC. Moreover, DDFC does not interfere with PCF or HCF because the IFS (Interframe Space) rules of DDFC are identical to those of DCF. We present the backoff algorithm in the next subsection.

3.2 Backoff Algorithm

For the purpose of suppressing delay fluctuation, we propose the backoff algorithm of DDFC specified by the following pseudocode:

if($RC = 0$){

Table 1 EDCF settings. p is BEB or MILD.

category (priority)	protocol		CW_{min}	CW_{max}	IFS	bit rate of a flow	frame size	queue length	flows	for which traffic
1 (high)	ED	p	15	255	50μs	64kbps	256B	4frames	$n$flows	real-time
2 (low)	CF	BEB	31	1023	70μs	—	1500B	8frames	4flows	non-real-time

// 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{

$$CW := (CW_{min}+1)2^{RC}-1$$

}
$CW := min(CW, CW_{max})$
}
$B := rand(1, CW)$

where t is waiting time of a frame, the time elapsing since the frame was enqueued into the interface queue; t_s is a protocol parameter functioning as the threshold used to determine whether t is considered to decide CW or not; and t_0 is a protocol parameter functioning as the scaling factor which adjusts the influence of t on CW.

If $RC = 0$ or $t \leqq t_s$, the algorithm lets CW be CW_{min} as well as BEB. Therefore, the algorithm performs as if it were BEB when traffic is light. If $RC > 0$ and $t > t_s$, the algorithm lets CW be monotonically decreased with increasing t. Thereby, a frame having larger waiting time is prompted to have smaller backoff time; small delay fluctuation is realized. Additionally, the factor, 2^{RC}, functions so as to double CW every retransmission similarly in BEB; contention resolution is performed. Therefore, the algorithm supports small delay fluctuation as it performs contention resolution when traffic is heavy, while it performs as well as BEB when traffic is light.

Because the proposed backoff algorithm decides CW not taking t into account if $RC = 0$ or $t \leqq t_s$, some readers perhaps think it cannot achieve small delay fluctuation when traffic is light. In fact, when traffic is light, small delay and small delay fluctuation are naturally achieved because almost all frames are transmitted successfully with short deferral and short backoff. Thereover, BEB can achieve high performance when traffic is light. Hence, we conclude that t does not need to be taken into account if $R = 0$ or t is small.

3.3 The Combination of EDCF with DDFC

We assume the combination of EDCF with DDFC as an application of DDFC. Vanilla EDCF can achieve small delay of real-time traffic owing to its differentiated services, but cannot achieve small delay fluctuation owing to the burst feature of BEB. The combination of EDCF with DDFC, which is realized by replacing the backoff algorithm used in a category for real-time traffic by the backoff algorithm of DDFC, can achieve not only relatively small delay of real-time traffic but also small delay fluctuation of real-time traffic.

Architecturally, DDFC can be used alone without EDCF, but we assume the case only where DDFC is used with EDCF because of the following reasons.

- Because real-time traffic and non-real-time traffic coexist in practical wireless LANs, traffic categorization is required to utilize DDFC for the suppression of delay fluctuation of real-time traffic.

Table 3 Parameter settings.

slottime	20μs
$SIFS$	10μs
$DIFS$	50μs
basic rate	1Mbps
data rate	2Mbps
$RetryLimit$	7

- Considering that EDCF is under standardization in IEEE802.11 task group e and that it has a framework of traffic categorization, using EDCF is an acceptable solution to categorize traffic.

Table 2 (a) shows an example of vanilla EDCF in which category 1 and category 2 are used for real-time traffic and for non-real-time traffic, respectively. This achieves relatively small delay of real-time traffic. Table 2 (b) shows an example of the combination of EDCF with DDFC. While all categories use BEB in vanilla EDCF, a category for real-time traffic uses DDFC and a category for non-real-time traffic uses BEB in the combination. The combination achieves not only small delay of real-time traffic but also small delay fluctuation of real-time traffic.

4. Performance

In this section, we examine the performance of DDFC by simulation. We use the settings shown in Table 3 and Table 4 for the simulation in this section. 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 until to be transmitted successfully.

4.1 Basic Characteristics

In this subsection, we present the basic characteristics of DDFC, which are obtained by the simulation in the settings shown in Table 3 and Table 4. Figure 3 shows a typical example of average delay, standard deviation of delay and throughput in an EDCF-based wireless LAN. p and n are defined in Table 4.

