

IEEE 802.11 네트워크에서 TCP 업링크 스테이션 간 공평성 지원

정회원 전 경 구*

TCP Uplink Station-Level Fairness Support in IEEE 802.11 Networks

Kyungkoo Jun* *Regular Member*

요 약

IEEE 802.11 네트워크에서 업링크와 다운링크 간 TCP 공평성 보장을 위한 여러 연구가 진행되어 왔다. 하지만 제안된 방법들은 하나의 스테이션이 여러 개의 TCP 업링크 스트림들을 동시에 사용하여 업링크 대역폭을 독점하는 불공평성 문제에는 효과적이지 못하다. 이러한 문제에 대해 본 논문에서는 AP가 각 업링크 스테이션마다 token bucket을 지정하여 전송 대역폭 독점을 막는 방법을 제안한다. 이 방법은 업링크와 다운링크 간의 공평성도 보장할 수 있다. 제안 방법은 token bucket을 사용할 때 나타날 수 있는 전송 대역폭의 utilization 저하를 막기 위해 스테이션 간에 잉여 token을 이동할 수 있도록 한다. 이러한 token 이동을 통해 공평성과 utilization간의 균형을 맞출 수 있다. 시뮬레이션을 통해 제안 방식이 업링크 스테이션간의 대역폭 사용의 공평성은 물론, 업링크와 다운링크간의 공평성도 보장함을 확인할 수 있었다.

Key Words : TCP Fairness, Token Bucket, IEEE 802.11 Networks

ABSTRACT

There have been various research efforts to support the fairness between uplink and downlink of TCP streams in IEEE 802.11 networks. Existing methods, which have been effective for the fairness, however could not provide the solution for the unfairness caused by the situation in which a station which is having multiple TCP uplink streams monopolizes the uplink bandwidth. This paper proposes a method that AP allocates token buckets for each uplink TCP station. The proposed method is also able to support the fairness between the uplink and downlink. To remedy the underutilization which may happen under token bucket-based schemes, it allows the movement of redundant tokens among the token buckets. By controlling the token movements, it can balance the fairness and the utilization. Simulation results show that the proposed method is able to support the fairness of the TCP uplink stations, as well as the fairness between the uplink and downlink.

I. 서 론

IEEE 802.11 네트워크의 인프라스트럭처 모드 (infrastructure mode)를 사용할 때, 하나의 액세스포

인트 (access point, 이하 AP)에 TCP 업링크와 다운링크 트래픽이 공존할 경우, 둘 간의 대역폭 사용에 있어 불공평성 문제가 있다¹⁾. 인프라스트럭처 모드로 데이터를 무선 송수신하는 여러 개의 스테이

※ 본 연구는 2008년도 인천대학교 교내연구비 지원으로 수행되었습니다.

* 인천대학교 멀티미디어시스템공학과 (kjun@incheon.ac.kr)

논문번호 : KICS2009-02-061, 접수일자: 2009년 2월 13일, 최종논문접수일자: 2009년 6월 1일

선(station, 이하 STA)들과 이를 외부 네트워크와 연결시키는 액세스포인트 (access point, 이하 AP)로 구성된다. TCP 업링크 트래픽이란 STA가 AP를 통해 외부 네트워크로 전송하는 것을 의미하고, 다운링크는 그 반대이다. 이러한 TCP 업링크와 다운링크 트래픽간 불공평성 문제에 대해서는 여러 가지 방법들^{[2]-[5]}이 연구되었다.

하지만 또 다른 불공평성 문제는 한 STA가 여러 개의 업링크 TCP 스트림을 동시에 사용해서 전송 대역폭을 독점하는 것이다. 하지만 이 문제는 거의 다루어지지 않았다. 예를 들어, 총 10개의 업링크 TCP 스트림이 한 AP에 연결되어 있고, 이 중 5개가 동일 STA에 속한 경우 50%의 업링크 대역폭이 해당 STA에 의해 점유되는 불공평성 문제가 발생한다. 즉, STA A가 m개의 업링크 TCP 스트림을 이용하여 전송할 경우, 점유하는 대역폭은 m에 비례하여 증가한다. 특히, 다른 STA들이 한 개씩의 업링크 스트림만을 가진다면, 할당되는 전송대역폭에는 심각한 불공평성 문제가 발생한다.

