

중앙 집중형 무선 매체 접근 제어 프로토콜을 위한 분할 및 합병 알고리즘을 이용한 다중 접근 방안

정회원 김민수*, 변태영**

A Multiple Access Scheme Using Split and Merge Algorithm for Centralized Wireless MAC Protocols

Min-su Kim*, Tae-young Byun** *Regular Members*

요 약

본 논문은 중앙 집중형 무선 MAC 프로토콜의 지연을 줄이기 위해 분할 및 합병 알고리즘(SMA(split and merge algorithm))를 사용한 새로운 다중 접근 기법을 제안한다. SMA의 아이디어는 각 단말의 트래픽 양에 따라 동적으로 단말을 분할과 병합을 한다는 것이다. SMA는 단말을 몇 개의 부그룹으로 나누어 각 그룹마다 서로 다른 타임 슬롯을 할당하는 방법이다. 본 알고리즘은 경쟁구간에서 충돌해결 기법과 아울러 사용될 수 있으며, 충돌 해결을 위해 분할 연산(split operation)을, 그리고 동적 그룹 관리를 위해 병합 연산(merge operation)을 사용한다. 시뮬레이션에 의하면, 지연시간, 처리율 그리고 충돌률 등의 면에서 기존의 분할(Split) 알고리즘보다 더 나은 성능을 나타낸다.

Key Words : Wireless MAC, SA, SMA, GAP, WiBro

ABSTRACT

In this paper, a novel multiple access scheme using SMA(Split and Merge Algorithm) is proposed to reduce the access delay of wireless MAC protocols. The key idea of the SMA is to dynamically split and merge the contending terminals depending on their traffic loads for effective control of access to the channels. The SMA divides the collided terminals into several sub-groups if a collision occurs and merges the idle terminals of the split sub-groups into their original group. The merge algorithm can be used to resolve collision and split operation can be used to dynamic group management. The simulation results indicate that our algorithm may provide a higher throughput and a lower collision rate than the existing algorithms, especially under heavy traffic loads.

1. 서론

MAC 프로토콜은 각 단말들이 효율적인 방법으로 공유매체에 접근할 수 있도록 하는 규칙들의 집합이다. 그러므로, MAC 프로토콜은 효율성을 보장하고 공평하게 무선 대역폭을 공유하게 하는데 중대

한 역할을 한다. 현대의 무선 MAC 프로토콜은 셀룰러 전화망, Wireless MAN, Wireless PAN, Wireless LAN 등을 포함하는데, 접근을 제어하는 중앙 장치의 유무에 따라 중앙 집중형 네트워크^[1-4]와 분산형 네트워크^[5-7]로 구분할 수 있다. 중앙 집중형 네트워크는 주로, 중앙 장치에 의해 대역이 스케줄링 되므로,

※ 이 논문(또는 저서)은 2005년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2005-214-D00154)

* Georgia Institute of Technology, ECE (kiunsen@gatech.edu),

** 대구가톨릭대학교 컴퓨터정보통신공학부 (tybyun@cu.ac.kr) (° : 교신저자)

논문번호 : KICS2006-07-293, 접수일자 : 2006년 7월 29일, 최종논문접수일자 : 2006년 8월 11일

이 장치는 단말이 가진 전송 큐의 상태를 파악할 수 있어야 한다. 각 단말은 자신의 큐 상태를 중앙 장치에 알리기 위하여 대역 예약 요구 패킷을 전송하는데, 이를 위해 랜덤 액세스 방식을 사용한다^[8]. 이러한 프로토콜의 대표적인 예는 무선 ATM MAC 프로토콜 계열의 DSA++^[1-2], DQRUMA^[3] 그리고 PRMA^[4] 등이며, 무선 PAN^[9] 혹은 무선 MAN^[10-11]의 MAC 프로토콜 등도 이러한 범주에 속한다. 특히, WiMAX나 WiBro^[12], HSDPA 등과 같이, OFDM을 사용하는 차세대 무선 인터넷에서 매체 접근을 제어하는 효율적인 MAC 프로토콜의 설계는 아주 중요한 이슈가 되고 있다.

랜덤 액세스 방식은 간단한 메시지를 전송하기에 효율적이므로 대역 요구 패킷을 전달하는데 적합하다. 그러나, 이 방식은 충돌이라는 고유한 문제를 가지고 있는데, 충돌은 서로 다른 다수의 단말이 동시에 패킷을 전송하여 그 정보를 분실하게 되는 상황으로, 성능 저하의 원인이 된다^[3-6]. 충돌에 대한 대책으로는 충돌 회피 알고리즘과 충돌 해결 알고리즘이 있는데, 본 논문에서는 무선 네트워크 환경에서 주로 채택하고 있는 CRP(Collision Resolution Periods) 구간의 스케줄링^[1-4]을 포함하는 충돌 해결 방안을 제안한다.

