

# 802.11e의 성능 향상을 위한 개별적 클래스 트래픽에 기반한 동적 충돌 윈도우 크기 조절 기법 (An Adaptive Contention- window Adjustment Technique Based on Individual Class Traffic for IEEE 802.11e Performance)

엄진영<sup>\*</sup> 오경식<sup>\*</sup>  
(Jinyeong Um) (Kyungsik Oh)

안종석<sup>\*\*</sup>  
(Jongsuk Ahn)

**요약** 본 논문은 IEEE 802.11e EDCA의 성능 향상을 위해서 클래스 트래픽에 기반하여 각 클래스의 경쟁 윈도우의 크기를 동적으로 조절하는 방안을 제안한다. IEEE 802.11e EDCA는 IEEE 802.11 DCF와는 달리 서비스 차별화를 위해 네 개의 클래스 AC마다 고정된 최소 경쟁 윈도우 크기인 CWmin 값과 최대 경쟁 윈도우 크기인 CWmax 값을 적용한다. 트래픽의 특성별로 나뉜 AC의 고정된 파라미터 값은 각 AC간의 차별화를 보장하지만 네트워크 환경이 혼잡할 경우 충돌 발생 확률을 높인다. 이를 해결하기 위해 채널과 네트워크 상태를 고려하여 일정한 주기마다 계산된 충돌 확률로 CWmin 값을 조절하는 CWminAS(CWmin Adaptation Scheme)가 제안되었다. 그러나 이 방식은 AC별 다른 파라미터 값으로 인해 발생하는 각 AC의 충돌 확률을 고려하지 않았다. 이러한 문제

를 해결하기 위해 본 논문에서는 AC별 다른 파라미터 값으로 인해 발생하는 각 AC의 충돌 확률을 계산하고, CWmin 값을 조절하는 ACATICT(Adaptive Contention-window Adjustment Technique based on Individual Class Traffic) scheme을 제안한다. 실험 결과 ACATICT가 CWminAS보다 효율이 최대 약 10% 향상되었다.

**키워드** : IEEE 802.11e, QoS, EDCA, 백오프 알고리즘, contention window (CW)

**Abstract** This paper proposes a technique for improving IEEE 802.11e EDCA's performance by dynamically adjusting each class's contention window size based on each class's traffic amount. For providing differentiated service differently from 802.11, 802.11e EDCA maintains four classes each of which specifies different static minimum and maximum contention window sizes. Since the initial window sizes significantly affect the 802.11e performance, several window adjustment schemes have been proposed. One of the schemes known as CWminAS(CWmin Adaptation Scheme) dynamically and synchronously determines the four windows' size based on the periodically measured collision rate. This method, however, can lower the send probability of high priority classes since it can't differentiate their collisions from those of low priority classes, leading to the channel underutilization. For solving this problem, we propose ACATICT(Adaptive Contention-window Adjustment Technique based on Individual Class Traffic) algorithm which adapts each class window size based on each individual collision rate rather than one average collision rate. Our simulation experiments show that ACATICT achieves better utilization by around 10% at maximum.

**Key words** : IEEE 802.11e, QoS, EDCA, backoff algorithm, contention window (CW)

## 1. 서론

무선 랜의 IEEE 802.11 표준안에서는 두 가지의 채널 획득 방안을 지원한다. 첫째는 CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) 기법과 랜덤 백오프(Random Backoff) 기법을 바탕으로 경쟁 기반의 채널 획득 방법인 DCF(Distributed Coordination Function)이며, 둘째는 단순 폴링을 통한 비경쟁 방식의 채널 분배 방법인 PCF(Point Coordination Function)이다[1]. DCF 방식은 기존의 이더넷 망에 쉽게 접근할 수 있지만, 멀티미디어 데이터 전송과 QoS(Quality of Service) 보장의 한계를 가지고, QoS 제공을 목적으로 제안된 PCF 방식은 구현의 복잡도와 일반적인 데이터 전송에서의 비효율성 때문에 거의 사용되지 않는다. 이에 무선망을 통해 비디오나 음성 신호등의

