

논문 2010-47TC-1-1

IEEE 802.11e HCCA 기반의 실시간 트래픽을 위한 스케줄링 알고리즘

(A Scheduling Algorithm for Real-Time Traffic in IEEE802.11e HCCA)

정진우*, 김종호**

(Jinoo Joung and Jongho Kim)

요약

본 논문에서는 멀티미디어 트래픽에 대한 성능 향상을 목표로 ASR-DRR, ASD-DRR이라 명명한 IEEE 802.11e HCCA기반에서 서비스 품질 보장이 가능한 스케줄링 알고리즘을 제안하였다. 또한 IEEE 802.11e HCCA(Hybrid Coordination Function Controlled Channel Access) 스케줄러와 관련된 다수의 개선된 형태의 스케줄러들이 실제 큐 사이즈 정보를 실시간으로 HC(Hybrid Coordinator)에 전달하지 못하여 생기는 불필요한 지연시간에 대해 지적하였다. 이러한 문제점을 해결하기 위해서 현재 큐 상태를 최대한 실시간으로 전달할 수 있는 간단한 해결책을 제안하였으며, 이를 바탕으로 두 가지 구현방안을 제시하였다. 이를 RTS(Request to Send)/CTS(Clear to Send)를 활용한 적응형 스케줄러(ASR)와 Data/Ack를 활용한 적응형 스케줄러(ASD)라고 한다. 이들 구현 방안들은 유선환경에서의 DRR(Deficit Round Robin) 스케줄러를 모방하여 최적화되었다. 또한 제안하는 스케줄러의 성능을 알아보기 위해 NS-2 시뮬레이터를 활용하여 기존 스케줄러와 비교한 결과 훨씬 좋은 성능을 보임을 확인하였다.

Abstract

In this paper we propose a scheduling algorithm for supporting Quality of Service(QoS) in IEEE 802.11e HCCA referred to as ASR-DRR and ASD-DRR, which aims at providing improved performance for the support of multimedia traffic. Although We identify the problem of the current IEEE 802.11e HCCA (Hybrid Coordination Function Controlled Channel Access) scheduler and its numerous variations, that the queue information cannot be notified to the Hybrid Coordinator (HC) timely, therefore the uplink delay lengthens unnecessarily. We suggests a simple solution and a couple of implementation practices, namely the Adaptive Scheduler with RTS/CTS (ASR) and Adaptive Scheduler with Data/Ack (ASD). They are both further elaborated to emulate the Deficit Round Robin (DRR) scheduler. They are also compared with existing exemplary schedulers through simulation, and shown to perform well.

Keywords : WLAN, HCCA, TXOP, DRR, Wireless DRR

I. 서론

무선랜(Wireless LAN or WLAN) 기술은 표준화와

시장에서의 성공을 통해 빠른 속도로 발전되었고, 사용자 수는 폭발적인 증가하였다. 이로 인해 무선랜 환경에서 오디오/비디오 스트리밍, 네트워크 게임, 원격회의와 같은 다양한 멀티미디어 응용이 효과적으로 지원될 것 기대하는 요구가 크게 증가하였다. 이러한 멀티미디어 실시간 응용을 원활하게 제공하기 위해서 높은 대역폭과 낮은 지연을 요구되고, 이를 위해 QoS(Quality of Service)의 지원이 필수적이다.

하지만 현재 표준화가 완료된 IEEE 802.11는 QoS를 지원하지 않도록 설계되었다^[1]. 이를 위해 개선된 무선

* 정회원, 상명대학교 컴퓨터과학부
(Sangmyung University)

** 정회원, (주)에어코드
(AIRCODE. Co.)

※ 이 논문 또는 저서는 2008년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(KRF-2008-331-D00495)

접수일자: 2008년9월29일, 수정완료일: 2010년1월18일

랜 기술인 IEEE 802.11e에서는 기존의 IEEE 802.11을 기반으로 단순히 효율을 높이는 것뿐만 아니라 QoS의 지원을 목적으로 하고 있다^[2].

IEEE 802.11e는 IEEE 802.11의 DCF(Distributed Coordination Function)와 PCF(Point Coordination Function) 프로토콜을 기반으로 하는 두 개의 MAC 매커니즘을 통해 채널에 접근하는 HCF(Hybrid Coordination Function)를 소개하였다. 구체적으로 경쟁 기반에서 트래픽의 상대적 우선순위를 활용하는 EDCA(Enhanced Distributed Channel Access)와 비경쟁기반에서 트래픽의 특성을 나타낸 파라미터를 활용하는 HCCA(HCF Controlled Channel Access)를 통해 QoS를 보장하였다.

또한 대부분의 멀티미디어 응용들은 느슨한 혹은 상대적 QoS 제공 보다는 엄격한 QoS 보장을 요구하고 있다. HCCA 모드는 HC(Hybrid Coordinator)에 의해 스테이션에서 생성되는 트래픽 스트림의 전송기회를 수락/제어하는 Polling 방식의 엄격한 QoS 보장을 추구한다. Polling 기반의 HC에서는 사전에 약속된 트래픽 스트림의 QoS 보장을 위해 적절한 전송기회를 할당하는 스케줄러를 가져야하고, 요구되는 QoS를 보장할 수 없다면 해당 트래픽 스트림의 전송을 제한하는 수락/제어 기능을 제공해야 한다. 하지만 표준에서 제안된 스케줄러는 규칙적으로 생성되는 CBR(Constant Bit Rate) 트래픽에는 좋은 성능을 보이지만 불규칙적으로 생성되는 VBR(Variable Bit Rate) 트래픽에 대해서는 만족할 만한 성능을 보여주지 못한다^[3]. VBR 트래픽의 요구의 증가와 그에 대한 QoS 보장이 어렵다는 점에서 표준 Reference 스케줄러의 한계점을 확인할 수 있다.

