

트리 기반 센서네트워크에서 전력 소모 균형을 위한 슬롯 스케줄링 알고리즘

정희원 김 제욱*, 종신회원 오 훈*

A Slot Scheduling Algorithm for Balancing Power Consumption in Tree-based Sensor Networks

Je-Wook Kim* *Regular Member*, Hoon Oh*° *Lifelong Member*

요 약

본 논문에서는 트리 기반의 센서네트워크에서 전력소모의 균형을 이루기 위한 시간 슬롯을 할당하는 알고리즘을 제안한다. 트리 토폴로지를 갖는 센서네트워크에서는 낮은 트리 깊이를 갖는 노드들이 더 많은 패킷을 처리하기 때문에 더 많은 전력을 소모한다. 따라서, 네트워크 생명이 단축되는 문제가 있다. 본 논문에서 제안하는 슬롯 스케줄링 알고리즘은 각 노드에게 수신 슬롯들을 먼저 할당하고 송신 슬롯들을 할당한다. 이러한 할당 방식은 패킷 결합과 필터링을 용이하게 하기 때문에 낮은 트리 깊이에 위치한 노드들의 부하를 감소시키는 효과가 있다. 시뮬레이션을 사용하여 제안한 알고리즘을 TreeMAC에서 제안한 프레임-슬롯 할당 알고리즘과 비교함으로써 전력 소모의 균형을 이룬다는 것을 확인하였다.

Key Words : Slot Scheduling, Power Balancing, Aggregation, Filtering, Tree topology

ABSTRACT

In this paper, we propose a slot scheduling algorithm for balancing power consumption in tree-based sensor networks. In this type of networks, nodes with lower depths tend to consume more energy than those with higher depths, thereby reducing the life time of the network. The proposed algorithm allocates a series of receiving slots first and then a series of sending slots. This way of slot allocation eases packet aggregation and filtering, and thus reduces traffic load on nodes near a sink. We compare the proposed algorithm and the frame-slot allocation algorithm employed in the TreeMAC by resorting to simulation. The simulation results showed that the proposed approach well achieves the balancing of power consumption.

1. 서 론

환경 데이터를 센싱하는 많은 노드와 노드들로부터 데이터를 수집하는 싱크노드들로 구성되는 무선센서 네트워크 (WSN)는 이벤트 기반의 환경관제를 위하여 주로 사용된다. WSN의 응용 분야는 시간에 민감한 산업용 감시 분야 혹은 제어 분야로 점점 확대되고 있

다. WSN이 위험한 작업환경의 안전 감시에 사용되는 경우에, 센싱된 데이터의 적시 전송 (Timely Delivery) 과 보장 전송 (Guaranteed Delivery)을 포함하는 신뢰성 통신이 요구된다. 적시 전송은 정의된 시간 내에 싱크노드에 데이터를 전달하는 것이고 보장 전송은 센싱 데이터를 분실 없이 싱크노드에 전달하는 것이다. 하지만, 무선 메쉬 네트워크에서 데이터 전달의

※ 본 연구는 울산시 및 교육과학기술부의 지역과학연구단지 개발 프로그램의 지원으로 수행되었습니다.

* 울산대학교 전기공학부 조선해양IT융합연구실, (° : 교신저자)

논문번호 : KICS2011-01-034, 접수일자 : 2011년 1월 12일, 최종논문접수일자 : 2011년 5월 2일

신뢰성을 보장하는 것은 간단한 문제가 아니다. 그 이유는 하나의 데이터 전송은 모든 이웃 노드의 유효 대역폭에 영향을 끼치기 때문이다.

MAC 프로토콜의 초기 연구는 대부분 전력소비를 줄이기 위하여 짧은 동작 사이클 (low-duty cycle)을 갖는 프로토콜 설계에 초점이 맞추어 졌다^{11,4)}. 이 중에서 D-MAC¹¹⁾은 S-MAC⁴⁾과 T-MAC³⁾ 프로토콜에서 데이터를 획득한 노드가 데이터를 싱크노드까지 전달하는 과정에서 이웃 노드들 간의 동작시간 싱크 문제로 데이터 전달 방해 (Interruption)가 발생하고 전송이 지연되는 문제를 해결하고자 하였다. D-MAC은 데이터 전달 방해와 전력 소모 문제를 개선하기 위하여 트리의 단말 노드로부터 싱크노드까지 실행 주기의 뒤틀린 (skewed) 동기 기법을 사용하였다.

최근에는 TDMA 기법을 도입하여 센싱 데이터의 적시 전달에 중점을 두는 프로토콜들이 제안되었다^{15,9)}. TRAMA¹⁷⁾는 시간을 슬랏의 형태로 나누고, 유니케스트와 브로드케스트 전송의 충돌을 방지하기 위하여 각 노드가 어떤 특정 슬랏에 전송해야 하는가를 결정하는데 있어서 노드의 트래픽 정보 기반 분산 선출 기법을 사용하였다. [8]에서는 TDMA 기반 프로토콜과 CSMA기반 프로토콜의 장점과 단점을 정의하고, 각 노드에 필요한 슬랏 할당량을 계산하기 위하여 DRAND¹⁰⁾를 사용한 Z-MAC을 제안하였다. Z-MAC은 TDMA기반의 프로토콜을 사용하였지만 CSMA기반의 슬랏 훔치기 (slot stealing) 기법을 사용하여 가변적인 트래픽 상황에서 전송효율을 개선하였다. 하지만, 깔대기 효과 (funneling effect) 문제⁵⁾를 다루지 못했다. 깔대기 효과는 트리 토폴로지에서 싱크노드에 가까운 노드들이 그들의 후손으로부터 전달되는 과도한 데이터 패킷을 효율적으로 처리하지 못하기 때문에 발생한다. 한편, 효율적인 통신을 위하여 트리 기반의 프로토콜이 제안되었다^{9,11,12)}. 이들 중에서, TreeMAC⁹⁾은 TDMA기반의 프로토콜이며 트리 토폴로지에서 프레임-슬랏 할당 (FSA) 알고리즘을 사용하여 각 노드에 프레임을 할당함으로써 신뢰성 통신을 보장하고 데이터 수집 트리 네트워크의 깔대기 효과를 해결할 수 있는 것으로 입증되었다.