SMA의 주요 아이디어는 각 단말이 발생시키는 트래픽을 고려하여, 하나의 경쟁 타임 슬롯에 부과되는 트래픽 양을 일정하게 유지하기 한다는데 있다. 이를 위해서는, 단말들을 몇 개의 그룹으로 나누어 서로 다른 타임 슬롯을 경쟁하도록 하는 방법이 있는데, 분할 시에는 각 단말의 트래픽 발생률이 고려되어야 한다. SMA는 이를 동적으로 해결하는 방법으로, 경쟁 타임 슬롯에 부과되는 트래픽이 많으면 그것을 사용하는 그룹을 분할하여 서로 다른 경쟁 타임 슬롯을 할당함으로써 슬롯에 부과되는 평균 트래픽을 줄이고, 경쟁 타임 슬롯에 부과되는 트래픽이 작으면 다른 그룹과 병합하여 하나의 경쟁 타임 슬롯을 할당함으로써 슬롯에 부과되는 평균 트래픽을 높인다. SMA의 분할 그룹 관리의 그룹 간에서 서로 독립적이므로, 최근 이슈로 떠오르는 OFDM 상의 WiMAX 혹은 WiBro를 위한 MAC 프로토콜 설계에 적용될 수 있다.

본 논문의 2장에서는 연구 배경과 관련 연구가 소개되며, 3장에서는 SMA 알고리즘을 기술한다. 4장에서는 SMA 알고리즘의 성능을 분석하고 이를 시뮬레이션 하였으며 마지막으로 5장에서 결론을 맺는다.

II. 관련 연구와 연구 배경

무선 MAC 프로토콜은 랜덤 액세스 방법을 통해 단말의 대역 예약 패킷을 전송하는 방법을 사용한다. 랜덤 액세스 방법은 서로 다른 다수의 단말이 동시에 전송할 경우 발생하는 충돌이 가장 큰 약점이다. 즉, 충돌이 일어나면 재전송하여야 하므로 추가적인 대역을 낭비하여야 하기 때문이다. 최근 제안된 MAC 프로토콜들은 충돌 해결 방안에서 조금씩 차이를 보인다. 본 논문에서는 이러한 충돌을 해결하기 위한 방안으로 SMA 알고리즘을 제안한다.

SMA 알고리즘은 단말이 발생시키는 트래픽에 따라 몇 개의 그룹으로 분배시키는 방안이다. 이렇게 분배된 각 단말 그룹은 개개의 타임 슬롯을 경쟁하게 되며, 경쟁에서 승리하는 단말이 대역을 예약할 수 있게 된다. 이러한 방법에서 중요한 요소는 (1) 그룹 내에서의 충돌 해결 알고리즘, (2) 그룹의 구성 방안이다.

SA(Split Algorithm)는 최근에 제안된 가장 뛰어난 충돌 해결 방안 가운데 하나이다. 이 알고리즘은 충돌한 단말들을 Q (2이상)개의 그룹으로 분산시키는 방법으로, 이러한 분산 과정을 분할(split)이라 부른다. 충돌이 일어난 단말은 Q 개의 면을 가진 주사위($Q=2$ 일 때는 동전)를 던져 전송할 순서를 결정하는데, 1이 나온 단말들부터 먼저 전송하며 나머지 숫자가 나온 단말들은 대기한다. 물론, 1이 나온 단말이 두 개 이상이면 역시 충돌이 발생할 것이며, 이 추가적인 충돌을 위해 다시 주사위를 던져서 전송 순서를 결정한다. 충돌한 단말들이 나뉘어지는 이런 과정은 Q -ary 트리(특히, block access), 또는 스택(특히, free access)을 통해 쉽게 볼 수 있다. 이러한 방법으로 충돌을 해결하는 시간을 충돌 해결 구간(CRP: Collision Resolution Period)라 부르며, 이 기간 동안 도착하는 새로운 패킷들은 충돌이 해결되는 동안 블록 된다. 물론, 이렇게 블록 되는 패킷이 두 개 이상인 경우에는 새로운 CRP가 시작될 것이다. 본 논문에서는 성능 분석의 간소화를 위해 $Q=2$ 인 경우만을 고려하기로 한다.

