

클럭 표류율과 기준 신호를 이용한 두 센서 노드간 시간 동기 알고리즘

김 현 수[†] · 전 중 남^{††}

요 약

무선 센서 네트워크에서의 시간 동기 알고리즘은 위치 추적, 데이터 암호화, 중복 이벤트 감지 인식, 정밀한 TDMA 스케줄링 등의 다양한 응용을 위해서 필수적이다. 본 논문에서는 두 노드 사이에서 시간 보정을 위한 클럭 표류율과 기준 신호를 이용한 시간 동기 알고리즘인 CDRS를 제안한다. CDRS는 시간 동기를 위해 두 단계로 구성된다. 첫 번째 단계에서는 LTS를 이용하여 시간 보정 값인 노드간의 시간 차이와 클럭 표류율을 구한다. 이 단계가 끝나면 두 노드는 시간이 맞추어진 상태가 되고 클럭 표류율로 시간 차이를 보정할 수 있게 된다. 두 번째 단계에서는 동기 노드는 주기적으로 기준 신호를 전송한다. 비동기 노드는 수신된 신호를 사용하여 두 노드간 시간 차이를 측정하고, 시간 차이가 최대 허용 오차 범위를 초과하면 다시 첫 번째 단계를 수행한다.

시뮬레이션을 통한 성능 분석 결과, CDRS는 LTS 대비 시간 정확도가 향상된다. 또한 메시지 발생량이 LTS 대비 50% 감소하고, 기준 신호는 타임스탬프를 사용하지 않기 때문에 CDRS는 LTS에 비하여 시간 동기화에 사용되는 에너지가 2.5배 정도 적게 사용된다.

키워드 : 클럭 표류율, 기준 신호, 시간 동기, 저전력, 무선 센서 노드

Time Synchronization Algorithm using the Clock Drift Rate and Reference Signals Between Two Sensor Nodes

Hyounsu Kim[†] · Joongnam Jeon^{††}

ABSTRACT

Time synchronization algorithm in wireless sensor networks is essential to various applications such as object tracking, data encryption, duplicate detection, and precise TDMA scheduling. This paper describes CDRS that is a time synchronization algorithm using the Clock Drift rate and Reference Signals between two sensor nodes. CDRS is composed of two steps. At first step, the time correction is calculated using offset and the clock drift rate between the two nodes based on the LTS method. Two nodes become a synchronized state and the time variance can be compensated by the clock drift rate. At second step, the synchronization node transmits reference signals periodically. This reference signals are used to calculate the time difference between nodes. When this value exceeds the maximum error tolerance, the first step is performed again for resynchronization.

The simulation results on the performance analysis show that the time accuracy of the proposed algorithm is improved, and the energy consumption is reduced 2.5 times compared to the time synchronization algorithm with only LTS, because CDRS reduces the number of message about 50% compared to LTS and reference signals do not use the data space for timestamp.

Keywords : Clock drift rate, Reference signals, Time synchronization, Power consumption, Wireless sensor nodes

1. 서 론

최근 무선 센서 네트워크에 대한 연구가 활발하게 진행되고 있다[1]. 무선 센서 네트워크는 컴퓨터, 통신, 네트워크, 지능 계산, 센서, 임베디드 시스템, 마이크로 전자 등 다양한

분야가 교차, 종합되어 다양한 센서 연결 포인트로 구성된 자체 통제 가능한 네트워크로 물리 세계의 동적 상태를 지능적으로 협동 감지 할 수 있는 시스템이다. 연구 분야로는 저가격의 저전력 무선 기술, 저전력을 고려한 데이터 기반 라우팅 프로토콜, 센싱 데이터의 가공, 보안, 시간 동기 기법 연구 등이 있다.

무선 센서 네트워크에서 시간 동기화 기술은 매우 중요하다. 시간 동기가 이루어진 무선 센서 노드의 타임 스탬프를 이용하면 시간 흐름에 따른 노드의 위치 추적과 암호화 기술, 중복 이벤트 메시지의 발생을 감지할 수 있다. 뿐만 아

※ 이 논문은 2007년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음 (This work was supported by the research grant of the Chungbuk National University in 2007)

† 준 회원 : 충북대학교 전자계산학과 석사과정

†† 종신회원 : 충북대학교 전기전자 컴퓨터공학부 교수(교신저자)

논문접수 : 2008년 10월 7일

수정일 : 1차 2008년 11월 18일

심사완료 : 2008년 11월 20일

나라 Time division multiple access(TDMA) 스케줄링을 통하여 노드를 사용하지 않는 시간 동안에는 특정 모듈에 공급되는 전원을 차단함으로써 노드의 소비 전력을 줄일 수 있다. 특히 무선 센서 노드와 같이 저가의 수정 발진자를 사용하는 경우에는 클럭 표류가 크게 발생하기 때문에 주기적인 시간 동기화 과정이 필요하다.