이러한 업링크 스테이션간의 공평성 문제는 모바일 인터넷 기반 어플리케이션 사용패턴이 변화하면서 그 중요성이 커지고 있다. Twitter^[10]와 같은 모바일 블로깅 서비스가 인기를 얻으면서 업링크 멀티미디어 데이터양이 증가할 것으로 예상되기 때문이다.

TCP 업링크와 다운링크간의 불공평성 문제를 해결하기 위해 제안되었던 방법들은 논문에서 다루고자 하는 TCP 업링크 스테이션들 사이의 공평성 문제에 적용하기는 적절하지 않다. Per-flow queueing^[2]을 이용한 방법, TCP 데이터 패킷과 ACK 패킷을 별도의 큐로 분리하는 방법^[3], IEEE 802.11e의 Enhanced Distributed Channel Access (EDCA)를 이용하는^[4], 사용가능한 WLAN 대역폭을 계속적으로 측정해서 각 flow별로 rate limiter를 적용하는 방법^[5] 등은 공통적으로 업링크 전송량을 감소시켜 다운링크 트래픽 전송에 더 많은 기회를 주도록 하는 것으로, 업링크 TCP 스테이션간의 공평성에는 관심을 두지 않았다.

TCP 업링크 스트림 간 공평성에 관해서는^[6]에서 다루고 있다. 업링크 패킷에 대한 다운링크 ACK 패킷을 필터링 하거나 지연시키는 방법으로 업링크 스트림 간의 공평성을 유지하는 방식을 제안하고 있다. 하지만 이 방식 역시 하나의 스테이션이 여러 개의 업링크 스트림을 이용하여 대역폭을 독점할 경우, 스테이션 간의 공평성을 지원하지는 못한다.

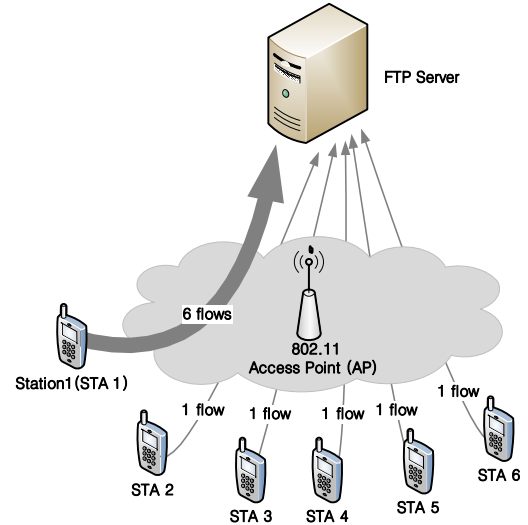


그림 1. 다수의 TCP 업링크 스트림을 갖는 STA에 의한 전송대역폭 독점 현상 재연을 위한 시뮬레이션 환경

본 논문에서는 TCP 업링크 스테이션 간의 대역폭 공평성 문제에 대한 해결책을 제안한다. 즉 여러 STA 들 중 하나의 STA가 다수의 업링크 TCP 스트림을 이용하여 대역폭을 독점하는 문제를 다룬다. 제안 방법의 핵심은 AP에 각 업링크 STA에 해당하는 token bucket을 설정하여, STA별 대역폭 사용량을 제어하는 것이다. 이 방법은 기존 업링크와 다운링크 간 공평성 문제에도 효과적이다. 또한 token bucket을 사용할 때 발생하는 대역폭의 utilization 저하를 최소화하기 위해 token 이동을 제한적으로 허용하도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 소수의 STA에 의해 업링크 대역폭이 독점되는 불공평성 문제를 자세히 소개한다. 3장에서는 이러한 문제에 대한 해결책을 제안한다. 4장에서는 제안 방식의 성능을 시뮬레이션을 통해 기존 방식들과 비교한다. 마지막으로 5장에서 결론을 도출한다.

