

무선 센서네트워크에서의 시각동기를 위한 실시간 클럭 스큐 추정

허 경[†]

요 약

무선 센서네트워크에서의 시각동기는 Wireless USB, WBAN 등의 MAC 계층에서부터 응용 계층에 이르기까지 거의 모든 계층에서 다양한 목적을 위해 매우 중요한 기술이다. 본 논문에서는 무선 센서네트워크에서의 시각동기를 위한 실시간 클럭 스큐 추정 방법을 제시한다. 재귀적 최소제곱법을 통해 오프셋 보정 정보들을 얻을 때마다 클럭 스큐가 실시간적으로 추정 및 갱신되며, 아울러 스큐 추정을 위해 각 센서노드에 저장해야 할 정보를 최소화한다. 제안한 클럭 스큐 추정 방법은 기존의 클럭 오프셋 보정 방법과 쉽게 통합될 수 있으며, 이 경우 보다 정확하고 효율적인 시각동기화가 가능해진다. 시뮬레이션 및 실험 결과를 통해 제안한 클럭 스큐 추정 방법을 통한 시각동기 정확도의 향상을 보인다.

Realtime Clock Skew Estimator for Time Synchronization in Wireless Sensor Networks of WUSB and WBAN

Kyeong Hur[†]

ABSTRACT

Time synchronization is crucial in wireless sensor networks such as Wireless USB and WBAN for diverse purposes from the MAC to the application layer. This paper proposes online clock skew estimators to achieve energy-efficient time synchronization for wireless sensor networks. By using recursive least squares estimators, we not only reduce the amount of data which should be stored locally in a table at each sensor node, but also allow offset and skew compensations to be processed simultaneously. Our skew estimators can be easily integrated with traditional offset compensation schemes. The results of simulation and experiment show that the accuracy of time synchronization can be greatly improved through our skew compensation algorithm.

Key words: Time Synchronization(시각동기), Skew Estimation(스큐 추정), Skew Compensation(스큐 보정), Wireless Sensor Networks(무선 센서네트워크)

1. 서 론

일반적으로 센서네트워크를 구성하는 노드들은 자체적으로 타이머(클럭)를 보유하고 있으며 MAC

(Medium Access Control) 계층, 네트워크(Network) 계층, 응용(Application) 계층에서의 요구로 인해 일부 또는 전체 노드의 타이머가 동기화되는 것이 필요하다. 예를 들어, TDMA를 사용하는 MAC 알고리즘

※ 교신저자(Corresponding Author) : 허경, 주소 : 경기도 안양시 만안구 삼막로 155 교사교육센터 410호(430-739), 전화 : 031)470-6292, FAX : 032)470-6299, E-mail : khur@ginue.ac.kr

접수일 : 2011년 11월 10일, 수정일 : 2012년 7월 11일
완료일 : 2012년 10월 17일

[†] 정회원, 경인교육대학교 컴퓨터교육과 부교수

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2010-0002366). 또한 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2011-0016145).

에서는 타임 슬롯(slot)을 나누고 할당하는 데에 있어 정확한 시각동기가 요구된다. 그리고 네트워크 계층에서 라우팅을 하는 데에 있어, 시각동기화가 되어 있다면 좀 더 빠르고 효율적으로 라우팅을 처리할 수 있게 된다. 또한 센서노드들에 의해 감지된 이벤트 정보가 싱크노드(sink node) 또는 서버로 보고될 때, 이 이벤트를 감지한 시각 정보가 해당 메시지에 포함되어 함께 전달된다면 싱크노드 또는 서버의 어플리케이션에서는 좀 더 고차원적인 처리가 가능해진다. 전달되는 이벤트가 다수의 센서노드에 의해 감지된 것이라면 시각 정보를 통해 동일 이벤트인지 또는 다른 시점에 발생한 이벤트인지를 판단할 수 있으며, 또한 침입 감지 이벤트가 발생했을 경우라면 감지된 시각 정보들을 바탕으로 침입된 물체의 추적(tracking)이 가능하게 된다. 아울러 통신 횟수를 줄이기 위한 데이터 퓨전(data fusion), 데이터 압축(data suppression) 등 네트워크 내부 처리(in-network processing)를 하는데 있어서도 센서노드들 간에 정확한 시각동기가 되어있어야 가능하다.