TreeMAC은 트리에 속한 모든 노드가 자신의 센싱 데이터를 간섭없이 싱크노드에 보낼 수 있는 시간 사이클로 정의되는 수퍼프레임 (SF)을 사용한다. 프레임-슬랏 할당 (FSA) 알고리즘은 기본적으로 SF를 다수의 프레임으로 나눈다. 이 때 각 프레임은 3개의 슬랏으로 구성된다. 각 노드는 자식 노드의 프레임 요구량에 따라서 프레임들을 할당하고 프레임을 할당 받은

각 노드는 트리의 수직방향으로 2홉 거리 내에 위치한 노드들 간에 동일한 슬랏 번호가 선택되지 않도록 프레임의 전송 슬랏 번호를 결정한다. 이런 식으로 FSA알고리즘은 수직방향으로 2-홉 거리에 있는 노드들의 전송 간섭을 근본적으로 차단한다. 이 프로토콜의 장점은 충돌 회피 패킷 전송 혹은 수신, 버퍼링없는 패킷 전달, 싱크노드에 대한 높은 전송 효율을 포함한다. 이러한 장점에도 불구하고 TreeMAC은 각 프레임을 구성하는 3개의 슬랏이 수면, 송신, 수신 등으로 번갈아 변하기 때문에 데이터 패킷의 결합 및 필터링이 어렵다. 따라서, 전력 소모 균형을 이루기 어려운 단점이 있다.

본 논문에서는 요구량 기반의 슬랏 할당 알고리즘을 제시함으로써 TreeMAC 프로토콜의 문제점을 다루었다. 제안하는 슬랏 요구량 기반 할당 알고리즘에서는 트리에 있는 각 노드는 자신과 자신의 후손들이 한 SF 동안에 수집한 모든 데이터를 자신의 부모 노드에게 어떠한 간섭없이 송신할 수 있을 만큼의 슬랏량을 요구한다. 싱크노드에서 시작하여 각 노드는 자신의 자식 노드들의 각자 요구하는 만큼의 슬랏수를 SF로부터 할당해 준다. 따라서, 슬랏의 낭비가 발생하지 않으며, 각 노드에 할당되는 슬랏들의 패턴은 데이터 패킷의 결합과 필터링에 적합하도록 수신 슬랏의 연속과 송신 슬랏의 연속으로 되어 있다. 패킷결합과 필터링을 통하여 싱크노드 가까이 있는 노드들에서 전력 소모를 크게 줄이고 전력 소모의 균형을 가능하게 함으로써 센서네트워크의 수명을 연장시킨다.

본 논문의 구성은 다음과 같다. 2장에서 본 연구의 동기와 함께 연구 배경을 논의한다. 3장에서는 제안하는 프로토콜에 대하여 기술하고, 4장에서는 시뮬레이션을 통해서 성능을 평가한다. 마지막으로 5장에서 결론을 맺는다.

II. 연구배경

2.1 네트워크 모델

본 논문의 네트워크 모델은 복수의 싱크노드와 복수의 센서노드, 즉 소스노드들로 구성된다. 네트워크는 싱크노드에서 시작하는 데이터 수집 트리를 구성한다. 서로의 통신 거리에 위치하는 두 개의 노드는 링크를 가지고 있다고 하고, 특히 부모노드와 자식노드 사이에 링크는 트리링크라고 한다.

각 센서노드는 데이터를 센싱하고 무선 멀티홉을 경유하여 획득한 데이터를 싱크노드에 보낸다. 모든 센서노드는 고정이며 하나의 데이터 수집 사이클에서

한 개의 전송할 데이터 패킷을 생산한다.

2.2 문제 정의

프레임-슬롯 할당 (FSA) 알고리즘은 SF를 프레임으로 나누고 각 프레임을 3개의 슬롯으로 나눈다. 각 노드는 부모-자식 관계를 이용하여 프레임-슬롯 할당을 수행한다. 노드는 수평적 간섭을 배제하기 위해서 각 자식노드에게 그 자식노드의 후손노드의 수에 하나 (즉, 자식노드 자체를 위한 프레임) 를 더한 개수에 해당하는 프레임을 할당한다. 할당된 프레임에 속하는 3개의 슬롯에 대하여, 각 노드는 “전송슬롯 # = 트리 깊이 mod 3”의 전송슬롯 결정공식을 사용하여 전송 슬롯 번호를 결정한다. 즉, 트리 깊이 $3i+1$, $i \geq 0$, 에 있는 노드는 슬롯 번호 1, 트리 깊이 $3i+2$ 에 있는 노드는 슬롯 번호 2, 트리 깊이 $3(i+1)$ 에 있는 노드는 슬롯 번호 0을 전송슬롯으로 할당한다. 이런 식으로 전송슬롯을 결정함으로써 수직 2홉 거리 내에서 동일한 전송슬롯 번호를 사용하는 것을 배제한다. 결과적으로, TreeMAC은 수평적으로는 서로 다른 프레임을 할당하고 수직적으로 2홉 거리 내에 동일한 전송슬롯의 사용을 배제함으로써 데이터 전송 시에 서로간의 간섭을 배제한다.