\* 이 논문은 2007 한국컴퓨터종합학술대회에서 '802.11e의 성능 향상을 위한 개별적 클래스 트래픽에 기반한 동적 충돌 윈도우 크기 조절 기법'의 제목으로 발표된 논문을 확장한 것이다

<sup>\*</sup> 학생회원 : 동국대학교 컴퓨터학과  
mog07@dgu.edu  
eoris81@dgu.edu

<sup>\*\*</sup> 종신회원 : 동국대학교 컴퓨터학과 교수  
jahn@dgu.edu

논문접수 : 2007년 10월 2일  
심사완료 : 2008년 1월 11일

Copyright©2008 한국정보과학회 : 개인 목적이나 교육 목적의 경우, 이 작품의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨터의 실제 및 래터 제14권 제2호(2008.4)

멀티미디어 데이터를 전송하는 어플리케이션을 지원하기 위해 IEEE 802.11e가 개발되었다.

IEEE 802.11e는 기존 IEEE 802.11 MAC 프로토콜의 DCF와 PCF를 기반으로 하는 HCF(Hybrid Coordination Function)를 제시하였다. HCF는 DCF와 유사한 경쟁 기반의 채널 접근 방법인 EDCA(Enhanced Distributed Channel Access)와 PCF와 유사한 단순 폴링 방식(Polled Channel Access)인 HCCA(HCF Controlled Channel Access)를 함께 사용한다. EDCA는 기존의 DCF를 보완한 것으로 802.1P CoS(Class of Service)에 따라 트래픽을 8개의 카테고리로 나누며, 이에 따라 IFS(Inter Frame Space)와 CWmin(Contention Windows minimum size)을 다르게 적용하여 트래픽 카테고리별 서비스를 차별화한다.

상위 계층으로부터 MAC 계층에 도착하는 각 프레임은 트래픽 카테고리별 사용자 우선순위를 가지게 되며 각각의 QoS 데이터 프레임 MAC 헤더에는 사용자 우선순위 값이 포함되어 있다. 우선순위를 포함하는 QoS 데이터 프레임의 전송을 위해 IEEE 802.11e QoS 스테이션은 4개의 AC(Access Category)를 구현한다. 모든 AC는 각각의 전송 큐와 AC 파라미터 값을 가지게 되는데, AC간의 우선순위 차이는 서로 다르게 설정된 AIFS[AC], CWmin[AC], CWmax[AC]와 같은 AC 파라미터 값으로 구현된다[2]. 각 AC마다 고정된 파라미터 값은 각 AC간의 차별화를 보장하지만 네트워크 환경이 혼잡할 경우 충돌을 발생시킬 확률을 높인다. 그리고 성공적인 패킷 전송 후 CW 값을 0과 (CWmin-1) 사이 값으로 감소시키는 백오프(back-off) 기법은 네트워크 환경이 혼잡할 때, 재충돌 확률을 높인다[3]. 이를 해결하기 위해 [4]에서 채널과 네트워크 상태를 고려하여 일정 주기마다 계산된 충돌 확률로 CWmin 값을 조절하는 CWminAS가 제안되었다. 그러나 이 방식은 AC별 다른 파라미터 값으로 인해 발생하는 각 AC의 충돌 확률을 고려하지 않았다[5]. 이러한 문제를 해결하기 위해 본 논문에서는 AC별 다른 파라미터 값으로 인해 발생하는 각 AC의 충돌 확률을 계산하고, AC별 CWmin 값을 조절하는 ACATICT를 제안한다. 2장에서는 EDCA 개념에 대하여 설명하며, 3장에서는 기존에 제안되었던 방식인 CWminAS에 대하여 기술한다. 4장에서는 본 논문에서 제안하는 ACATICT에 대하여 기술하며, 5장에서는 ACATICT와 CWminAS의 성능을 비교 분석한다. 6장에서는 결론 및 향후 연구 과제를 기술한다.