시각동기의 문제는 인터넷 또는 LAN 환경에서 활발히 연구되기 시작해왔으며, NTP[1] 등의 프로토콜을 통해 인터넷 분산 환경 상에 연결된 단말들 간의 클럭을 동기화시킬 수 있다. 그러나 Wireless USB 및 WBAN 등과 같은 무선 센서네트워크는 인터넷 또는 LAN 환경과는 다르기 때문에 무선 센서네트워크를 위한 시각동기 프로토콜은 그만의 특수한 요구사항을 필요로 한다. 첫째, 배터리로 동작하는 센서노드의 가용 에너지는 한정되어 있기 때문에 시각동기는 반드시 에너지 효율적인 방법으로 동작할 수 있도록 구현되어야 한다. 둘째, 센서노드는 보통 크기가 작으므로 컴퓨팅 능력과 저장 공간에도 제약이 따른다. 따라서 NTP 또는 GPS와 같은 전통적인 시각동기 방법들은 복잡도와 에너지 관련 이슈, 경제성, 제한된 노드 크기 등으로 인해 무선 센서네트워크에 적용하기에는 적합하지 않다[2].

앞서 언급한 요구사항들을 고려하여, 본 논문에서는 실시간 클럭 스쿼 추정을 통한 무선 센서네트워크에서의 시각동기 기법을 제시한다. 클럭 스쿼는 재귀적 최소제곱법(recursive least squares)을 통해 추정 및 갱신되며, 오프셋 보정과 스쿼의 보정은 동시에 수행되도록 한다. 제안한 추정 방법을 통해 스쿼 보정을 위해 각 센서노드에 저장되는 데이터의 양을

최소화할 수 있다. 아울러 시각동기의 정확도를 향상시켜 빈번한 재동기에 따른 통신 부담을 줄일 수 있도록 한다.

2. 관련연구

센서네트워크에서의 시각동기를 위한 많은 프로토콜들이 연구되어 왔으며, 이 프로토콜들은 대체로 공통적인 특징을 갖고 있다. 센서노드들 간 클럭(타임스탬프) 정보의 교환을 통해 시각동기를 수행하고, 메시지 송수신 과정에서 비결정적인(nondeterministic) 요소들의 영향을 최소화한다는 것이 그것이다. 이들은 receiver-receiver (R-R) 방식, sender-receiver (S-R) 방식, 단방향 메시지 확산(one-way dissemination) 방식의 세 가지로 분류할 수 있다.

R-R 동기화 방식에서는 한 노드가 비컨 메시지를 이웃 노드들에게 주기적으로 브로드캐스트 전송한다. 그림 1(a)와 같이 이를 수신한 각 노드들은 메시지 수신 시점을 클럭 비교를 위한 참조 포인트로 삼으며, 각자의 로컬 클럭에 의한 수신 시각을 서로 교환한다. 그 후 각 노드들은 수신 시각의 차이를 이용하여 클럭 오프셋을 구할 수 있으며 시각을 동기화시킬 수 있다. Reference Broadcast Synchronization (RBS)[3]은 R-R 동기화 방식의 대표적인 프로토콜이다.

S-R 동기화 방식에서는 한 쌍의 노드 간에 핸드셰

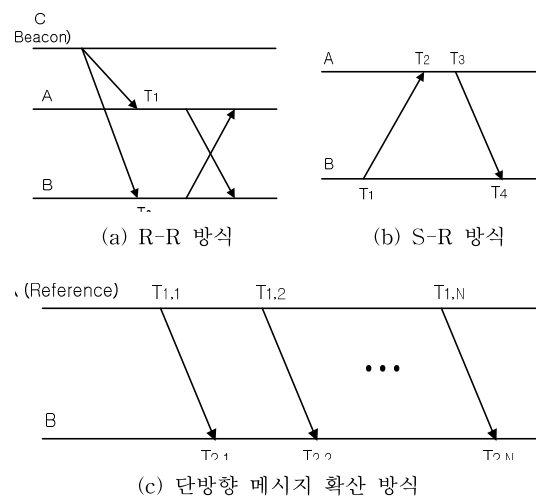


그림 1. 시각동기 기본 접근 방식

이크(handshake) 프로토콜을 통해 동기화가 이뤄진다. 이 방식을 따르는 대표적인 시각동기 프로토콜에는 Timing-Sync Protocol for Sensor Networks (TPSN)[4], Tiny-Sync/Mini-Sync (TS/MS)[5]가 있다. 그림 1(b)는 TPSN에서의 two-way 메시지 교환 과정을 나타낸다. 여기에서 T_1 은 노드 B의 송신 시각, T_2 는 이 메시지를 노드 A가 수신한 시각, T_3 는 노드 A가 Ack 메시지를 송신하는 시각, T_4 는 Ack 메시지를 노드 B가 수신한 시각 정보를 나타낸다. Two-way 메시지 교환이 끝나게 되면, 노드 B는 수식 (1)에 의해 타이머 오프셋(Δ)과 전파지연 시간(d)을 구할 수 있으며, 자신의 타이머값에서 Δ 만큼 더하여 보정을 하게 되면 노드 A의 타이머와 동기화할 수 있다.