SA는 한 경쟁 타임 슬롯에 부과되는 트래픽 양 s 에 따라 CRP의 길이가 달라진다. 즉, s 가 크면 CRP가 그만큼 길어지게 된다. 그러나 s 에 적응적으로 동작할 수 있다면 더 나은 성능을 얻을 수 있다. s 에 적응적으로 동작한다는 말은 더 나은 성능을 내기 위한 s 의 값을 찾고, 그것을 계속 유지하게 하도록 한다는 것을 의미한다. QoS를 고려한 많은 연

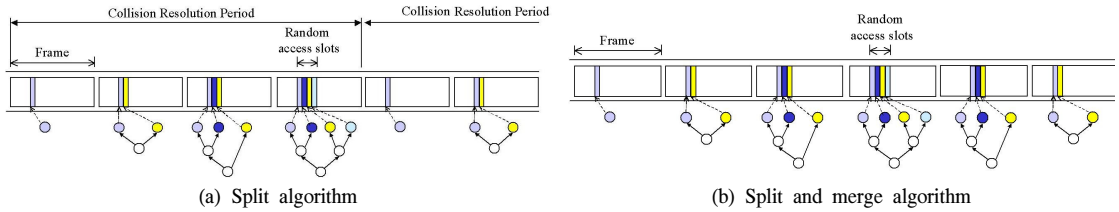


그림 1. SA와 SMA의 동작 차이

구에서 트래픽의 종류(CBR, VBR 등)에 따라 서로 다른 채널을 경쟁하게 하는 방법들이 소개되고 있다. 이러한 경우 경쟁하는 트래픽들은 채널 접근 패턴이 유사하므로 적응적으로 접근하는 것이 유리하다. 예를 들어, CBR 트래픽들이 서로 경쟁한다면 고정적인 접근 패턴을 가지게 되므로 적정한 s 를 유지하게 되면 그만큼 지연 시간을 감소시킬 수 있다.

s 의 유지는 분할된 상태를 유지하게 함으로써 가능하며 이를 통해 다음 충돌 해결 구간의 길이를 감소시켜 지연 시간을 줄이거나 동일한 시간 동안 더 많은 타임 슬롯에서 전송에 성공하도록 하여 처리율을 높일 수 있다. SMA는 이를 가능하게 하기 위해 본 논문에서 제안하는 프로토콜으로써, s 에 적응적으로 동작한다. 이를 위해 고려해야 할 사항은 다음과 같다.

- Merge과 stay
- s 의 추정

분할 상태를 유지하기 위한 방법으로 merge 연산과 stay 연산을 고려한다. Merge 연산은 s 의 값이 너무 작아져서 높일 필요가 있을 경우에 수행된다. 즉, s 의 값이 낮추어야 할 만큼 크다면 분할하고, 적당한 값이면 분할 상태를 유지하고, 그렇지 않다면 병합하여 s 에 적응적인 상태를 유지하게 되는 것이다. 고려해야 할 두 번째 사항은 s 를 찾아내는 일이다. 본 논문에서는 s 의 값을 추정하기 위해서 이전 프레임의 타임 슬롯 접근 정보를 이용한다. 즉, 이전 경쟁 타임 슬롯에서 충돌하였으면 그 타임 슬롯의 s 값을 감소시키기 위해 split 연산을 수행하고, 전송에 성공하였으면 현재의 s 값을 유지하기 위해 stay 연산을 수행하며, 그 슬롯에 접근하는 단말이 없으면 merge 연산을 수행하여 s 를 높인다. 직관적으로, 최적의 s 값은 1임을 알 수 있다. 그림 1은 SA와 SMA를 비교한 것이다. 앞서 언급한 바와 같이 타임 슬롯에 접근하는 각 그룹을 Q-ary 트리로 나타내었다. 타임 슬롯에 대한 접근 결과는 기지국에서 파악하여 매 프레임마다 각 단말에게 알려

주므로 트리는 프레임마다 변화하며, 한 프레임마다 각 트리의 단말 노드마다 하나의 타임 슬롯을 할당 받게 된다.

III. SMA 알고리즘

SMA의 알고리즘은 split, stay 그리고 merge의 세가지 연산들로 구성된다. 이 연산들은 기지국에서 수행되며 연산의 결과는 하향 링크를 통해 단말들에게 피드백 된다. 전송한 바와 같이 이 연산들은 단말들의 상향 접근 정보(collision, success 혹은 empty)에 따라 수행된다. 피드백 메시지는 그룹 접근 권한(GAP: Group Access Permission) 정보를 포함하게 되어 각 단말이 자신이 속한 그룹에 할당된 타임 슬롯을 찾을 수 있도록 한다.

