

저 에너지의 취침 기상 사이클로 작동하는 무선 센서 네트워크 노드들을 위한 시간 동기화

(Time Synchronization for WSN Nodes Operating on Low-Energy Sleep-Wake Cycles)

윤 호 중 † 윤 주 성 †
(Hojung Yun) (Joosung Yun)

이 승 구 ††
(Sunggu Lee)

요 약 무선 센서 네트워크에서 중요한 이슈 중 하나는 배치되는 센서 노드들이 최대한 오래 지속되도록 전력 소비를 줄이는 것과, 네트워크를 시간 동기화 하는 것이다. 기존의 저 전력 시간 동기화 알고리즘은 취침모드에 대한 고려 없이 교환되는 패킷의 수를 줄이거나, 네트워크 토폴로지에 따라 다른 시간 동기화 방법을 사용하는 하이브리드 방법이 전부였다. 무선 센서 네트워크에서 효율적인 에너지 절약 방법은 노드의 취침을 통한 duty cycle의 감소이기 때문에, 본 논문에서는 노드의 취침을 통해 duty cycle을 줄일 수 있는 시간 동기화 알고리즘을 제안하였다. 알고리즘 제안 시, 실험을 통해 노드 구성요소의 특성을 파악하여 모델을 세운 후 접근하였다. 제안한 알고리즘의 성능 파악을 위해 네트워크에 요구되는 시간 동기화 정확도가 있을 때, 이를 만족시키기 위한 시간 동기화 주기 값을 구해보았으며, 노드 사이의 편차율을 구할 때 걸리는 시간

을 확률적인 개념을 사용해 분석하여 보았다. 제안된 알고리즘을 3×3 격자(grid) 구조에서 구현해본 결과, duty cycle 5%, 시간동기 주기 10초일 때, 107.57 μs의 평균 시간동기 오차를, duty cycle 2.5%, 시간동기 주기 20초일 때, 130 μs의 평균 시간동기 오차를 보여주었다.

키워드 : 무선 센서 네트워크, 시간동기화, 저 에너지

Abstract Previous low-energy time synchronization methods have mainly focused on reducing the number of transmission or reception packets. However, this paper proposes a method that reduces the percentage of time a node has to be awake (the duty cycle), assuming that a periodic sleep-wake cycle is used to conserve energy. Based on our experience with actual WSN devices, a system model is proposed, and the potential performance of the proposed method, with different parameter values, is analyzed. To further demonstrate the feasibility of our method, experiments were conducted using nine WSN devices in a 3×3 grid network topology. The results show the average synchronization error is 107.57 μs in duty cycle 5% and synchronization period 10 sec, and 130 μs in duty cycle 2.5% and synchronization period 20 sec.

Key words : Wireless Sensor Network, Time Synchronization, Low-Energy

1. 서 론

무선 센서 네트워크를 가능하게 해주는 핵심적인 기술 중 하나는 시간 동기화 기술이다. 모든 분산 시스템의 실시간 시간이 특정 기준 시간과 완벽히 동기화가 된다면, 분산 시스템에서 발생하는 모든 사건들을 기준 시간으로 기록함으로써 사건 간의 전후 관계와 시간 간격을 알아 낼 수 있다. 또한 사용되는 통신 방식들의 패킷 지연 시간을 측정할 수 있다. 시간 동기화는 분산 처리를 하는 시스템이라면 어느 곳에서나 필요하기 때문에 여러 시간 알고리즘이 제안되었다. 하지만 제한된 처리 능력과 전력을 가진 센서 노드들로 구성된 무선 센서 네트워크에서 기존의 분산 시스템에서 사용되는 시간 동기화 기술을 그대로 적용하기에는 문제가 있다 [1]. 따라서 무선 센서 네트워크를 위해 기존의 동기화 기법이 수정되거나 새로운 방법들이 제안되었다 [2-5].

기존의 제시된 시간 동기화 알고리즘은 모두 취침모드에 대한 고려 없이 제안된 알고리즘이며, 기존의 에너지 절약을 위한 시간 동기화 알고리즘도 교환되는 패킷의 수를 줄이거나, 주변 노드의 상황에 따라 다른 시간 동기화 알고리즘을 사용하는 하이브리드 방법이 전부였다. 무선 센서 네트워크에서 효율적인 에너지 절약 방법은 노드의 취침을 통한 duty cycle의 감소이기 때문에, 본 논문에서는 노드의 취침을 통해 duty cycle을 줄일 수 있는 시간 동기화 알고리즘을 제안하였다.