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \quad (1)$$

$$d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$

단방향 메시지 확산 방식에서는 그림 1(c)와 같이 기준 노드가 이웃 노드들에게 시각 정보를 브로드캐스트 전송하고, 이를 수신한 각 노드는 브로드캐스트 메시지의 수신 시각들을 기록한다. 모든 시각값을 얻은 후 각 노드는 선형회귀(linear regression) 테이블을 통해 로컬 클럭과 기준 노드의 클럭 간 변환이 가능하게 된다. Flooding Time Synchronization Protocol (FTSP)[6]은 단방향 메시지 확산 방식의 대표적인 프로토콜이다. 본 논문에서의 스큐는 동기화를 위해 보정을 해야 하는 클럭에 기준을 두어 정의를 한 것이다[7-10].

3. 오프셋 보정을 통한 실시간 클럭 스큐 추정 방법

시각동기를 위해서는 오프셋 보정과 스큐 보정이 동시에 수행되어야 한다. 즉 노드 입장에서 모든 클럭 데이터를 얻은 이후의 시점이 아닌, 클럭 데이터를 얻을 때마다 이들 두 보정 작업이 함께 이루어져야 바람직하다. 예를 들어, 응용 계층에서 생성된 탐지 보고 메시지의 경우와 같이, 타 계층에서 현재의 시각 정보를 필요로 할 경우에 있어서 보정이 이뤄지지 않은 클럭으로는 해당 시점에서의 정확한 시각 정보를 제공할 수 없기 때문이다. 센서노드는 또한

그 특성상 제한된 자원으로 동작하기 때문에, 메모리 점유율과 연산에 따른 부담을 줄이기 위해 각 노드에 저장되어야 할 정보는 최소화되어야 한다.

이러한 사항을 고려하여 본 논문에서는 재귀적 최소제곱법을 이용한 실시간 클럭 스큐 추정 방법을 제공한다. 오프셋 보정과 관련해서는 기존의 방식들 (TPSN, RBS, FTSP 등)이 수정 없이 그대로 적용될 수 있으며, 클럭 오프셋 정보를 얻을 때마다 클럭 스큐는 재귀적으로 추정 및 갱신되어 저장해야 할 데이터 정보를 최소화할 수 있다. 먼저 오프셋 보정이 주기적으로 이루어지는 상황에서 실시간 클럭 스큐 추정에 대해 알아보고, 이를 오프셋 보정이 비주기적으로 이루어지는 일반적인 상황으로 확장하여 살펴보도록 한다.

노드 B가 노드 A와 동기화하려는 상황을 가정하자. 노드 A와 B가 하나 혹은 그 이상의 동기 메시지를 서로 교환하게 되면, 노드 B는 오프셋 보정을 통해 노드 A와 잠시 동기화할 수 있다. 그러나 실제 무선 네트워크 환경에서는 다양한 지연 요소들이 메시지 전송 과정에 영향을 미치게 되어 시각동기의 문제를 더욱 어렵게 만들고, 따라서 노드 간 클럭 오프셋과 스큐를 추정하기 위해서는 연속적인 동기 메시지의 교환이 필요하다. 메시지 전송 과정에서의 비결정적인 요소들의 영향으로 인해, 노드 B는 i -번째 메시지 교환 단계에 의한 노드 A와의 상대 클럭 오프셋을 식(2)과 같이 추정할 수 있다.

$$\hat{\theta}_{B,i}^A = \theta_{B,i}^A + X_i \quad (2)$$

식 (2)에서 $\theta_{B,i}^A$ 는 i -번째 오프셋 추정 시점에서 실제 두 로컬 클럭의 정확한 오프셋값을 의미한다. 그리고 X_i 는 송수신 과정에서의 가변적인 지연 요소들의 영향으로 인한 오프셋 추정 상의 노이즈를 의미하며, 이는 평균이 0, 분산이 σ_x^2 인 정규분포 확률변수로 가정한다. 전체 지연이 충분히 많은 수의 독립적인 확률변수들의 합으로 표시된다고 하면, 중심극한 정리(central limit theorem)에 의해 이를 정규분포 확률변수로 모델링할 수 있다. 이는 가변적인 지연 요소들이 신뢰도 99.8%의 정규분포 확률변수로 모델링될 수 있음을 실험실 테스트를 통해 보인 참고문헌 [3]에서의 결과에 의해서도 뒷받침된다. 아울러 메시지 송수신 과정에서의 지연 요소들에 대해서는 참고문헌 [4]에서 자세히 설명되어 있다.

