

# Unicast 통신에서의 S-MAC Protocol의 전력 효율성 연구

A study on the Power Efficiency of S-MAC Protocol for Unicast Communication

백 세현, 김 영범

(건국대학교 정보통신대학 전자공학부)

Key Words : MAC, S-MAC, IEEE802.11, Energy efficiency, Ns2

## 요약

유비쿼터스 센서 네트워크에서 전력 효율을 높이는 것은 매우 중요한 과제이다. 노드의 전력 소비를 줄이기 위해 다양한 mac protocol 들이 제안되어 왔고, 본 논문에서는 ns2(network simulator) tool을 이용하여 Wei Ye, John Heidemann, Deborah Estrin 이 2002년에 발표한 S-MAC Protocol 의 전력효율을 기존의 IEEE 802.11 MAC 과 비교하였다.

## 1. 서론

MAC(Medium Access Control) Protocol은 노드의 매체접속을 제어하는 Protocol 이다. 다수의 노드가 하나의 매체를 공유하는 경우, 두 개 이상의 노드가 동시에 매체를 사용하게 되면 패킷간 충돌로 인해 통신은 실패하게 될 것이다. 이를 방지하기 위해 MAC Protocol 은 언제 어떤 노드가 매체를 사용해도 되는지 결정하는 제어 역할을 하게 된다.

MAC 계층에서의 주된 전력 소비요인은 크게 4가지를 꼽을 수 있다.

첫 번째는 idle listening 이다. 매체가 idle 상태일 때 전송될지 여부가 불확실한 패킷을 받기 위해 노드가 매체를 감지하는 것으로, 가장 주된 전력 소비 요인이다.

두 번째는 collision이다. 동시에 두 개 이상의 노드가 패킷을 전송할 경우 패킷은 충돌하여 소실되며, 제

전송이 이루어져야 되므로 전력소비가 증가하게 된다.

세 번째는 overhearing 이다. 이는 목적노드가 아닌 노드가 패킷을 수신하는 것을 의미한다.

마지막으로 control packet overhead 는 control packet(RTS,CTS등의 제어패킷)을 송수신하는 과정도 전력소비를 증가시키는 것을 의미한다.

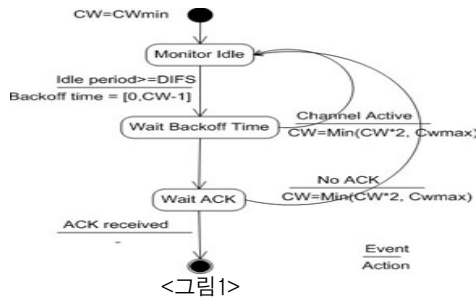
## 2. IEEE 802.11 MAC

IEEE802.11 MAC은 contention-based channel access function인 DCF를 기반으로 한다.

DCF(Distributed Coordination Function)는 basic access mechanism 과 RTS/CTS(Request to send/Clear to send)메커니즘을 사용한다. DCF에서 패킷을 보내고자 하는 노드는 일정시간 동안 매체의 idle 여부를 감지하게 되는데, 이 과정은 매체가 DIFS(Distributed inter-frame space)보다 긴 시간동안 사용되지 않을때까지 지속된다. idle>=DIFS를 만족하면 노드는 전송하기 전에 임의지연시간(random backoff interval)동안 기다린다. backoff time counter 는 매체가 idle로 감지되는 동안 지속적으로 감소한다. 이 counter는 매체내 전송이 감지되면 멈추며, 이후 DIFS보다 큰 idle 상태가 감지되면 다시 감소하기 시작한다. 이 backoff time이 0이 될 때 노드는 패킷전송을 시작한다. <그림1>

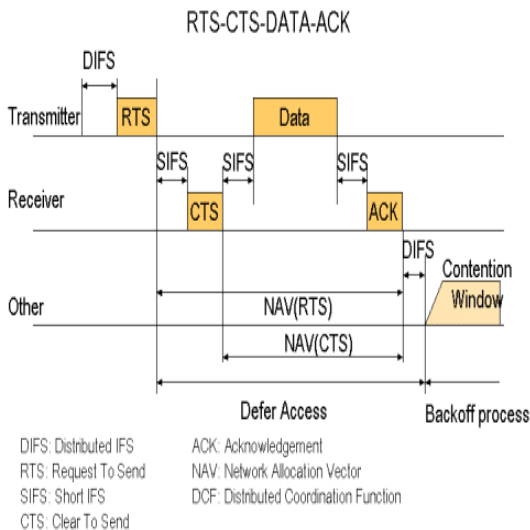
목적지 노드가 패킷을 성공적으로 수신한 경우 ACK 패킷을 송신노드에 전송하게 되는데 만약 송신노드가 지정된 시간내에 ACK패킷을 받지 못하면 위의

backoff 규칙에 따라 패킷을 재전송하게 된다. 이 방식을 basic access mechanism 이라 한다.