이 문제를 개선하기 위해 다양한 해결책이 제시되었다. 하지만 이전에 제시된 대부분의 접근 방법은 이전 슬롯에서 스테이션의 큐 사이즈 정보를 획득하고, 현재의 슬롯에 예상되는 큐 사이즈를 추정하여 TXOP을 할당하는 것이다. 이와 같은 방법 또한 lag 문제를 발생시킨다. lag는 스테이션의 큐 상태를 HC가 즉시 확인할 수 있는 것이 아니라 약간의 시간 뒤에 확인할 수 있을 때 발생하는 정보의 부정확성 말하며 SI 크기와 밀접하게 연관되어 있다. 또한 데이터 전송 지연은 lag에 비례적으로 증가할 것이다. 그러므로 실시간, 멀티미디어 트래픽의 요구가 증가되는 환경에서는 해당 트래픽의 특성을 고려하여 전송기회를 유동적으로 적용할 수 있는 스케줄러 개발이 필요하다.

따라서 본 논문에서는 간단한 아이디어를 통해 lag를 줄이기 위한 두 가지의 방안을 제안 할 것이다. 첫 번째는 RTS(Request to Send)/CTS(Clear to Send) 프레임을 활용한 방안이고, 두 번째는 Data/Ack 프레임을 활용한 방안이다. 구체적인 동작 과정은 III장에서 설명하도록 한다.

본 논문 구성을 살펴보면, II장에서는 IEEE 802.11e Reference 스케줄러와 기존에 제안된 해결방안에 대해 설명하고, III장에서는 앞에서 언급한 문제점을 해결하기 위해 새로운 스케줄러를 제안한다. IV장에서는 제안한 스케줄러와 Reference 스케줄러의 성능을 비교 분석하고, 마지막 V장에서는 결론으로 논문을 마무리한다.

II. IEEE 802.11e HCCA와 개선된 스케줄러

본 장에서는 IEEE 802.11e HCCA의 Reference 스케줄러의 구체적인 동작방법을 포함한 HCCA 대해서 알아본다. 또한 앞에서 언급한 Reference 스케줄러의 문제점을 개선하기 위해 제안된 몇 가지 스케줄러에 대해 알아본다.

1. IEEE 802.11e HCCA

IEEE 802.11e의 HCCA는 PCF 기반의 개선된 형태이고, 스테이션이 전송한 TSPEC 정보를 기반으로 각각의 스테이션에게 전송할 수 있는 시간(TXOP)을 할당하는 Polling 매커니즘을 사용한다.

HCCA 모드에서 TXOP 할당은 전체 HCCA 동작과정에서 매우 중요한 부분이다. Reference 스케줄러의 TXOP 할당 방법을 살펴보면, HC는 TXOP을 요청하는 스테이션의 TSPEC을 전달 받고 TSPEC에 포함된 Maximum SI, Mean Data Rate, Maximum MSDU Size, Nominal MSDU Size를 기반으로 TXOP를 할당한다. 구체적으로 HC는 스테이션에게 TXOP을 할당하기 위해 두 단계를 거친다. 우선 SI를 결정하고, 두 번째로 계산된 SI를 사용 하여 각각의 스테이션의 별로 TXOP을 계산하는 것이다. SI는 HCCA 모드에서 전송을 희망하는 모든 TS(Traffic Stream)의 MSI(Maximum Service Interval)를 활용하여 계산한다. SI는 Beacon Interval의 약수 중 각각의 스테이션으로부터 전달받은 MSI의 가장 작은 값 이하로 SI로 결정한다. 예를 들어 MSI1=15ms MSI2=20ms일 때, beacon interval은 100ms이라고 하자. 여기서 SI는 Beacon

Interval의 약수이면서 MSI 보다 작은 10ms로 결정된다. 다음에는 앞에서 계산한 SI 값을 활용하여 SI 동안 전송될 MSDU의 개수를 추정한다.

$$N_i = \left\lceil \frac{SI \cdot \rho_i}{L_i} \right\rceil \quad (1)$$

위의 공식에 따라서 앞에서 계산한 SI와 ρ_i (Mean Data Rate), L_i (Nominal MSDU Size)를 이용하여 SI 동안 전송할 수 있는 MSDU의 개수를 계산한다.

$$TXOP_i = \max\left[\frac{N_i \cdot L_i}{R_i} + O, \frac{M}{R_i} + O\right] \quad (2)$$

마지막으로 이전에 계산한 N_i 과 ρ_i (Mean Data Rate), L_i (Nominal MSDU Size), M (Maximum MSDU Size)을 이용하여 스테이션에 할당하는 TXOP를 결정하게 된다. R 은 최소물리전송률이고 O 는 IFS(Inter Frame Space)와 Ack 또는 CF-Poll의 오버헤드이다. 최종적으로 새롭게 생성되는 TS의 전송의 가능여부를 판단한다. 그 방법은 (3)과 같다.