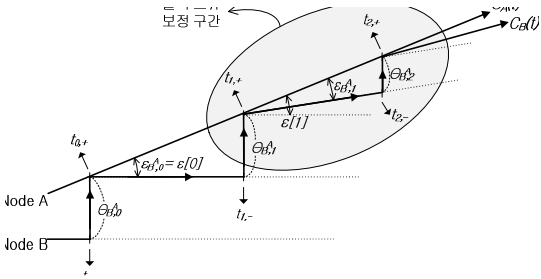


그림 2. 오프셋 보정 및 실시간 클럭 스큐 추정

그림 2는 동기 메시지 교환을 통해 노드 B의 클럭이 노드 A의 클럭과 동기화하는 과정을 나타낸다. 이 절에서 오프셋 추정 및 보정은 동기 메시지의 교환을 통해 주기적으로 수행된다고 가정한다. i -번째 동기 메시지의 교환이 끝난 후 노드 B는 상대 클럭 오프셋 $\hat{\theta}_{B,i}^A$ 을 계산하게 되고, 곧바로 현재 자신의 클럭값에 이 오프셋을 더함으로써 노드 A의 클럭과 동기화할 수 있다. 노드 B가 상대 클럭 스큐를 추정하기 위해서는, 로컬 클럭에 의한 오프셋 보정 시각을 알아야만 한다. 그런데 오프셋 보정 순간에 노드 B의 클럭은 점프하므로, 두 개의 오프셋 보정 시각이 존재한다. 보정하기 바로 전 시각을 $t_{i,-}$ 라 하고, 보정하고 난 이후의 시각을 $t_{i,+}$ 라 한다면 두 시각은 식(3)과 같은 관계를 갖는다.

이후 $(i+1)$ -번째 오프셋 보정이 수행되면, 노드 B 입장에서는 i -번째 단계와 $(i+1)$ -번째 단계 사이의 상대 클럭 스큐를 계산할 수 있게 된다. 0-번째 오프셋 보정 시점에서는 아직 스큐 추정을 할 수 있는 충분한 정보가 없기 때문에 오프셋 보정만 수행된다. 1-번째 오프셋 보정이 수행되고 난 후에야, 상대 클럭 스큐 $\epsilon_{B,0}^A$ 의 계산이 식(3)을 통해 가능해진다.

$$t_{i,+} = t_{i,-} + \hat{\theta}_{B,i}^A, \quad \epsilon_{B,i}^A = \frac{\hat{\theta}_{B,i+1}^A}{t_{i+1,-} - t_{i,+}} \quad (3)$$

1-번째 오프셋 보정 이후 $\epsilon_{B,0}^A$ 을 계산하게 되면, 노드 B 클럭의 동작 주파수는 식(4)와 같이 추정될 수 있다.

$$f_A = f_B(1 + \epsilon_{B,0}^A) \quad (4)$$

여기서, 노드 A 입장에서는 자신의 클럭 주파수 f_A 를 알 수 없기에 f_B 도 알 수 없다는 점, 다만 스큐 보정을 위해 필요한 정보는 상대 클럭 스큐(주파수 오프셋)임에 유의하자. 현 단계에서 노드 B는 클럭

주파수의 조정을 통해 스큐 보정을 수행할 수 있게 된다.

1-번째 오프셋 보정 이후 상대 클럭 스큐값을 계산하였으나, 현 시점에서 이 값은 추정 신뢰도가 높지 않다. 식(3)의 분자인 $\hat{\theta}_{B,i+1}^A$ 은 실제의 상대 클럭 오프셋이 아닌 노이즈 X_{i+1} 가 포함되어 있는 값이기 때문에, 계산한 상대 클럭 스큐 $\epsilon_{B,i}^A$ 역시 오차를 포함하고 있는 값이다. 실제 클럭 스큐와 추정치와의 오차를 줄이기 위해 선형회귀 분석에 사용되는 최소제곱법을 사용하여 접근하도록 한다. 여기에서 상대 클럭 스큐를 나타내는 새로운 변수 $\epsilon[i]$ 를 정의하자. 변수 $\epsilon[i]$ 역시 i -번째 오프셋 보정 시점부터 $(i+1)$ -번째 오프셋 보정 시점까지의 상대 클럭 스큐를 나타낸다는 점에서 $\epsilon_{B,i}^A$ 와 비슷하다. 그러나 $\epsilon[i]$ 는 노드 B 클럭의 본래의 주파수로부터의 노드 A 클럭의 주파수 오프셋을 의미하며, $\epsilon_{B,i}^A$ 는 노드 B 클럭의 보정된 주파수로부터의 주파수 오프셋을 의미한다(그림 3 참조). 1-번째 오프셋 보정 이전까지는 주파수 보정이 수행되지 않으므로, 두 변수의 초기값은 같다. ($\epsilon[0] = \epsilon_{B,0}^A$)