3.1 그룹 접근 권한(GAP)

GAP은 분할된 그룹의 식별자이며, 그 길이 I_{GAP} 은 n 이하이다. 여기에서 n 은 분할될 수 있는 최대 길이이며, 단말의 총 개수는 $2n$ 이 된다. 각 단말은 자신에게 할당된 id와 GAP를 비교하여 자신의 그룹에 할당된 슬롯을 찾을 수 있다. 그림 2는 GAP를 사용하는 예를 보인 것인데, GAP은 노드를 표현하는 원 안에 이진수로 나타나 있다.

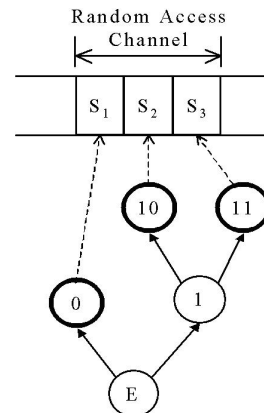


그림 2. GAP 사용의 예

```

algorithm split(IN GAPO, IN lGAPO, OUT lGAPN, OUT
GAPL, OUT GAPR)
GAPL :=(GAPO << 1) + 0
GAPR :=(GAPO << 1) + 1
lGAPN = lGAPO + 1
    
```

그림 3. Split algorithm

```

algorithm merge(IN GAPL, OUT GAPR, IN lGAPO, OUT
lGAPN, OUT GAPN)
if(GAPL = GAPR)
    GAPN := GAPL >> 1
    lGAPN = lGAPO - 1
    return true
return false
    
```

그림 4. Merge 연산

이 예에서 사용된 GAP는 0, 10, 11이며 각 단말 들은 자신의 식별자와 GAP를 비교하여 접근하게 된다. 즉, 0으로 시작하는 식별자를 가진 단말들은 s1에, 10으로 시작하는 식별자를 가진 단말들은 s2에 그리고 11로 시작하는 식별자를 가진 단말들은 s3에 각각 접근할 수 있다.

3.2 Split, Stay 그리고 Merge 연산

Split, stay 그리고 merge 연산은 이전 프레임의 경쟁 결과에 따라 실행 된다. 즉, 해당 경쟁 슬롯에 충돌이 일어났으면 split 연산을, 전송에 성공하였으면 stay 연산을, 그리고 빈 슬롯이면 merge 연산을 수행하는 것이다. 각 연산은 프로시저의 형식으로 기술하였으며 파라미터 전달 방법은 각 파라미터 앞에 IN(값에 의한 전달)과 OUT(참조에 의한 전달)을 통해 기술하였다. Split 연산은 그림 3에 기술되어 있다.

충돌이 일어났을 경우에는 해당 슬롯에 접근한 단말들을 두개의 그룹으로 분할하는데, 이들 각각을 위한 GAP(GAP_L and GAP_R)가 필요하게 된다. Split 연산은 GAP_L and GAP_R 을 계산하기 위해 기지국에서 사용이 되며 그 결과는 하향 링크를 통해 각 단말에게 알려진다. l_{GAP_O}은 이전 프레임에서 해당 슬롯을 사용하는 그룹의 GAP의 길이이며 GAP_L와 GAP_R는 이전 프레임에서 사용된 GAP 값이다. 또한 l_{GAP_N}는 다음 프레임에서 해당 슬롯을 위해 적용할 GAP의 길이이며, '<<'는 비트 쉬프트 연산자이다. 그림 4는 merge 연산을 기술하고 있다.

Merge 연산은 해당 경쟁 슬롯에 대한 접근이 없었던 그룹을 다른 그룹과 병합하기 위해 수행되므로 병합되어야 할 두 그룹의 GAP(GAP_L과 GAP_R)을 비교한 후 병합한다. GAP의 시퀀스가 서로 다른 두

```

algorithm stay(IN GAPO, OUT GAPN, IN lGAPO, OUT
lGAPN)
GAPN := GAPO
lGAPN := lGAPO
    
```

그림 5. Stay 연산

그룹은 병합될 수 없으므로 이를 먼저 검사한 후 병합해야 한다. GAP_L과 GAP_R은 이전 프레임에서 사용된 GAP 값을 저장하고 있으며 병합할 수 없는 경우에는 false를 리턴한다. 그림 5는 stay 연산을 기술한다.

