

최적 Backoff Stage 선택을 통한 IEEE 802.11 성능 개선

윤희돈¹, 김응성^{2*}, 조성환³

IEEE 802.11 Performance Improvement by Selecting Optimal Backoff Stage

Hee-Don Yoon¹, Eung-Sung Kim^{2*} and Seong-Hwan Cho³

요약 본 논문에서는 IEEE 802.11 DCF의 성능을 향상시키기 위해 채널의 혼잡상태를 이용하여 충돌을 해결하는 방법을 제안한다. 이 기법은 혼잡 이벤트와 단일 스테이지 모델을 통해 스테이션들의 스테이지 이동 횟수를 감소시켜 무선채널에서의 충돌 발생량을 감소시킨다. 시뮬레이션을 통해 DCF, ADCF 그리고 SD 기법과 성능을 비교하였다.

Abstract In this paper we propose a new method which adjusts the collision resolution depending on busy status of the channel to improve performance of IEEE 802.11 DCF. This scheme reduces the number of collisions in wireless channel by decreasing the amount of movements of each station's backoff stage based on busy event count and the one stage backoff model. Through simulation, we compared the performance of our scheme with that of DCF, ADCF and SD.

Key Words : IEEE 802.11 DCF, collision avoidance, performance evaluation, multiple access control, ADCF, SD

1. 서론

IEEE 802.11[1]은 무선 통신 네트워크에서 가장 많이 보급된 MAC(Medium Access Control) 프로토콜 중 하나이다. IEEE 802.11 MAC의 기본적인 접근 방법은 CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) 기반의 DCF(Distributed Coordination Function)[1-5] 모드이다.

DCF 모드에서 채널 내 각 스테이션(Station)들은 초기 경쟁 윈도우(Initial Contention Window) $CW_{min}(=W)$ 에서 랜덤하게 백오프 카운트(Backoff Count)를 할당받는다. 각 스테이션들은 채널이 유희상태(Idle State)일 경우에만 백오프 카운트를 감소시키고 채널에서 전송이 감지될 경우에는 백오프 카운트를 그대로 유지시킨다. 백오프 카운트가 0이 되면 해당 스테이션은 전송을 시도한다. 전송에 실패하면 스테이지(Stage)를 하나 증가시키고 경쟁윈도우 크기를 두 배로 늘려 랜덤하게 백오프 카운트를 할당

받는다. 한 스테이션이 연속적으로 전송에 실패할 경우 최대 m 스테이지에 도달하게 되고 경쟁윈도우 사이즈는 최대 사이즈인 $2^m W$ 까지 증가하고 랜덤한 백오프 카운트를 할당받는다. 스테이션이 전송을 성공하면 0 스테이지로 이동하고 초기 경쟁윈도우 크기(Initial Contention Window Size) W 에서 랜덤하게 백오프 카운트를 할당받는다. 이때 채널 내 스테이션이 많을 경우 전송성공 시 0 스테이지로 이동시키고 다시 전송이 가능한 적절한 스테이지로 이동하기까지 충돌이 발생하게 된다. 이로 인해 시스템의 처리율(throughput)은 감소되고, 지연시간(delay)은 증가하게 된다.

이러한 문제점을 해결하기 위해 제시된 접근 방법은 IEEE 802.11의 백오프 알고리즘(Backoff Algorithm)을 개선하거나, Data-ACK (혹은 RTS-CTS-Data- ACK)의 기존의 프레임 교환 방식을 수정하여 한 번 전송에 성공한 스테이션이 채널을 오랫동안 소유하여 채널의 활용도를 높이는 방법들이 대표적이다.

본 논문에서는 IEEE 802.11 표준을 개선한 백오프 알고리즘으로서, 스테이션이 전송을 성공한 경우 0 스테이지로 이동하여 다시 전송이 가능한 적절한 스테이지(Proper Stage)까지 이동하는 동안 발생하는 불필요한 충

¹성균관대학교 정보통신공학부

²경기공업대학 컴퓨터정보시스템과

³금강대학교 교양학부

*교신저자: 김응성(imagecap@kinst.ac.kr)

돌을 방지하기 위해 채널 내 스테이션 수를 예측하여 전송이 성공한 후 바로 적절한 스테이지로 이동시키는 방법을 제안한다.