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=1}^k \frac{TXOP_i}{SI} \leq \frac{T - T_{cp}}{T} \quad (3)$$

그 방법은 이전에 누적 할당된 TXOP의 합과 이번 에 할당된 TXOP의 합하여 SI 이내의 전송 가능여부를 판단하여 수락/제어를 결정한다. 여기서 T_{cp} 는 EDCA에 사용되는 시간이다. (3)에 적합하다면 TXOP을 할당 받지만 적합하지 않다면 스테이션은 TXOP을 할당 받지 못하고 HCCA 모드에서 데이터 전송이 허가되지 않는다.

Reference 스케줄러의 TXOP 할당방법은 TS 생성 초기에 HC에 전달한 TSPEC에 기반으로 TXOP을 할당하고, 그 이후에 변화하는 어떠한 상태 정보도 활용하지 않는다. 그렇기 때문에 VBR 트래픽과 같이 SI 마다 트래픽의 생성되는 정도가 가변적인 트래픽에는 부적합하다. 그렇기 때문에 QoS 보장 가능한 스케줄러는 임의의 간격으로 일정하지 않는 길이로 생성된 트래픽을 스케줄 할 수 있어야 하며, 각각의 스테이션의 실제 트래픽 양을 기반으로 TXOP이 할당되어야 한다.

2. 개선된 스케줄러

ARROW^[4-5]는 가변적으로 생성되는 트래픽에 적합

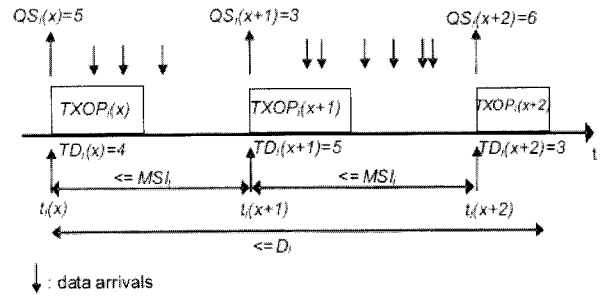


그림 1. ARROW 스케줄러
Fig. 1. ARROW Scheduler.

한 해결책 중 하나이다. ARROW에서 스테이션은 각각의 QoS-Data 프레임의 QoS 컨트롤 필드를 통해 현재 스테이션의 큐에 담겨있는 데이터의 양을 HC에게 알려준다. 이러한 정보를 받으면 스케줄러는 다음번 TXOP 할당 시 이전에 남아있는 데이터를 고려한 충분한 TXOP을 할당하여 스테이션의 요구를 만족시킨다. 그림 1은 ARROW 스케줄러의 아이디어를 명확히 나타낸다.

다른 접근 방안은 SI를 변화시켜 네트워크 환경에 적용하는 것이다^[6-7]. 이 방안에서는 스테이션이 동적으로 mSI(Minimum SI) & MSI(Maximum SI)를 재설정하고, 스테이션은 재설정된 SI 값을 HC에 알려준다. HC는 SI를 재계산 후 변화된 SI 사용하여 서비스를 수행하게 된다. 이 접근 방안은 긴급하거나 임박한 프레임에게 빠른 서비스를 제공할 수 있으나 SI의 다시 계산하는 과정이 복잡하고 많은 시간을 소비하는 프로세스가 될 수 있다.

PRO-HCCA 스케줄러^[8]은 스테이션의 요청과 HC의 응답 사이의 시간 간격의 최소화를 목적으로 한 접근 방안이다. PRO-HCCA는 두 개의 아이디어의 조합으로 설계되었다. 첫 번째는 각기 다른 delay bound의 계산이다. TS의 패킷은 'degree of delay'를 나타낸 기록을 가지고 있다. TS은 degree of delay에 의해 구분할 수 있다. 이것은 SI를 변화시키는 접근 방안과 비슷하다. 두 번째 이전의 정보를 이용하여 VBR 트래픽에서 생성 비율이 높은 지점을 예측을 하는 것이다. PRO-HCCA 스케줄러는 Feedback 기반의 접근 방안이라고 할 수 있다.

Higuchi et. Al^[9]는 잘 알려진 LR server의 개념과 leaky-bucket을 기반으로 제작한 HCCA uplink 스케줄러를 소개하였다. 여기서는 사용되지 않은 TXOP을 QoS-Null 프레임에 HC에 전송하므로써 반환하고, 반

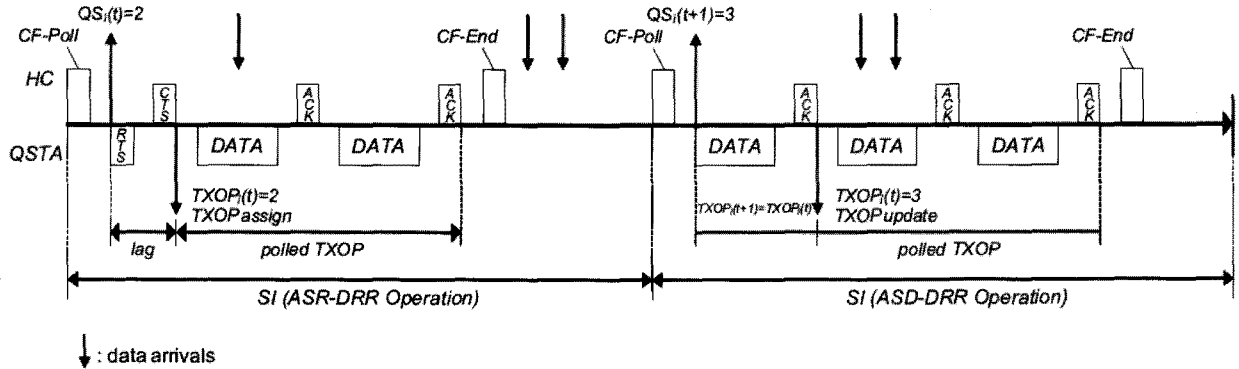


그림 2. ASR-DRR 과 ASD-DRR의 TXOP 할당 과정
Fig. 2. TXOP assignment with ASR-DRR and ASD-DRR.