Stay 연산은 전송이 성공한 경쟁 슬롯을 사용한 그룹들을 위해 사용된다. 즉, 전송에 성공하였다는 것은 이 그룹의 s가 적절하다는 것을 의미하므로 GAP를 그대로 유지하기 위해 사용된다. 이전 프레임에서 가졌던 GAP(GAP_O)와 l_{GAP_O}를 다음 프레임의 GAP_N와 l_{GAP_N}에서 그대로 유지한다.

IV. 성능 평가

4.1 수학적 분석

최근 몇 년간의 성능 분석 연구에서, 중요한 평가 요소 가운데 하나는 바로 지연 시간이므로 본 논문에서는 SMA의 지연 시간을 수학적으로 분석하고 시뮬레이션을 통하여 그 성능을 평가한다. SMA의 지연 시간은 하나의 경쟁 타임 슬롯을 할당 받은 한 그룹 내에서 일어나므로, 각 그룹의 평균 CRP의 길이가 바로 지연 시간이 될 것이다. SMA는 SA와 마찬가지로, split 연산을 통해 충돌을 해결하는데, 한 그룹의 크기(즉, 그룹에 속한 단말의 개수)와 각 단말의 평균 패킷 발생률이 주어졌을 경우의 지연 시간은 SA의 분석 논문인^{1), 2)}에서 소개된 바 있다. 따라서, 본 논문에서는 분할 그룹의 크기를 추정하는데 초점을 맞추기로 한다.

분할 그룹의 크기는 그룹의 분할 회수를 통해 알 수 있다. 즉, 한 번도 분할하지 않은 그룹의 크기는 2ⁿ이며, 한 번 분할한 그룹의 크기는 2ⁿ⁻¹, i번 분할한 그룹의 크기는 2ⁿ⁻ⁱ가 되는 것이다. 각 그룹의 평균 분할 회수 \bar{l}_d 는 SMA 프로토콜이 안정적인 상태에 이르렀을 경우의 이전 트리 깊이를 의미하며 식 (1)에 나타내었다.

$$\bar{l}_d(n, \bar{k}) = \sum_{l=0}^{n-1} l \cdot p_c(n, \bar{k}, l) \quad (1)$$

$p_c(n, \bar{k}, l)$ 는 평균 분할 회수가 l일 확률이며 그림 6의 상태 천이 다이어그램을 통해 구할 수 있다.

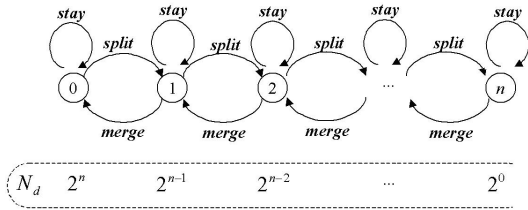


그림 6. 평균 분할 회수를 위한 상태 천이 다이어그램

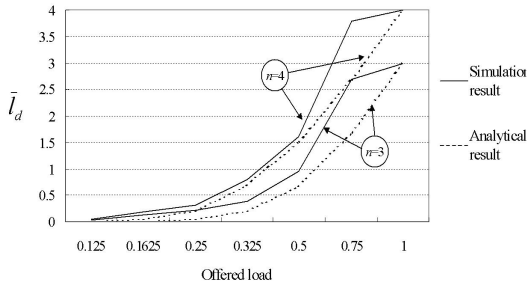


그림 7. 평균 분할 횟수

각 상태에서 접근할 수 있는 단말의 개수, N_d 는 분할 깊이가 깊어지는 만큼 감소한다. 각 상태에서 분할이 일어날 확률, 즉 그룹의 멤버들에 의해 충돌이 발생할 확률 p_{split} 은 식 (2)와 같다.

$$p_{split}(N_d, \bar{k}) = \sum_{i=2}^{N_d} \binom{N_d}{i} \left(\frac{\bar{k}}{2^n}\right)^i \left(1 - \frac{\bar{k}}{2^n}\right)^{N_d-i} \quad (2)$$

각 상태에서 merge할 확률 p_{merge} 와 stay할 확률 p_{stay} 은 될 확률은 식 (3)과 식 (4)를 통해 얻을 수 있다.

$$p_{merge}(N_d, \bar{k}) = 2 \binom{N_d}{0} \left(\frac{\bar{k}}{2^n}\right)^0 \left(1 - \frac{\bar{k}}{2^n}\right)^{N_d} \quad (3)$$

$$p_{stay}(N_d, \bar{k}) = 1 - (p_{split}(N_d, \bar{k}) + p_{merge}(N_d, \bar{k})) \quad (4)$$

그림 7은 식 (1)의 평균 분할 깊이를 시뮬레이션 결과와 비교하고 있다. 그림에서 볼 수 있듯이, 수식에 의한 결과와 시뮬레이션 결과가 거의 일치함을 알 수 있다. 여기에서 주의할 것은 그래프의 가로축이 한 타임 슬롯에 부과되는 트래픽 양이라는 점이다.