<그림1>

RTS/CTS 메커니즘은 DCF에서 사용하는 또 하나의 기법이다. 노드는 데이터 패킷을 보내기 전에 짧은 RTS 패킷을 전송한다. RTS 패킷도 backoff 규칙에 따라 전송한다. 이 RTS 패킷이 성공적으로 전송되면, 수신노드는 짧은 CTS 패킷으로 응답한다. CTS 패킷이 에러없이 송신노드에 전달되면, 송신노드는 데이터 패킷을 보내고, 수신노드는 데이터 패킷을 받은 후 ACK 패킷을 송신노드에 보냄으로써 통신을 마치게 된다. 즉, RTS/CTS 메커니즘은 데이터 패킷 전송을 위해 매체를 예약하는 기능을 한다. <그림2>



<그림2>

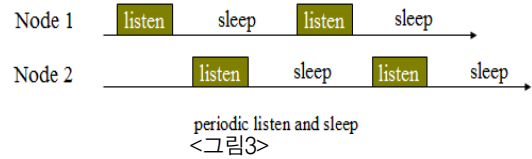
### 3. S-MAC Protocol

S-MAC Protocol 은 Wei Ye, John Heidemann, Deborah Estrin이 2002년에 발표한 MAC Protocol 이며, 주요 특징은 다음과 같다.

#### 3.1 Periodic Listen and Sleep

각 노드는 주기적으로 listen/sleep mode를 반복하게

된다. 노드는 sleep mode 일 때 전력 소비가 매우 낮다.<그림3>



만약 두 개 이상의 노드가 특정노드로 패킷을 보내고자 할 때는 매체를 예약하기 위해 송신노드들은 경쟁해야할 것이다. 이 경쟁기법(contention mechanism)은 RTS/CTS 메커니즘을 사용하는 IEEE 802.11 의 그것과 동일하다. RTS를 제일 먼저 보내는 노드가 매체를 사용할 권한을 얻게 되고, 수신노드는 CTS 패킷으로 응답하게 된다. 전송이 시작된 후, 전송이 모두 끝날때까지 송,수신노드는 그들의 sleep schedule 을 무시하게 된다.

각 노드는 동작을 시작하기 전에 먼저 listen/sleep schedule을 선택하고 이 정보를 이웃노드들에 알려주어야 한다. 모든 노드는 이웃노드들의 listen/sleep schedule 정보를 저장하는 schedule table을 가지고 있으며, 노드가 그들의 listen/sleep schedule을 설정하고 schedule table을 완성,수정하는 것은 노드간 주기적인 sync패킷 교환을 통해 이루어진다.

sync 패킷에는 해당 노드의 주소와 다음 sleep time 정보가 포함되어 있으며 몇주기마다 sync패킷을 전송할 것인지는 사용자가 설정할 수 있다(한번의 listen/sleep:1주기). 구체적인 과정은 다음과 같다.

먼저, 노드는 매체를 일정 시간동안 감지하게 되고, 이웃노드로부터 sync 패킷을 받지 못하면 sleep mode로 동작할 시간을 독자적으로 설정하여 sync 패킷을 통해 t초 후에 sleep mode로 동작할 것임을 이웃노드들에게 알려주게 된다. 이 노드를 synchronizer라 부른다.

만약 노드가 독자적으로 schedule을 설정하기 전에 이웃노드로부터 schedule 정보를 받게 되면 이 schedule 정보에 맞추어 자신의 listen/sleep schedule을 셋팅하게 된다. 이러한 노드를 follower라 부른다. 임의 지연시간 td 후, 그 노드는 자신의listen/sleep schedule을 이웃노드들에 알려주게 된다.

이 메커니즘을 통해 통신하고자 하는 두 개의 노드의 listen/sleep 주기가 틀려도 상대 노드의 next listen time을 알고 있기 때문에 통신이 가능하게된다.

IEEE 802.11 MAC에서 노드는 sleep mode로 동작하지 않는다. listen/receiving 과 transmitting mode만이 지원된다.

### 3.2 Collision and Overhearing Avoidance

IEEE 802.11 MAC에서 각 노드는 이웃노드들의 모든 전송을 listening 하게 되고, 이는 각 노드가 자신과 상관없는 많은 패킷들을 overhearing 하게 됨으로써 전력 효율이 떨어지게 된다.

S-MAC Protocol 은 송,수신노드의 이웃노드들을 RTS혹은 CTS 패킷을 수신한 직후 sleep mode로 동작하게 한다. 보통 데이터 패킷의 길이는 RTS, CTS 등의 제어패킷보다 훨씬 길게 마련이고, 이 데이터 패킷과 뒤따르는 ACK 패킷을 이웃노드들이 overhearing하는 것을 방지함으로써 전력효율을 더 좋게 하는 것이 가능하다. 이는 네트워크 할당 백터(NAV)를 통해 가능한 기법이다.

각 노드는 이웃노드의 매체 사용여부를 알 수 있는 NAV정보를 가지고 있다. 자신이 아닌 타 노드에 전송되어야 할 패킷을 받은 경우, 그 패킷의 duration field에 저장된 NAV 값을 참조하여 자신의 NAV 값을 수정하게 된다. 이 NAV 값이 0 이 아니면, 이는 이웃노드가 매체를 사용 중이라는 것을 의미하며 NAV값이 0으로 감소할 때까지 노드는 sleep mode로 동작하게 된다.