환된 TXOP 시간은 EDCA 메커니즘을 사용하도록 하였다. 지연속도는 delay bound가 계산 될 수 있도록 LR 서버모델을 사용하였다. 이 방안으로 SI 길이의 최적화가 가능하다는 것을 주장한다.

마지막으로 특허로 발표된 흥미로운 접근방안이 있다^[10]. 스케줄러는 EDF(Earliest Deadline First)에서 착안하였다. 스테이션의 Flow m 은 mean data rate (R_m)과 평균 패킷크기 S_m , 지연 요구 δm 과 함께 TSPEC을 제공한다. t_{cm} 은 일반적으로 큐의 맨 앞에 있는 패킷이 생성되는 시간으로 정의된다. 마감시한에 도착하면, TXOP은 $(t_{cm} + \delta m)$ 보다 먼저 받아야 하고, $minagem$ 과 $maxagem$ 은 TXOP을 받을 때 가장 오래된 패킷에 aging한다. Flow는 $minagem$ 보다 빠르거나 $maxagem$ 보다 늦으면 안 된다. $Maxagem$ 이 $(t_{cm} + \delta m)$ 로 정의 될 때 $minagem$ 은 Flow에서 다음번 TXOP 기간에 전송될 패킷을 수집하길 원하는 시간이다. AP는 부여할 TXOP 결정한다. 또한 t_{cm} , $minagem$ 그리고 $maxagem$ 은 의 결정한다. t_{cm} , $minagem$, $maxagem$ 은 의사결정에 사용되는 중요한 파라미터이다. 그러나 핵심 아이디어는 것은 TXOP을 선택되는 순간에 큐에 있는 패킷의 개수(N_{qp})를 추정하는 것이다. N_{qp} 는 아래의 공식에 따라서 추정할 수 있다.

$$N_{pq} = INT\left(\frac{(t - t_{cm}) \cdot R_m}{S_m}\right) + 1 \quad (4)$$

여기서 S_m 은 Flow m 의 평균 패킷 크기이고, t_{cm} 은 큐에 남은 패킷의 개수에 의해 추정된다.

앞에서 언급한 몇 가지의 개선된 스케줄러들은 TXOP을 할당하기 위해 이전 슬롯($t-1$)의 정보를 활

용한다. 그러나 실제 TXOP 할당은 현재 슬롯인 (t)에 일어난다. 여기서 실제 TXOP을 할당하는 시간과 활용하는 정보의 차이가 있음을 확인할 수 있다. 여전히 lag 문제를 해결하지 못한 상황에서 TXOP이 할당되는 것이다. 앞에서 언급한 문제에 대한 해결 방안을 III장에서 제안하는 스케줄러를 통해 소개한다.

III. 적응형 스케줄러

III장에서는 기존의 스케줄러에서 지적된 VBR 트래픽에 대한 QoS 보장 위해 새로운 형태의 적응형 스케줄러를 제안한다. 제안하는 적응형 스케줄러는 간단한 아이디어를 통해 lag를 획기적으로 줄일 수 있다. 또한 스테이션이 요구하는 정확한 TXOP 할당이 가능하고, 더 많은 스테이션에게 전송기회를 보장하므로 네트워크 자원 사용의 효율성을 높였다. 다음 절에서는 제안하는 적응형 스케줄러의 기본 동작 원리에 대해서 설명하고 스케줄러의 구체적인 TXOP 할당 알고리즘에 대해 설명한다.

1. 적응형 스케줄러의 기본 동작 원리

IEEE802.11e HCCA uplink시스템을 살펴보면 Wired 스위치 내부에 output port와 유사한 output queue와 스케줄러의 조합으로 구성된 server 형태라고 할 수 있다. 하지만 wired 와 wireless의 차이는 wired 환경에서는 스케줄러가 항상 큐의 상태를 알 수 있지만 wireless 환경에서는 큐의 상태 실시간으로 알 수 없고, 일정한 시간이 간격 뒤에 알게 된다. 이와 같은 시간 간격을 lag라고 한다. HCCA 모드에서의 lag는 SI라고 할 수 있다. 제안하는 적응형 스케줄러는 CFP 또는 CAP 구

간 내에서의 선택적 오퍼레이션인 RTS/CTS 또는 Data/Ack 프레임을 활용하여 최소화된 lag를 가지는 것을 기반으로 한다.

앞에서 언급한 아이디어를 통해 최소한 두 가지 방법으로 적응형 스케줄러를 구현할 수 있다. 첫 번째 구현 방법은 RTS/CTS를 활용한 적응형 스케줄러(ASR)라고 부르고, RTS/CTS lag를 가지게 된다. 구체적으로 스테이션에서 RTS 프레임을 통해 스테이션의 큐 사이즈를 알리고, HC는 이를 확인하고 스테이션의 요청이 반영된 TXOP을 CTS 프레임을 통해 전달하게 된다. 이와 같은 방법을 활용하면 기존의 표준에서 정의된 일반적인 lag인 SI와 비교하면 무시할 수 있을 정도인 RTS/CTS 전송 시간 만큼의 lag로 가지게 된다. 두 번째 구현 방법은 Data/Ack를 활용한 적응형 스케줄러(ASD)라고 부르고, 아주 작은 lag를 가지게 된다. 또한 표준에서 정의된 동작을 그대로 활용가능하고, 약간의 응용만으로 구현이 가능하다. 동작 방법은 스테이션에게 TXOP을 알려주고 Poll을 받은 스테이션은 전송하고자 하는 데이터와 QoS-Data 프레임의 QoS 컨트를 필드에 큐 사이즈를 넣어서 HC에 전송한다. HC는 전달받은 큐 사이즈 정보를 기반으로 TXOP을 재설정 후 Ack 프레임을 통해 스테이션의 TXOP을 전달하게 된다. 하지만 Block-Ack을 사용할 경우 올바르게 동작하지 않는다.