그룹의 크기와 단말의 트래픽 발생률이 주어졌을 경우의 CRP 길이를 분석하여 보자. 이는 앞서 언급한 바와 같이, SA 혹은 SMA 알고리즘의 지연 시간과 같다. 먼저, 충돌이 일어난 터미널의 수를 $k(k \geq 2)$, 분할 단계의 최고 깊이를 n 이라 할 때, k 개의

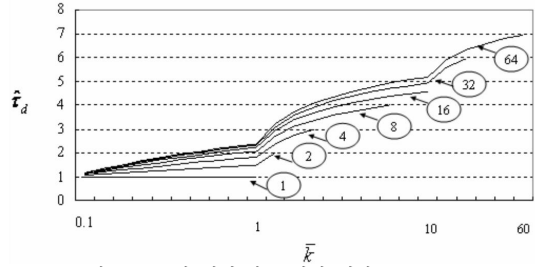


그림 8. 평균 CRP의 길이(평균 지연 시간)

단말이 충돌하여 k_i 개가 왼쪽 그룹, k_i 개는 오른쪽 그룹을 선택할 확률 $Q(n, k, k_i)$ 은 식 (5)를 통해 구할 수 있다.

$$Q(n, k, k_i) = \frac{\binom{2^{n-1}}{k_i} \binom{2^{n-1}}{k-k_i}}{\binom{2^n}{k}} \quad (5)$$

k 개의 충돌을 해결하는데 필요한 구간의 길이가 1일 확률 $p(n, k, 1)$ 은 식 (6)을 통해 재귀적으로 구할 수 있다.

$$p(n, k, l) = \frac{1}{k} \sum_{i=0}^k Q(n, k, i) (i \cdot p(n-1, i, l-1) + (k-i) \cdot p(n-1, k-i, l-1)) \quad (6)$$

충돌 해결 과정에서, 그룹의 초기 크기 k 는 $0 \leq k \leq 2^n$ 이며, 평균이 \bar{k} 인 이항 분포를 따른다. 이에 따른 pdf는 식 (7)과 같다.

$$p_b(n, k, \bar{k}) = \binom{2^n}{k} \left(\frac{\bar{k}}{2^n}\right)^k \left(1 - \frac{\bar{k}}{2^n}\right)^{2^n-k} \quad (7)$$

k 개의 충돌을 해결하기 위해 필요한 평균 슬롯의 개수 \bar{l}_s 는 식 (8)에서 구할 수 있다.

$$\bar{l}_s(n, k) = \sum_{l=0}^{n+1} l \cdot p(n, k, l) \quad (8)$$

이 식들을 이용하여, 평균 지연 시간 $\bar{\tau}_d$ 를 구할 수 있다.

$$\hat{\tau}_d(n, \bar{k}) = \frac{1}{k} \sum_{k=0}^{2^n} (p_b(n, k, \bar{k}) \cdot k \cdot \bar{l}_s(n, k)) \quad (9)$$

그림 8은 $\hat{\tau}_d$ 를 도시한 것이다. 각 원 안에 수는 터미널의 수(2^n)를 나타낸다. 이를 통하여 우리는 $\hat{\tau}_{SA}$ 와 $\hat{\tau}_d$ 가 동일함을 알 수 있다.