· 이 논문은 2009 한국컴퓨터종합학술대회에서 '저 에너지 Sleep-Wake Cycle로 작동하는 무선 센서 네트워크 노드들을 위한 시간 동기화'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : POSTECH 전자과
hojung1903@postech.ac.kr
acajuri@postech.ac.kr

†† 종신회원 : POSTECH 전자과 교수
slee@postech.ac.kr
논문접수 : 2009년 8월 13일
심사완료 : 2009년 11월 19일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용하지 않으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨터의 실제 및 레터 제16권 제3호(2010.3)

2. 관련 연구

무선 센서 네트워크에서 저 전력을 위한 연구는 크게 에너지 절약을 위한 시간 동기화 알고리즘[2,3]과 노드의 취침을 통한 저 전력 알고리즘[4,5]으로 나눌 수 있다.

에너지 절약을 위한 시간 동기화 알고리즘으로는 교환되는 패킷의 수를 줄이는 방식의 Energy Efficient Time Synchronization Protocol(ETSP)[2], 이웃 노드의 개수에 따라 다른 알고리즘을 적용하는 하이브리드 방식의 Hybrid Energy-Aware Synchronization Algorithm[3]이 있다. 노드의 취침을 고려한 알고리즘은 네트워크에 필요 이상으로 많은 노드가 존재할 때, 일부 노드를 취침모드로 두어 네트워크의 지속시간(lifetime)을 늘이는 ASCENT[4], 네트워크를 정사각형 형태의 클러스터로 나누고 클러스터 내에 하나의 노드를 제외한 나머지 노드를 모두 취침모드로 전환하는 GAF[5] 등이 있다. 하지만 이러한 알고리즘들은 시간 동기화, 노드의 스케줄링을 각각 고려하였고, 저 전력을 위해 취침 모드를 고려한 시간 동기화 알고리즘은 없는 실정이다.

3. 플랫폼 모델(platform model)

무선 센서 네트워크에서 사용되는 센서 노드의 각각의 구성 요소(component)가 실제 동작 시 보이는 특성을 기초로 하여 모델을 세워 보았다. 모델에는 임베디드 소프트웨어 지원센터(CEST)에서 개발된 CFR-PP100 RF 모듈이 사용되었다. 다음은 각각의 구성 요소의 실제 동작 시 특성에 기초한 모델이다.

3.1 MPU model

MPU로 사용되는 Atmega 128L은 에너지 절약을 위해 idle, ADC Noise Reduction, power-down, power-save, standby, extended standby의 6가지 취침 모드를 제공한다[6]. 이중에 에너지를 가장 절약할 수 있는 power-down모드는 노드의 모든 타이머가 동작하지 않아서 시간 동기화에 사용하기 힘들다. 따라서 센서가 이벤트를 감지한다 하여도 이벤트 발생시간을 알 수 없으므로 시간 동기화에 사용하기 힘들다. 우리는 시간 동기화를 이룬 후에 노드는 자고 있지만 센서를 통해 실시간으로 발생하는 이벤트를 감지하는 상황을 구현하였기 때문에 power-down모드 다음으로 에너지 절약이 되고, 노드의 취침 시 MPU는 대기모드로 들어가 있지만, 타이머는 켜져 있는 power-save모드를 사용하였다.

또한 Atmega 128L은 높은 주파수(8MHz)로 동작하는 내부 타이머와 낮은 주파수(32.768 KHz)로 동작하는 외부 타이머를 가지고 있다. 취침 모드가 되면 내부 타이머는 동작을 멈추고, 외부 타이머만 동작하게 된다. 취침모드에서 노드가 깨어나게 되면 내부 타이머는 꺼

져있었기 때문에 자신의 시간을 잃어버리게 되고, 또한 새로 시작되어도 초기에 일정 시간 동안 타이머의 동작이 불안정하게 동작하는 start up delay를 가진 후에 타이머가 안정화된다. 이와 같은 이유로 취침과 기상이 반복되는 상황에서는 내부 타이머를 사용하는 것은 어려움이 따른다. 따라서 우리는 취침과 기상에 상관없이 안정적으로 동작하는 외부 타이머를 사용하였다.