2-번째 오프셋 보정이 수행된 후에, 노드 B는 새로운 상대 클럭 스큐 $\epsilon_{B,1}^A$ 를 얻을 수 있게 된다. 1-번째 오프셋 보정 단계 이후 노드 B는 $\epsilon_{B,0}^A$ 에 따라 주파수 보정을 수행하였기 때문에, 2-번째 오프셋 보정 이후 노드 B가 계산한 현 단계에서의 노드 A의 주파수와 상대 클럭 스큐는 식 (5), (6)과 같다.

$$f_A = f_B(1 + \epsilon_{B,0}^A)(1 + \epsilon_{B,1}^A) \quad (5)$$

$$\epsilon[1] = \epsilon_{B,0}^A + \epsilon_{B,1}^A + \epsilon_{B,0}^A \epsilon_{B,1}^A \quad (6)$$

2-번째 오프셋 보정 이후, 노드 B는 획득한 두 개의 상대 클럭 스큐값들($\epsilon[1], \epsilon[2]$)을 바탕으로 최소제곱법을 사용하여 보다 신뢰적인 클럭 스큐를 추정할 수 있다. 2-번째 오프셋 보정 이후 최소제곱법을 사용하여 최종적으로 추정된 클럭 스큐를 $\hat{E}[1]$ 이라 하면, 이는 식 (7)과 같이 표현될 수 있다.

$$\hat{E}[1] = \frac{\epsilon[0] + \epsilon[1]}{2} = \epsilon_{B,0}^A + \frac{1}{2} \epsilon_{B,1}^A (1 + \epsilon_{B,0}^A) \quad (7)$$

따라서 2-번째 오프셋 보정 이후 노드 B는 새로운 상대 클럭 스큐 추정치 $\hat{E}[1]$ 에 따라 주파수 보정을 수행하게 되며, 이후 계속적으로 오프셋 보정이 수행

될 때마다 노드 B는 스큐 추정치를 갱신시키고 그 값에 따라 주파수 보정을 수행할 수 있게 된다.

그런데 여기서 식 (6)은 재귀적 형태의 수식이 아니며, 단계 i 가 증가할 때마다 저장해야 할 정보들은 계속 늘어나게 된다. 따라서 재귀적 형태의 스큐 추정 식이 필요하며, 먼저 식 (6)을 일반적인 형태로 표현하면 식(8)과 같다.

$$\epsilon[i] = \hat{E}[i-1] + \epsilon_{B,i}^A + \hat{E}[i-1] \epsilon_{B,1}^A \quad (8)$$

여기서 $\hat{E}[i]$ 과 $\epsilon[i]$ 의 초기값은 식(9)와 같다.

$$\hat{E}[0] = \epsilon[0] = \epsilon_{B,0}^A \quad (9)$$

식(13)를 바탕으로, $(i+1)$ -번째 오프셋 보정 이후 추정되는 상대 클럭 스큐 $\hat{E}[i]$ 에 대한 재귀적 형태의 수식을 정리하면 식(10)과 같다.

$$\begin{aligned} \hat{E}[i] &= \frac{i}{i+1} \hat{E}[i-1] + \frac{1}{i+1} \epsilon[i] \\ &= \hat{E}[i-1] + \frac{1}{i+1} \epsilon_{B,i}^A (1 + \hat{E}[i-1]) \end{aligned} \quad (10)$$

수식 (10)을 통해, 상대 스큐 추정치를 갱신시키기 위해 필요한 정보는 세 가지임을 확인할 수 있다. 첫째는 이전 단계에서의 스큐 추정치 $\hat{E}[i-1]$, 둘째는 $(i+1)$ -단계에서의 오프셋 보정을 통해 얻은 상대 클럭 스큐 $\epsilon_{B,i}^A$, 셋째는 관측 스큐 정보의 수 i 이다. 여기서 $\epsilon[i]$ 에 대한 정보는 필요치 않다는 점에 유의하자. 기존의 스큐 추정 알고리즘의 경우, 클럭 스큐의 추정을 위해 노드가 다수 개의 과거 오프셋 보정 정보들을 저장해야만 한다. 아울러 충분한 개수의 샘플들을 얻고 난 이후에야 비로소 클럭 스큐가 추정될 수 있었다. 본 논문에서는 클럭 스큐를 추정하기 위해 재귀적 최소제곱법을 사용함으로써, 노드가 동기 메시지의 교환을 할 때마다 스큐 추정치를 갱신하여 실시간적으로 클럭 스큐의 보정을 수행하게 하였다. 또한 각 노드에 저장되어야 할 데이터의 양은 최소화될 수 있도록 하였다. 이러한 점은 자원이 한정적인 센서네트워크에 있어 매우 바람직한 특징이라 여겨질 수 있다.