그림 2는 그림 1의 트리에 대한 FSA 알고리즘의 적용 예를 보여준다. 노드 1은 5개의 프레임 즉, 자식들에게서 수신한 데이터를 전달하기 위한 4개의 프레임과 자신이 획득한 데이터를 전송하기 위한 1개의 프레임에 해당하는 프레임 요구량을 가지고 있다. 노드 1의 프레임 요구량인 5개의 프레임은 5개의 전송 슬롯, 4개의 수신슬롯, 6개의 수면슬롯으로 구성되어 있다. 노드 1의 트리 깊이는 1이기 때문에 전송슬롯 결정공식에 따라서 슬롯 번호 1을 전송슬롯으로 사용한다. 유사하게, 노드 2는 4개의 프레임, 노드 3과 6은 각각 2개의 프레임을 할당 받는다. 터미널 노드 5와 7

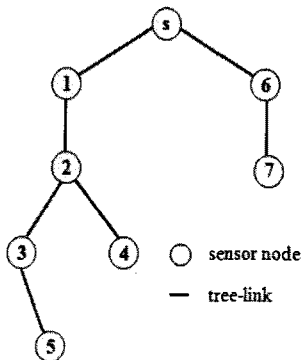


그림 1. 트리토폴로지의 예

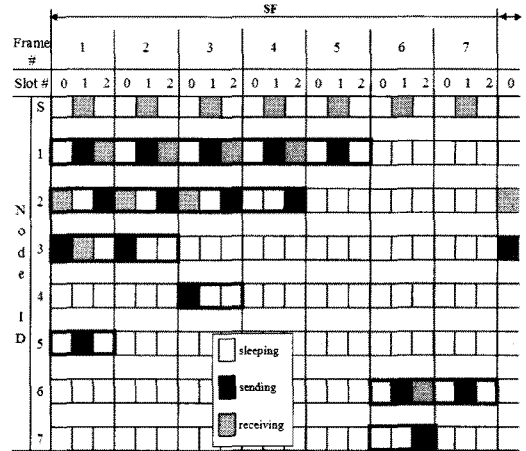


그림 2. 그림 1의 트리에 대한 프레임-슬롯 할당

은 자신의 데이터를 부모 노드에게 보내기만 하면 되니가 1개의 프레임을 할당 받으면 충분하다. 싱크노드 S의 프레임 요구량은 자식노드인 노드 1과 노드 6의 프레임 요구량의 합에 해당하는 7개의 프레임이 되며, 싱크노드의 프레임 요구량은 수퍼프레임(SF)가 된다.

TreeMAC은 싱크에 가까운 노드가 더 많은 데이터를 처리함으로써 발생하는 데이터 패킷의 손실 혹은 지연 문제 즉, 깔때기 효과 (funneling effect)^[5]를 제거하는데 기여했다. 또한 적응적 동작사이클 (duty cycle) 기법에서 존재하는 데이터 전달 간섭 문제^[1]를 없앴다. TreeMAC의 이러한 장점에도 불구하고, 데이터 패킷의 결합 및 필터링이 매우 어렵다. 패킷결합과 필터링을 이용하여 패킷 헤더 및 데이터의 중복전송을 줄임으로써 대역폭의 효율성을 높이고 전력소모를 줄일 수 있다. 하지만, FSA 알고리즘을 사용하면 프레임을 구성하는 슬롯들의 상태가 번갈아 변하기 때문에 데이터 결합이 매우 어렵다. 예를 들면, 그림 2의 노드 1에 할당된 각 프레임에서는 수면, 전송, 수신 형태의 상태를 반복하고 있다.

FSA에 의해서 제기된 단점을 극복하기 위하여 모든 노드가 성공적으로 자신의 센싱 데이터를 싱크노드에 전송할 수 있는 식으로 자신의 슬롯 요구량을 진술하고, 그 요구량에 따라서 슬롯을 할당하는 새로운 요구량 기반 슬롯 할당 알고리즘 (DSA)을 제안한다. DSA 알고리즘에서는 각 노드에게 각 노드가 자신의 데이터와 자신의 후손노드가 전달한 데이터를 자신의 부모노드에게 전송하기 위하여 필요한 슬롯량을 다른 노드의 슬롯들과 중복되지 않게 할당한다. 따라서, 노드들 사이에 수직 및 수평

적인 간섭이 배제된다. 또한, DSA 알고리즘은 일련의 수신 슬랏을 할당하고 일련의 송신 슬랏을 할당하는 식으로 각 노드에 슬랏들을 할당한다. 이러한 슬랏 할당 방식은 데이터 패킷을 먼저 수집하고 결합 및 필터링을 수행한 후에 전송을 수행할 수 있게 한다. 따라서, 싱크에 가까운 노드의 부하를 줄여서 전력소모의 균형을 추구할 수 있다.