3.2 Radio model

RF 송수신기로 사용되는 CC2420은 power down, transmission, reception, idle의 4가지 모드로 나누어진 데[7]. Power down 모드에서 송수신기를 켜 후 1ms의 시간이 지나면 idle 모드 상태로 전환된다. Idle 모드에서 상황에 따라 메시지를 전송할 수 있는 전송 모드와 메시지 수신 가능한 수신 모드로 바꿀 수 있다. Idle 모드, 전송 모드, 수신 모드 사이에는 0.192 ms의 스위칭 딜레이가 존재하며, idle, 전송, 수신 모드에서 power down 모드로는 딜레이 없이 모드 변경이 가능하다.

3.3 Power Source model

CFR-PP100 RF 모듈의 경우 배터리에서 공급되는 전원이 레귤레이터(regulator)를 통해 정격 전압으로 바뀌어 노드의 각 구성 요소에 사용된다. 각 구성요소의 전력 소모를 측정해 보기 위해서 MPU를 켜둔 채로 송수신기를 on/off시켰을 때와 MPU를 power save 모드를 8초, active 모드를 2초의 10초를 주기로 반복시킨 경우에 대해 노드의 지속시간을 측정해 보았다. 위의 경우를 레귤레이터가 있는 경우와 없는 경우로 나누어 실험해 보았다. 실험 결과 power save 모드와 active모드를 반복시킨 경우 레귤레이터가 있을 시에는 1.3V 떨어지는데 480시간이 걸렸던 것이, 레귤레이터가 없을 시에는 1,344시간이 걸리는 것을 알 수 있었다(그림 1). 이는 레귤레이터가 있는 것이 전력을 2.8배 정도 더 소모하는 것이다. 이로서 우리는 레귤레이터가 전력 소모의 큰 원인이라는 것을 알 수 있었으며, 따라서 우리는 노드의 지속시간을 늘이기 위해서 시간 동기화 알고리즘 구현 시에는 레귤레이터를 제거하고 구현하였다.

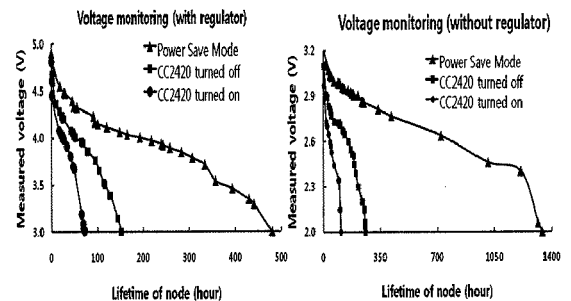


그림 1 레귤레이터의 유무에 따른 노드의 전압 변화

4. 제안된 알고리즘에 대한 설명

무선 센서 네트워크의 노드들이 취침과 기상을 반복적으로 하는 상황에서 사용될 수 있는 시간 동기화 알고리즘에 대해 제안한다. 제안된 알고리즘은 시간 동기화를 이루기 위해 노드간의 정보를 교환하고, 교환된 정보를 사용하여 시간 동기를 이루는 INITSYNC 단계, 노드의 취침과 기상에 따른 INITSYNC 단계를 반복하는 RESYNC 단계의 두 단계로 구성된다.

4.1 INITSYNC 단계

INITSYNC 단계는 시간 동기를 이루기 위해 노드간의 정보를 교환하고, 이를 사용해 무선 센서 네트워크의 초기 시간 동기화를 이루는 단계이다. 기준 노드가 자신의 ID와 시간 값이 담긴 INITSYNC 패킷을 전송하면서 시작된다. 이를 받은 노드는 자신의 시간 값과 노드 ID를 추가로 포함한 후, 이 패킷을 다른 노드로 전송한다. 네트워크의 모든 노드에 이러한 패킷 전송의 과정이 2번 반복된다. 이 후에 각 노드는 기준 노드의 전송 시간 값과 자신의 수신 시간 값을 사용하여 클럭 편차율(Δ)을 구해서 자신의 시간을 기준 노드의 시간에 맞춘다(식 (1)).

$$\Delta = \frac{T_2 - T_1}{R_2 - R_1} (local_timer - R_2) + T_2 \quad (1)$$

Where T_x : Transmission Timestamp in The Packet

R_x : Reception Timestamp of Local Timer

노드들은 시간 동기화 직후 MPU를 power save 모드, 송수신기를 power down 모드로 전환시켜 취침모드로 들어간다.