2. 백오프 알고리즘

2.1 DCF의 백오프 알고리즘과 기존의 향상된 백오프 알고리즘

IEEE 802.11 DCF는 포화상태(Saturation Condition)에서 심각한 처리율 감소와 불안정한 지연시간(Delay)이 발생한다[2]. 이는 채널 내 스테이션의 수가 증가함에 따라 전송을 위한 스테이션들 간의 경쟁이 매우 심해지므로 발생한다. 이를 해결하기 위해 제안된 백오프 알고리즘 중에는 SD(Slow CW Decrease) 기법[6]과 ADCF(Adaptive DCF) 기법[7]이 있다.

이 기법들의 차이는 각각의 백오프 알고리즘에서 스테이션이 전송을 성공한 직후 어떤 스테이지로 이동시키는지에 있다.

2.1.1 DCF의 백오프 알고리즘

DCF에서는 스테이션이 전송을 성공하면 스테이지를 0으로 이동시킨다. 전송에 실패한 경우, 그 스테이션의 스테이지가 k 였다면 $\min(k+1, m)$ 스테이지로 이동한다. 이때 m 은 최대 스테이지를 의미한다. 전송 성공 후 0 스테이지로 이동하기 때문에 다음 전송을 위해 전송이 가능한 스테이지로 이동하기까지 또다시 많은 충돌을 거쳐서 이동하게 된다. 채널 내 스테이션들이 많다면 더 많은 충돌을 거쳐야 한다. 이러한 충돌로 인해 시스템의 처리율은 낮아지고, 지연시간은 늘어나게 된다.

2.1.2 SD의 백오프 알고리즘

SD에서는 스테이션이 전송에 성공했을 때 0 스테이지로 보내지 않고, 파라미터 g 에 따라 0에서 $(k-1)$ 사이의 스테이지로 이동시킨다. 파라미터 g 는 시간에 따라 변하지 않는 고정된 값이다. 스테이션이 k 스테이지에서 전송에 성공했을 때, 다음 전송을 위해 이동할 초기 스테이지는 $(k-g)$ 스테이지가 된다. Qiang Ni가 제안한 논문[6]에서는 채널 내 스테이션의 수가 많은 경우 g 가 1 일 때 가장 좋은 성능을 보여주고 있다. 스테이션이 전송에 실패한 경우에는 SD기법에서도 DCF와 같이 $\min(k+1, m)$ 스테이지로 이동시킨다.

2.1.3 ADCF의 백오프 알고리즘

IEEE 802.11 DCF에서는 각 스테이션들이 전송을 위해 항상 채널의 상태를 모니터링(Monitoring)하고 있다. 백오프 프로시저(Backoff Procedure)가 수행되는 동안 각 스테이션들은 자신이 전송에 성공할 때(Successful Transmission), 전송에 실패할 때(Collision), 또는 다른 스테이션들이 채널을 사용하여 대기할 때(Freeze)의 정보를 모두 알 수 있다.

ADCF에서는 이러한 정보들을 이용해서 각 스테이지마다 성공률(τ_k)을 계산한다. 각 스테이지별 성공률은 그 스테이지에서 발생한 전송 성공 횟수와 전송 실패 횟수, 그리고 대기(Freeze)한 횟수로 구성되는 함수이다. 두 개의 스테이지 i 와 j 에서 만약 $\tau_i > \tau_j$ 라면 스테이지 i 에서 스테이지 j 보다 더 짧은 시간동안 더 많은 전송에 성공했다는 것을 의미한다. 따라서 전송성공 후 이동할 스테이지로 j 스테이지보다는 i 스테이지를 선택함으로써 더 높은 성공률을 기대할 수 있다. 그러므로 ADCF에서는 스테이션이 전송을 성공하면 전체 스테이지들 중에서 성공률이 가장 높은 스테이지로 이동시킨다. 전송에 실패한 경우에는 ADCF기법에서도 DCF와 같이 $\min(k+1, m)$ 스테이지로 이동시킨다.