2. EDCA 개념

경쟁 기반 채널 접근 방식인 EDCA는 기존의 DCF를 강화해 8가지 종류의 사용자 우선순위를 가지는 프레임

에 대해 차별화된 매체 접근을 허용한다. 상위 계층으로부터 MAC 계층에 도착하는 각 프레임은 특정 사용자 우선순위 값을 지니게 되며 각각의 QoS 데이터 프레임 MAC 헤더에는 사용자 우선순위 값이 포함되어 있다. 우선순위를 포함하는 QoS 데이터 프레임의 전송을 위해 IEEE 802.11e QoS 스테이션은 4개의 AC를 구현한다. 모든 AC는 각각의 전송 큐와 AC 파라미터를 가지게 되며, AC간의 우선순위 차이는 각 AC마다 다르게 설정된 AC 파라미터 값으로 인해 발생된다. 즉, EDCA는 각 AC에 속한 프레임을 전송하기 위해 DCF에서 사용되는 DIFS, CWmin, CWmax를 대신하여 AIFS[AC], CWmin[AC], CWmax[AC]를 사용한다.

표 1은 IEEE 802.11e EDCA에서의 파라미터 셋을 나타낸다. AIFS[AC], CWmin[AC], CWmax[AC] 등의 값은 AP(Access Point)에 의해 비콘 프레임에 실려 각 스테이션에 통보될 수 있다. 기본적으로 AIFS[AC]와 CWmin[AC] 값이 작을수록 높은 우선순위를 가지며, 이에 따라 채널 접근 지연이 짧아져 주어진 트래픽 환경에서 보다 많은 대역을 사용한다. 이런 EDCA 파라미터들은 다양한 사용자 우선순위 트래픽에서의 채널 접근을 차별화하기 위해 사용한다[2].

그림 1은 IEEE 802.11e의 기본 MAC 프로토콜인 EDCA의 동작을 보여준다. STA 1이 STA 2보다 높은 우선순위를 가진다고 가정한다. 높은 우선순위에 해당하는 AC 0에 속한 패킷을 전송하려는 STA 1은 낮은 우선순위의 AC 3에 속한 패킷 전송을 시도하려는 STA 2보다 먼저 매체에 접근하게 된다. AIFS의 차이 때문에 AC 0은 AC 3보다 먼저 백오프 타이머를 감소시켜 나가며, 백오프 타이머가 0이 되면 데이터를 전송한다. STA 1이 데이터를 전송하게 되면, STA 2는 백오프 값을 감소시키는 중 채널이 busy한 상태라고 판단하고

표 1 802.11e EDCA에서의 기본 파라미터 셋

AC	CWmin	CWmax	AIFSN
AC_VO(Voice)	7	15	2
AC_VI(Video)	15	31	2
AC_BE(Best Effort)	31	1023	3
AC_BK(Background)	31	1023	7

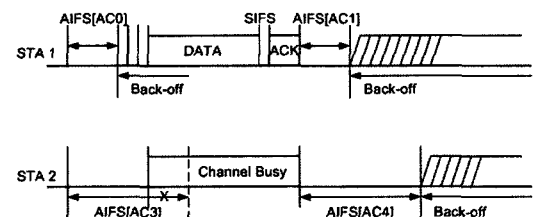


그림 1 EDCA 동작

전송을 지연한다. 이러한 EDCA 동작의 특성상 우선순위가 낮은 AC는 우선순위가 높은 AC에 비해 AIFS의 차이와 CWmin 값의 차이로 인해 천천히 백오프 타이머를 감소시키며, 각 AC 마다 채널의 상태를 다르게 인식하게 된다[6,7].

### 3. 기존의 CWminAS

네트워크 상황이 혼잡한 경우 발생하는 충돌 확률을 낮게 하기 위해 채널 상태와 네트워크 환경을 주기적으로 알아보고 충돌 확률에 따라 CWmin 값을 조절하는 CWminAS가 제안되었다[4].