2.3 정의 및 정리

다음과 같은 기호를 정의하여 사용한다.

- h_i : 싱크노드로부터 노드 i 의 트리 깊이 (홉 수)
- $i.T$: 노드 i 에서 시작되는 트리의 노드 집합
- $i.P$: 노드 i 의 부모노드
- $i.C$: 노드 i 의 자식노드들의 집합
- $d(i, j)$: 노드 i 로부터 노드 j 까지 홉 수에 의한 거리이며 노드 i 와 노드 j 가 동일한 경로에 있는 경우에도 $d(i, j) = |h_i - h_j|$

정의 1: 슬랏은 어떤 센서 노드가 하나의 패킷을 자신의 부모 노드에게 전달하기 위하여 배타적으로 사용하는 시간 길이를 나타낸다. 슬랏의 시간 싱크오류로부터 패킷 전송의 신뢰성을 보장하기 위하여 RTS, CTS, ACK와 같은 제어 메시지를 사용할 수도 있다.

정의 2: SF (Superframe)은 트리의 싱크노드가 자신의 후손노드들로부터 그들이 생성한 모든 센싱 데이터를 안전하게 수신하기 위하여 필요한 슬랏의 최소 개수로 구성되는 프레임이다

정의 3: 한 SF동안에 노드 j 로부터 노드 i 의 슬랏 요구량을 $D_i(j)$ 로 표기하며, 이 요구량은 $i.T$ 에 속하는 모든 노드가 어떤 간섭을 받지 않고 자신의 센싱 데이터를 노드 j 까지 전달하는데 필요한 슬랏의 수를 나타낸다.

보조정리 4: 노드 i 가 하나의 패킷을 노드 j 까지 전달하는데 요구되는 슬랏의 수는 $d(i, j)$ 와 같다.

증명: 정의 1에 의하여, 데이터 패킷을 1홉 떨어진 노드에 전달하기 위해서는 하나의 슬랏이 필요하다. 결과적으로, 노드 i 에서 노드 j 까지 데이터 패킷의 전달하기 위해서는 거리 $d(i, j)$ 에 해당하는 슬랏의 수가 필요하다. □

정리 1: 노드 j 로부터 노드 i 의 슬랏 요구량을 구하는 공식은 다음과 같다.

$$D_i(j) = \sum_{k \in i.T} d(k, j)$$

증명: 보조정리 4에 의하면, 하나의 데이터 패킷을 노드 k 에서 노드 j 까지 전달하기 위해서 요구되는 슬랏의 수는 $d(k, j)$ 와 같다. 정의 3에 의해서, 노드 i 의 슬랏 요구량은 $i.T$ 에 속하는 모든 노드 k 가 자신의 데이터 패킷을 노드 j 까지 전달하기 위하여 요구되는 슬랏의 수 즉, $d(k, j)$ 의 합이 된다. □

추론 1: 노드 i 는 자신의 부모노드 $i.P$ 로부터의 슬랏 요구량을 다음 식을 사용하여 계산할 수 있다.

$$D_i(i.P) = \sum_{k \in i.C} D_k(i) + |i.T|$$

증명: 정리 1에 의해서,

$$\begin{aligned} D_i(i.P) &= \sum_{k \in i.T} d(k, i.P) = \sum_{k \in i.T} (d(k, i) + 1) \\ &= \sum_{k \in i.T} d(k, i) + \sum_{k \in i.T} (1) = \sum_{k \in i.T} d(k, i) + |i.T| \end{aligned}$$

오른 편 of 첫 번째 항을 다시 정리하면 다음과 같다.

$$\begin{aligned} \sum_{k \in i.T} d(k, i) &= \sum_{k=i} d(k, i) + \sum_{k \in i.C} \left(\sum_{n \in k.T} d(n, i) \right) \\ &= 0 + \sum_{k \in i.C} D_k(i) = \sum_{k \in i.C} D_k(i) \end{aligned}$$

이 등식을 위의 공식에 대입하면 추론식은 증명된다. □

추론 1에서, 오른 편 of 첫 번째 항은 노드 i 로부터 노드 i 의 자식노드들의 슬랏 요구량의 합이 되고, 두 번째 항은 노드 i 자신이 $|i.T|$ 패킷 (자신이 생성한 패킷 1개와 후손 노드들로부터 수신한 패킷의 합인 $(|i.T| - 1)$ 패킷) 들을 $i.P$ 에 전달하기 위하여 필요한 슬랏 요구량이다.

따라서, $D_i(i.P)$ 는 $i.P$ 가 자신에게 할당된 전체 슬랏에서 아들 노드 i 에게 할당해야 하는 슬랏의 수를 나타낸다. 할당하고 남은 슬랏의 수는 노드 $i.P$ 가 자신의 아들 노드들에게서 수신한 패킷들과 자신이 생성한 패킷을 자신의 부모 즉, $(i.P).P$ 에게 보내기 위해서 사용한다.

추론 2: 노드 i 가 부모 노드들 가지고 있지 않다면, 그것의 슬랏 요구량은 다음 D_i 와 같다.

$$D_i = \sum_{k \in i.C} D_k(s)$$

Proof: 노드 i 가 부모 노드를 가지고 있지 않다면, 자신은 패킷을 전송할 필요가 없다. 따라서, 추론 1의 두 번째 항은 0이 된다. 부모를 가지지 않는 유일한 노드는 싱크노드이다. □

추론 2에 의하면, 싱크노드 s 의 슬랏 요구량은 D_s 이고, 해당 트리의 SF가 된다.