4.2 RESYNC 단계

취침모드에 들어간 노드들은 MPU와 송수신기가 꺼져 있지만, 센서는 켜져 해당 지역을 감지하고 있기 때문에, 전체 네트워크의 에너지 소모는 크게 줄어든다. 하지만 노드들이 사용하는 외부 타이머는 낮은 동작 주파수로 동작하는데다가, 타이머 간에 클럭 편차율이 다르기 때문에 일정 시간이 지나면 서로의 시간이 달라지는 현상이 발생한다. 따라서 주기적으로 노드들의 시간을 다시 맞추어주는 RESYNC 과정이 필요하다. RESYNC 과정은 기준 노드로 부터 다시 한번 INITSYNC 패킷이 발생하면서 시작된다. RESYNC 단계에서는 노드들은 자신과 기준 노드와의 클럭 편차율을 이미 알고 있기 때문에 동기화를 위해 기준 노드의 시간 값 패킷을 2번 받을 필요 없이 1번만 받으면 시간 동기화를 이룰 수 있다. 하지만, 각 노드는 INITSYNC 단계에서 구해진 클럭 편차율과 주어진 시간동기 주기를 가지고 패킷 수신 시간을 예상할 수 있는데, 이 예상 값과 실제 수신 시간 값의 오차가 크게 되면, 기존의 수신 값과 이번에

받은 수신 값을 사용하여 클럭 편차율을 다시 구해서 사용하게 된다.

각 노드는 받은 패킷을 이웃 노드로 전달한 후 바로 취침모드로 전환한다. 정해진 주기마다 RESYNC 단계를 반복하여 네트워크의 시간 동기화를 유지한다.

RESYNC 단계에서는 노드들이 INITSYNC 패킷을 전달하고, 바로 취침모드로 들어가 정해진 주기 후에 일어나 시간 동기화를 하는 방식을 사용한다. 이러한 방식의 사용은 주기가 기준 노드에서 멀어지면서 자동으로 전송의 지연시간만큼 이동(shift)되는 현상을 발생시킨다(그림 2). 이렇게 주기가 자동으로 이동되는 방식의 장점은 메시지 교환 과정에서의 전송 지연시간에서 오는 불확실성 때문에 노드들이 불필요하게 깨어있을 필요가 없다는 것을 의미한다. INITSYNC 단계에서 이미 한번의 전송 지연 시간을 지난 후에 패킷을 받고 이 시간을 기준으로 주기만큼 취침과 기상을 반복하므로 RESYNC 단계에서는 자동으로 지연시간들이 고려되어 있는 것이다. 이는 네트워크의 규모가 커질수록 보다 많은 에너지 절약을 가져올 수 있는 방식이다.

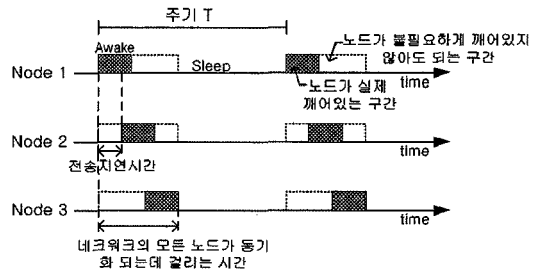


그림 2 RESYNC 단계에서 자동으로 주기가 패킷 교환 지연시간만큼 이동(shift)되는 현상

5. 제안된 알고리즘의 수학적 분석

2가지 측면에서 제안된 알고리즘의 성능을 수학적으로 분석하였다. 첫째는 시간 동기화 과정에서 노드사이의 편차율을 구할 때 걸리는 시간을 확률적인 개념을 사용해 분석하여보고, 둘째는 무선 센서 네트워크에 요구되는 정확도(최대오차)가 있을 때, 이를 만족시키는 시간 동기의 주기 값을 구해보았다.