위에서 기술된 재귀적 가중최소제곱법을 사용한 시각동기의 전체 절차를 정리하면, 그림 3과 같다. 오프셋 보정이 주기적으로 수행되는 경우와 비교하였을 때, 비주기적 상황에서의 재귀적 스큐 추정을 위해 부가적으로 저장해야 할 정보는 두 가지이다.

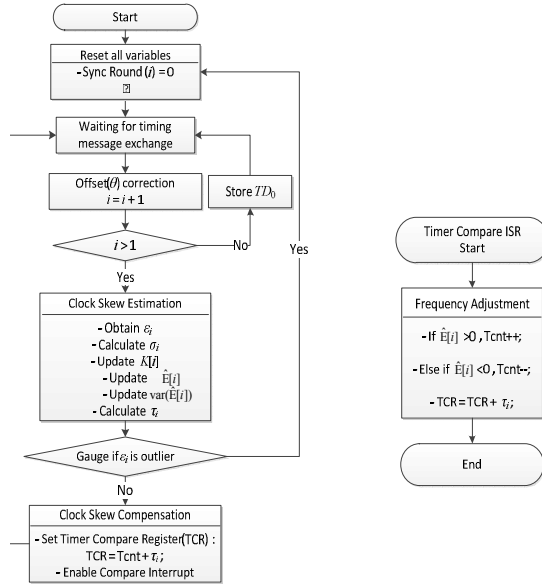


그림 3. 재귀적 가중최소제곱법을 사용한 시각동기 순서도

*첫째는 첫 번째 오프셋 보정 간격 $TD_0 (= t_{i+1,-} - t_{i,+})$ 이며, 둘째는 이전 단계에서의 추정치의 분산값 $var(\hat{E}[i-1])$ 이다.

4. 시뮬레이션 결과 분석

앞서 제시한 재귀적 클럭 스큐 추정 방법을 통해 클럭 스큐의 추정이 정확하게 이뤄지는지 알아보기 위해 먼저 몬테카를로(Monte Carlo) 컴퓨터 시뮬레이션을 수행하여 확인하였다.

여기서 몬테카를로 컴퓨터 시뮬레이션을 수행한 이유는, 시각동기를 수행하는 각 단계마다 클럭 스큐의 추정 값이 변화하기 때문에 최종적인 클럭 스큐 추정 정확도를 이론적으로 계산하기 어렵기 때문이다. 따라서 초기 클럭 스큐 추정 결과값을 재귀적으로 다음 단계의 클럭 스큐 추정 과정 상의 입력값으로 사용하여 시각동기를 수행하는 각 단계별 클럭 스큐 추정 정확도를 구하고자 몬테카를로 컴퓨터 시뮬레이션을 수행하였다.

각 노드의 타이머 주파수는 32.768kHz, 오프셋 추정 상 노이즈 X_i 의 분산 σ_X^2 은 1이라 가정하였으며, 30초 주기로 오프셋 보정을 수행하도록 하였다. 그림 4는 관측 데이터 샘플 수, 즉 동기 메시지 교환을 통한 오프셋 보정 횟수에 따른 추정치의 평균 제공근 오차

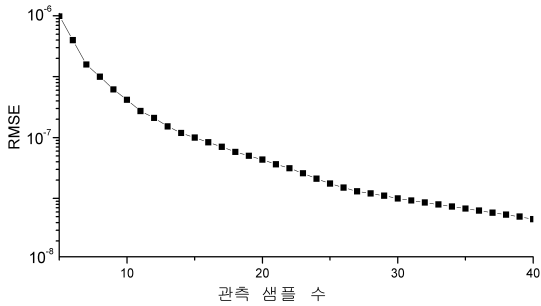


그림 4. 클럭 스큐 추정 성능

(root mean square error)를 보여준다. 기대했던 것과 마찬가지로 관측 샘플 수가 증가할수록 오차는 0에 가까워짐을 확인할 수 있다. 주기적으로 오프셋 보정을 수행하였기 때문에 최소제곱법을 통한 결과와 가중최소제곱법을 통한 결과는 동일하다.