3. 제안 백오프 알고리즘

3.1 시스템 모델링

본 논문에서는 G. Bianchi가 제안한 바와 같이 채널 내 고정적인 n 개의 스테이션이 경쟁하고 있고, 각 스테이션들은 한 번의 전송 성공 후 바로 전송할 또 다른 패킷이 있는 포화상태(Saturation status)를 가정하였다.[2]

$b(t)$ 는 슬롯 타임(Slot Time)이 t 일 때의 백오프 윈도우 크기를 나타내는 확률 프로세스(Statistic Process)이다. 이 프로세스는 non-Markovian이지만 편의상 $0 \leq i \leq m$ 일 때 $W = Z^m$ 라하고 i 를 백오프 스테이지라 하겠다. $s(t)$ 는 슬롯 타임이 t 일 때의 백오프 스테이지($0, \dots, m$)을 나타낸다. 이 모델에서 핵심적인 사항은 패킷 전송 시 충돌이 날 확률 p 를 스테이션의 상태 $s(t)$ 에 독립적이라고 근사화시키는 것이다. 이 조건에서 이차원 프로세스인 $\{s(t), b(t)\}$ 는 [그림 1]과 같이 discrete-time Markov chain이다.

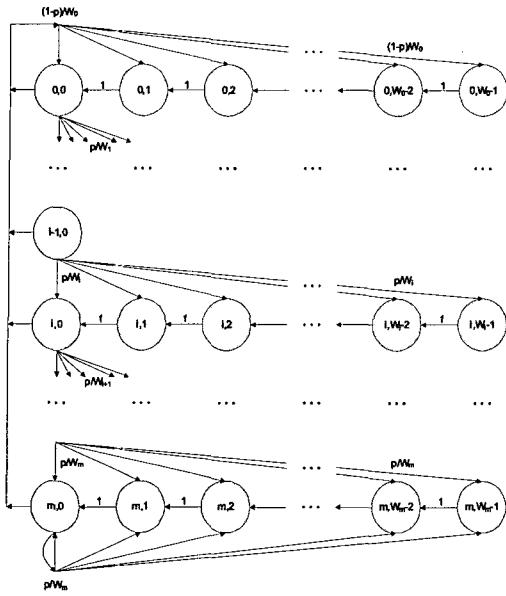


그림 1. 백오프 윈도우 크기의 Markov Chain 모델

이 모델에서 한 단계의 상태 변이 확률은 다음의 식(1)과 같다.

$$\begin{cases} P\{i, k | i, k+1\} = 1, & k \in (0, W_i - 2); \\ & i \in (0, m) \\ P\{0, k | i, 0\} = (i - p) / W_0, & k \in (0, W_0 - 1); \\ & i \in (0, m) \\ P\{i, k | i - 1, 0\} = p / W_i, & k \in (0, W_i - 1); \\ & i \in (i, m) \\ P\{m, k | m, 0\} = p / W_m, & k \in (0, W_m - 1). \end{cases} \quad (1)$$

첫 번째 식은 백오프 카운트의 감소를 나타내는 식이고, 두 번째 식은 전송 성공 후 다시 0 스테이지로 이동하는 것을 나타내는 식이며, 세 번째와 네 번째 식은 전송 실패 시 스테이지 이동을 설명하고 있다. $b_{i,k} = \lim_{t \rightarrow \infty} P\{s(t)=i, b(t)=k\}$, $i \in (0, m)$, $k \in (0, W_i - 1)$ 를 정상분포(stationary distribution)를 가진다고 하면 연쇄정칙(chain regularities)에 의해 다음 식(2)가 성립한다[2, 4].

$$\begin{aligned} b_{i,0} &= p^i b_{0,0}, & i \in (0, m-1) \\ b_{m,0} &= \frac{p^m}{1-p} b_{0,0} \\ b_{i,k} &= \frac{W_i - k}{W_i} b_{i,0}, & k \in (0, W_i - 1). \end{aligned} \quad (2)$$

$b_{0,0}$ 를 구하면 다음 식(3)과 같다.

$$\begin{aligned} 1 &= \sum_{i=0}^m \sum_{k=0}^{W_i-1} b_{i,k} = \sum_{i=0}^m b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} \\ &= \sum_{i=0}^m b_{i,0} \frac{W_i + 1}{2} \\ &= \frac{b_{0,0}}{2} \left[W \left(\sum_{i=0}^{m-1} (2p)^i + \frac{(2p)^m}{1-p} \right) + \sum_{i=0}^{m-1} p^i + \frac{p^m}{1-p} \right] \end{aligned} \quad (3)$$

식(3)으로부터 다음 식(4)를 계산할 수 있다.

$$b_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W+1) + pW(1-(2p)^m)}. \quad (4)$$

τ 를 스테이션이 전송을 할 확률이라 하면 τ 를 계산하면 식(5)와 같다.