II. 다수의 TCP 업링크 스트림을 갖는 STA에 의한 대역폭 독점 문제

다수의 TCP 업링크 스트림을 가진 소수의 STA에 의한 대역폭 독점 문제를 Qualnet 4.0^[7]을 이용한 시뮬레이션을 통해 살펴본다. 그림 1과 같이 6개의 STA들(STA 1~6)이 있고, STA와 AP는 모두 IEEE 802.11b PHY와 MAC을 사용한다. STA와 AP간의 거리는 약 60 M로 line-of-sight를 가정하

며 최대 전송률은 1 Mbps이다. 각 STA들은 FTP 트래픽을 AP를 거쳐 외부 서버로 전송한다. STA 1은 6개의 TCP 업링크 스트림을 가지고 있고, 나머지 STA들은 각 1개씩을 가지고 있다. FTP 트래픽은 4 킬로바이트 패킷을 연속적으로 전송하는 것으로 간섭이나 경쟁이 없을 경우 최대 850Kbps의 전송량을 가진다.

실험은 AP에 아무런 변경을 가하지 않은 경우와 업링크와 다운링크 간 공평성 지원을 위해 token bucket을 이용한 방식⁵⁾을 사용 하였다. Token bucket 방식의 경우에는 업링크 최대 전송률을 440 Kbps로 설정하였다. 시뮬레이션을 60초간 진행하면서 외부 FTP 서버에서 수신되는 업링크 throughput을 STA별로 측정하였다.

그림 2는 각 STA별 외부 FTP 서버에서의 throughput 측정 결과를 보여준다. Token bucket을 사용한 경우 (token bucket으로 표시)와 그렇지 않은 경우 (normal이라고 표시) 에서 공통적으로 STA 1의 throughput이 다른 STA들보다 약 6배 정도 크다. 이것은 STA 1이 다른 STA들에 비해 6배 많은 업링크 스트림을 가지기 때문이다. Token bucket을 사용한 경우의 throughput이 normal 경우보다 전체적으로 작은 것은 업링크 전송량을 440 Kbps로 제한했기 때문이다.

측정 결과에서 보듯이, 스트림 개수에 비례하여 STA가 차지하는 업링크 throughput은 증가하게 된다. 이러한 현상은 token bucket을 사용하는 경우에도 마찬가지로 발생한다. 이것은 token bucket방식이 업링크 전체 throughput에 대해서만 제한을 하고, STA 개별에 대한 제한이 없기 때문이다.

이러한 불공평성 문제는 Van Jacobson이 제안한

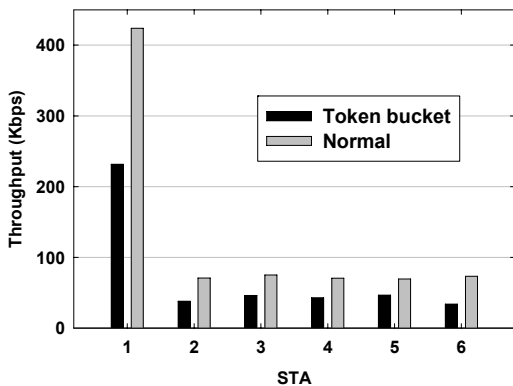


그림 2. FTP 서버에서 STA별로 측정된 업링크 throughput

Additional Increase Multiplicative Decrease (AIMD) 방식의 TCP congestion control의 잘못된 가정에 기인한다⁸⁾. AIMD는 개별 TCP 스트림이 자신의 전송 윈도우 크기를 congestion 상황에 따라 조절하도록 하여 특정 스트림이 대역폭을 독점하는 것을 방지하도록 하는 것이다. AIMD는 ACK 패킷을 받을 때 마다 전송 윈도우를 수신 ACK 갯수에 비례하여 증가시키고, 반대로 패킷 소실이 발생하면 전송 윈도우 크기를 현재의 1/2 혹은 1 Maximum Segment Size (MSS)로 급격하게 감소시킨다.