시간 동기 알고리즘 중에서 Lightweight time synchronization(LTS)[2]는 주기적으로 매번 타임스탬프 메시지를 주고 받는다. 이것은 시간 동기의 정확도는 항상 시킬 수 있지만, 전체 네트워크의 수명을 단축시키는 결과를 초래한다. Reference broadcast synchronization(RBS)[3]는 참조 노드를 중심으로 서로간의 시간을 동기화 시킬 수 있지만, 역시 주기적으로 서로의 시간 값을 주고 받아야 한다. 이 외에도 주고 받은 누적 시간을 이용하여 더 정확한 시간 동기를 구현하는 방법도 있으나, 여전히 주기적으로 데이터를 주고 받아야 한다는 문제가 있다. 즉, 시간 동기 과정에서 발생하는 시간 동기화 메시지 발생량을 줄임으로써 센서 노드들이 소모하는 전력을 줄이기 위한 노력이 필요하다. 이를 위해서 시간 동기 과정에서의 메시지 발생량과 메시지가 가지는 데이터의 크기를 줄이는 방법을 연구해야 한다.

본 논문에서는 이동 환경의 무선 센서 네트워크에서 정확성을 고려하고 저전력으로 운영하는 시간 동기 알고리즘을 제안한다. 일반적으로 센서 노드는 각각의 노드마다 가지고 있는 고유의 클럭 표류로 인해 지속적인 동기화 과정이 필요하다. 따라서 각 센서 노드가 가지는 고유의 물리 시간을 보정하기 위해서 로컬 시간을 사용한다. 그리고, 클럭 표류로 인해 발생하는 시간 차이를 역 이용하여, 노드 스스로가 주기적으로 클럭 표류율을 이용하여 동기화를 지속적으로 유지할 수 있도록 한다. 그래서 이 과정이 정확한지 확인하기 위해 최소 크기의 데이터를 가지는 신호를 사용하여 동기 정확도를 지속적으로 확인할 수 있는 시간 동기 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 시간 동기 연구의 동기와 관련 연구에 대해 논한다. 3장에서는 제안하는 클럭 표류율과 기준 신호를 이용한 두 센서 노드 간의 시간 동기 알고리즘을 적용하는 기법에 대해 기술한다. 4장에서 제안한 기법을 시뮬레이션을 통해 성능을 분석하고, 5장에서는 결론과 더불어 향후 연구 방향을 제시한다.

2. 관련 연구

제한된 에너지와 처리 능력을 가지는 센서 노드의 특성으로 인해 기존의 컴퓨터 네트워크에서 사용하는 시간 동기화 기법을 바로 적용하는 것은 비효율적이다. 따라서 기존의 동기 기법을 수정하거나, 새로운 동기 기법들이 제시되고 있다. 관련 연구를 통해 무선 센서 네트워크 환경에서 시간 동기화 과정에서 발생하는 여러 발생 원인과 클럭 표류에 대하여 알아본다. 그리고 주로 사용되고 있는 동기화 프로토콜과 최근 제시되는 기법들에 대해서 언급한다.

2.1 시간 동기 정확도를 떨어뜨리는 원인

시간 동기에서의 정확도는 시간 동기를 진행하는 과정에서 다양한 비결정적 요소들에 의해 떨어진다. 정확도를 떨어뜨리는 오류의 원인은 다음의 4가지 기본 구성요소로 분류할 수 있다.

- 전송시간(Send Time): 응용 프로그램에서 데이터를 전송하기 위해 전송 명령을 수행했을 때, 실제로 전송을 시도하는 과정에서 지연되는 시간으로 커널 프로세싱, 스케줄링, 시스템 콜 등과 같은 현재 시스템 부하에 따라 달라진다.
- 접근시간(Access Time): 생성된 메시지가 radio 전송을 하기 위해 전송 채널로 접근하기 위해 기다리는 시간으로 사용되는 MAC 기법의 종류에 따라 달라진다.
- 전달시간(Propagation Time): radio 신호가 송신 측에서 수신 측에 도착하는데 걸리는 시간으로 두 노드간의 거리에 따라 달라지며, 센서 네트워크에서 빛의 속도로 전송 되기 때문에 전송 시간과 접근 시간에 비해 무시해도 될 정도로 작다.
- 수신시간(Receive Time): 수신한 데이터가 어플리케이션에 도달하는데 걸리는 지연 시간이다.