CWminAS는 일정한  $T_{update}$  주기마다 하나의 스테이션이 경험하는 충돌 확률을 계산하여 CWmin 값을 조절한다. 식 (1)은  $j$  번째 주기에서 발생되어지는 충돌 확률로 해당 주기 동안 스테이션  $p$ 에서 전송한 패킷 수인  $Num(data\_sent_j[p])$ 와 스테이션  $p$ 에서 전송한 패킷이 충돌이 발생한 수인  $Num(collisions_j[p])$ 을 이용하여 계산한다.

$$f_{curr}^j = \frac{Num(collisions_j[p])}{Num(data\_sent_j[p])} \quad (1)$$

식 (2)는 random fluctuation을 최소화하기 위해 EWMA(Exponentially Weighted Moving Average)를 사용하여 충돌 확률의 평균을 계산한다. 여기에서의  $\alpha$ 는 0에서 1사이 범위 값으로 smoothing factor를 의미한다.  $f_{avg}^{j-1}$ 는 이전 주기인  $j-1$ 번째 주기의 충돌 확률을 의미한다.

$$f_{avg}^j = (1-\alpha) \times f_{curr}^j + \alpha \times f_{avg}^{j-1} \quad (2)$$

$j$  번째 주기에 발생되어지는 충돌 확률을 사용하여  $T_{update}$  시간 후인  $j+1$  번째 주기에 사용될 CWmin 값을 식 (3)을 사용하여 도출하게 된다. 식 (3)은 충돌이 많이 발생하여 식 (2)에서 도출한 충돌 확률 평균이 1일 경우,  $(CW_{max}[i] - CW_{min}[i]) \times 2^{i-2}$  만큼 반영하여  $CW_{min}[i]$  값을 크게 하여 재충돌 확률을 줄인다. 또한 충돌이 발생하지 않아서 충돌 발생 확률이 0일 경우, 기존의  $CW_{min}[i]$  값을 사용하여 빠른 전송을 유도한다. 여기에서의  $CWmin[i]$ 는 AC[i]의 CWmin 값을 의미하고,  $CWmax[i]$ 는 AC[i]의 CWmax 값을 의미한다.

$$DCW_{min}[i] = (1 - f_{avg}^j) \times CW_{min}[i] + f_{avg}^j \times (CW_{max}[i] - CW_{min}[i]) \times 2^{i-2} \quad (3)$$

CWminAS로 CWmin 값을 도출할 경우 네트워크 환경에 따른 충돌 확률을 고려하여 CWmin 값을 조절함으로써 고정된 파라미터 값을 가지는 EDCA에 비해 충돌 확률을 줄이게 된다.

### 4. ACATICT scheme

CWminAS는 일정한 주기별 충돌 확률을 고려하여 CWmin 값을 조절하지만, 각 AC별 AIFS[AC], CWmin[AC], CWmax[AC]가 다름으로 인해 각 AC별 매체 접근 빈도수와 충돌 확률이 다르게 나타나는 것을 고려하지 않았다[5].

그림 2는 충돌 구간[7]을 보여준다. AC 1은 AC 2보다 높은 우선순위를 가진다고 가정할 때 contention zone 1은 오로지 AC 1에 해당하는 패킷을 전송하려는 스테이션들만이 매체 접근을 위한 경쟁을 시도한다. 다시 말해 contention zone 1은 낮은 우선순위의 AC들, 즉 AC 2는 경쟁하지 않는다. contention zone 1에서는 AC 1에 해당하는 스테이션만 경쟁을 하게 되고, contention zone 2에서는 AC 1과 AC 2에 해당하는 스테이션들이 경쟁을 하기 때문에 각 구간마다 경쟁을 하는 스테이션의 수가 다르게 된다. 따라서 CWminAS와 같이 충돌 확률을 하나의 스테이션이 한 주기 동안에 충돌이 일어난 패킷 수 / 전송한 패킷 수로 단순히 계산하여 정확한 충돌 확률을 고려할 수 없다. 따라서 본 논문에서는 클래스별 충돌 확률 측정 방식을 이용하여 각 AC당 발생하는 충돌 확률을 계산하여 CWmin 값을 조절한다.