$$\begin{aligned} \tau &= \sum_{i=0}^m b_{i,0} = \frac{b_{0,0}}{1-p} \\ &= \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)}. \end{aligned} \quad (5)$$

최종적으로 전송 시 충돌이 발생할 확률 p 는 다음 식(6)과 같이 구해진다.

$$p = 1 - (1 - \tau)^{n-1}. \quad (6)$$

3.2 포화상태의 처리율(Saturation throughput)

n 개의 스테이션이 활동하고 있는 채널에서 적어도 1개 이상의 채널이 전송을 시도할 확률을 P_g 이라 하고 스테이션이 전송에 성공할 확률을 P_s 라 하면 다음 식(7), 식(8)과 같이 나타낼 수 있다.

$$P_g = 1 - (1 - \tau)^n \quad (7)$$

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_g} = \frac{n\tau(1-\tau)^{n-1}}{1 - (1-\tau)^n}. \quad (8)$$

Ψ 를 연속적인 두 전송 사이에 발생하는 연속적인 유휴 슬롯들의 수라고 하면, 평균 값인 $E[\Psi]$ 는 다음 식(9)와 같다.

$$E[\Psi] = \frac{1}{P_g} - 1. \quad (9)$$

이러한 조건 내에서 정규화된 시스템의 처리율 S 를 식 (10)과 같이 구할 수 있다. 처리율 S 는 전송에 성공한 실제 데이터의 평균 전송 시간과 패킷을 성공적으로 전송하는 데 소요된 평균 시간의 비율로 구해진다[2-5].

$$S = \frac{E[\text{time used for successful transm. in interval}]}{E[\text{length of a renewal interval}]} = \frac{P_s E[P]}{E[\Psi] + P_s T_s + (1 - P_s) T_c} \quad (10)$$

식(10)에서 $E[P]$ 는 평균 패킷 길이 이고, T_s 는 전송 성공 시 소요되는 평균 시간이며, T_c 는 전송 실패 시 소요되는 평균 시간이다.

처리율 S 를 구하기 위해 T_s 와 T_c 를 정의하면, $H = PHY_{hdr} + MAC_{hdr}$ 을 패킷 헤더, δ 를 지연이라 했을 때, 다음 식(11)과 같이 표현할 수 있다.

$$\begin{cases} T_s = H + E[P] + SIFS + \delta + ACK + DIFS + \delta \\ T_c = H + E[P'] + DIFS + \delta \end{cases} \quad (11)$$

식(11)에서 $E[P']$ 은 충돌 발생 시 전송을 시도했던 패킷들 중 가장 긴 패킷들의 평균을 의미하는데, 본 논문에서는 패킷들의 길이가 모두 동일한 길이를 가지고 있다고 가정하였기 때문에 $E[P'] = E[P] = P$ 이다.

3.3 스테이션 수 예측

본 논문에서는 시스템의 처리율을 향상시키기 위해서 채널 내 활동 중인 스테이션의 수를 예측하여 전송 성공 후 적절한 스테이지로 이동시켜 불필요한 충돌을 방지하는 기법을 제안하고 있다. 이때 스테이션의 수를 예측하기 위해 BEC(Busy Event Count)[6]를 이용한다.

그림 2에서 BEC는 A라는 스테이션이 전송에 성공한 후 다시 한 번 전송에 성공할 때 까지를 한 주기라고 할 때, 한 주기 내에 다른 스테이션들이 전송 성공을 하거나 실패한 수와 A 스테이션이 전송에 실패한 수의 총합을 의미한다.

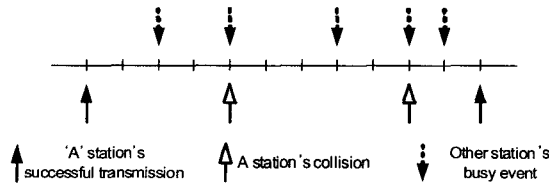


그림 2. BEC(Busy Event Count)

한 주기 내에서 일어나는 총 이벤트 수를 TEC(Total Event Count)라 하고 한 주기 내 채널이 유휴상태인 슬롯의 총 수를 IEC(Idle Event Count)라 한다. 한 주기 내에서의 TEC는 식 (12)와 같다. 매 주기마다 성공 이벤트의 수는 1이 된다.

$$TEC = IEC + BEC + 1 \quad (12)$$

채널 내 n 개의 스테이션 중 A 스테이션이 전송에 성공할 확률 $P_{A_success}$ 는 식(13)과 같다.

