

IEEE 802.11 무선랜의 업링크와 다운링크간 공정성 향상을 위한 Contention Window 조절 알고리즘

정회원 임완선*, 김동욱*, 종신회원 서영주**°, 정회원 권동희***

A Contention Window Adjustment Algorithm for Improving Fairness between Uplink and Downlink in IEEE 802.11 WLANs

Wan-Seon Lim*, Dong-Wook Kim* *Regular Members*, Young-Joo Suh**° *Lifelong Member*,
Dong-Hee Kwon*** *Regular Member*

요약

본 논문은 IEEE 802.11 무선랜에서의 업링크와 다운링크 플로우간의 공정성 문제를 다룬다. 기존 연구에서는 AP의 최소 Contention Window (CWmin) 값을 단말의 CWmin 값보다 작게 설정하여 무선랜에서의 업링크와 다운링크 트래픽 간의 공평한 채널 사용을 보장하고자 하였다. 기존 논문의 해결책과 달리 본 논문에서 제안하는 알고리즘은 공정성 개선뿐만 아니라 동시에 시스템의 전체 throughput을 높일 수 있는 CWmin 값을 찾는 것을 목적으로 한다. 이를 위해 제안 알고리즘에서는 AP가 다운링크 플로우를 가지고 있는 단말의 수와 업링크 플로우를 가지고 있는 단말의 수를 실시간으로 확인하고 최적의 CWmin 값을 계산하여 단말에게 알린다. 실험 결과를 통해 제안 알고리즘이 단말간의 공평한 채널 사용을 보장함과 동시에 시스템의 throughput을 향상시키는 것을 확인할 수 있었다.

Key Words : IEEE 802.11, 무선랜, 공정성, MAC 프로토콜

ABSTRACT

This paper addresses the fairness issue between uplink and downlink traffic in IEEE 802.11 WLANs. Some solutions in existing work try to solve this issue by giving smaller minimum contention window (CWmin) value to an AP compared to stations. In contrast to the existing solutions, a proposed algorithm in this paper aims at finding CWmin values that not only provides fairness between uplink and downlink traffic among stations but also achieves high throughput. For this, in the proposed algorithm, an AP checks the number of stations that have uplink and downlink traffic, respectively. Based on this information, the AP calculates optimal CWmin values and announces it to stations. Our simulation results show that the proposed algorithm outperforms existing algorithms in terms of fairness and throughput.

I. 서론

IEEE 802.11 기반의 무선랜은 공공장소와 가정에서 널리 사용되고 있다. 이에 발맞추어 대부분의 노트

* 본 연구는 한국연구재단을 통해 교육과학기술부의 세계수준의 연구중심대학육성사업(WCU)과 (R31-2010-000-10100-0) 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업과 (2010-0024938) 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업과 (NIPA-2011-C1090-1131-0009) 2011년도 두뇌한국21사업의 지원을 받아 수행되었음.

* 포항공과대학교 컴퓨터공학과 모바일 네트워크 연구실({kiki, shkim80}@postech.ac.kr)

** 포항공과대학교 컴퓨터공학과/정보전자융합공학부 yjsuh@postech.ac.kr, (° : 교신저자), *** SK 텔레콤 (donghee.kwon@sk.com) 논문번호: KICS-2010-11-556, 접수일자: 2010년 11월 12일, 최종논문접수일자: 2011년 4월 6일

북, PDA, 스마트폰 등의 단말들은 무선랜 인터페이스를 기본으로 장착하여 출시되고 있다. 비록 이동통신망과 WiMAX 등 다양한 무선 접속 기술들이 발전하고 있지만 빠른 속도와 저렴한 비용이라는 장점으로 인해 무선랜은 향후에도 널리 쓰일 것으로 전망된다.

IEEE 802.11 표준에서는 distributed coordination function (DCF)와 point coordination function (PCF) 두 가지의 채널 접근 방식이 정의되어 있다^[1]. 현재 대부분의 무선랜 장비들은 구현이 용이한 DCF 모드로 동작하고 있다. DCF는 carrier sense multiple access with collision avoidance (CSMA/CA)를 기반으로 하고 있으며 모든 단말들이 동등한 채널 접속 기회를 가지도록 설계되어 있다. 하지만 DCF 기반의 무선랜에서는 업링크와 다운링크 플로우간의 채널 사용 불균형 문제가 존재하며, 이는 VoIP, P2P 등의 어플리케이션 사용시 심각한 성능 저하를 가져온다. 이러한 불균형 문제는 다운링크 트래픽이 집중되는 AP가 다른 단말과 동등한 채널 접속 기회를 가지기 때문에 발생한다.