그림 3은 ACATICT를 보여준다. AC 1이 AC 2보다 높은 우선순위를 가지며,  $P_1$ 은 AC 1의 충돌 확률,  $P_2$ 는 AC 2의 충돌 확률이라고 가정한다. 만약 AC 1에서 데이터 전송 시 충돌이 많이 발생하여  $P_1$ 이 높고, AC 2에서 충돌이 적게 발생하여  $P_2$ 가 낮을 경우에 각 클래스별로 충돌 확률을 다르게 계산하지 않는다면, AC 2는 자신이 데이터를 보내는 시간의 충돌 확률인  $P_2$ 가 낮음에도 불구하고  $P_1$ 가 높음으로 인해 AC 2의

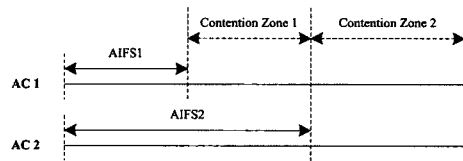


그림 2 충돌 구간

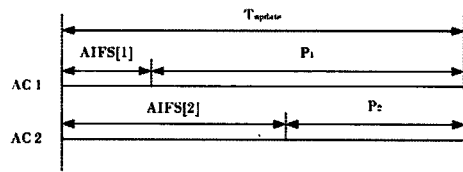


그림 3 ACATICT scheme

CWmin 값이 커지게 되어 데이터를 전송하기 어려워지게 된다. 즉, 공평성의 문제가 발생하게 되므로  $P_1$ 와  $P_2$ 를 각각 계산하여 각 클래스별로 충돌 확률을 다르게 주어야 한다.

ACATICT는 CWminAS와 유사하게 동작하는데, AC별 데이터의 전송 수와 충돌이 일어난 수를 계산하고, 이를 이용하여 각 AC별 충돌 확률의 평균을 구한다는 점에서 다르다. 식 (4)는 j 번째 주기에서 발생되어지는 AC[i]의 충돌 확률로 해당 주기 동안 스테이션 p에서  $T_{update}$  주기 동안 AC[i]로 전송한 패킷 수인  $Num(data\_sent_j[p_i])$ 와 스테이션 p에서  $T_{update}$  주기 동안 AC[i]로 전송한 패킷이 충돌이 발생한 수인  $Num(collisions_j[p_i])$ 을 이용하여 계산한다.

$$f_{curr}^j[i] = \frac{Num(collisions_j[p_i])}{Num(data\_sent_j[p_i])} \quad (4)$$

식 (5)는 random fluctuation을 최소화하기 위해 EWMA를 사용하여 충돌 확률의 평균을 계산한다. 여기에서의  $\alpha$ 는 0에서 1사이 범위 값으로 smoothing factor를 의미한다.  $f_{avg}^{j-1}[i]$ 는 이전 주기인 (j-1)번째 주기의 AC[i]의 충돌 확률을 의미한다.

$$f_{avg}^j[i] = (1-\alpha) \times f_{curr}^j[i] + \alpha \times f_{avg}^{j-1}[i] \quad (5)$$

j번째 주기에 발생되어지는 AC[i]의 충돌 확률을 사용하여 (j+1)번째 주기, 즉  $T_{update}$  시간 후 사용될 AC[i]의 CWmin 값을 식 (6)을 사용하여 도출하게 된다.

$$DCW_{min}[i] = (1-f_{avg}^j[i]) \times CW_{min}[i] + f_{avg}^j[i] \times (CW_{max}[i] - CW_{min}[i]) \times 2^{i-2} \quad (6)$$

ACATICT를 사용할 경우 각 AC의 실제적인 채널 상태에 따른 충돌 확률을 계산하여 CWmin 값을 조절함으로써 CWminAS보다 성능이 향상된다.

### 5. 성능 분석

본 장에서는 ACATICT의 성능을 분석하기 위해 CWminAS와 ACATICT의 성능을 비교 분석한다.

실험 환경은 네트워크 시뮬레이터 NS-2[8]를 사용하였다. NS-2에 IEEE 802.11e EDCA가 동작하도록 [9,10]에서 구현한 EDCA 모듈을 추가하였다.