$$P_{A_success} = \tau(1 - \tau)^{n-1} \quad (13)$$

그러면 한 주기 내에서 다음 식(14)가 성립하게 된다.

$$TEC \times P_{A_success} = TEC \times \tau(1 - \tau)^{n-1} = 1 \quad (14)$$

채널이 유휴 상태일 확률 P_{idle} 은 식(15)이고, 식(15)로부터 식(16)을 계산할 수 있다.

$$P_{idle} = 1 - P_{busy} = (1 - \tau)^n \quad (15)$$

$$(1 - \tau)^{n-1} = \frac{1 - P_{busy}}{1 - \tau}, \quad (\tau \neq 0) \quad (16)$$

식 (15)를 식 (16)에 대입하면 다음 식(17)이 구해진다.

$$\begin{aligned} TEC \times \tau(1 - \tau)^{n-1} &= 1 \\ TEC \times \tau \frac{1 - P_{busy}}{1 - \tau} &= 1 \end{aligned} \quad (17)$$

이로부터 τ 를 계산할 수 있다.

$$\tau = \frac{1}{TEC(1 - P_{busy}) + 1} \quad (18)$$

식(15)를 변형하여 n 을 구하는 식 (19)를 얻을 수 있다.

$$\begin{aligned} P_{idle} &= (1 - \tau)^n \\ n &= \frac{\log P_{idle}}{\log(1 - \tau)} \end{aligned} \quad (19)$$

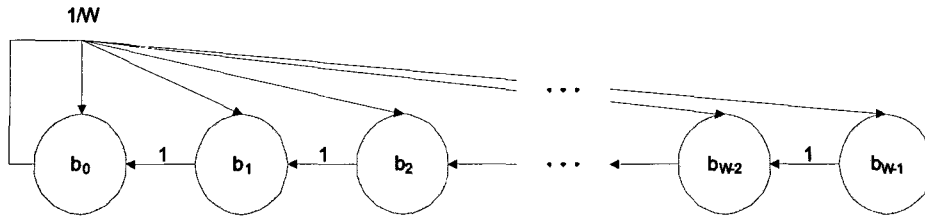


그림 3. 단일 스테이지 모델

3.4 적절한 백오프 스테이지

IEEE 802.11 DCF에서 스테이션은 충돌이 발생하거나 전송이 성공하면 백오프 스테이지를 하나 증가시키거나 0으로 리셋 시킨다. 혼잡한 상황에서 이러한 동작들은 매우 빈번하게 발생하게 된다. 이러한 동작들은 시스템의 처리율(Throughput)을 떨어뜨리고 지연시간(Delay)을 증가시킨다. 본 논문에서는 이러한 동작의 발생 횟수를 줄이기 위하여 채널상태를 고려한 적절한 백오프 스테이지(Proper Backoff Stage)를 제안한다. 이를 구하기 위해 [그림 3]과 같은 단일 스테이지 모델(Single Stage Model)을 사용하였다.

$$\tau' = b_0' = \frac{2}{W+1} \tag{20}$$

$$P_b' = 1 - (1 - \tau')^n \tag{21}$$

식(20)과 식(21)으로부터 스테이션의 수별로 최대의 처리율을 갖는 적절한 백오프 스테이지를 구하였다. [그림 4]는 포화상태(Saturation Condition)에서의 분석 결과이다. 포화상태에서의 처리율은 스테이션의 수($n=1,2,3, \dots, 50$)와 경쟁 윈도우 크기($m=0,1,2, \dots, 6$)의 범위 내에서 계산하였다.

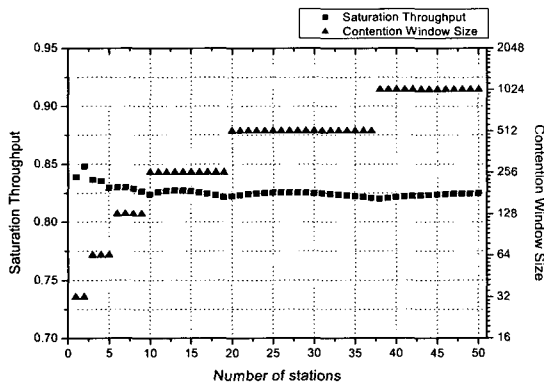


그림 4. 최대 처리율과 적절한 백오프 스테이지

4. 실험 결과

본 논문에서 제안하는 기법의 성능 테스트를 위해 이벤트 드리븐(Event-driven)방식의 시뮬레이션 프로그램을 작성하고 [표 1]의 파라미터를 이용하여 n 을 예측하는 시뮬레이션을 수행하였다. 그 결과가 [표 2]에 나타나 있는데, 여기서 n 은 실제 스테이션 수이고, $E[n]$ 은 예측한 스테이션 수를 의미한다.