제안하는 스케줄링 알고리즘의 스테이션이 전송하고자 하는 트래픽의 양을 실시간으로 정확히 파악한다는 데 중점을 두었다. 제안하는 스케줄러에서는 정확한 트래픽양, 즉 TXOP를 할당하는 시점에서 스테이션의 정확한 큐 사이즈를 HC가 인지하므로 정확한 TXOP을 할당할 수 있다. 그림 2는 RTS/CTS를 활용한 적응형 스케줄러(ASR)와 Data/Ack를 활용한 적응형 스케줄러(ASD)를 나타내었다. 또한 제안하는 스케줄러는 유선 환경에서의 성능을 검증 받은 DRR(Deficit Round Robin) 스케줄러^[11]를 모방하여 복잡성을 최소화하고 fairness를 최적화하였다.

2. 적응형 스케줄링 알고리즘

제안하는 적응형 스케줄러는 IEEE802.11e HCCA 모드의 무선 환경에서 DRR 알고리즘을 적용하여 uplink에 대한 스케줄링을 고려하였다. 기본적으로 HCCA 모드와 동일하게 스테이션들은 트래픽에 대한 정보를 TSPEC을 통해서 HC에게 전달하고, HC는 스테이션이

전달한 TSPEC에 포함된 평균 전송률을 기반으로 스테이션의 quantum size를 할당한다. quantum size는 최대 MSDU 보다 크게 할당하도록 한다. 만약 RTS 프레임을 통해 스테이션으로부터 전송받은 스테이션의 큐 크기가 0이라면 deficit count $D_i(t)$ 는 0으로 리셋되고, HC는 CTS 프레임을 전송하는 대신에 다른 스테이션에 poll을 전달하게 된다. 그렇지 않다면 AP는 이전의 SI의 deficit count($D_i(t-1)$)을 기반으로 TXOP을 재계산 후, 그 결과를 CTS 프레임에 포함시켜 스테이션에게 알려준다. 또한 재할당하는 TXOP은 스테이션의 큐의 양이나 해당 트래픽 스트림의 MBS(Maximum Burst Size)를 넘을 수 없다. 공식화하면 아래와 같다.

$$Q_i(t) = \min[\phi_i + D_i(t-1), q_i(t), MBS_i] \quad (5)$$

(5)를 통해 재할당되는 TXOP 최대 값을 제한하므로 일부 스테이션의 채널 독점을 방지할 수 있다.

$$N_i(t) = \lceil \frac{Q_i(t)}{L_i} \rceil \quad (6)$$

그리고 (6)과 같이 L_i (Nominal MSDU Size)를 이용하여 패킷의 개수($N_i(t)$)를 추정한다.

$$TXOP_i(t) = \frac{Q_i(t)}{R_i} + O_D + O_O, \quad (7)$$

$$O_D = N_i(t) \cdot \left[2 \cdot SIFS + \frac{M_{Ack}}{R_i} \right],$$

$$O_O = 2 \cdot SIFS + \left(\frac{M_{RTS} + M_{CTS}}{R_i} \right)$$

(7)에서는 (5),(6)을 기반으로 TXOP을 할당한다. 여기서 R_i 은 최소물리전송률이고, O_D 는 데이터 전송 과정에서, O_O 는 ASR-DRR 스케줄러 동작 시 생성되는 오버헤드를 나타낸다. ASD-DRR 스케줄러 생성되지 않는다. 구체적으로 (7)를 통해 확인할 수 있다. (7)에서 M_{Ack} 는 Ack 프레임의 크기를 M_{RTS} , M_{CTS} 는 각각 RTS와 CTS 프레임의 크기를 나타낸다.

$$D_i(t) = \phi_i + D_i(t-1) - TXOP_i(t-1) \quad (8)$$

마지막으로 HC는 deficit count($D_i(t)$)를 (8)과 같이 재설정 한다.

두 가지 구현 방안인 ASR-DRR, ASD-DRR 모두 동일한 TXOP 할당 방법을 사용한다. 하지만 스테이션의

큐 상태를 알아오는 과정에 따라 ASR-DRR 스케줄러는 동작 시 RTS/CTS 전송 시간만큼의 오버헤드를 가지게 되지만, ASD-DRR 스케줄러는 어떠한 오버헤드도 발생시키지 않는다.