Figure 3 (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 EDCF using DDFC for category 1 traffic works as well as vanilla EDCF. On the other hand, the standard deviation of category 1 traffic in DDFC much smaller than that in BEB. This illustrates DDFC's ability to suppress delay fluctuation.

Figure 3 (b) shows that the throughput characteristics when p = DDFC are almost as same as those when p = BEB. However, 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, because DDFC's feature of making backoff time smaller causes more collisions. This is a compensation for suppression of delay fluctuation, but the compensation is little when traffic is almost nothing and a little even when traffic is heavy.

4.2 Impacts of parameters

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

4.2.1 Impacts of t_0

Figure 4 shows the impacts of t_0 on the delay characteristics of DDFC. In Fig. 4 (a), we can find that the average delay characteristics of DDFC is much the same as that of BEB, but the characteristics when traffic is heavy and that

Table 2 An example of vanilla EDCF and an example of the combination of EDCF with DDFC.

(a) An example of vanilla EDCF

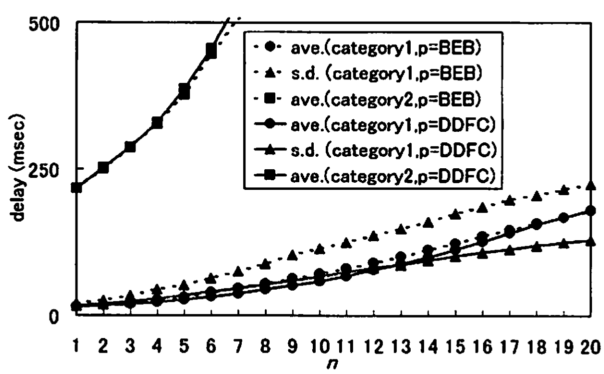
category (priority)	protocol		parameters					service	for which traffic
			CW_{min}	CW_{max}	IFS	t_s	t_0		
1 (high)	EDCF	BEB	15	255	50μs	—	—	small delay	real-time
2 (low)		BEB	31	1023	70μs	—	—		non-real-time

(b) An example of the combination of EDCF with DDFC

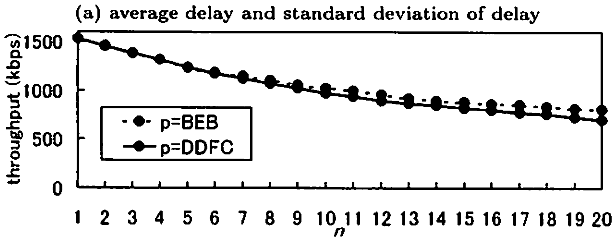
category (priority)	protocol		parameters					service	for which traffic
			CW_{min}	CW_{max}	IFS	t_s	t_0		
1 (high)	EDCF	DDFC	15	255	50μs	10ms	100ms	small delay, small delay fluctuation	real-time
2 (low)		BEB	31	1023	70μs	—	—		non-real-time

Table 4 EDCF settings. p is BEB or DDFC. t_s and t_0 are used only if p = DDFC.

category (priority)	protocol		CW_{min}	CW_{max}	IFS	t_s	t_0	bit rate of a flow	frame size	queue length	flows
1 (high)	ED	p	15	255	50μs	t_s	t_0	64kbps	256B	4frames	$n$flows
2 (low)	CF	BEB	31	1023	70μs	—	—	—	1500B	8frames	4flows