이를 해결하기 위해서 다양한 해결책들이 제시되었는데, 그 중 대표적인 접근 방식은 AP의 최소 Contention Window (CW_{min}) 값을 단말의 CW_{min} 값보다 상대적으로 작게 설정하여 AP의 채널 접속 기회를 확보하는 것이다^[2,3]. 이러한 CW_{min} 조절 방식은 DCF의 동작을 크게 수정하지 않고 효과적으로 업링크와 다운링크 플로우간의 불균형 문제를 해결할 수 있다는 장점을 가진다. 하지만 기존 연구에서^[2,3] 제안한 알고리즘은 공평성에만 초점을 맞추었고 시스템 전체의 throughput은 고려하지 않았다는 한계를 가진다.

기존 연구 결과에 따르면 CW_{min} 값의 설정에 따라 무선랜 시스템의 전체 throughput이 크게 영향을 받는다는 것이 증명된 바 있다^[4]. 본 논문에서는 업링크와 다운링크 플로우간의 불균형 문제를 해결함과 동시에 시스템의 throughput을 높일 수 있는 효과적인 CW_{min} 조절 알고리즘을 제안한다. 제안하는 알고리즘에서는 AP와 단말의 CW_{min} 값을 조절할 때 업링크 플로우와 다운링크 플로우를 가지고 있는 단말의 수를 기반으로 최적의 값을 계산한다. 그리고 계산된 값을 단말에게 알려서 현 상황에 맞게 CW_{min} 값을 변경하여 적용하도록 한다. 실험 결과를 통해 제안하는 알고리즘을 적용했을 경우 기존 CW_{min} 조절 알고리즘과 마찬가지로 단말간 공평성이 보장되면서도 보다 높은 throughput을 낼 수 있다는 것이 확인되었다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 IEEE 802.11 DCF의 업링크와 다운링크간 불균형

문제를 개선하기 위해서 제안된 기존 연구 결과들을 살펴본다. 3장에서는 제안하는 알고리즘을 설명하고, 4장에서는 시뮬레이션을 통해 제안하는 알고리즘의 성능을 평가하도록 한다. 마지막으로, 5장에서 본 논문의 결론을 맺는다.

II. 관련 연구

IEEE 802.11 표준에 정의된 DCF 모드에서는 AP 혹은 단말이 데이터 전송을 시도하기 전에 백오프 동작을 다음과 같이 수행한다. 백오프 카운터는 $[0, CW]$ 임의의 값으로 설정되며 최초 CW 값은 CW_{min} 이고 i 번째 재전송의 경우 다음과 같이 계산된다.

$$CW = \min(2^i(CW_{min} + 1) - 1, CW_{max}) \quad (1)$$

한 슬롯타임마다 백오프 카운터는 1씩 감소되며, 채널이 사용중인 경우에는 카운터 감소가 일시 중지된다. 카운터가 0이 되면 해당 AP 또는 단말은 데이터 전송을 시작한다.

무선랜에서의 업링크 플로우와 다운링크 플로우간의 불균형 문제는 [5]에서 처음 언급되었다. 무선랜의 구조상 AP에 접속해 있는 단말로 향하는 모든 다운링크 플로우는 해당 AP의 큐에 쌓이게 되지만 업링크 플로우는 각각의 단말의 큐에 별도로 쌓이게 된다. 위에서 살펴본 DCF의 백오프 동작에 따르면 AP와 단말은 동일한 채널 접속 기회를 보장 받기 때문에 다운링크 플로우는 제한된 대역폭을 나누어서 써야 하는 문제가 생긴다. 예를 들어 5개의 단말이 업링크 플로우를 하나씩 가지고 5개의 단말이 다운링크 플로우를 하나씩 가지고 있는 경우를 가정해보자. 이 때 AP는 5개의 다운링크 플로우를 번갈아 가면서 전송해야 하기 때문에 업링크 플로우의 throughput이 다운링크 플로우에 비해 약 5배 커지게 된다.

이러한 불균형 문제의 해결책은 AP가 보다 많은 채널 접속 기회를 가지게 하는 것이다. 논문 [2]와 [3]에서는 CW_{min} 조절 알고리즘을 제안하였다. [2]에서는 AP의 CW_{min} 값을 고정하고 다운링크 플로우의 수에 비례해 단말의 CW_{min} 값을 늘리는 방식을 취했고 [3]에서는 반대로 단말의 CW_{min} 값을 고정하고 AP의 CW_{min} 값을 줄이는 방식을 취했다. 논문 [6]에서는 업링크와 다운링크 플로우 간의 불균형을 완화하기 위해 AP가 백오프 동작 없이 곧바로 데이터를 전송할 수 있도록 하였다. [2, 3]에 비해서 AP에 복잡한 구현

이 필요하다는 단점이 있다. [7]에서는 AP의 transmission opportunity (TXOP) 값을 늘리는 방식을 제안하였다. 이 방식은 IEEE 802.11e 기반의 무선랜에서는 효율적으로 동작하지만 802.11e를 지원하지 않는 경우에는 적용하기 힘들다는 단점이 있다.