5.1 시간동기 안정화 시간 분석

시간 동기를 이루기 위해서 노드는 다른 노드에 대한 시간 값과 그에 대한 자신의 시간 값을 이용해 편차율 방정식을 구한다. 편차율 방정식은 직선의 방정식으로 기울기 값과 y절편 값을 갖는다. 기울기를 편차율, y절편을 오프셋이라 한다. 이 때 계산하는 시간 값은 전송의 지연 혹은 랜덤 하게 발생하는 클럭의 잠움으로 인

하여 오차가 발생할 수 있다. 이 시간 값의 오차는 매우 작은 값이더라도 편차를 방정식을 구하게 되면 큰 오차를 갖는 편차를 방정식이 된다. 따라서 확률적으로 올바른 편차를 방정식을 구해 시간동기 오차가 안정화 되는 시간을 분석하고자 한다.

노드의 시간동기 편차율과 오프셋을 계산하기 위해 편차를 방정식의 시작점, 편차를 방정식의 끝점을 잡는다. 시작점은 먼저 얻은 시간 값이고, 끝점은 시작점 보다 상대적으로 이후에 얻은 값이다. 편차를 방정식을 구하고, 이 후에 시간이 편차를 방정식으로 예상한 시간보다 큰 오차가 발생하게 되면 끝점의 좌표를 새로운 시간 값으로 바꾸고 다시 편차를 방정식을 구한다. 만약 이후에도 앞선 상황이 계속 반복된다면 시작점이 잘못되었다는 판단으로 시작점을 바꾼다.

확률적 분석을 위해 노드의 개수를 n , 특정 시간에서 시간 값에 대한 오차가 발생할 확률을 e , 편차를 방정식이 잘못되었을 때 끝점을 다시 잡는 수행횟수를 N , 시간 값을 주고받는 주기를 T 라고 한다. 또한 새로운 시작점을 잡는 과정을 반복하는 것을 주기 P 라고 한다.

특정 시점에서 n 개의 노드가 올바른 시간 값을 갖는 확률 P_t 는 $P_t = (1-e)^n$ 이고, 하나라도 올바르지 못한 시간 값을 갖는 확률 q^t 는 $q^t = 1 - P_t$ 이 된다. 주기 P 안에서는 시작점이 올바른 값이어야 하고, 이후 N 번까지 끝점에서 오차가 발생해도 된다. 따라서 주기 P 안에서 k 번째 점에서 제대로 된 편차를 방정식을 구할 확률 변수를 X 라 하고 X 의 확률을 구하면,

$$P[X] = p_i^2 q_i^{N-1}, P[X \leq k] = p_i (1 - (q_i)^k) \quad (2)$$

이후 주기 P 가 K 번만큼 반복해 제대로 된 편차를 방정식을 구하는 확률 변수를 Y 라 하고 Y 의 확률을 구하면,

$$P[Y] = X(1-X)^{K-1}, P[Y \leq K] = 1 - (1-X)^K \quad (3)$$

따라서 시간 t 에 대해 올바른 편차를 방정식을 구할 확률 Z 는

$$Z = TNY, P[Z \leq t] = (1 - (1-X)^{NT})^{\frac{t}{T}} \quad (4)$$

실제 구현과 같은 상황을 만들기 위해 9개의 노드로, 노드당 시간 값 오차가 0.01의 확률로 발생하고, 편차를 방정식의 끝점을 바꾸는 횟수를 3일 때 확률 값을 구해보았다. 99.9%의 확률로 제대로 된 편차를 방정식을 구하기 위해 시간 동기 주기가 $T=10$ 일 때 88초, $T=20$ 일 때 178초, $T=30$ 일 때 266초가 걸린다는 것을 계산 할 수 있었다.

5.2 요구되는 정확도(최대오차)를 만족시키는 시간 동기화 주기에 대한 분석

분석에 앞서 2.3절에서 설명한 4가지 요인[4]에 의해 클럭의 증가는 직선 형태가 아닌 잡음(noise)이 있는 상

태로 나타나게 되는데 분석에서는 단순화를 위해서 1차 선형으로 증가한다고 가정한다.