III. 슬랏 요구량 기반 슬랏 할당 (DSA) 알고리즘

3.1 수퍼 프레임과 슬랏 요구량

트리 토폴로지는 이미 구성되어 있으며, 모든 노드는 자신의 트리 깊이를 알고 자신의 부모와 아들 노드들에 대한 정보를 가지고 있다고 가정한다. 또한, 모든 트리 노드들은 시간적으로 동기화 되어 있다고 가정한다. 추론 2에 의하면, SF는 해당 트리에서 모든 노드들의 슬랏 요구량을 계산함으로써 얻어진다. 추론 2에 따라서 $D_s = SF$ 를 구할 수 있고,

추론 1에 따라서, 싱크노드에서 터미널 노드까지 순환적으로 슬랏 요구량을 계산할 수 있다.

그림 3을 보면, 3개의 터미널 노드 4, 5, 7이 있으며, $D_4(2) = D_5(3) = D_7(6) = 1$. 노드 3에 대하여, $D_3(2) = D_5(3) + \beta \cdot T_1 (= 2) = 3$. 이런 식으로 $SF = D_s = D_1(s) + D_6(s) = 16$.

3.2 DSA알고리즘

모든 노드 i 는 슬랏 요구량 $D_i(i.P)$ 와 α 와 β 가 추론 1의 첫째 항과 둘째 항에 각각 대응하는 식으로 (α, β)를 진술한다. α 는 자신의 아들 노드에게 재분배되고 β 는 자신의 데이터를 부모 노드에게 전송하기 위해서 예약된다. 싱크노드의 경우에, 추론 2에 따라서 α 는 SF와 같고 β 는 0이 된다는 것을 의미한다.

DSA (Demand-based Slot Scheduling) 알고리즘은 모든 노드가 자신의 자식 노드들에 대하여 슬랏 요구

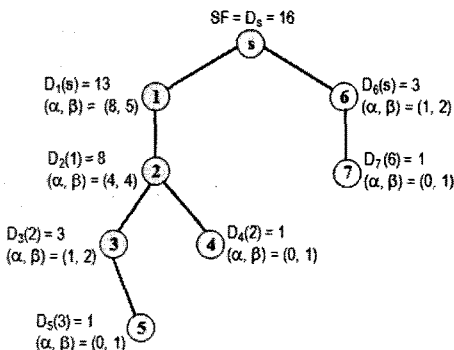


그림 3. 슬랏 요구량 계산 예

량을 알고 있을 때 다음과 같이 동작한다. 각 노드에게 슬랏 번호가 1, 2, 3 등으로 매겨지는 SF를 기준으로 시작 번호를 할당한다. 자식은 자신의 슬랏 요구량에 따라서 SF의 일부분을 할당 받는 식으로 연속적으로 각각의 자식 노드에게 시작 번호를 할당한다. 각 노드가 SF의 일부분을 얻게 되면, 자신의 전송을 위해서 β 슬랏을 보유하고 α 슬랏을 싱크노드가 했던 식으로 자신의 자식 노드들에게 재분배한다. 그림 4는 DSA 알고리즘을 기술한다.

그림 5는 그림 1에 대한 DSA 알고리즘의 수행 결과를 보여준다. SF는 슬랏 요구량 계산 방식에 따라서 16개의 슬랏을 할당 받는다. DSA 알고리즘의 첫 번째 파트에 따라서 노드 1은 13개의 슬랏, 노드 6은 세

```
// We assume that node i knows Dk(i) for its child k
// startPos : the start position of node's demand w.r.t. SF
// SS-REQ(x): synchronization schedule request message
// that includes start position x
At a sink node s:
startPos = 1;
for each node k in s.C do
    Send SS-REQ(startPos) to node k
    startPos = startPos + Dk(s)
endFor
At an intermediate node i:
wait for SS-REQ(x) startPos = x
for each node k in i.C do
    send SS-REQ(startPos) to node k
    startPos = startPos + Dk(i)
endFor
```

그림 4. DSA 알고리즘

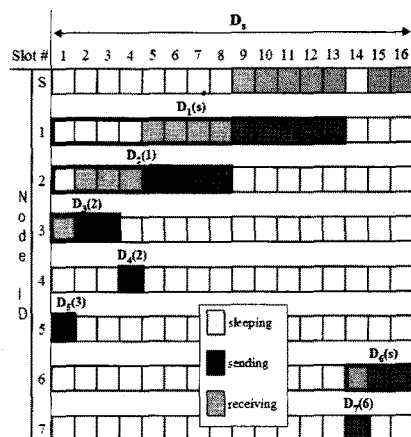


그림 5. 그림 3에 대한 DSA 알고리즘의 적용

개의 슬롯을 할당 받는다. 노드 1은 자신의 전송을 위하여 5개의 슬롯을 확보하고 8개의 슬롯을 자신의 유일한 자식 노드인 노드 2에게 재 분배한다. 이런 식으로 각 노드는 자신의 슬롯 요구량에 대한 시작 슬롯 번호를 알게 된다.