III. 문제 분석 및 새로운 알고리즘 제안

3.1 기존 알고리즘의 문제점

앞서 살펴본 바와 같이 [2, 3]에서 제안한 CW_{min} 알고리즘을 적용한다면 기존 AP나 단말의 큰 수정 없이 효율적으로 업링크와 다운링크 플로우간의 불균형 문제를 해결할 수 있다. 하지만 이러한 기존 CW_{min} 조절 알고리즘은 공평성에만 초점을 맞추었고 시스템 전체의 throughput은 고려하지 않았다는 한계를 가진다. 논문 [4]를 비롯한 다양한 기존 연구 결과에 따르면 업링크 플로우를 가진 단말의 수에 따라 시스템의 throughput을 높일 수 있는 CW_{min} 값이 존재한다. 본 논문에서는 업링크와 다운링크 플로우간의 공평성을 보장함과 동시에 높은 throughput을 낼 수 있는 CW_{min} 값을 찾는 새로운 알고리즘을 제안하고자 한다.

3.2 CW_{min} 값에 따른 throughput 모델링

알고리즘 설계에 앞서 CW_{min} 값에 따른 AP와 단말의 throughput 값을 예측하기 위해 [8]의 모델을 바탕으로 새로운 수식 모델을 설계한다. 분석을 단순화하기 위해 n_d 개의 단말은 업링크 플로우 하나씩을, n_d 개의 단말은 다운링크 플로우 하나씩을 가지고 있다고 가정하며 항상 보내거나 받을 트래픽이 큐에 차있는 상태라고 가정한다. 또한 최대 CW 값 CW_{max} 는 상당히 큰 값이라고 가정한다. 본 논문에서는 RTS/CTS 프레임이 사용되지 않는 경우를 가정했으나 RTS/CTS 프레임이 적용되는 경우에도 쉽게 확장하여 적용할 수 있다.

그림1은 논문 [8]에서 쓰인 Markov 체인 모델이다. 이 모델에 따르면 채널을 busy로 센스할 확률이 p , $CW_{min}+1$ 이 W 인 AP 혹은 단말의 채널 접속 확률 τ 를 아래 식으로 표현할 수 있다.

$$\tau(p, W) = \frac{1 - p^{L+1}}{(1-p) \sum_{j=0}^L \left[1 + \frac{1}{1-p} \sum_{k=1}^{2^j W - 1} \frac{2^j W - k}{2^j W} \right] p^j} \quad (2)$$

이 때 L 은 최대 재전송 시도 횟수이다. 다시 정리하면 AP가 채널을 busy로 센스할 확률을 p_{AP} 라고 하고

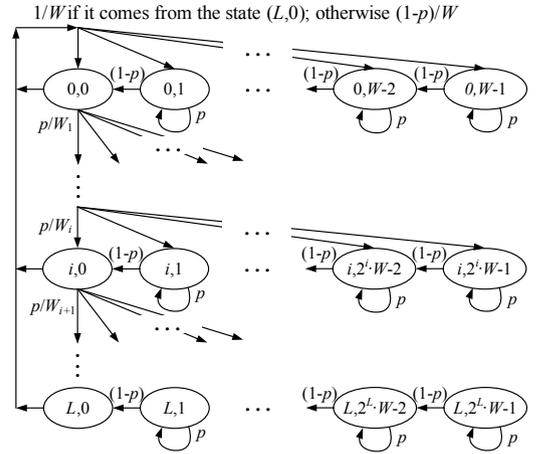


그림 1. Markov chain model

AP의 $CW_{min}+1$ 이 W_{AP} 인 경우 채널 접속 확률 τ_{AP} 는 아래 식으로 표현되고

$$\tau_{AP} = \tau(p_{AP}, W_{AP}) \quad (3)$$

마찬가지 방식으로 단말의 채널 접속 확률 τ_{STA} 는 아래 식으로 표현된다.

$$\tau_{STA} = \tau(p_{STA}, W_{STA}) \quad (4)$$

또한 채널을 busy로 센스한다는 것은 자신 이외의 트랜스미터가 있는 경우이기 때문에 아래의 식을 유도할 수 있다.

$$p_{AP} = 1 - (1 - \tau_{STA})^{n_d} \quad (5)$$

$$p_{STA} = 1 - (1 - \tau_{AP})(1 - \tau_{STA})^{n_d - 1} \quad (6)$$

AP가 전송에 성공할 확률 $p_{s,AP}$ 와 단말이 전송에 성공할 확률 $p_{s,STA}$ 는 아래와 같이 정리할 수 있다.

$$p_{s,AP} = \tau_{AP}(1 - \tau_{STA})^{n_d} \quad (7)$$

$$p_{s,STA} = \tau_{STA}(1 - \tau_{AP})(1 - \tau_{STA})^{n_d - 1} \quad (8)$$

이 때 업링크와 다운링크 플로우간의 공평한 채널 사용을 보장하기 위해서는 아래의 식을 만족해야 할 것이다.