하지만 AIMD 기반 congestion control이 개별 STA들에게 대역폭을 공평하게 분배할 수 있는 경우는 STA들이 모두 동일한 수의 TCP 스트림을 사용할 때뿐이다. 그 동안 AIMD방식이 효과적이었던 것은 개별 STA들이 한 번에 한 개의 TCP 스트림을 사용하는 경우가 많았기 때문이다. 또한 여러 개의 스트림을 사용한다 할 지라도 텍스트 위주로 데이터 전송량이 많지 않았기 때문에 큰 문제가 되지 않았다. 하지만 최근에는 하나의 STA가 여러 개의 TCP 스트림을 가지며, 또한 대용량 멀티미디어 콘텐츠가 증가함에 따라 각 스트림의 데이터량이 많은 경우가 대부분이다. 대표적으로 P2P 기반 파일공유 프로그램은 10개에서 100개의 TCP 스트림을 동시에 사용한다. 그 결과 전체 10% 미만의 인터넷 사용자들이 네트워크 트래픽의 75%를 차지하는 심각한 불공평성 문제가 발생한다⁹⁾.

III. 제안방식: Separate Token Buckets

제안 방식의 핵심은 AP에서 TCP 업링크 스트림을 가진 STA마다 별도의 token bucket을 사용하여 STA별로 대역폭 사용량을 공평하게 제한하는 것이다. AP는 STA로부터 TCP 업링크 패킷을 수신할 때 마다 해당 STA의 token bucket에서 token을 감소시킨다. Token이 모자랄 경우 해당 패킷은 drop된다. 기존 방식⁵⁾가 모든 업링크 STA가 하나의 token bucket을 공유하는 shared token bucket방식인데 비해, 제안 방식은 separate token bucket방식이다.

STA별로 token bucket을 설정하는 것은 대역폭 utilization에 좋지 않은 영향을 준다. 시간에 따라 STA의 전송량이 달라지므로 할당 받은 token보다 적게 전송하는 경우도 발생하므로 이 때 남은 대역폭의 낭비가 발생한다. 반면 다른 STA는 할당 받은 token보다 더 많이 전송할 필요가 있을 경우 대역폭 부족이 발생한다. 따라서 token의 잉여와 부족을

효율적으로 처리하는 방법이 필요하다.

이를 위해 제안 방식에서는 STA들의 token bucket 들 간에 token 이동을 조건적으로 허용한다. 즉, 할 당 받은 token이 남을 것으로 예상되는 경우 부족한 STA 쪽으로 이동시킬 수 있도록 한다. 하지만 이동 token량을 결정하는 것은 쉽지 않다. 예를 들어, token을 다른 STA로 이동 후 자신의 전송에 token이 모자라는 경우가 생길 수 있다. 반대로 모 자랄 것으로 예상하여 token을 이동하지 않았는데 token이 남는 경우가 발생할 수 있다. 이러한 이동 결정에 대해서는 뒤에서 자세히 설명한다.

제안 방식이 AP에서 동작하는 방식은 다음과 같 다. 매 주기마다 업링크 전체에 할당된 전체 token 량을 tkn_i 라고 하자. 이 때 AP는 매 주기 시작시 점에서 각 STA i 의 token bucket에 token을 tkn_i 만큼씩 할당한다.

$$tkn_i = \frac{tkn}{n}, i = 1 \dots n$$

여기에서 n 은 TCP 업링크 스트림을 가진 STA의 개수이다. STA i 의 현재 가용 token량을 cur_i 라고 할 때, cur_i 는 매 주기마다 tkn_i 로 초기화된다.

AP가 업링크 데이터 패킷 pkt_j 를 수신하면, 우선 IP 헤더부분의 source address를 이용하여 어느 STA 인지 확인한다. AP는 STA i 의 사용가능 토큰량을 확인하기 위해, STA와 해당 cur_i 값들을 저장하고 있는 테이블에서 cur_i 값을 검색하고 pkt_j 의 크기인 $sizeof(pkt_j)$ 값을 비교한다. 이 때 만약 pkt_j 가 ACK-only 패킷일 경우 비교 없이 외부 유선링크로 전달 한다. ACK 패킷은 다운링크 전송된 패킷에 대한 것이므로 업링크 공평성과는 관계가 없기 때문이다.

만약 $cur_i \geq sizeof(pkt_j)$ 인 경우 AP는 cur_i 값 을 $sizeof(pkt_j)$ 만큼 감소시킨 후, pkt_j 를 외부 유선 링크로 전달한다. $cur_i < sizeof(pkt_j)$ 인 경우에는 다른 STA로부터 token 이동을 시도하고, 이동이 불 가능할 경우 pkt_j 를 drop시킨다.