### 3.3 Message Passing

S-MAC Protocol 은 긴 message를 다수의 작은 fragments로 분할하여 한꺼번에 전송하게 된다. 이 때 오직 하나의 RTS 패킷과 CTS 패킷이 사용된다. 한번의 RTS, CTS 패킷 교환으로 모든 fragments를 전송하기 위해 필요한 시간만큼 매체를 사용할 수 있다. 송신노드는 각 data fragment를 전송할 때마다 수신노드로부터 ACK 패킷을 받게 되며, 만약 ACK패킷을 받지 못하면 해당 data fragment 의 전송을 실패한 것으로 간주하여 필요한 시간만큼 매체 사용시간을 늘린후, 다시 해당 data fragment를 전송하게 된다.

IEEE 802.11 MAC 도 message를 다수의 fragments로 분할하는 기법을 지원하지만, RTS 와 CTS 패킷을 각각의 data fragments 와 ACK 패킷을 주고받을 때마다 매체 사용을 위해 주고 받아야 한다.

## 4. S-MAC과 IEEE 802.11의 전력소비량 비교

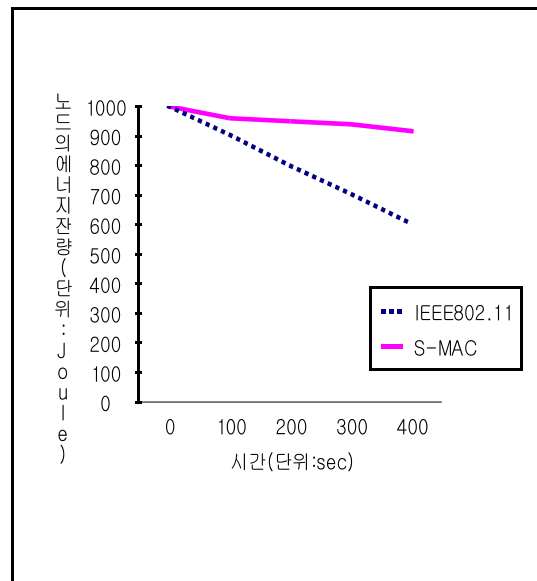
S-MAC과 IEEE 802.11의 전력소비량을 ns2 tool을 이용하여 simulation해보았다.

### 4.1 네트워크 시나리오

2개의 노드를 생성, 총 400초 동안 송신노드에서 수신노드로 매 5초마다 512Byte의 패킷을 전송하게 된다. 400초간 송신노드가 소비한 전력량을 S-MAC을 사용할 때와 IEEE 802.11 MAC을 사용할 때로 나누어 서로 비교하였으며, ns2 tool을 이용하여 simulation하였다. 송신노드의 기본에너지는 1000 Joule로 설정하였다.

### 4.2 Simulation Result

송신노드의 에너지 잔량을 S-MAC을 사용했을 때와 IEEE 802.11을 사용하였을 때로 나누어 비교해 보면 아래 그림과 같다. <그림4><표1>



<그림4>

<표1>

시간(sec)	송신노드의 에너지잔량(J)	
	IEEE802.11	S-MAC
0	1000	1000
100	903.9	960
200	798.9	949.7
300	703.9	939.6
400	603	916.5

<그림4>,<표1>를 살펴보면 400초가 지난후, S-MAC을 사용했을 때 송신노드가 소비한 전력

84J 정도인 반면, IEEE 802.11 MAC을 사용했을 때 송신노드가 소비한 전력은 397J 정도임을 확인할 수 있다. 2개의 노드가 존재하며, 송신노드가 수신노드로 일정 주기마다 일정한 크기의 패킷을 지속적으로 전송하는 환경에서 S-MAC 이 IEEE 802.11 에 비해 전력효율이 다소 좋다는 ns2 simulation 결과를 확인할 수 있었다.

## 5. 결론

본 논문에서는 두 개의 노드가 존재하며, 하나의 송신노드에서 수신노드로 패킷을 전송하는 간단한 네트워크 시나리오를 토대로 S-MAC 과 IEEE 802.11 의 전력 효율을 ns2 로 비교하였다. 네트워크 시나리오를 다양하게 설정하여 그에 따른 S-MAC 과 IEEE 802.11의 전력효율을 고찰해볼 필요가 있다.

## 참고문헌

- [1] Data Communications and Networking (by Behrouz A. Forouzan)
- [2] "Architecture, Mobility Management, and Quality of Service for Integrated 3G and WLAN Networks" (in Wiley Journal of WCMC 2005. by Yang Xiao)
- [3] NS2.26 SourcesOriginal File List(doxygen)
- [4] S-MAC Related Papers(by Wei Ye, John Heidemann, Deborah Estrin)
- [5] Tutorial for the Network Simulator "ns"(by MarcGreis)