IV. 성능평가

제안된 DSA 프로토콜을 TreeMAC 프로토콜^[9]에서 사용된 FSA 알고리즘과 비교한다. FSA 알고리즘은 Funneling-MAC^[5]과 CSMA^[13]에 비해서 좋은 성능향상을 보인 바 있다. 내용은 트리구성 방식, 시뮬레이션 모델, 그리고 시뮬레이션 결과에 대한 평가로 구성된다.

4.1 트리구성

노드들은 두 가지 노드 상태를 가지는데, 트리에 속하는 노드를 **멤버**, 속하지 않는 노드는 **비멤버**라고 칭한다. 트리는 초기에 트리의 유일한 멤버인 하나의 싱크노드로 구성된다. 멤버는 solicitation 메시지 즉, SOL = (ID, depth)를 브로드캐스트 한다. SOL을 받으면, 비멤버는 join request message 즉, JREQ = (ID)를 멤버에게 보낸다. 해당 멤버는 그 비멤버 노드를 자식으로 받아들인다. 그 비멤버 노드가 J-REQ에 대한 응답으로 ACK를 받으면, 그 멤버를 부모로 취하고 그 자신은 멤버노드로 전환된다. 그 비멤버 노드가 복수의 멤버들에게서 SOL을 받으면, 가장 짧은 트리 깊이를 제공하는 멤버 노드와 짝을 이룬다. 이런 식으로, 모든 노드는 싱크노드까지 최단 거리를 확보할 수 있다.

4.2 시뮬레이션 모델

본 시뮬레이션에서 센서 노드들은 정적이고 100 x 100 (m)의 영역에 분포된다. 싱크노드는 영역의 최상단의 중앙에 위치시킨다. 모든 센서는 데이터를 센싱하고, 헤드를 포함하여 100 바이트 길이의 패킷으로 변환하여 싱크노드에게 보낸다. 시뮬레이션의 주요 파라메타들과 그 값들은 표 1에 정리되어 있다.

4.3 시뮬레이션 결과 및 논의

4.3.1 시뮬레이션 입력 및 측정

모든 노드는 SF 마다 하나의 패킷을 생성한다. 각 노드는 자신에게 할당되는 슬롯 내에서 데이터를 전송할 수 있다. DSA 알고리즘의 효율성을 평가하기 위해서 필터링과 패킷 결합을 다루는 알고리즘의 능력

표 1. 시뮬레이션 파라메터 및 값

파라메타	값
슬롯 크기	8 ms
FSA에서 프레임(f)	3 슬롯
FSA에서 SF	$n \times f$, (n = 노드 수)
DSA에서 SF	토폴로지 종속적
전송 세기	-25 dBm
채널 주파수	2.4 GHz
Pathloss 모델	Two rays
센서 에너지 모델	MICAZ
영역	100 × 100 (m ²)
노드 수	싱크 1개, 노드 25개
시뮬레이션 시간	600 초

을 평가한다.

본 시뮬레이션에서, 패킷은 전송 동안에 손실되지 않는다고 가정한다. 따라서, 본 논문에서는 전송률을 비교 평가하지 않는다. 주어진 네트워크가 모든 생성된 데이터 패킷들을 처리할 수 있는 그런 식으로 노드들이 패킷을 생성한다. 시뮬레이션 입력 데이터는 다음과 같이 생성된다.

복수의 센서 노드들 (nNodes)을 가진 트리가 주어지면, 각 노드는 하나의 SF 시간 길이 동안에 패킷 ID 번호를 가진 하나의 패킷을 생성한다. 패킷 ID를 하기 위하여 다음 함수를 사용한다.

$$rand[1, k*nNodes / |s.C|], k = 1, 2/3, 1/2,$$

여기서 s.C는 싱크노드 s의 자식들의 집합이고, k 값은 패킷의 중복 정도를 나타낸다. k값이 적으면 유일한 ID를 갖는 패킷의 수는 감소하고 패킷의 필터링 수는 증가할 것이다.

동일한 ID를 갖는 복수의 패킷은 동일한 데이터를 센싱하였다고 가정하고 하나를 제외하고 필터링 된다. k값은 데이터 패킷의 중복 정도를 나타낸다. k값이 적으면 유일한 ID를 갖는 패킷의 수는 감소하고 더 많은 수의 패킷이 필터링 될 것이다. 여기서, 동일한 ID를 갖는 패킷의 수를 증가시키기 위해서 상위 범위를 |s.C|로 나눈다. 이것은 동일한 패킷을 갖는 두 개 이상의 패킷이 싱크노드 s의 서로 다른 서브트리(subtree)들에 속해 있으면, 결코 필터링이 이루어지지 않기 때문에 합리적이라고 생각한다. 즉, $k * nNodes / |s.C|$ 를 사용하는 것은 $k * i.nNodes$ 를 사용하는 효과와 평균적으로 동일하다. 여기서, i.nNodes는 i번째 서브트리의 노드의 수를 나타낸다.

4.3.2 패킷 결합 및 필터링 효율성

효율성 지표 (Efficiency Index)로서 EI(A_M)을 사용하여 알고리즘 A와 알고리즘 A의 패킷 결합 (Aggregation)과 필터링 (Filtering) 능력을 평가한다.

$$EI(A_M) = \frac{nBytes - nBytesRcvdAtSink(M)}{nBytes}$$

A ∈ {FSA, DSA}, M ∈ {F, AF} 이며, F는 필터링 기법을 나타내고 AF는 패킷 결합 및 필터링 기법을 동시에 사용하는 것을 의미한다. nBytes는 네트워크의 모든 센서노드에서 생성된 바이트의 수이며, nBytesRcvdAtSink(M)는 기법 M이 사용될 때 싱크 노드에서 수신된 바이트의 수를 나타낸다.