2.2 클럭 표류

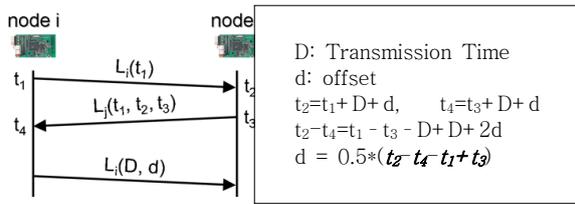
무선 센서 노드는 일반 컴퓨터의 클럭과 마찬가지로 클럭 표류를 가지고 있다. 클럭 표류 또는 클럭 이탈이라고 하는 이 증상은 센서 노드의 수정 발진자가 정해진 주파수로 동작하지 못하고 일정 편차를 갖는 것을 말한다. 센서 노드와 같이 저가의 수정 발진자를 가진 경우에는 큰 클럭 표류율을 가지고 있다. 버클리 모드의 경우 최대 40ppm(평균 20ppm)의 클럭 표류율을 갖는 것으로 명시한다[4]. 1ppm은 매 11.6일마다 1초의 오류가 누적되는 것을 의미하고, 20ppm의 경우 약 14시간 마다 1초씩 오류가 누적됨을 의미한다.

또한 수정 발진자의 주파수는 시간에 따라 변한다. 단기간의 변화는 환경적인 요인으로써 온도, 습도 그리고 대기압의 변화, 전압의 불안정 등에 기인할 수 있다. 장기간의 변화는 오실레이터 자체의 노화에 기인한다.

따라서 클럭 표류와 시간에 따른 주파수의 변화로 인해 주기적인 재동기화를 해주지 않는다면, 자연스럽게 동기화 상태는 풀어지게 된다. 실제 무선 센서 네트워크에서 Sensor-MAC[5]을 구현하기 위해서는 msec 단위의 정밀한 TDMA 스케줄링이 요구되는데, 클럭 표류로 벌어지는 시간의 틈을 메우기 위해서는 재동기화에 대한 부담이 더욱 가중된다. 따라서, 실제 무선 센서 네트워크에서 정확도를 제고하기 위해서 클럭 표류를 반드시 고려하여 설계해야 한다[6].

2.3 시간 동기화 기법 분류

대표적인 무선 센서 네트워크에서의 시간 동기화 기법으로는 Timing-sync Protocol for Sensor Networks(TPSN)[7], RBS가 있다.

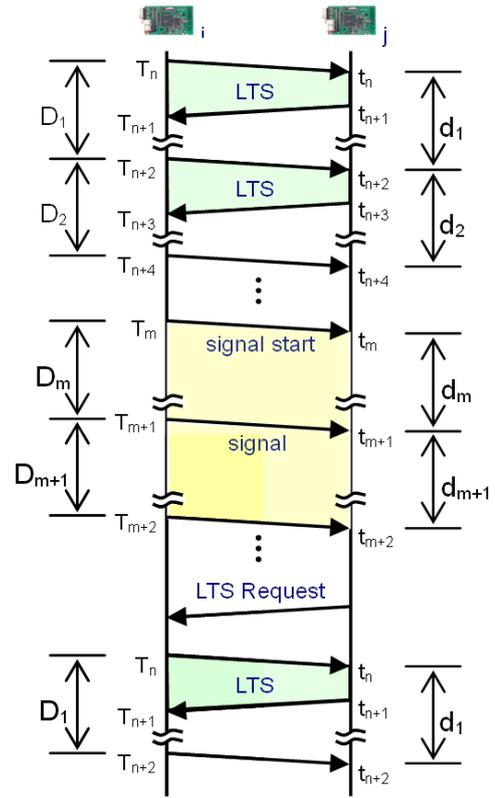


(그림 1) 두 노드간의 시간 동기

TPSN은 LTS기법을 이용하여 무선 센서 네트워크를 동기화하는 기법이다. LTS 기법은 (그림 1)과 같이 두 노드간에 서로의 송수신 시간 값을 주고 받음으로써 노드간의 오프셋을 측정할 수 있다. TPSN은 두 단계로 나누어서 동기화를 진행한다. 첫 번째 단계에서는 네트워크의 각 노드에 레벨을 할당하여 계층적 토폴로지를 형성한다. 레벨 0에 해당하는 하나의 루트 노드를 중심으로 트리 구조가 형성된다. 두 번째 단계는 하위 레벨의 노드가 상위 레벨의 노드에 시간 동기를 수행한다. 동기화 단계의 마지막 시점에 모든 노드는 루트 노드의 시간에 동기 되어 네트워크 전체의 동기가 이루어진다.