STA간의 token 이동이 필요할 경우에는 다음과 같 이 처리한다. 즉, $cur_i < sizeof(pkt_j)$ 인 경우, 모자 라는 token량 $need_i$ 를 우선 다음과 같이 계산한다.

$$need_i = pkt_j - cur_i$$

이를 이용해서 모든 STA k ($k \neq i$)에 대해서 다

음 두 조건이 모두 성립하는 STA k 를 선택한다.

$$\begin{cases} cur_k \geq need_i & \dots\dots\dots (1) \\ \alpha_k = \frac{cur_k}{tkn_k} \geq \alpha_{thres} & \dots\dots\dots (2) \end{cases}$$

조건 (1)은 STA k 가 pkt_j 를 전송하는데 필요한 token량 이상을 가지고 있는지를 판단한다. 조건 (2) 는 STA k 의 잉여 token 비율 α_k 가 지정된 α_{thres} 보다 큰지를 확인한다. 따라서 α_{thres} 는 token 이동을 허용하는 기준이 된다. 예를 들어, α_{thres} 가 클 경우, token 이동 확률이 감소하며, 반대의 경우에는 증가한다.

한 가지 눈 여겨 볼 점은 α_{thres} 를 조절함으로써 전송 대역폭의 utilization과 업링크 STA간의 공평 성 중 어느 쪽에 더 비중을 둘 지를 결정할 수 있 다는 것이다.

Token 이동을 결정하는 과정에서 조건 (1)과 (2) 를 만족시키는 STA k 가 여러 개 있을 경우에는 α_k 가 가장 큰 것을 선택한다. 이 때 선택된 STA k 의 cur_k 값은 $need_i$ 만큼 감소된다.

제안방법을 실제 AP에 적용할 때 고려해야 할 사항들은 다음과 같다. 우선 AP에서 STA별로 해당 토큰량을 저장하기 위한 메모리 오버헤드가 발생한다. 각 STA별로 IP 주소 (4 바이트)와 토큰량 (4 바이트 unsigned integer)을 저장하는 정도의 오버헤 드가 발생한다. 프로세싱 측면에서는 업링크 패킷마 다 해당 cur_i 를 갱신, 패킷 크기와의 비교, 토큰 이 동 결정 및 수행에 관련된 오버헤드가 발생한다. 이 오버헤드는 업링크 트래픽 패턴과 STA 수에 의해 영향을 받는다.

IV. 시뮬레이션을 통한 성능 검증

본 장에서는 시뮬레이션을 통해 제안방식이 TCP 업링크 공평성을 보장하는지 여부와 전송 대역폭 utilization에 끼치는 영향을 평가한다. 비교대상은 기존 AP를 사용하는 경우와 shared token bucket^[7] 방식이다.

시뮬레이션은 Qualnet 4.0을 이용하여 그림 1에 서와 동일한 환경을 구축하여 60초간 진행하였다.

그림 3은 업링크와 다운링크 간의 throughput 비 율을 비교하여 제안방식이 업링크와 다운링크 간의 공평성을 보장할 수 있는지 확인한다. X축에는 비

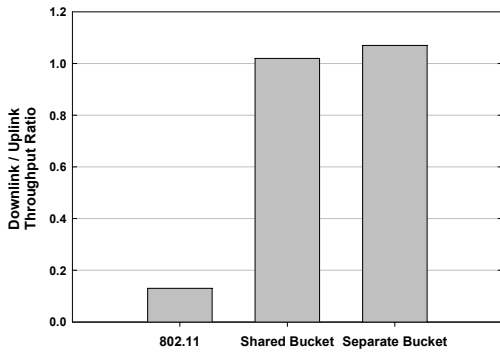


그림 3. TCP 업링크와 다운링크 throughput간 비율

교 대상인 기존 AP, shared bucket 방식, 그리고 제안 방식인 separate bucket 방식이 표시되었다. Y축은 다운링크 throughput을 업링크 throughput으로 나눈 값을 표시한다. 제안방식의 경우 α_{thres} 는 60%로 설정했다. 기존 AP의 경우, 다운링크 throughput이 업링크 대비 13%에 불과하여 다운링크와 업링크 대역폭 간에 심각한 불공평성 문제가 있다. 제안 방식은 shared bucket 방식과 비슷한 수준으로 공평성을 지원할 수 있음을 보여준다.