$$n_d P_{s,AP} = P_{s,STA} \quad (9)$$

수식 (7)와 (8)을 (9)에 넣어서 정리하면 τ_{AP} 와 τ_{STA} 가 아래와 같은 관계를 가질 때 업링크와 다운링크간의 공평성이 보장되는 것을 알 수 있다.

$$\tau_{AP} = \frac{n_d \tau_{STA}}{1 - \tau_{STA} + n_d \tau_{STA}} \quad (10)$$

수식 (10)을 (6)에 대입하면 p_{STA} 를 아래의 식으로 정리할 수 있다.

$$p_{STA} = 1 - \left(1 - \frac{n_d \tau_{STA}}{1 - \tau_{STA} + n_d \tau_{STA}}\right) (1 - \tau_{STA})^{n_u - 1} \quad (11)$$

이제 수식 (4)와 (11)을 통해서 n_d , n_u , W_{STA} 값이 주어질 경우에 p_{STA} 와 τ_{STA} 를 구해낼 수 있다. 마지막으로, 구해진 τ_{STA} 값을 수식 (5)와 (10)에 대입하면 p_{AP} 와 τ_{AP} 의 값이 각각 나오고 이때 수식 (3)을 만족시키는 W_{AP} 값을 찾아낼 수 있다.

결론적으로, n_d 와 n_u 값을 알고 있고 W_{STA} 이 정해져 있으면 이때 업링크와 다운링크 플로우간의 throughput을 공평하게 맞출 수 있는 W_{AP} 값을 결정할 수 있다.

이제 시스템 전체의 throughput을 모델링한다. 논문 [8]을 참조하면 전체 throughput S 는 아래와 같이 표현할 수 있다.

$$S = \frac{E(\text{payload transmission time in a slot time})}{E(\text{length of a slot time})} \quad (12)$$

$$= \frac{p_s T_{E(P)}}{(1 - p_b)\sigma + p_s T_s + (p_b - p_s)T_c}$$

이때 p_b 는 하나 이상의 단말 혹은 AP가 특정 슬롯 타임에 전송을 시도할 확률이고 p_s 는 특정 슬롯 타임에 성공적인 전송이 발생할 확률이다. 또한 $T_{E(P)}$ 는 데이터 payload의 평균적인 전송 시간, σ 는 슬롯타임의 길이, T_s 는 성공적인 전송이 발생했을 때 채널이 busy로 판단되는 평균적인 시간, T_c 는 프레임간 충돌이 발생했을 때 채널이 busy로 판단되는 평균적인 시간을 나타낸다. p_b 와 p_s 는 각각 아래와 같이 구할 수 있고

$$p_b = 1 - (1 - \tau_{AP})(1 - \tau_{STA})^{n_u} \quad (13)$$

$$P_s = P_{s,AP} + n_u P_{s,STA} \quad (14)$$

T_s 와 T_c 는 각각 아래와 같이 구할 수 있다.

$$T_s = T_H + T_{E(P)} + SIFS + T_{ACK} + DIFS \quad (15)$$

$$T_c = T_H + T_{E(P^*)} + SIFS + T_{ACK} + DIFS \quad (16)$$

여기서 T_H 과 T_{ACK} 은 각각 헤더(PHY, MAC 헤더)의 전송 시간, ACK 프레임의 전송시간을 나타내며 $T_{E(P^*)}$ 는 프레임간의 충돌 발생 시 가장 긴 프레임의 payload의 전송 시간을 나타낸다. $T_{E(P^*)}$ 을 구하는 방법은 [4]에 설명되어 있다.

3.3 CW_{min} 조절 알고리즘 제안 및 적용 방법

Algorithm1은 본 논문에서 제안하는 CW_{min} 조절 알고리즘을 수도코드 형식으로 표현한 것이다. W_{STA} 의 값을 16부터 512까지 두 배씩 증가시키면서 가장 높은 throughput을 보장하는 W_{STA} 와 W_{AP} 의 값을 찾게 된다.

알고리즘의 동작 자세히 살펴보면 다음과 같다. 네 번째 줄의 while 루프가 시작되면 먼저 W_{STA} 에 대응하는 W_{AP} 의 값을 수식 (2)~(11)을 이용해서 찾아낸다. 그리고 throughput을 계산하여 이전 결과보다 더 큰 값을 가지면 지금의 throughput을 최대 throughput(S_{max})으로 설정하고 그 때의 W_{STA} 와 W_{AP} 을 기억한다. W_{STA} 는 매 루프마다 두 배씩 증가하며 최종적으로 512에 도달하면 루프가 끝나게 된다.