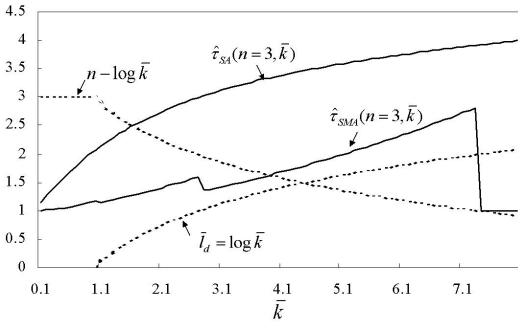


그림 9. SA와 SMA의 평균 지연시간 비교

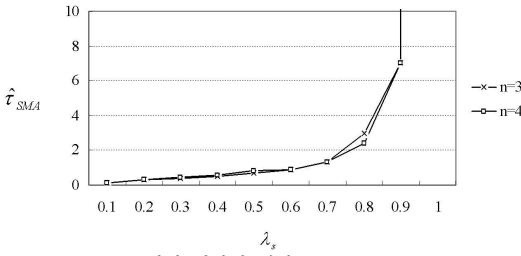


그림 10. SMA 지연 시간의 변화

CRP의 길이는 단말의 개수가 많아질수록 그리고 단말의 패킷 발생률이 높아질수록, 높아지는 것을 알 수 있다. 식 (1)의 결과를 고려하면 SMA의 평균 지연 시간을 식 (10)과 같이 정의할 수 있다.

$$\hat{\tau}_{SMA}(n, \bar{k}) = \hat{\tau}_d(n - \bar{l}_d, \frac{\bar{k}}{2^{n - \bar{l}_d}}) \quad (10)$$

그림 9는 SA와 SMA의 평균 지연 시간을 비교하고 있다. 두 알고리즘 모두 $n=3$ 일 때의 경우이다. 두 개의 실선은 두 알고리즘의 평균 값을 나타내고 있으며 두 사선은 각각 $n - \log \bar{k}$ 와 $\log \bar{k}$ 를 도시하고 있다. 이 그림에서, τ_{SMA} 는 선형적인 증가 추세를 보이는데, 이것은 도착률이 증가함에 따라 분할 깊이가 깊어지기 때문이다.

4.2 시뮬레이션 결과

시뮬레이션에서, 단말들은 포아송 분포에 따라 트래픽을 발생시킨다. 주어진 시뮬레이션 결과에서도 알 수 있지만, SMA는 n 에 독립적인 특성을 가지므로 n 이 3과 4인 경우만을 비교하였다.

그림 10은 λ_s 에 따른 SMA의 지연 시간을 시뮬레이션한 결과이다. 전술한 바와 같이, $n=3$ 일 때의 지연과 $n=4$ 일 때의 지연이 거의 같음을 알 수 있는데, 이는 SMA의 경우 그 지연 시간이 n 의 값에 무관함을 의미한다. 즉, 단지 λ_s 의 값만이 SMA의 지연

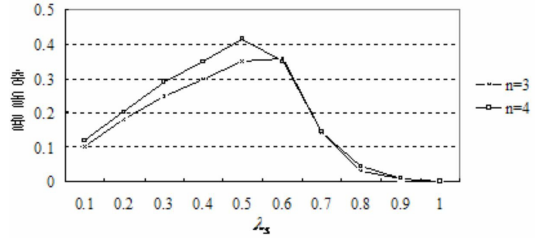


그림 11. SMA의 충돌률의 변화

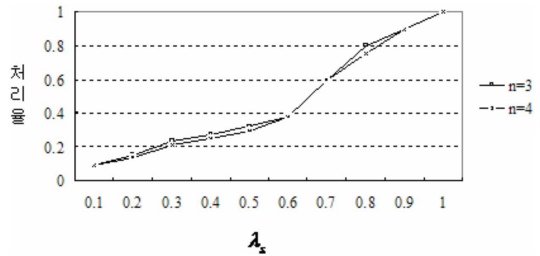


그림 12. SMA의 처리율

시간에 영향을 주는데, 이는 SMA의 충돌 해결이 안정 상태에서 시작되기 때문이다.

그림 11은 SMA의 충돌률에 대한 시뮬레이션 결과이다. SMA의 충돌률은 n 이 아니라 λ_s 에 영향을 받는다. SMA의 충돌률은 λ_s 의 값이 0.5를 넘어서면서부터 감소하기 시작한다. 이는 이진 트리의 깊이, 즉 분할 횟수 \bar{l}_d 가 0.5 이하에서는 별로 변화가 없다가 그 이후에 급속히 길어짐을 의미한다. 이러한 결과는 그림 7에서 나타난 \bar{l}_d 의 변화 추세와 연관 지어 생각할 수 있다.

그림 12는 SMA의 처리율에 대한 시뮬레이션 결과를 나타낸 것이다. 지연의 증가 시점과 처리율의 증가 시점이 같으며 증가 추세 또한 같음을 알 수 있다.

V. 결론

본 논문에서는 단말이 발생시키는 트래픽의 양에 따라 단말을 다수의 그룹으로 분할 및 병합을 반복적으로 수행하여, 하나의 그룹이 최적화된 트래픽 양을 적응적으로 유지하도록 하는 SMA 알고리즘을 제안하였다.