TPSN 기법은 기존 무선 센서 네트워크 시간 동기 기법들에 비해 정확도가 높은 장점을 가진다. 하지만, 한번의 LTS를 통한 동기화 기법은 2개의 메시지를 사용해서 동기화된 노드가 비동기 노드의 오프셋과 전파지연을 구할 수는 있지만, 이 값들을 비동기 노드는 알 수 없다는 문제점이 있다. 따라서 오프셋과 전파지연 값을 비동기 노드로 송신해 주는 과정이 필요하며 이로 인해 총 3개의 메시지가 동기화 과정에 필요하다. 클럭 표류가 큰 무선 센서 노드는 매번 동기화를 해 주어야 하므로 다수의 메시지가 필요하다. 그렇지 않으면 비동기 노드는 LTS를 진행한 이후에 클럭 표류에 의하여 임의의 특정 시간에서는 시간 동기의 정확도가 시간에 선형적으로 낮아지는 문제점을 안고 있다. 이 외에 TPSN의 응용 프로토콜로 누적 시간 정보를 이용한 시간 동기화 기법[8]이 있다. LTS를 이용하는 알고리즘은 정확성이 높다는 장점이 있다. 하지만 이동 환경의 무선 센서 네트워크에서 실시간의 변화에 능동적으로 대처를 할 수 없는 단점이 있다.

RBS는 참조 노드로부터 받은 동기 신호에 맞추어 수신 노드들간에 동기화를 하는 기법으로 송신 시간과 접근 시간 에러를 없애는 방법이다. RBS는 노드가 밀집하여 있을 경우 TPSN 보다 이웃 노드끼리 많은 메시지 교환을 하는 문제가 있다. 또한 참조 노드가 보낸 시그널을 수신하고 난 직후 자신의 송신 시간을 실어서 보내는 사이에서 발생 가능한 시간이 고려 되지 않았기 때문에 TPSN보다 정확도가 떨어진다. 이 외에 RBS의 응용 프로토콜로 Reference Interpolation Protocol (RIP)[9]가 있다. RIP는 RBS가 가진 참조 노드의 비동기화 문제, 전역 시간으로 동기화 되지 않는 문제, 그리고 노드의 밀집도가 높을 때 많은 메시지가 생성되는 문제에 대한 해결책을 제시한다. 이 방법은 RBS 대비 90% 가까운 메시지 발생량의 감소 효과가 있지만, 노



(그림 2) CDRS 동기화 과정

드가 밀집된 특정 상황에서 진행된 테스트의 시뮬레이션 결과이며, 정확도에 대한 추가 연구가 필요하다.

그 밖에 Hybrid Energy-Aware Synchronization Algorithm[10]은 앞에서 언급한 TPSN과 RBS 두 프로토콜이 가지는 특성에 따라 혼용하는 기법이다. 이 알고리즘은 클러스터의 밀집도를 분석하여 TPSN과 RBS를 혼용하여 시간 동기화를 선택적으로 수행한다. 이 기법은 TPSN보다 더 적은 메시지 소모와 더불어 에너지 소모의 표준 편차를 줄일 수 있는 장점을 가지고 있다. 하지만, TPSN 기법을 그대로 사용하고 있기 때문에 앞서 언급한 메시지 발생량을 줄이지는 못하고 있다.

3. CDRS 시간 동기 알고리즘

Clock drift rate and reference signals(CDRS) 알고리즘은 기존의 TPSN에서 언급한 LTS의 두 노드간의 시간 동기 기법과 유선 네트워크에서 사용하는 RBS 기법을 응용한 싱글 홉 기반의 시간 동기화 기법을 이용한 다. 동기화 과정은 (그림 2)와 같이 두 가지 단계로 나누어진다. 1단계에서는 LTS를 반복적으로 이용하여 누적된 시간 값을 이용하여 시간 보정을 위한 클럭 표류율과 오프셋을 구한다. 그리고 2단계에서는 1단계를 통해 구한 클럭 표류율과 오프셋 값을 이용하여 스스로 동기화를 진행한다. 이때, 비동기 노드의 동기화가 적정 허용 범위 내에서 이루어지는지 확인하

기 위해서 동기 노드에서 신호를 주기적으로 보내어 확인한다.