표 1. 시뮬레이션에서 사용한 파라미터

Packet payload	8184 bits
MAC header	272 bits
PHY header	128 bits
ACK length	112 bits + PHY header
RTS length	160 bits + PHY header
CTS length	112 bits + PHY header
Channel Bit Rate	1 Mbit/s
Propagation Delay	1 μ s
SIFS	28 μ s
DIFS	128 μ s
Slot Time	50 μ s
Initial CW size	16
Maximum stage	6

표 2. IEEE 802.11 DCF 에서의 스테이션 수 예측 결과

n	$E[n]$
5	5.01
10	10.01
15	15.02
20	20.00
25	25.04
30	30.00
35	34.96
40	39.98
45	44.92
50	49.97

4.1 충돌확률 비교

[표 3]에서 보면 ADCF와 SD가 DCF에 비해 충돌 확률이 준 것을 볼 수 있다. 그러나 본 논문에서 제안한 기법이 기존의 기법들 보다 충돌 확률을 현저히 줄인 것을 볼 수 있다.

표 3. DCF, ADCF, SD 와 제안한 기법의 충돌 확률 비교

n	DCF	ADCF	SD	Proposed
5	0.12	0.08	0.04	0.04
10	0.18	0.14	0.06	0.04
15	0.21	0.18	0.07	0.05
20	0.24	0.21	0.08	0.05
25	0.26	0.23	0.09	0.05
30	0.28	0.25	0.10	0.05
35	0.30	0.27	0.10	0.05
40	0.31	0.29	0.11	0.05
45	0.32	0.30	0.11	0.05
50	0.34	0.31	0.12	0.05

4.2 포화상태의 처리율 비교

4.1 절에서 본 것과 같이 SD나 ADCF보다 제안된 기법의 충돌 확률이 현저히 줄었다. 이로 인해 처리율도 [그림 5]와 같이 현격히 향상된 것을 볼 수 있다.

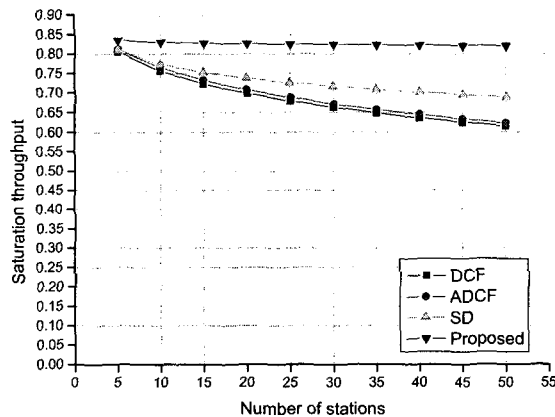


그림 5. 포화상태의 처리율

4.3 포화상태의 지연시간 비교

처리율과 마찬가지로 충돌확률의 감소로 인해 제안된 기법은 지연시간도 DCF, SD, ADCF 보다 감소한 것을 [그림 6]을 통해 볼 수 있다.