시간 동기화가 이루어진 직후의 오차를 E_INIT이라고 한다. E_INIT은 시간 동기화 과정 중 패킷의 지연에 의해 생긴 오차 E_d 와 편차율이 잘못 구해져 생긴 오차 E_t 로 나눌 수 있다. 패킷에 지연에 의해 생긴 오차 E_d 는 다음과 같이 구하여 진다. 노드 1에 의해 T1의 시간에 전달된 패킷이 노드 2에서 T2의 시간에 수신된 경우에 $T2=T1+S1+P1 \rightarrow 2+R2$ (여기에서 S1은 노드 1의 send time과 access time, $P1 \rightarrow 2$ 는 노드 1에서 노드 2로의 propagation delay, R2는 노드 2의 receive time)와 같이 나타낼 수 있다. SYNC단계와 RESYNC단계에서 노드들은 패킷을 전달만 하기 때문에 노드 2가 패킷을 노드 3로 전달하는 경우 $T3=T2+S2+P2 \rightarrow 3+R3$ 와 같이 나타낼 수 있다. 또한 노드들이 같은 하드웨어를 사용한다고 하면 receive time은 모두 같고, propagation delay는 전송 범위 내에서 같다고 볼 수 있기 때문에 이를 일반화 하면, 기준 노드로부터 주어진 무선 센서 네트워크

의 최대 홉수가 m 이라고 했을 때, $E_d = \sum_{i=1}^m S_i + m(P+R)$ 와 같이 나타내어진다. 편차율이 잘 못 구해져 생긴 오차 E_t 는 5.1절에서 다루었기 때문에 여기서는 다루지 않기로 한다.

주어진 네트워크의 노드 중 클럭 편차율이 가장 큰 노드의 편차율을 D_{max} , 가장 작은 노드의 편차율을 D_{min} , 요구되는 시간 동기화 정확도를 δ 라고 한다면 시간 동기화 주기 T 는 다음과 같이 구해질 수 있다.

$$T = \frac{\delta - E_d - E_t}{D_{max} - D_{min}}, \text{ where: } E_d = \sum_{i=1}^m S_i + m(P+R) \quad (5)$$

6. 실험 결과

제안된 알고리즘의 성능 확인에 앞서서 실험에 사용될 노드의 클럭 편차율을 측정하였다. 각 노드 타이머의 현재시간을 정확히 알기 어렵기 때문에, 하나의 노드를 기준노드로 선정하여 기준노드와의 상대 클럭 편차율을 구하였다. 실험의 사용될 9개의 노드들 중에서 최대 클럭 편차율은 6.79 μ s/sec, 최소 클럭 편차율은 0.049 μ s/sec 였다. 따라서 시간 동기가 정확히 이루어지더라도 10초 후에는 약 67.4 μ s의 시간 오차가 벌어진다는 것을 알 수 있다.

실험에는 CFR-PP100 RF 모듈을 사용하였으며, 9개의 노드를 3×3 격자(grid)구조로 배치하고, 시간동기 주기를 10초, 20초로 설정했다. 각각의 경우에 대해 duty cycle은 5%, 2.5%로 설정하여 100번의 실험을 진행하였다(그림 3,4).

시간동기 주기 10초의 경우, 노드가 시간동기를 위해 깨어있는 시간은 0.5ms이다. Duty cycle이 5%이기 때문에 sleep-awake cycle을 사용하지 않는 경우(duty cycle 100%)에 비해 노드의 사용시간이 20배로 늘어나는 것을 알 수 있다. 반면에 시간동기 평균 오차는 136.2 μ s였다. 또한 초기에 클럭 편차율을 제대로 구하는 시간은 72초가 걸렸으며, 이 이후에 평균 시간동기 오차는 108.8 μ s이다(표 1, 그림 3). 시간동기 주기 20초의 경우, duty cycle은 2.5%이기 때문에 sleep-awake cycle을 사용하지 않는 경우에 비해 노드의 사용시간이 40배로 늘어나는 것을 알 수 있다. 반면에 시간동기 평균 오차는 167.3 μ s였다. 초기에 클럭 편차율을 제대로 구하는 시간은 152초가 걸렸으며, 이 이후에 평균 시간동기 오차는 127.1 μ s이다(표 1, 그림 4). 이러한 오차는 duty

cycle의 감소로 노드의 사용시간이 20배, 40배로 증가한 것에 비하여, 상대적으로 크게 증가하지 않은 결과이다 (기존의 시간동기 알고리즘의 오차 20-50 μ s [2-5]).