그림 4는 업링크 STA별로 throughput을 측정할 결과이다. 한 개의 STA가 다수의 업링크 스트림을 갖는 상황에서 공평성을 보장하는지 여부를 보여준다. X축은 각 업링크 STA를 번호로 표시한다. Y축은 각 STA의 업링크 throughput이다. 제안 방식에서는 α_{thres} 를 60%로 설정하였다. 결과를 보면 제안 방식이 shared bucket 방식보다 공평성이 현저하게 개선되었음을 볼 수 있다. 이는 STA마다 별도의

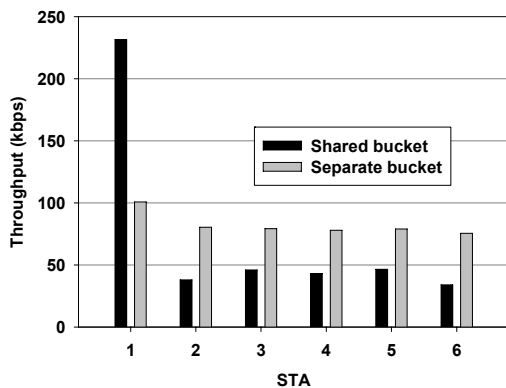


그림 4. TCP 업링크의 STA별 throughput

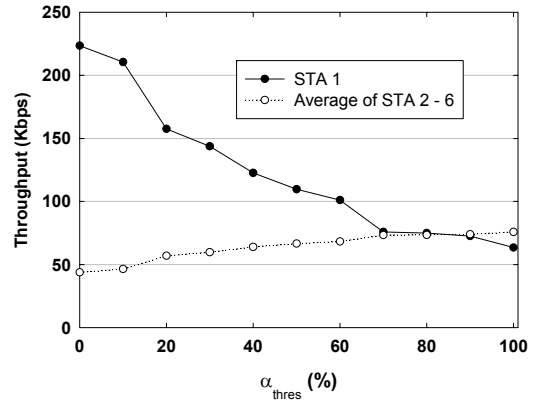


그림 5. α_{thres} 값 설정에 따른 STA별 throughput 변환

token bucket을 사용하여 전송대역폭 독점을 방지하기 때문이다. 기존 shared bucket 방식에서는 여섯 개의 업링크 스트림을 갖는 STA 1이 다른 STA들보다 약 6배 큰 throughput을 갖는다.

그림 5에서는 α_{thres} 를 변화시켜 가면서 STA 1과 STA 2~6 들 간의 평균 업링크 throughput을 측정하였다. 이를 통해 α_{thres} 가 공평성에 끼치는 영향을 알 수 있다. X축은 α_{thres} 의 변화를 나타내며, Y축은 throughput을 나타낸다. 이 때, STA 1의 throughput (STA 1으로 표시)과 나머지 업링크 STA(2~6)들의 평균 throughput을 별도로 표시하였다. α_{thres} 가 커질수록 STA 1의 throughput이 점차 낮아져 다른 STA들의 평균에 가까워지는 것을 볼 수 있다. 이는 α_{thres} 의 증가는 token 이동 허용 확률을 낮추기 때문에 STA 1의 token 부족이 심화되기 때문이다. α_{thres} 가 낮은 경우에는 token 이동이 자유롭기 때문에 STA 1이 다른 STA들로부터 token을 빌려 전송할 수 있다. 이를 통해 α_{thres} 가 STA 간 token 이동 허용 확률을 조절하여 공평성에 영향을 준다는 것을 알 수 있다.