그림 4는 실험 토폴로지로서 모든 스테이션에 똑같은 접속 기회를 부여하기 위해 링 토폴로지를 사용한다. 각 스테이션은 UDP CBR 데이터 스트림 방식으로 패킷을 전송하며, 이 데이터 스트림은 각각 오디오(high), 비디오(medium), 백그라운드(low)의 트래픽 카테고리를 가진다.

표 2는 각 AC의 파라미터를 나타낸다. 우선순위가 높은 오디오는 64kb/s의 전송률, 160byte의 패킷 사이

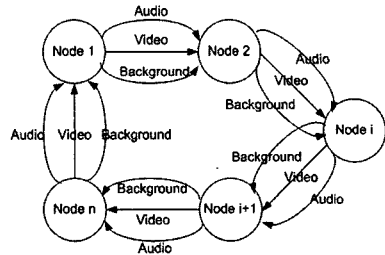


그림 4 실험 토폴로지

표 2 각 AC의 MAC 파라미터

Parameters	High	Medium	Low
CWmin	7	15	31
CWmax	200	500	1023
AIFS(μs)	34	43	52
PF	2	2	2
Packet size (byte)	160	1280	200
Packet Interval (ms)	20	10	12.5
Sending rate (kb/s)	64	1024	128

즈 그리고 20ms의 패킷 간격으로 설정하며, 중간 우선순위를 가지는 비디오는 1024kb/s의 전송률, 1280byte의 패킷 사이즈 그리고 10ms의 패킷 간격을 설정한다. 그리고 가장 낮은 순위를 가지는 백그라운드는 128kb/s의 전송률, 200byte의 패킷 사이즈 그리고 12.5ms의 패킷 간격으로 설정한다.

smoothing factor  $\alpha$ 를 구하기 위해 스테이션의 수를 20개로 두고,  $T_{update}$  주기를 8000 time slot으로 고정하여  $\alpha$  값을 0과 1사이의 범위로 실험한 결과 0.1이 가장 높은 성능을 나타내어, 본 실험에서는  $\alpha$ 를 0.1로 하여 실험하였다.  $T_{update}$  주기를 구하기 위해 스테이션의 수를 20개로 두고, smoothing factor를 0.1로 고정하여 1000 time slot과 35000 time slot 사이의 범위로 실험한 결과 1000 time slot이 가장 높은 성능을 나타내어 본 실험에서는  $T_{update}$  주기를 1000 time slot으로 하여 실험하였다.

그림 5는 CWminAS와 ACATICT의 클래스별 데이터 전송 시 전송 성공한 패킷 수를 보여준다. 그림 5에 의하면 ACATICT가 클래스별 충돌 확률을 계산함으로써 적용적 CWmin 조절 방식에 비하여 각 클래스별로 더 많은 데이터를 성공적으로 전송하였다.

그림 6은 CWminAS와 ACATICT의 효율을 보여준다. 그림 6에 의하면 ACATICT가 CWminAS보다 스테이션의 수가 25개일 때, 효율이 최대 10%의 향상을 보이며, 평균 약 3%가 향상된 것을 보여준다.