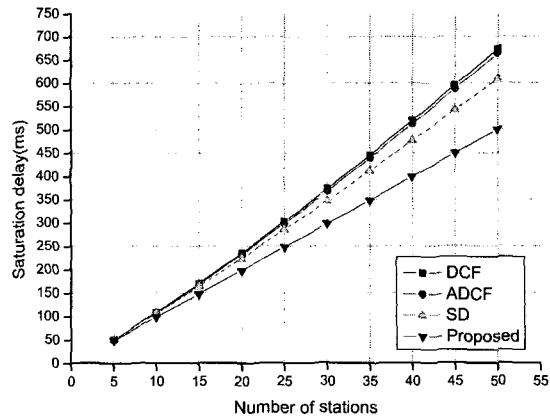


그림 6. 포화상태의 지연시간

5. 결론

본 논문에서는 기존의 802.11 DCF 방식이 채널 내 스테이션의 수가 많아질수록 충돌이 많이 발생하여 성능이 저하되는 문제점을 해결하기 위하여 개선된 백오프 알고리즘을 제안하였다. 백오프 프로시저(Backoff Procedure)가 수행되는 동안 각 스테이션들은 자신이 전송에 성공한 횟수(Successful Transmission), 전송에 실패한 횟수(Collision), 또는 다른 스테이션들이 채널을 사용하여 대기한 횟수(Freeze)의 정보를 모두 알 수 있다. 이 정보를 이용하여 혼잡 채널 내 스테이션 수를 예측하고, 스테이션이 전송에 성공한 후 예측한 스테이션 수에 적절한 백오프 스테이지로 이동함으로써 충돌 확률을 낮추어 시스템의 성능을 향상시키는 기법이다. 이 기법을 시뮬레이션한 결과, 기존의 802.11 DCF, SD(Slow CW Decrease), ADCF(Adaptive DCF)보다 처리율과 지연시간 성능에 향상이 있었음을 볼 수 있었다. 그리고 채널 내 스테이션의 수를 예측하는 하는 기법은 관련 연구에도 도움이 될 수 있을 것이다. 향후에는 포화상태가 아닌 상황(Normal Condition)에 본 논문에서 제안한 기법을 적용해 볼 것이다.

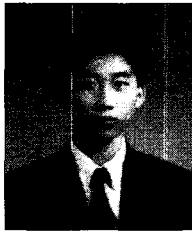
참고문헌

- [1] IEEE, "Standard for wireless LAN medium access control (MAC) and physical layer (PHY) specification," IEEE 802.11 WG, Aug. 1999.
- [2] G. Bianchi, "IEEE 802.11 ~ Saturation throughput analysis," IEEE Commun. Lett., vol.2, no.12, pp.318-320, Dec. 1998.

- [3] Y. Xiao, "A simple and effective priority scheme for IEEE 802.11," IEEE Commun. Lett., vol.7, no.2, pp.70-72, Feb. 2003.
- [4] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE Journal on, vol.18, Issue 3, pp.535-547, Mar. 2000.
- [5] Y. Xiao, "An analysis for differentiated services in IEEE 802.11 and IEEE 802.11e wireless LANs," Distributed Computing Systems, 2004. Proceedings. 24th International Conference on, pp.32-39.
- [6] Qiang Ni, Imad Aad, Chadi Barakat and Thierry Turletti, "Modeling and analysis of slow CW decrease for IEEE 802.11 WLAN," Proceedings of PIMRC, September 2003.
- [7] Kuppa, S and Prakash, R., "Adaptive IEEE DCF scheme with knowledge-based backoff," IEEE Wireless Communications and Networking Conference, vol. 1, pp.63-68, 2005.

윤 희 돈(Hee-Don Yoon)

[정회원]



- 1996년 8월 : 성균관대학교 전자공학과 (공학사)
- 2001년 8월 : 성균관대학교 대학원 전기전자 및 컴퓨터공학과 (공학석사)
- 2001년 8월 ~ 현재 : 성균관대학교 전기전자 및 컴퓨터공학과 (박사과정)

<관심분야>

영상통신, 무선통신, 영상처리 등

김 응 성(Eung-Sung Kim)

[정회원]



- 1989년 2월 : 성균관대학교 전자공학과 (공학사)
- 1992년 2월 : 성균관대학교 대학원 전자공학과(공학석사)
- 1998년 2월 : 성균관대학교 대학원 전자공학과(공학박사)
- 1995년 3월 ~ 2000년 2월 : 성균관대학교 과학기술연구소 연구전담요원
- 2000년 3월 ~ 현재 : 경기공업대학 컴퓨터정보시스템과 부교수

<관심분야>

영상처리, 영상통신, 신경회로망 등

조 성 환(Seong-Hwan Cho)

[정회원]



- 1980년 2월 : 성균관대학교 전자공학과 (공학사)
- 1982년 2월 : 성균관대학교 대학원 전자공학과(공학석사)
- 1991년 8월 : 성균관대학교 대학원 전자공학과(공학박사)
- 1982년 ~ 1985년 : 해군사관학교 전기 및 전자공학과 전임강사
- 1997년 : 미국 Columbia 대학 CATT Visiting Scholar
- 1985년 ~ 2002년 : 동서울대학 컴퓨터공학과 부교수
- 2003년 ~ 현재 : 금강대학교 교양학부 컴퓨터전공 부교수

<관심분야>

영상처리, 신경회로망, 패턴인식, DRM 등