그림 6는 k값에 따라서 알고리즘의 패킷 결합 및 필터링의 능력을 나타낸다. DSA_F와 FSA_F는 DSA와 FSA의 필터링 효율을 나타낸다. DSA가 FSA보다 필터링 능력이 뛰어나다는 것을 보여준다.

FSA_F와 FSA_AF를 비교할 때, FSA 알고리즘은 노드가 데이터 패킷의 수신 및 송신을 번갈아 수행하기 때문에 패킷 결합을 잘 수행하지 못한다. 하지만, DSA_F와 DSA_AF 커버의 차이는 DSA 알고리즘이 패킷 결합을 효과적으로 수행한다는 것을 나타낸다. 차이가 비교적 적은 것은 패킷 결합에 의해서 줄어든 패킷 헤드의 수만을 이득에 반영했기 때문이다. 하지만, 패킷 결합이 증가하면 RTS, CTS, SAK과 같은 제어메시지의 수가 비례하여 줄어든다는 것을 고려하면 이득은 이보다 훨씬 커다는 것을 알 수 있다.

다음 논의에서는 패킷 결합과 필터링을 둘 다 고려하는 DSA_AF와 FSA_AF만을 비교한다. 따라서, AF를 생략하고 DSA와 FSA와 같이 단순 표기를 사용하였다.

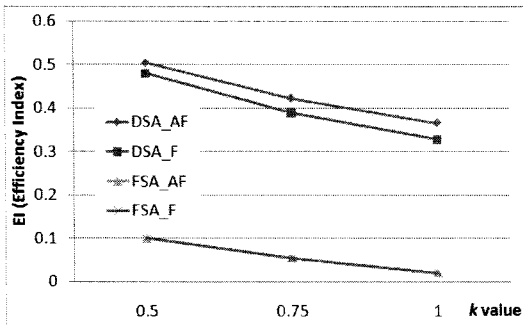


그림 6. 중복 정도에 따른 필터링 및 결합의 효율성 지표 (노드 수: 25개)

4.3.3 부하 균등 및 전력 효율

트리 토폴로지에서는 낮은 트리 깊이에 위치한 노드는 더 많은 패킷을 처리하기 때문에 더 많은 전력을 소모하게 된다. 전력 소모의 불균형은 네트워크 수명을 단축시킨다. FSA와 DSA가 사용될 때, 노드의 트리 깊이에 따라서 처리 부하와 전력 소모의 균형을 평가한다. 트리 깊이의 변화에 따른 트리 상의 노드들이 처리하는 바이트의 수를 비교한다. 다음 평가모델을 사용한다.

$$LOAD(d) = \frac{\sum_{i=1..k_d} (R(n_{d,i}) + T(n_{d,i}))}{k_d}$$

여기서, n_{d,i}는 트리 깊이 d에 위치한 ith 노드이며, k_d는 트리 깊이 d에 위치한 노드의 수, R(n_{d,i})는 n_{d,i}에서 수신하는 패킷의 수, T(n_{d,i})는 n_{d,i}에서 송신하는 패킷의 수, 그리고 LOAD(d)는 트리 깊이 d에 있는 노드의 평균 부하를 나타낸다.

그림 7은 트리 깊이의 변화에 따라 노드의 처리 부하 (수신 및 송신 패킷)를 나타낸다. 일반적으로, 트리 토폴로지에서 노드의 트리 깊이가 더 낮으면, 노드의 처리 부하가 더 높음으로 알려져 있다. 트리 깊이 1의 노드 (D1-Node)의 처리 부하를 생각해보자. DSA_0.50 (k=0.50을 갖는 DSA)은 FSA_0.50에 비해서 약 40%의 처리 부하를 줄인다. 0.75와 1.00의 다른 k값에 대하여, DSA는 FSA에 비해 약 35%정도 처리부하를 줄일 수 있다는 것을 알 수 있다. 트리 깊이가 증가하면 더 적은 수의 패킷을 수신하고 패킷 결합과 필터링의 효과가 줄어들게 된다. 그 결과 DSA와 FSA의 차이가 점차 감소한다는 것을 알 수 있다.

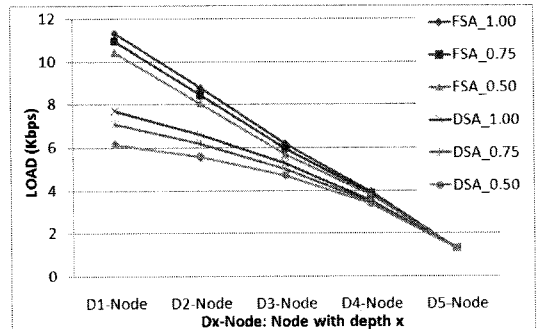


그림 7. 트리 깊이와 k값의 변화에 따른 처리 부하 (노드 수: 25개)

4.3.4 전력 소모

그림 8은 DSA 및 FSA의 노드 위치에 따른 전력 소모를 나타낸다. 커버 모형은 처리 부하의 그것과 유사하다. FSA의 커버를 보면, 트리 깊이가 낮을수록 훨씬 더 많은 데이터를 처리하기 때문에 커버의 기울기가 급격히 감소하는 패턴을 보여준다. 하지만, DSA는 평균 전력 소모에 대하여 FSA에 비해 훌륭한 개선을 보여준다. 서로 다른 k 값에 따른 전력소모의 차이는 FSA의 경우에 거의 없다는 것을 알 수 있다. 이것은 FSA의 필터링 능력 부족에 기인한다. DSA 커버들 사이의 작은 차이는 그림 10에 처리 부하의 적은 차이에서 비롯된다.