Algorithm1 CW adaptation

- 1: Input: n_u , n_d , L
- 2: **begin procedure**
- 3: Initialize $W_{STA} \leftarrow 16$, $S_{max} \leftarrow 0$
- 4: **while** ($W_{STA} \leq 512$) **do**
- 5: Find τ_{AP} , τ_{STA} , and W_{AP} using (2)~(11)
- 6: Calculate S using (12)
- 7: **if** $S > S_{max}$ **then**
- 8: $S_{max} \leftarrow S$
- 9: $W_{STA,max} \leftarrow W_{STA}$; $W_{AP,max} \leftarrow W_{AP}$
- 10: **end if**
- 11: $W_{STA} \leftarrow 2 \cdot W_{STA}$
- 12: **end while**
- 13: **return** $W_{STA,max}$ and $W_{AP,max}$
- 14: **end procedure**

표1, 표2, 표 3은 n_d 와 n_u 값이 다양하게 변할 때 제안 알고리즘을 통해 얻은 결과이다. 이 때 최대 재전송횟수 L 은 4로 설정하였다. 먼저 표1은 n_d 가 10일 때 n_u 의 변화에 따른 W_{STA} 와 W_{AP} 의 값을 보여준다. n_u 가 증가하게 되면 평균 백오프 카운터의 값을 늘려서 전송 충돌을 줄여야하기 때문에 W_{STA} 가 증가하는 것을 볼 수 있다. 그리고 W_{AP} 도 같이 증가하게 된다.

표 2는 n_u 가 10일 때 n_d 의 변화에 따른 W_{STA} 와 W_{AP} 의 값을 보여준다. 우선 n_d 가 W_{STA} 의 값에는 큰 영향을 주지 않는 것을 확인할 수 있다. 반면 n_d 가 증가하는 경우 AP가 보다 많은 트래픽을 보내야 하기 때문에 W_{AP} 가 증가하는 것을 볼 수 있다.

표3은 n_d 와 n_u 가 같을 때의 W_{STA} 와 W_{AP} 의 값을 보여준다. n_d 와 n_u 가 증가함에 따라 W_{STA} 가 증가하고 이

에 맞추어 W_{AP} 가 증가하는 것을 볼 수 있다. 결과적으로 최적의 W_{STA} 와 W_{AP} 를 결정하기 위해서는 업링크 플로우를 가진 단말과 다운링크 플로우를 가진 단말의 수를 모두 고려해야 한다는 것을 확인할 수 있다.

제안하는 알고리즘은 AP에서 수행된다. 알고리즘의 동작을 위해 AP는 n_d 와 n_u 그리고 L 값을 알아야 한다. L 은 고정된 값이기 때문에 AP가 미리 알고 있다고 가정한다. n_d 와 n_u 의 값을 얻기 위해서 AP는 정해진 시간 T 동안에 송신한 데이터 프레임과 수신한 데이터 프레임의 MAC 헤더에 포함되어 있는 destination 주소를 검사한다. T 시간 동안의 관찰이 끝나면 송신한 프레임에서 발견된 주소의 수를 n_d , 수신한 프레임에서 발견된 주소의 수를 n_u 로 하여 제안 알고리즘을 수행한다. 이 때 주목할 점은 n_d 와 n_u 가 플로우 수가 아닌 하나 이상의 플로우를 가진 단말의 수라는 점이다. 따라서 하나의 단말이 여러 개의 다운링크 플로우 혹은 업링크 플로우를 가지고 있는 경우에도 n_d 와 n_u 값에 중복되어 적용되지 않는다.

알고리즘 수행 후 결정된 W_{AP} 는 그 즉시 AP에 반영되며 W_{STA} 는 주기적으로 전송되는 AP의 비콘 메시지에 포함되어 단말에게 전달된다. 비콘 메시지를 들은 단말들은 W_{STA} 값을 갱신한다. 이를 위해 IEEE 802.11 표준에 정의되어 있는 비콘 메시지의 포맷에 새로운 필드의 추가가 필요하다.

IV. 성능평가

제안하는 알고리즘의 성능을 측정하기 위해 ns-2 시뮬레이터를 기반으로 실험을 수행하였다. 실험에 쓰인 파라미터는 IEEE 802.11b 표준을 기반으로 하였고 표 4에 정리되어 있다. 무선 구간에서의 데이터 전송 속도는 11Mbps이다. 시뮬레이션에서 구성된 토폴로지는 그림 2와 같다. 각 단말은 CN과 UDP 기반의 통신을 하며 다운링크 혹은 업링크 중 하나의 플로우

표 1. n_d 가 10일 때 n_u 의 변화에 따른 W_{STA} 와 W_{AP}

n_u	1	2	3	4	5
W_{STA}	64	128	128	128	128
W_{AP}	64	66	46	35	29
n_u	6	7	8	9	10
W_{STA}	128	128	128	128	128
W_{AP}	25	22	20	18	17

n_u	1	2	3	4	5
W_{STA}	32	32	64	64	64
W_{AP}	7	7	10	10	10
n_u	6	7	8	9	10
W_{STA}	128	128	128	128	128
W_{AP}	17	17	17	17	17