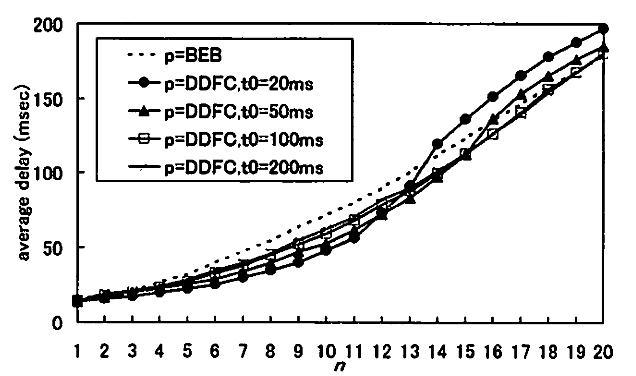


(a) average delay and standard deviation of delay

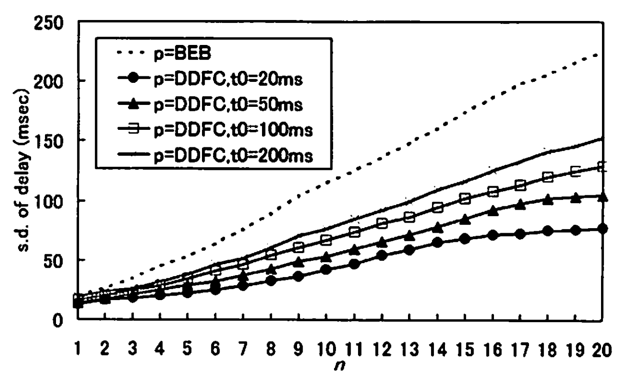


(b) wireless LAN throughput

Figure 3 The characteristics of EDCF. t_s = 20ms and t_0 = 100ms.



(a) average delay



(b) standard deviation of delay

Figure 4 The delay characteristics of category 1 traffic. t_s = 20ms.

when traffic is light are a little bit different.

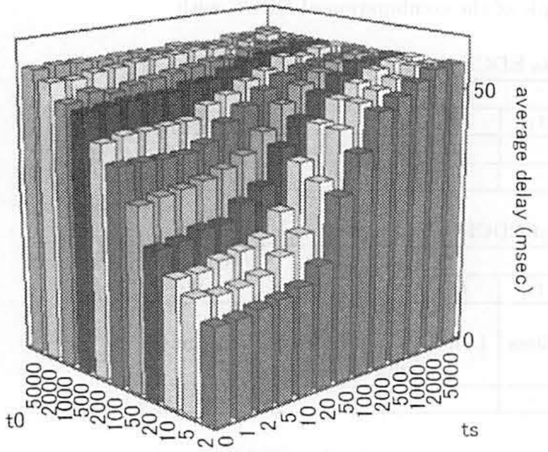
When n is small, namely when traffic is light, the average delay in DDFC is a little bit smaller than that in BEB. This is because of DDFC's feature to prompt frames waiting longer to be transmitted earlier. Additionally, we can see in Fig. 4 (a), the smaller t_0 is, the smaller the average delay is.

On the other hand, when n is large, namely when traffic is heavy, we can see in Fig. 4 (a), the smaller t_0 is, the larger the average delay is. Especially, if $t_0 \leqq$ 50ms, 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.

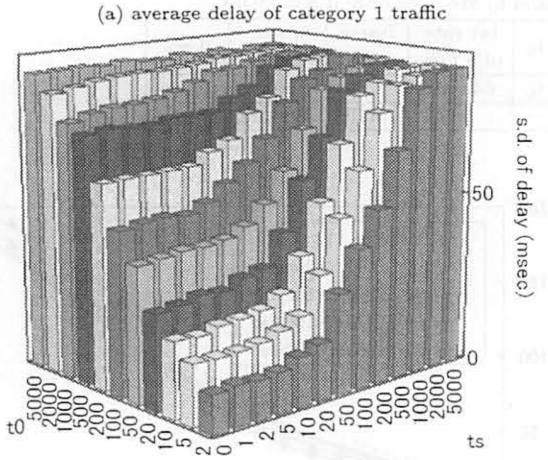
In Fig. 4 (b), we can find that the standard deviation of delay in DDFC is much smaller than that in BEB. We can see in the figure the smaller t_0 is, the smaller the standard deviation of delay is. This is because DDFC with smaller t_0 assigns delayed frames to smaller CW, and prompts the frames to 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.

4. 2. 2 Impacts of t_s

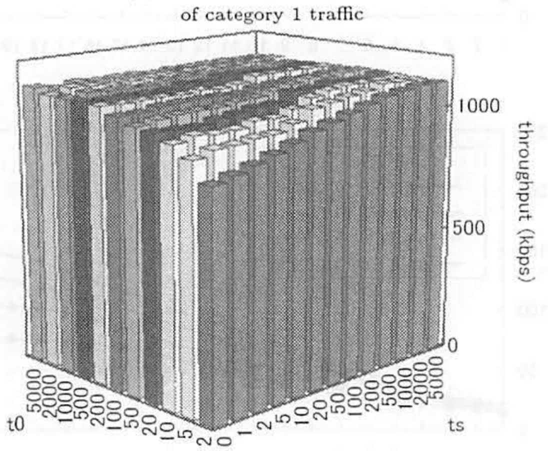
Figure 5 shows the impacts of t_s and t_0 on the performance. If t_0 is large, t_s has little influence on the performance. Except in the case, DDFC with smaller t_s achieves smaller average delay and smaller standard deviation of delay. However, Fig. 5 (a) and (b) show that t_s has little influence on the delay characteristics when t_s is small. This means too small t_s has no additional advantages. Additionally, as shown in Fig.