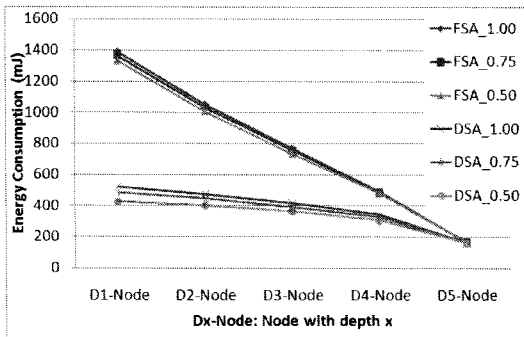


그림 8. 트리 깊이와 k값의 변화에 따른 전력 소모 (25 노드)

V. 결 론

모든 노드가 시간 슬랏을 계층적으로 할당 받는 새로운 DSA 알고리즘을 제안하였다. DSA 알고리즘의 특징은 각 노드에게 수신 슬랏들을 먼저 할당하고 송신 슬랏들을 할당한다. 이러한 슬랏 할당 방식은 패킷의 필터링 및 결합을 용이하게 하고 그 결과 낮은 트리 깊이의 노드들의 패킷 처리부하를 줄일 수 있도록 한다. 시뮬레이션을 통해서 DSA는 패킷 결합, 필터링, 채널 효율성의 측면에서 FSA와 비교하였으며, DSA가 FSA보다 훨씬 뛰어나다는 것을 보였다.

참 고 문 헌

[1] G. Lu, B. Krishnamachari, C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks," *Wiley InterScience: Wireless Communications and Mobile Computing*, Vol.7, No.7, pp.863-875, May 2007.

[2] J. Polastre, J. Hill, D. Culler, "Versatile low power media access for wireless sensor networks," *In Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems*, ACM SenSys'04, pp.95-107, Nov. 2004.

[3] T. V. Dam, K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," *In Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems*, ACM SenSys'03, pp.171-180, November 2003.

[4] W. Ye, J. Heidemann, D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *In Proceedings of the 21st Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Vol.3, pp.1567-1576, June 2002.

[5] G. S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, F. Cuomo, "Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks," *In Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems*, ACM SenSys'06, pp.293-306, Nov. 2006.

[6] J. Kim, J. Lim, C. Pelczar, B. Jang, "RRMAC: A Sensor Network MAC for Real Time and Reliable Packet Transmission," *In Proceedings of International Symposium Consumer Electronics*, pp.1-4, April 2008.

[7] V. Rajendran, K. Obraczka, J. J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," *Wireless Networks*, Vol.12, No.1, pp.63 - 78, Feb. 2006.

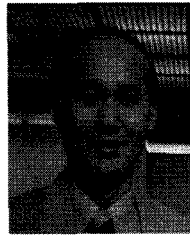
[8] I. Rhee, A. Warriar, M. Aia, J. Min, M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, Vol.16, No.3, pp.511-524, Jun. 2008.

[9] S. Wen-Zhan, H. Renjie, S. Behrooz, L. Richard, "TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks," *Pervasive and Mobile Computing*, Vol.5, No.6, pp.750-765, Dec. 2009.

- [10] I. Rhee, A. Warriier, J. Min, L. Su, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, Vol.8, No.10, pp.1384-1396, Oct. 2009.
- [11] T. D. Han, H. Oh, "A Topology Management Routing Protocol for Mobile IP Support of Mobile Ad Hoc Networks," LNCS 5793, pp.341-346, Sep. 2009.
- [12] H. Oh, "A tree-based approach for the Internet connectivity of mobile ad hoc networks," *Journal of Communications and Networks*, Vol.11, No.3, pp.261-270, Jun. 2009.
- [13] A. Woo, D. E. Culler, "A transmission control scheme for media access in sensor networks," *In Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp.221-235, Jul. 2001.

오 훈 (Hoon Oh)

중신회원



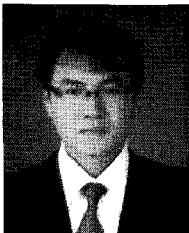
1981년 성균관대학교 전자공학
학사
1993년 텍사스A&M대학교 전
산학 석사
1995년 텍사스A&M대학교 전
산학 박사
1996년 삼성전자 중앙연구소
수석

2005년~현재 울산대학교 컴퓨터정보통신공학부 부
교수

<관심분야> 실시간 시스템, 임베디드 시스템, 상황
인식컴퓨팅, 이동 애드 혹 및 센서 네트워크 프
로토콜

김 제 욱 (Je-Wook Kim)

정회원



2004년 울산대학교 공학사
2007년 울산대학교 정보통신공
학석사
2010~현재 울산대학교 자동차
선박대학원 박사과정
<관심분야> 상황인식컴퓨팅, 무
선통신