표 2. n_u 가 10일 때 n_d 의 변화에 따른 W_{STA} 와 W_{AP}

n_u	1	2	3	4	5
W_{STA}	64	128	128	128	128
W_{AP}	64	66	46	35	29
n_u	6	7	8	9	10
W_{STA}	128	128	128	128	128
W_{AP}	25	22	20	18	17

표 3. n_d 와 n_u 가 같을 때의 W_{STA} 와 W_{AP}

n_u	1	2	3	4	5
W_{STA}	16	32	32	64	64
W_{AP}	16	18	13	19	16
n_u	6	7	8	9	10
W_{STA}	64	128	128	128	128
W_{AP}	14	22	20	18	17

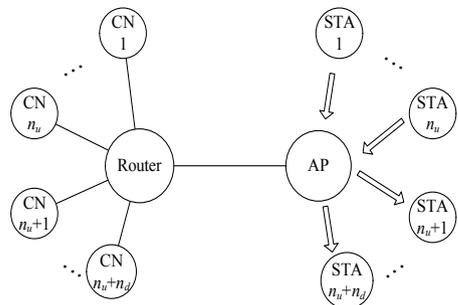


그림 2. 시뮬레이션 토폴로지

만을 가지고 있다. 패킷 사이즈는 1024 바이트로 고정되어 있다. 실험에서는 n_d 와 n_u 가 같은 상황을 가정하였다. 비교 대상으로는 IEEE 802.11 DCF와 [3]의 알고리즘이 적용된 결과를 비교하였다. 제안 알고리즘의 파라미터인 T 는 1초로 설정하였다.

먼저 각 단말이 보내거나 받을 패킷이 큐에 항상 존재하는 saturated 환경에서의 실험을 수행하였다.

그림 3은 n_d 와 n_u 의 변화에 따른 전체 throughput을 보여준다. 제안하는 알고리즘의 경우 플로우의 개수가 증가해도 throughput의 저하가 미미한 반면 DCF와 [3]의 경우 플로우 개수의 증가에 따라 급격하게 시스템의 throughput이 감소하는 것을 확인할 수 있다.

그림 4는 n_d 와 n_u 의 변화에 따른 업링크 플로우의 throughput을 포함한 결과이다. DCF의 경우 n_d 와 n_u 의 값이 증가함에 따라 업링크 throughput이 증가하다가 8개가 되는 시점에서 서서히 감소한다. 반면 [3]의 알고리즘의 경우 업링크 throughput이 다소 변동이 있지만 일정 범위를 유지하는 것을 볼 수 있다. 제안하는

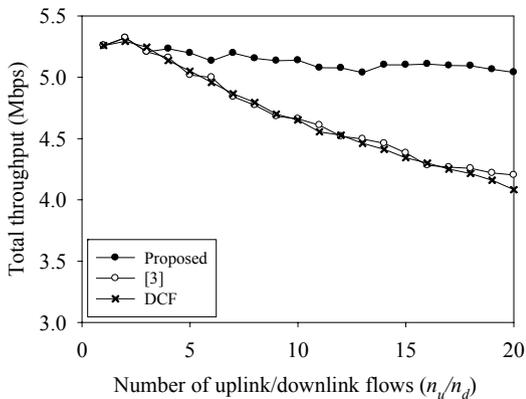


그림 3. n_u 와 n_d 의 변화에 따른 전체 throughput

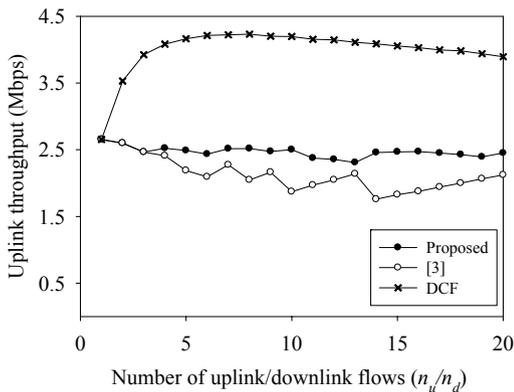


그림 4. n_u 와 n_d 의 변화에 따른 업링크 throughput

알고리즘의 경우 n_d 와 n_u 의 값에 상관없이 거의 일정한 값을 보여준다.

그림 5는 n_d 와 n_u 의 변화에 따른 다운링크 플로우의 throughput을 포함한 결과이다. 먼저 DCF를 살펴보면 n_d 와 n_u 의 값이 증가함에 따라 다운링크 throughput이 계속 감소한다. [3]의 알고리즘의 경우 업링크 throughput과 마찬가지로 다운링크 throughput이 다소 변동이 있지만 일정 범위를 유지하는 것을 볼 수 있다. 제안하는 알고리즘의 경우 업링크 throughput에서도 n_d 와 n_u 의 값에 상관없이 거의 일정한 값을 보여준다.