수학적 분석을 통해 평균 분할 깊이와 이에 따른 평균 지연 시간을 도식화 하였으며, SA와의 비교를 통해, SMA의 지연시간에 관한 성능이 더 우수함을 보였다. 특히 유입 트래픽이 점점 더 많아 질수록 SMA의 지연시간에 대해 더욱 유리한 상황을 가지게

됨을 알 수 있다. 또한 컴퓨터 시뮬레이션을 통해 SMA의 성능은 단말의 개수와는 무관하고 단지 트래픽의 발생률에만 영향을 받는다는 사실을 밝혀내었다. 평균 분할 깊이는 완만하게 증가하다가 트래픽이 0.5를 넘어서면서 급속히 증가하는 추세를 보이는데, 이러한 추세는 시뮬레이션 결과와도 일치한다.

참 고 문 헌

[1] D. Petras and A. Hettich, "Performance Evaluation of a Logical Link Control Protocol for an ATM air interface," *PIMRC.*, Nov. 1996.

[2] D. Petras, A. Hettich and A. Krämling. "Design Principles for a MAC Protocol of an ATM Air Interface." *PIMRC'96*, Nov. 1996.

[3] Kantawong, S.; Sivamok, N.; Wuttisittikulkij, L, "Investigation of Contention Resolution Algorithms for Wireless Communication Systems," *Vehicular Technology Conference*, June 2001.

[4] Charoenpanitkit, A.; Sivamok, N.; Wuttisittikulkij, L, "An analysis of a new access control technique for channel request in wireless communications," *IEEE International Symposium on Circuits and Systems*, June 2001.

[5] Chao-Ming Teng, Kwang-Cheng Chen. "A unified algorithm for wireless MAC protocols," *Vehicular Technology Conference*, June 2002.

[6] Wen-Tsuen Chen, Bo-Bin Jian, Shou-Chih Lo, "An adaptive retransmission scheme with QoS support for the IEEE 802.11 MAC enhancement," *Vehicular Technology Conference*, Aug. 2002.

[7] M. J. Karol, Z. Liu, and K. Y. Eng, "Distributed queueing request update multiple access for wireless packet(ATM) networks," Sept. *ICC*, 1995.

[8] D.J. Goodman, R.A. Valenzuela, K.T. Gayliard and B. Ramamurthi, "Packet Reservation Multiple Access for Local Wireless Communications," *Vehicular Technology Conference*, June 1988.

[9] C. Cordeiro, S. Abhyankar, R. Toshiwal, and D. Agrawal, "BlueStar: Enabling efficient integration between Bluetooth WPANs and IEEE 802.11 WLANs," *Mobile Networks and Applications*,

Vol.9, No.4, Aug. 2004.

[10] C. Eklund, R. Marks, K. Starwood, and S. Wang, "IEEE Standard 802.16: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access," *IEEE Communication Magazine*, Volume: 40, Issue: 6, June 2002.

[11] D. Cho, J. Song, M. Kim, and K. Han, "Performance Analysis of the IEEE 802.16 Wireless Metropolitan Area Network," *IEEE DFMA'05*, Feb. 2005.

[12] A. Doufexi, S. Armour, M. Butler, A.Nix and D. Bull, "A Study of the Performance of HIPERLAN/2 and IEEE 802.11a Physical Layers," *Vehicular Technology Conference*, May, 2001.

김 민 수 (Min-su Kim)

정회원



1998년 2월 동국대학교 컴퓨터 과학과 학사
 2000년 2월 경북대학교 컴퓨터 공학과 석사
 2004년 2월 경북대학교 컴퓨터 공학과 박사
 2005년 (주)모빌랩 근무

2005년 9월~현재 Postdoctor, ECE, Georgia Institute of Tech.

<관심분야> 무선 MAC, 차세대 이동통신

변 태 영 (Tae-young Byun)

정회원



1994년 2월 경북대학교 컴퓨터 공학과 학사
 1997년 2월 경북대학교 컴퓨터 공학과 석사
 2000년 2월 경북대학교 컴퓨터 공학과 박사
 1994년 SK그룹 (주)SKC 근무

1998년 3월~2000년 2월 (주)새빛정보 대표이사

2000년 3월~2003년 8월 경주대학교 컴퓨터전자공학부 조교수

2003년 9월~현재 대구가톨릭대학교 컴퓨터정보통신 공학부 부교수

<관심분야> 무선 MAC 프로토콜, 차세대 인터넷기술, 차세대 이동통신