다음으로는 제안한 클럭 스큐 추정 방법을 적용한 시각동기의 실제 성능을 확인하기 위해 동기 오차를 측정하기 위한 실내 실험을 수행하였다. 이를 위해 자체 제작 센서노드 플랫폼에 제안한 알고리즘을 구현하였다. 센서노드 플랫폼은 MSP430 MCU와 CC1100 RF 트랜시버의 하드웨어로 구성되며, 초소형 OS와 네트워크 프로토콜 스택이 탑재되어 있다. 또한 ±20ppm의 주파수 오차를 갖는 클럭 기반의 32.768kHz 타이머를 사용하였으며, MAC 계층에서 타임스탬핑이 수행되도록 하였다. 한 개의 노드를 기준 클럭을 가진 기준 노드로 동작할 수 있도록 하였으며, 다른 한 개의 노드는 싱크 노드로 동작하여 다른 노드들의 동기 오차를 수집하고 이를 시리얼 통신을 통해 PC에 전송할 수 있도록 하였다. 두 가지의 네트워크 토폴로지를 구성하여, 기준 노드와 다른 노드들 간의 시각동기 오차를 측정하였다.

먼저 트리 기반의 멀티홉 네트워크 토폴로지 상에서 오차를 측정하였으며, 이를 위해 20개의 노드로 구성된 5-홉의 IEEE 802.15.6 WBAN (Wireless Body Area Networks) 멀티홉 네트워크를 구성하였다. 동기 정확도를 측정하기 위해 모든 노드가 오프셋 보정을 수행한 후 곧바로 보정 오프셋 정보를 싱크 노드로 보고하도록 설정하였다. 이 오프셋값은 오프셋 보정을 수행하기 전 해당 노드의 클럭과 기준 노드의 클럭 간 시각 차이를 의미한다. 물론 측정 과정에서 오차로 인해 이 값이 실제 두 클럭의 동기 오차와 정확히 일치하지는 않지만, 동기 오차를 파악

하는데 있어서는 어느 정도 의미 있는 값이라 할 수 있다. 오프셋 보정을 위한 방법으로는, 멀티홉 환경에 적용하기 적합한 TPSN에서의 two-way 메시지 교환을 사용하였다. 부모 노드와 자식 노드 간 two-way 메시지 교환은 싱크 노드에 의해 시작되며, 이 과정은 레벨 별로 네트워크의 모든 에지(edge)에서 수행되게 된다.

클럭 스큐의 영향을 살펴보고자 모든 노드가 원래의 TPSN에서처럼 오프셋 보정만을 수행하도록 하였다. 모든 노드는 30분 또는 1시간 간격으로 오프셋 보정을 수행하도록 하고, 홉 수에 따라 각 노드로부터 보고되는 동기 오차의 평균편차를 그림 5에 나타내었다. 참고문헌 [4]에서의 결과와 마찬가지로 동기 오차는 홉 수가 커짐에 따라 조금씩 증가하는 경향을 나타내었다. 아울러 30분 경과 후의 데이터와 1시간 경과 후의 데이터를 비교해보면 시간이 경과함에 따라 클럭 스큐가 동기 오차에 미치는 영향은 점점 커짐을 확인할 수 있다.

앞선 방법과 비슷하게, 다만 노드가 오프셋 보정 뿐만 아니라 스큐 보정도 수행할 수 있게 하여 동기 오차를 측정해 보았다. 또한 재귀적 최소제곱법에 의한 추정과 재귀적 가중최소제곱법에 의한 추정을 서로 비교해보기 위해 이들을 각각 테스트하였다. 재귀적 최소제곱법을 사용하는 경우 싱크 노드가 5분의 간격으로 two-way 메시지 교환을 4번 수행하도록 하였으며, 재귀적 가중최소제곱법을 사용한 추정에서는 첫 번째 오프셋 보정 이후 20분 동안 랜덤한 시점에 4번의 two-way 메시지 교환을 수행하도록 하였다. 이 두 경우에 대해, 마지막 오프셋 보정 이후 30분 경과 후와 1시간 경과 후의 동기 오차를 측정하

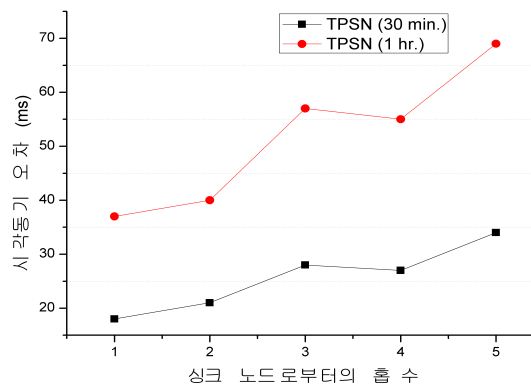


그림 5. 오프셋 보정만 수행했을 시의 동기 오차

여 그 평균편차를 그림 6에 나타내었다. 주기적(Periodic)으로 오프셋 보정이 수행될 때가 비주기적(Aperiodic)으로 오프셋 보정이 수행될 때 보다 다소 동기 성능이 좋게 나타났지만 그 차이는 크지 않았다. 또한 그림 5와 비교해보면, 클럭 스큐 추정 및 보정을 통한 동기화가 전체 동기 정확도를 상당히 향상시킴을 확인할 수 있다.