그림 6은 그림 4와 그림 5에서 보여준 업링크 throughput과 다운링크 throughput의 차이를 나타낸다. 다른 연구결과에서 보고된 바와 같이 DCF의 경우 업링크와 다운링크간의 심각한 불균형 현상을 관찰할 수 있다. [3]의 알고리즘을 적용했을 때는 DCF보다 불균형 현상이 많이 감소한 것을 볼 수 있다. 제안 알

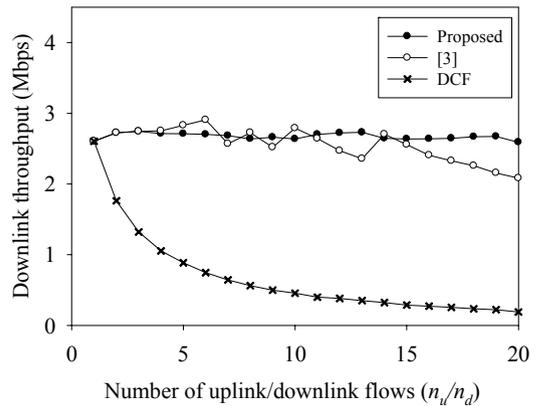


그림 5. n_u 와 n_d 의 변화에 따른 다운링크 throughput

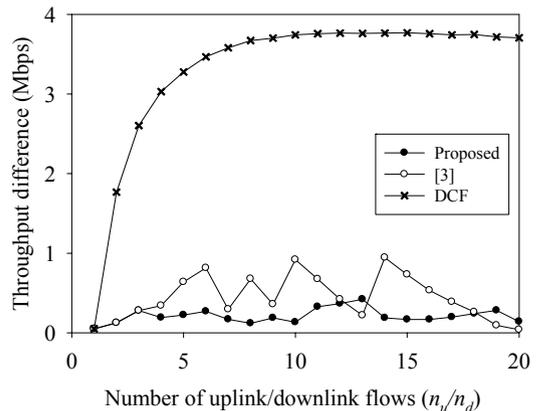


그림 6. n_u 와 n_d 의 변화에 따른 업링크 throughput과 다운링크 throughput의 차이

고리즘을 썼을 때는 대부분의 경우에 [3]의 알고리즘보다도 더 공평하게 업링크 throughput과 다운링크 throughput을 분배한다. 결론적으로 그림 3과 그림 6을 보았을 때 제안하는 알고리즘이 업링크 throughput과 다운링크 throughput간의 불균형 문제를 해결할 뿐만 아니라 시스템의 전체 throughput을 높인다는 것을 확인할 수 있다.

두 번째로 n_d 와 n_u 가 10으로 고정되어 있는 상황에서 각 단말의 전송률을 변화시키며 실험을 하였다. 이번 실험에서는 전체 throughput 측정 결과와 업링크 throughput과 다운링크 throughput의 차이만을 정리하였다. 그림 7과 그림 8의 결과를 통해 분석해 보면 전송률이 150kbps 이하인 경우에는 알고리즘의 종류와 상관없이 동일한 성능을 보이는 것을 확인할 수 있다. 이때는 데이터의 손실 없이 모든 패킷이 성공적으로 전달되는 경우이다. 하지만 전송률이 증가할수록 제안하는 알고리즘을 적용하였을 때 다른 경우와 비

교하여 전체 throughput이 높아짐과 동시에 업링크 throughput과 다운링크 throughput 간의 차이가 줄어든다. 이 결과를 통해 제안하는 알고리즘이 네트워크 사용량이 많은 환경에서 더 높은 이득을 보인다는 것을 확인할 수 있다.

V. 결 론

본 논문에서는 IEEE 802.11 기반의 무선랜을 위한 CW_{min} 조절 알고리즘을 제안하였다. 이를 위해 먼저 업링크와 다운링크 플로우간의 공평한 채널 사용이 보장되는 AP와 단말간 CW_{min} 값을 수식으로 모델링하였고 이와 더불어 전체 시스템 throughput을 모델링하였다. 위의 분석 결과를 바탕으로, 채널 사용의 공평성을 보장함과 동시에 높은 시스템 throughput을 얻을 수 있는 CW_{min} 조절 알고리즘을 설계하였다. 실험 결과를 통해 우리는 새롭게 제안하는 알고리즘이 IEEE 802.11 DCF 뿐만 아니라 기존에 제안되었던 공평성 보장을 위한 CW_{min} 조절 알고리즘과 비교해 높은 성능을 보이는 것을 확인할 수 있었다.

향후 이번 연구 결과를 바탕으로 VoIP 등 특정 어플리케이션에 특화된 CW_{min} 조절 알고리즘을 개발하고, IEEE 802.11e 기반의 무선랜 시스템으로의 적용 방안에 대해 연구할 계획이다.