Algorithm 1: ASR-DRR and ASD-DRR

Require: Confirm current queue condition through RTS or Data frame

```

1:  $D_i \leftarrow 0$ 
2:  $Q_i \leftarrow \alpha \cdot [(MEAN\ RATE_i \cdot SI) + MAC\ Header\ Size]$ 
3:  $MBS_i \leftarrow MAX\ BURST\ SIZE_i$ 
Ensure: CTS or Ack frame notify updated  $TXOP_i(t)$  to polled Station
4: if queue is not empty then
5:    $Q_i(t) \leftarrow \min[\phi_i + D_i(t-1), q_i(t), MBS_i]$ 
6:    $TXOP_i(t) = \frac{Q_i(t)}{R_i} + O_D + O_O$ 
7: else
8:    $D_i(t) \leftarrow 0$ 
9:    $TXOP_i(t) \leftarrow O_O$ 
10: end if
11: for  $i=0$  to  $n$  do
12:    $TXOP\ SUM \leftarrow TXOP\ SUM + TXOP_i(t)$ 
13:    $i \leftarrow i+1$ 
14: end for
15: if  $TXOP\ SUM > SI$  then
16:    $TXOP_i(t) \leftarrow TXOP_i(t-1)$ 
17: end if
18:  $D_i(t) = D_i(t-1) + \phi_i - TXOP_i(t-1)$ 

```

알고리즘 1은 ASR-DRR과 ASD-DRR 스케줄러의 구체적인 동작 과정을 나타낸다. 스케줄링 알고리즘 동작 이전에 사용되는 모든 파라미터의 초기화를 필요하고, 또한 Quantum factor α 는 항상 1보다 크게 설정해야한다. 그 이유는 SI 기간 내에 하나의 MSDU 전송을 보장하기 위해 최대 MSDU 크기 이상의 Quantum Size 할당이 필요하기 때문이다.

IV. 적응형 스케줄러의 성능 평가

이번 장에서는 제안하는 스케줄러의 성능 평가를 위해서 수행한 시뮬레이션 결과를 설명하였다. 성능 향상을 비교하기 위해 다양한 트래픽에 전송되는 환경에서 HCCA의 reference 스케줄러와 비교하였다.

1. 시뮬레이션 환경 및 가정

시뮬레이션은 Network Simulator 2(NS-2)^[12]를 사용하였고, 구체적으로 ns-allinone-2.29.3 package에 HCCA를 구현한 ns2hcca patch를 적용하였다. ns2hcca는 draft amendment 802.11e/D13.0 을 기반으로 구현되

표 1. 시뮬레이션 파라미터

Table 1. Parameter values used in the simulation.

Parameter	Value
SlotTime	0.000020s (20us)
SIFS	0.000010s (10us)
PreambleLength	144bit
PLCPHeaderLength	48bit
PLCPDataRate	1.0e6 (1Mbps)
dataRate	11.0e6 (11Mbps)
basicRate	1.0e6 (1Mbps)

었다. 시뮬레이션에서는 uplink 스케줄링만을 고려하였고, 토폴로지는 하나의 HC와 다수의 스테이션으로 구성하였다. 또한 스케줄러의 직접적인 스케줄링 성능 비교를 위해서 admission control은 수행하지 않았다. 스테이션에서 전송되는 Flow는 대표적인 Voice 트래픽인 G.711을 모방한 CBR 트래픽과 Video 트래픽인 H.261, H.263 코덱으로 인코딩된 VBR 트래픽을 사용하였다. VBR 트래픽은 real movie clip(The Firm, Starship Troopers)^[13]을 활용하였고, UDP 프로토콜을 이용하여 전송하였다. 생성하는 Flow의 평균 전송률은 각각 64Kbps, 256Kbps, 256Kbps 이다. 추가적으로 생성하는 Flow가 동일한 패턴으로 생성되지 않게 하기 위해서 Flow의 생성 시간을 조금씩 다르게 설정하였다.

우선적으로 제안된 스케줄러에 최적화된 quantum size를 찾아내는 시도를 하였다. 일반적으로 Wired 네트워크에서의 DRR 스케줄러는 quantum size가 작을수록 fairness 한 성능을 나타내었으나 복잡성은 증가하였다. AP에 uplink flows를 스케줄 할 권한을 주고, quantum factor에 다양한 변화를 주어 최대 허용 가능한 스테이션의 수를 관찰하였다. 그 결과는 quantum factor가 2.5 또는 그 이상으로 설정하였을 때 가장 좋은 성능을 보여주었다. 이와 같은 결과를 바탕으로 MSDU 사이즈의 3배수로 Quantum size를 최적화 하였다.

표 1은 시뮬레이션에서 사용되는 주요한 변수의 설정 값을 나타내었다.

2. 시뮬레이션 결과 분석

이번 절에서는 다양한 트래픽 타입별 Reference 스케줄러와 제안하는 스케줄러(ASR-DRR, ASD-DRR)을 처리량과 평균 지연시간 측면에서 비교하였다.