7. 결론

기존의 에너지 절약을 위한 시간 동기화 알고리즘은 취침모드에 대한 고려 없이 교환되는 패킷의 수를 줄이거나, 주변 노드의 상황에 따라 다른 시간 동기화 방법을 사용하는 하이브리드 방법이 전부였다. 무선 센서 네트워크에서 효율적인 에너지 절약 방법은 노드의 취침을 통한 duty cycle의 감소이기 때문에, 본 논문에서는 노드의 취침을 통해 duty cycle을 줄일 수 있는 시간 동기화 알고리즘을 제안하였다. 알고리즘 제안 시, 실험을 통해 CFR-PP100 RF 모듈의 구성요소 특성을 파악하여 모델은 세운 후 접근하였다. 제안한 알고리즘의 성능 파악을 위해 센서 네트워크에 요구되는 정확도가 있을 때, 이를 만족시키기 위한 시간 동기화의 주기 값을 구해보았으며, 시간 동기화 과정 중에 노드 사이의 편차율을 구할 때 걸리는 시간을 확률적인 개념을 사용해 분석하여 보았다. 제안된 알고리즘을 3×3 격자(grid)구조에서 구현해본 결과, duty cycle 5%, 주기10초일 때, 107.57 μ s의 평균 시간동기 오차율, duty cycle 2.5%, 주기 20초일 때, 130 μ s의 평균 시간동기 오차를 보여주었다. 이러한 오차는 duty cycle의 감소로 network lifetime이 크게 증가한 것에 비해서 상대적으로 크게 증가하지 않은 결과이다.

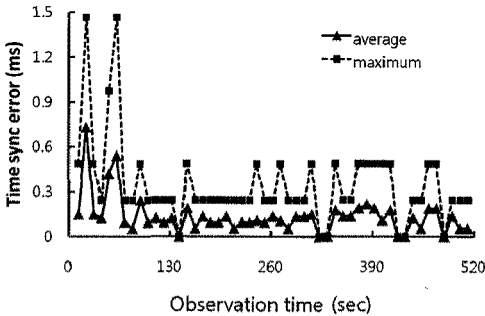


그림 3 9개의 노드, 3 대 3 격자(grid)구조의 네트워크에서 시간 동기 주기 10초일 때, 시간 오차 그래프

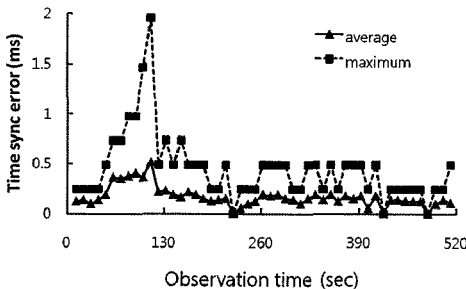


그림 4 9개의 노드, 3 대 3 격자(grid)구조의 네트워크에서 시간 동기 주기 20초일 때 시간 오차 그래프

표 1 Duty cycle과 시간 동기 오차

Duty cycle(%)	100	5	2.5
The initial stabilizing time (sec)		72	152
The average time synchronization error (μ s)	20 50	136.2	167.3
The average time synchronization error after initial stabilizing time (μ s)	20-50	108.8	127.1
Network lifetime	1	20	40

참고 문헌

- [1] J. Elson and D. Estrin, "Time Synchronization for Wireless Sensor Networks," *15th IEEE IPDPS*, San Francisco, CA, USA, pp.186, April 2001.
- [2] K. Shahzad, A. Ali and N. D. Gohar, "ETSP: An Energy-efficient Time Synchronization Protocol for Wireless Sensor Networks," *22nd International Conference on Advanced Information Networking and Applications*, Mar. 2008.
- [3] R. Akl and Y. Saravanos, "Hybrid Energy-Aware Synchronization Algorithm in Wireless Sensor Networks," *18th IEEE PIMRC'07*, Sep. 2007.
- [4] K. Alzoubi, X. Li, Y. Wang, P. Wan and O. Frieder, "Geometric spanners for wireless ad hoc network," *IEEE Transactions On Parallel and Distributed System*, vol.14, May. 2003.
- [5] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," *In Proceedings of ACM MobiCom*, pp. 70-84, 2001.
- [6] Atmega 128L, <http://www.atmel.com>
- [7] Chipcon CC2420 radios. <http://www.chipcon.com>.