(a) average delay of category 1 traffic



(b) standard deviation of delay of category 1 traffic



(c) wireless LAN throughput

Figure 5 Impacts of t_s and t_0. $n = 8$.

5 (c), smaller t_s deteriorates wireless LAN throughput. This is because smaller t_s encourages more collisions. Therefore, using too small t_s is undesirable.

5. Conclusion

We proposed decentralized delay fluctuation control mechanism, called DDFC, to suppress delay fluctuation in CSMA/CA networks. DDFC can be used in IEEE802.11-based wireless LANs, and DDFC does not interfere with existing IEEE802.11-compliant protocols such as PCF and HCF.

We examined the performance of DDFC, which is used for real-time traffic in an EDCF-based wireless LAN, by simulation. The results of simulation confirmed that we can achieve not only small delay but also small delay fluctuation in EDCF-based wireless LANs by controlling real-time traffic according to DDFC.

References

[1] Transmission control protocol. RFC793, September 1981.

[2] IEEE. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999 edition, 1999.

[3] Leonard Kleinrock and Fouad A. Tobagi. Packet switching in radio channels: Part i - carrier sense multiple access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, Vol. COM-23, No. 12, pp. 1400–1416, December 1975.

[4] Brian P. Crow, Indra Widjaja, Jeong Geun Kim, and Prescott T. Sakai. Investigation of the IEEE 802.11 medium access control (MAC) sublayer functions. In *Proceedings of IEEE INFOCOM'97*, pp. 126–133, Kobe, Japan, April 1997.

[5] Brian P. Crow, Indra Widjaja, Jeong Geun Kim, and Prescott T. Sakai. IEEE 802.11 wireless local area networks. *IEEE Communications Magazine*, Vol. 35, No. 9, pp. 116–126, September 1997.

[6] Constantine Coutras, Sanjay Gupta, and Ness B. Shroff. Scheduling of real-time traffic in IEEE 802.11 wireless LANs. *ACM Wireless Networks*, Vol. 6, No. 6, pp. 457–466, December 2000.

[7] Malathi Veeraraghavan, Nabeel Cocker, and Tim Moors. Support of voice services in IEEE 802.11 wireless LANs. In *Proceedings of IEEE INFOCOM 2001*, pp. 488–497, Anchorage, Alaska, USA, April 2001.

[8] Kazuyoshi Saitoh, Yasuhiko Inoue, Masataka Iizuka, and Masahiro Morikura. 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, September 2001.

[9] Michael Barry, Andrew T. Campbell, and Andras Veres. Distributed control algorithms for service differentiation in wireless packet networks. In *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, USA, April 2001.

[10] Andras Veres, Andrew T. Campbell, Michael Barry, and Li-Hsiang Sun. Supporting service differentiation in wireless packet networks using distributed control. *IEEE Journal of Selected Areas in Communications*, Vol. 19, No. 10, pp. 2081–2093, October 2001.

[11] Greg Chesson, Wim Diepstraten, Maarten Hoeben, Aman Singla, Harold Teunissen, and Menzo Wentink. IEEE P802.11 wireless LANs EDCF proposed draft text. IEEE 802.11-01/131r1, March 2001.

[12] IEEE. Draft D4.0 IEEE std. 802.11e. 802.11e/D4.0, Nevember 2002.

[13] Michael Fischer. IEEE P802.11 wireless LANs hybrid coordination function (HCF) proposed updates to normative text of D0.1. IEEE 802.11-01/110r1, March 2001.

[14] Yoshiteru Takeshima, Kensuke Fukuda, Hideki Takayasu, and Misako Takayasu. An analysis on the critical phenomena in CSMA/CD network traffic model by computer simulations. *The Transactions of the Institute of Electronics, Information and Communication Engineers B*, Vol. J84-B, No. 5, pp. 840–848, May 2001.

[15] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A media access protocol for wireless LAN's. In *Proceedings of ACM SIGCOMM'94*, pp. 212–225, London, UK, September 1994.