그림 7은 CWminAS와 ACATICT의 goodput을 보여준다. 그림 7에 의하면 ACATICT가 CWminAS보다

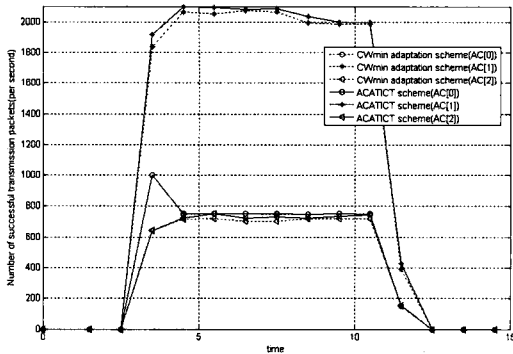


그림 5 데이터 전송 시 전송 성공한 패킷 수

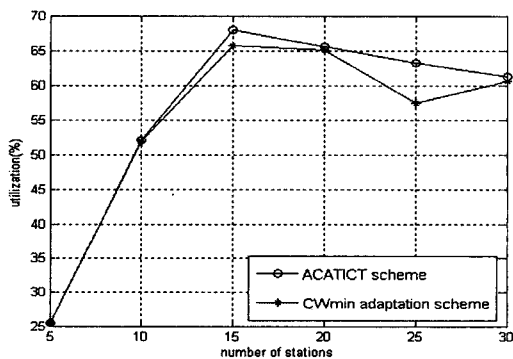


그림 6 CWminAS와 ACATICT의 효율

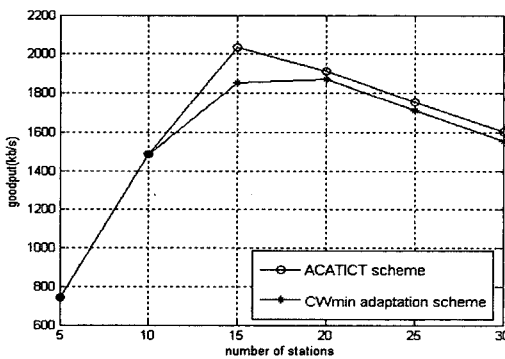


그림 7 CWminAS와 ACATICT의 goodput

스테이션의 수가 15개일 때, 최대 약 10%의 향상을 보이며, 평균 약 3%의 성능이 향상된 것을 보여준다. 참고로 goodput은 수신측에 성공적으로 전송된 실제 데이터 양을 의미하며, 바이트 단위의 수신 데이터의 총 양을 전체 시뮬레이션 시간으로 나누어 구한다.

### 6. 결론 및 향후 연구 과제

본 논문에서는 AC별 다른 파라미터 값으로 인해 발

생되는 클래스별 충돌 확률을 고려하여 계산된 충돌 확률 값으로 CWmin 값을 조절한다. ACATICT와 CWminAS의 효율과 goodput을 비교하였다. 실험에 의하면 평균적으로 ACATICT가 CWminAS보다 효율과 goodput이 각각 약 3%, 3% 향상되었다.

향후 연구 과제로는 AIFS의 길이의 차이가 성능에 미치는 영향에 대하여 분석하고, 더 나아가 네트워크 환경에 따라 CWmax 값을 적용적으로 조절하여 CWmax 값이 성능에 미치는 영향에 대하여 살펴볼 것이다.

### 참고 문헌

- [1] Qiang Ni, Lamia Romdhani, Thierry Turletti, "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN," Journal of Wireless Communications and Mobile Computing, Wiley, 2004: Volume 4, Issue 5: pp. 547-566.
- [2] IEEE Std 802.11e; Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications; Amendment: Medium Access Control (MAC) Quality of Service Enhancements, IEEE Std 802.11e-2005.
- [3] Vasilios A. Siris, George Stamatakis, "Optimal CWmin Selection for Achieving Proportional Fairness in Multi-Rate 802.11e WLANs: Test-bed Implementation and Evaluation," WiNTECH'06,
- [4] Lassaad Gannoune and Stephan Robert. "Dynamic Tuning of the Contention Window Minimum (CWmin) for Enhanced Service Differentiation in IEEE 802.11 Wireless Ad-Hoc Networks," IEEE PIMRC 2004.
- [5] Mounir Frikha, Fatma Ben Said, Lilia Maalej, Faiza Tabbana, "Enhancing IEEE 802.11e standard in congested environments," AICT/ICIW 2006.
- [6] Kyoung-Ju Noh, Woo-Yong Choi, Sok-Kyu Lee, "Adaptive and Dynamic Tuning of the Operation Parameter Value for QoS and Fairness in Wireless LAN," IEEE 2004.
- [7] Haitao Wu, Xin Wang, Qian Zhang, Xuemin Shen, "IEEE 802.11e Enhanced Distributed Channel Access(EDCA) Throughput Analysis," ICC 2006. Vol.1, June 2006.
- [8] <http://www.isi.edu/nsnam/ns>
- [9] <http://www-sop.inria.fr/planete/qni/Research.html>
- [10] Lamia Romdhani, Qiang Ni, and Thierry Turletti, "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks," IEEE 2003.