참 고 문 헌

- [1] IEEE, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Reference number ISO/IEC 8802-11:1999(E), IEEE Std 802.11, 1999 edition.
- [2] M. Bottigliengo, C. Casetti, C.-F. Chiasserini and M. Meo, "Smart Traffic Scheduling in 802.11 WLANs with Access. Point," *IEEE VTC*, Oct. 2003.
- [3] B.A.H.S. Abeysekera, T. Matsuda, and T. Takine, "Dynamic Contention Window Control to Achieve Fairness Between Uplink and Downlink Flows in IEEE 802.11 WLANs," *IEEE WCNC*, Mar. 2007.
- [4] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, Vol.18, No. 3, pp.535-547, Mar. 2000.

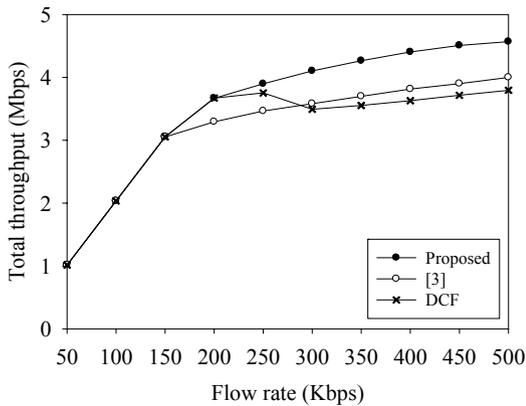


그림 7. 전송률의 변화에 따른 전체 throughput

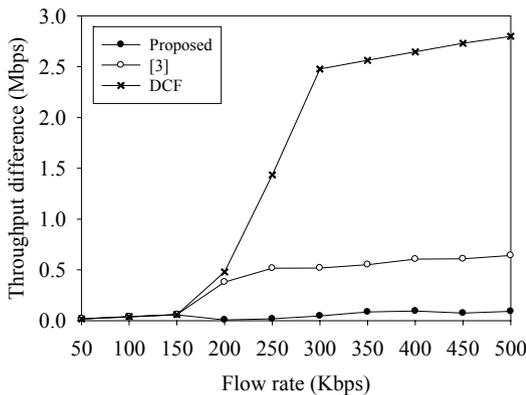


그림 8. 전송률의 변화에 따른 업링크 throughput 다운링크 throughput의 차이

- [5] A. Grilo and M. Nunes, "Performance evaluation of IEEE 802.11e," *IEEE PIMRC*, Sep. 2002.
- [6] S.W. Kim, B.-S. Kim, and Y. Fang, "Downlink and Uplink Resource Allocation in IEEE 802.11 Wireless LANs," *IEEE Transactions on Vehicular Technology*, Vol.54, No.1, Jan. 2005.
- [7] P. Clifford, K. Duffy, D. Leith, and D. Malone, "On improving voice capacity in 802.11 infrastructure networks," *IEEE WirelessCom*, June 2005.
- [8] Y. Xiao, "Performance Analysis of Priority Schemes for IEEE 802.11 and IEEE 802.11e Wireless LANs," *IEEE Transactions on Wireless Commun.*, Vol.4, No.4, pp.1506-1515.

서 영 주 (Young-Joo Suh)

중신회원



1985년 2월 한양대학교 전자공학과
 1987년 2월 한양대학교 전자공학과 석사
 1996년 미국 조지아 공대 (Georgia Tech) 컴퓨터공학 박사

1988년~1990년 LG전자 연구원
 1990년~1993년 충청대학 교수
 1996년~1997년 미국 Georgia Tech 연구원
 1997년~1998년 미국 University of Michigan 연구원
 1998년~현재 포항공대 컴퓨터공학과 교수
 <관심분야> 무선랜 프로토콜, 이동 IP, 이동 멀티캐스트, ad-hoc/sensor 네트워크, 차세대 이동 네트워크

임 완 선 (Wan-Seon Lim)

정회원



2004년 2월 포항공과대학교 컴퓨터공학과
 2006년 2월 포항공과대학교 컴퓨터공학과 석사
 2010년 2월 포항공과대학교 컴퓨터공학과 박사
 2010년 3월~현재 포항공대 정보통신연구소 연구원

<관심분야> 무선랜, 차세대 네트워크, 이동성 지원 기술

권 동 희 (Dong-Hee Kwon)

정회원



1997년 2월 한국과학기술원 전자공학과 학사
 2000년 2월 포항공과대학교 정보통신공학과 석사
 2006년 2월 포항공과대학교 컴퓨터공학과 박사
 2006년~2009년 POSDATA 연구원

2010년~현재 SK 텔레콤 연구원
 <관심분야> 무선 통신, 모바일 네트워크

김 동 욱 (Dong-Wook Kim)

정회원



2005년 2월 경북대학교 컴퓨터공학과
 2005년 3월~현재 포항공대 컴퓨터공학과 통합과정
 <관심분야> 무선랜, 무선 메쉬 네트워크, 무선 네트워크, 애드혹 네트워크