그림 3의 결과를 분석하면 실험한 모든 트래픽에 대해서 제안하는 스케줄러가 기존의 Reference 스케줄러

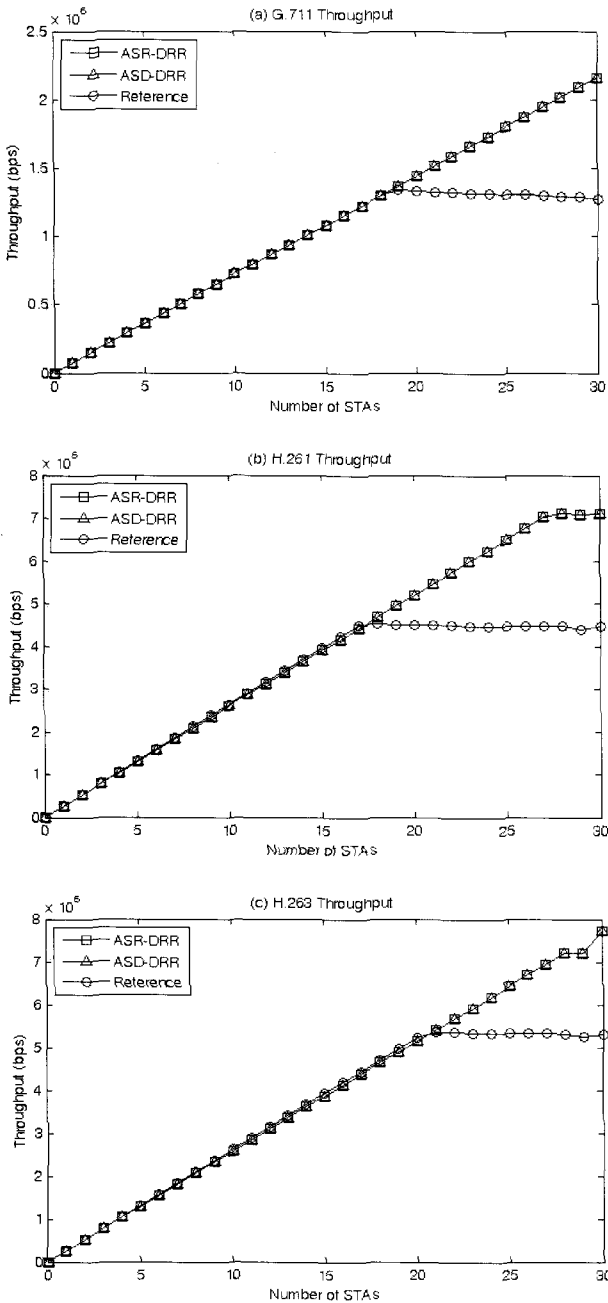


그림 3. 다양한 트래픽 타입에 따른 처리량
Fig. 3. Throughput per various traffic type.

에 비해 더 많은 양의 트래픽을 처리함으로써 더 많은 양의 스테이션의 데이터 전송이 가능함을 확인하였다. 이와 같은 결과는 제안하는 스케줄러가 무선 채널 환경에서 효율적으로 동작함을 확인할 수 있다.

다음으로 평균 지연시간 측면에서 제안하는 스케줄러의 성능을 평가하였다. 그 결과 그림 4와 같이 제안하는 스케줄러가 기존의 Reference 스케줄러에 더 많은 스테이션에게 작은 평균 지연시간을 보임을 확인하였다. Reference 스케줄러의 경우, 스테이션의 개수가 적

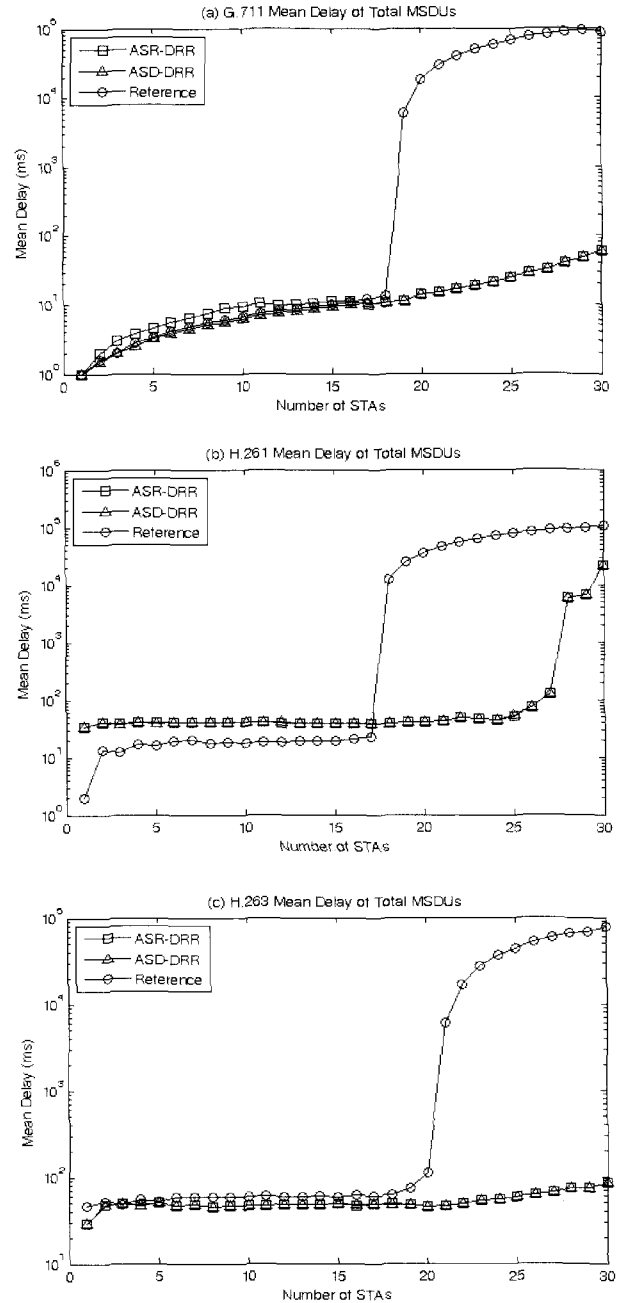


그림 4. 다양한 트래픽 타입에 따른 평균 지연시간
Fig. 4. Mean Delay of various traffic type.