그림 6은 제안방식의 토큰이동 기능이 성능에 미치는 영향을 알아보기 위한 실험의 결과이다. α_{thres} 값의 변화에 따른 스테이션별로 TCP 업링크 throughput 변화와 그에 따른 utilization을 관찰하였다. 다운링크 스트림에 의한 영향을 없애기 위해 앞 실험들과는 달리 여기에서는 다운링크 스트림을 사용하지 않았다. 또한 토큰 이동을 유발시키기 위해 STA 2~6 들 각각의 TCP 업링크 전송량을 token rate 75Kbps보다 작은 45 Kbps로 한정하였다. 따라서 30 Kbps만큼의 잉여 token이 발생한다. X축은 α_{thres} 의 변

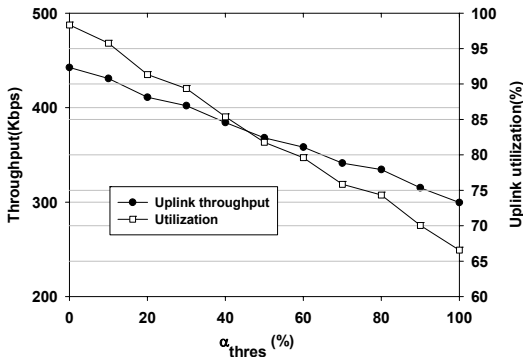


그림 6. α_{thres} 값 설정에 따른 업링크 throughput과 utilization

화를 나타내며, 왼쪽 Y축은 STA 1부터 6까지의 전체 업링크 throughput 합을 표시했으며, 오른쪽 Y축은 업링크 가용 throughput 450 Kbps의 utilization을 나타낸다. α_{thres} 가 증가할수록 업링크의 throughput이 감소한다. 이는 token 이동 허용 확률이 감소하기 때문에, 잉여 token이 낭비되어 utilization이 감소하기 때문이다. 반대로 α_{thres} 가 감소할수록 throughput이 증가된다. 이것은 STA 1이 다른 STA들의 잉여 token을 이용하여 전송할 수 있기 때문이다.

V. 결 론

본 논문은 IEEE 802.11 네트워크에서 다수의 TCP 업링크 스트림을 갖는 하나의 STA에 의해 대역폭이 독점되는 문제에 대한 해결책을 제안하였다. TCP 업링크 스트림을 가지는 STA마다 AP가 token이동이 가능한 token bucket을 설정하여 전송 대역폭 사용의 공평성과 utilization 감소방지를 지원한다. 시뮬레이션을 통해 제안방법이 업링크 STA간 공평성 지원은 물론, 기존의 업링크와 다운링크 간 공평성도 보장함을 알 수 있었다.

참 고 문 헌

- [1] S. Pilosof, R. Ramjee, Y. Shavitt, P.Sinha, "Understanding TCP fairness over wireless LAN," in Proc. IEEE INFOCOM 2003.
- [2] Y. Wu, Z. Niu, and J. Zheng, "Study of the TCP Upstream/Downstream Unfairness Issue with Per-flow Queueing over Infrastructure-mode WLANs," Wireless Communications and Mobile Computing, pp.459-471, June, 2005.
- [3] J. Ha, C. Choi, "TCP Fairness for Uplink and Downlink Flows in WLANs," in Proc. IEEE Globecom 2006.
- [4] D. Leith, P. Clifford, D. Malone, A. Ng, "TCP Fairness in 802.11e WLANs," IEEE Communications Letters, Vol.9, No.12, December, 2005.
- [5] N. Blefar-Melazzi, A. Detti, I. Habib, A. Ordine, S. Salsano, "TCP Fairness Issues in IEEE 802.11 Networks: Problem Analysis and Solutions Based on Rate Control," IEEE Transactions on Wireless Communications, Vol.6, No.4, April, 2007.
- [6] F. Keceli, I. Inan, E. Ayanoglu, "TCP ACK Congestion Control and Filtering for Fairness Provision in the Uplink of IEEE 802.11 Infrastructure Basic Service Set," in Proc. of ICC 2007.
- [7] <http://www.scalable-networks.com>
- [8] B. Briscoe, "Flow Rate Fairness: Dismantling a Religion," Internet-draft, July, 2007.
- [9] G. Ou, "Fixing the Unfairness of TCP Congestion Control," <http://blogs.zdnet.com/Ou>
- [10] <http://www.twitter.com>

전 경 구 (Kyungkoo Jun)

정회원

한국통신학회 논문지 제31권 제4B호 참조

현재 인천대학교 멀티미디어시스템공학과 조교수