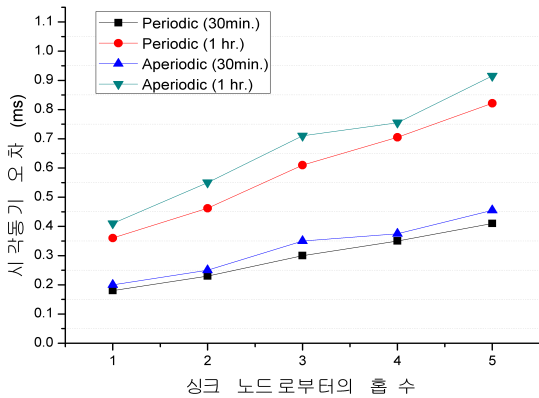


그림 6. 오프셋 보정과 스큐 보정을 동시에 수행했을 시의 동기 오차

5. 결 론

자원이 제한적인 센서네트워크 환경에서 효율적이고 정확한 시각동기를 유지할 수 있도록 하는 것은 매우 어려운 과제이다. WUSB 및 WBAN 등과 같은 센서네트워크에서의 이러한 특성을 고려하여 본 논문에서는 재귀적 최소제곱법을 사용한 클럭 스큐 추정 방법을 제시하였다. 제안한 방법을 통해 오프셋 보정과 스큐 보정이 동시에 수행되도록 하며, 아울러 이 과정에서 각 노드가 저장해야 할 데이터의 양을 최소화할 수 있도록 하였다. 오프셋 보정과 관련하여 기존의 방법들이 수정 없이 제안한 클럭 스큐 추정 방법에 적용될 수 있다. 실험 결과를 통해 제안한 방법이 클럭 스큐의 영향에 따른 클럭 오프셋 오차를 상당히 감소시키고 장기적인 시각동기의 안정성을 향상시킴을 확인하였다. 이는 적정 수준의 동기 수준을 유지하기 위한 재동기 주기를 길게 가져갈 수 있도록 하여, 재동기에 따른 에너지 소모를 줄일 수 있는 효과가 있다.

참 고 문 헌

- [1] D. Mills, "Internet Time Synchronization: the Network Time Protocol," *IEEE Transactions on Communications*, Vol. 39, No. 10, pp. 1482-1493, 1991.
- [2] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock Synchronization for Wireless Sensor Networks: a Survey," *Ad-Hoc Networks*, Vol. 3, No. 3, pp. 281-323, 2005.
- [3] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," *Proc. 5th Symposium on Operating System Design and Implementation*, pp. 147-163, 2002.
- [4] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-Sync Protocol for Sensor Networks," *Proc. First International Conference on Embedded Network Sensor Systems (SenSys)*, pp. 138-149, 2003.
- [5] M.L. Sichitui and C. Veerarittiphan, "Simple, Accurate Time Synchronization for Wireless Sensor Networks," *Proc. IEEE Wireless Communication and Networking*, Vol 2, pp. 1266-1273, 2003.
- [6] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol," *Proc. Second International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 39-49, 2004.
- [7] B.R. Hamilton, X. Ma, Q. Zhao, and J. Xu, "ACES: Adaptive Clock Estimation and Synchronization using Kalman Filtering," *Proc. International Conference on Mobile Computing and Networking (MobiCom)*, pp. 152-162, 2008.
- [8] K. Noh, Q. M. Chaudhari, E. Serpedin, and B. W. Suter, "Novel Clock Phase Offset and Skew Estimation using Two-Way Timing Message Exchanges for Wireless Sensor Networks," *IEEE Transactions on Communications*, Vol. 55, No. 4, pp. 766-777, 2007.

- [9] 엄정남, 이금분, 박정진, 조범준, “GPS와 가속도계를 이용한 이동 물체의 위치 추정 시스템,” 멀티미디어학회논문지, 제12권, 제4호, pp. 600-607, 2009.
- [10] K.I. Kim, “Adjusting Transmission Power for Real-Time Communications in Wireless Sensor Networks,” *Journal of Information and Communication Convergence Engineering*, Vol. 10, No. 1, pp. 21-26, 2012.



허 경

1998년 고려대 전자공학과 학사
2000년 고려대 전자공학과 석사
2004년 8월 고려대 전자공학과
통신공학 박사
2004년 8월~2005년 8월 삼성중
합기술원(SAIT) 전문연
구원

2005년 9월~현재 경인교대 컴퓨터교육과 부교수
관심분야: QoS, Wireless MAC, 컴퓨터교육