은 상황에서는 좋거나 비슷한 지연시간을 보이지만, 일정 범위 이상의 스테이션이 생성되면 평균 지연시간이 기하급수적으로 커짐을 확인할 수 있다. 이 결과를 통해 제안하는 스케줄러를 사용할 경우 Reference 스케줄러에 비해 더 많은 스테이션에게 품질 보장된 서비스 제공이 가능하다는 것을 알 수 있다.

시뮬레이션을 통해 제안하는 스케줄러가 기존의 Reference 스케줄러에 비해 오디오/비디오 트래픽 전송에 훨씬 좋은 성능을 보임을 확인할 수 있다. 또한 제안

하는 스케줄러는 스테이션의 정확한 큐 사이즈 정보를 활용하여 불필요하거나 부족한 TXOP 할당 문제를 감소시켜 무선 채널의 효율적인 사용하고 있음을 확인하였다.

IV. 결 론

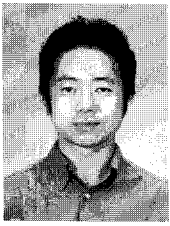
본 연구는 IEEE 802.11e HCCA 스케줄러의 취약점으로 지적된 VBR 트래픽의 QoS 문제에 접근하였다. 그 문제를 간략히 언급하면 스테이션의 큐 정보를 HC가 필요한 시간에 알아낼 수 없는 문제로 인해 uplink 딜레이가 불필요하게 길어졌던 문제를 밝혔고 그에 대한 대안으로 본 연구에서는 ASR-DRR 과 ASD-DRR이라는 간단한 솔루션과 두 가지의 구현 방안을 제안하였다. 또한 무선 uplink 스케줄러는 유선 네트워크를 위해 개발된 스케줄러(WFQ, DRR 등등)를 활용하였다. 기존에 발전된 많은 형태의 유선 네트워크 환경에서의 스케줄링 알고리즘을 활용하여 낮은 Delay와 Jitter를 기대할 수 있다. 제안하는 ASR-DRR은 IEEE 802.11e 표준에 근소한 수정만을 요구한다. ASD-DRR은 표준에 어떠한 수정도 필요하지 않고 WLAN 칩셋의 펌웨어 업그레이드만으로 구현 가능한 Wired-equivalent QoS 스케줄러를 제안하였다.

참 고 문 헌

- [1] IEEE Std. 802.11, "IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Nov. 1997.
- [2] IEEE Std. 802.11e-2005, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications Amendment 8: medium access control (MAC) quality of service enhancements," Nov. 2005.
- [3] M. M. Rashid, E. Hossain, and V. K. Bhargava, "Queueing Analysis of 802.11e HCCA with Variable Bit Rate Traffic," in Proc. IEEE ICC'06, Istanbul, Turkey, June 2006.
- [4] Dimitris Skyrianoglou, Nikos Passas, and Apostolis K. Salkintzis, "Support of IP QoS over Wireless LANs" Vehicular Technology Conference, 2004. VTC 2004-Spring.
- [5] Dimitris Skyrianoglou, Nikos Passas, and Apostolis K. Salkintzis, "ARROW: An Efficient Traffic Scheduling Algorithm for IEEE 802.11e

- HCCA", IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 5, NO. 12, DECEMBER 2006.
- [6] A. Grilo, M. Macedo, and M. Nunes, "A scheduling algorithm for QoS support in IEEE 802.11e networks," IEEE Wireless Commun. Mag., pp.36-43, June 2003.
- [7] Inanc Inan, Feyza Keceli, and Ender Ayanoglu, "An Adaptive Multimedia QoS Scheduler for 802.11e Wireless LANs" IEEE ICC 2006.
- [8] Mohammad M. Rashid, Ekram Hossain, and Vijay K. Bhargava, "HCCA Scheduler Design for Guaranteed QoS in IEEE 802.11e Based WLANs" IEEE WCNC 2007.
- [9] Yuhi Higuchi, Augusto Foronda, Chikara Ohta, Masahiko Yoshimoto, and Yoji Okada, "Delay Guarantee and Service Interval Optimization for HCCA in IEEE 802.11e WLANs" IEEE WCNC 2007.
- [10] M. Eiger, M. Elaoud, and R. Merera, (Toshiba America Inc. and Telcordia Inc.), "Latency-Aware Service Opportunity Window based (LASO) scheduling", Patent Application US 2007/0036116 A1, published Feb. 15, 2007.
- [11] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," IEEE/ACM Trans. Networking, vol. 4, no.3, pp. 375--385, June 1996.
- [12] IEEE 802.11e HCCA simulation using the Network Simulator 2. [Online] Available : <http://info.iet.unipi.it/~cng/ns2hcca/>
- [13] MPEG-4 and H.263 Video Traces for Network Performance Evaluation. [Online] Available : <http://www.tkn.tu-berlin.de/research/trace/trace.html>

저 자 소 개



정 진 우(정회원)
 1992년 KAIST 전기전자공학과
 공학사.
 1997년 Polytechnic Univ., NY,
 USA, 공학박사 (Ph.D.)
 1997년~2001년 삼성전자
 중앙연구소

2001년~2005년 삼성종합기술원
 2005년~현재 상명대학교 컴퓨터과학부
 <주관심분야 : 네트워크, 유무선통신, 임베디드
 시스템>



김 종 호(정회원)
 2007년 상명대학교 소프트웨어
 학과 졸업.
 2009년 상명대학교 컴퓨터과학과
 석사 졸업.
 2009년~현재 (주)에어코드

<주관심분야 : 방송·통신, 유무선통신, 임베디드
 시스템>