3.1 [1단계] 비동기 상태

- ① 동기 노드 i 는 시간 T_n 에 $LTS_Req_i(T_{n-1}, T_n)$ 를 송신한다. 이때 T_{n-1} 은 이전 LTS_Req_i 메시지의 Ack 신호를 수신한 시간으로 최초 송신에서 T_{n-1} 은 0 값을 가진다.
- ② $LTS_Req_i(T_n, T_0)$ 를 수신한 비동기 노드 j 는 수신 시간 t_n 을 측정 한다.
- ③ 비동기 노드 j 는 LTS_Req_i 에 대한 응답 메시지를 만들어서 송신하기 직전 시간인 t_{n+1} 을 측정하여 $LTS_Ack_j(t_n, t_{n+1})$ 를 송신한다.
- ④ 비동기 노드 j 는 t_n 과 t_{n+1} 를 자신의 시간 테이블에 저장하고, T_n 과 T_{n+1} 을 동기화 노드 시간 테이블에 저장한다.
- ⑤ 최초 전송되는 LTS_Req_i 메시지는 쿨 스타트 효과로 인해 지연 값이 크므로 첫 번째 LTS를 통해 교환되는 시간 값은 저장하지 않고, 두 번째 교환부터 사용되는 시간을 시간 테이블에 저장한다.
- ⑥ 노드 i 에서 동기화 메시지 LTS_Req_i 를 보내고 나서 일정 시간 D 가 경과하면, 다시 송신 시간을 측정하여 동기화 메시지 $LTS_Req_i(T_{n+1}, T_{n+2})$ 를 송신한다.
- ⑦ ②번에서 ⑥번까지의 과정을 n 번 반복하면서 비동기 노드 j 는 자신의 시간 테이블과 이웃 노드 시간 테이블에 시간 값을 채운다.
- ⑧ n 번의 LTS 과정을 통해 테이블에 값을 다 채우면, 동기 노드 i 는 2단계로 넘어가고, 비동기 노드 j 는 자신의 시간 테이블과 동기화 노드 시간 테이블의 시간을 이용하여 시간 보정 값을 구하고 2단계로 넘어간다.

3.2 [2단계] 동기 유지 상태

- ① 동기 노드 i 는 T_m 에 시간 동기 기준 신호의 시작을 알리는 $start_signal$ 을 보낸다.
- ② 비동기 노드 j 는 $start_signal$ 을 t_m 에 수신하면 2단계가 정상적으로 시작되었음을 확인한다.
- ③ 노드 i 에서 $signal$ 을 보내고 난 후 D 의 시간이 경과하면 T_{m+1} 에 $signal$ 을 다시 보낸다. 이때 동기화 $signal$ 안에는 어떤 데이터 값도 들어가지 않는다.
- ④ 노드 j 는 $signal$ 의 수신 시간 t_{m+1} 을 측정 하고 현재 수신한 시간이 동기화를 위한 최대 허용 오차 범위를 만족 하는지 확인한다.
- ④ 1 만약 허용 오차 범위를 벗어나는 경우 비동기 노드 j 는 $LTS_Request$ 메시지를 만들어 노드 i 에게 송신하고 자신을 1단계로 만들고 자신의 상태를 초기화 한다. 동기 노드 i 가 $LTS_Request$ 메시지를 수신하면, 동기 노드 j 는 자신을 1단계로 만들고 시간 D 가 경과하면, 1단계의 ①부터 다시 시작한다.
- ④ 2 최대 허용 오차 범위를 만족하면, 시간 동기 참조

테이블에 수신한 시간을 기록한다. 만약 시간 동기 참조 테이블에 n 개 만큼 수신 시간이 기록 되었다면, 기존의 클럭 표류율을 주기적으로 갱신한다. 이를 통해 더욱 최신의 값으로 정확도를 구한다.

- ⑤ 노드 i 와 j 가 동기화 유지 상태라면, 지속적으로 ③~④의 과정을 수행하면서 동기화를 유지한다.

4. 실험 및 성능 평가

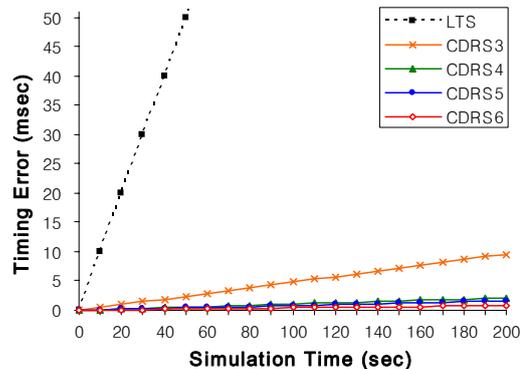
제안한 기법의 성능 평가를 위해 NS2.31에서 2개의 노드를 <표 1>과 같이 Zigbee 환경에 맞추었다[11]. 전력 소비량 기준은 버클리 모트 전력 소모량 측정표를 이용하였다. 제시된 에너지 소비 기준 모드는 Mica2이고 무선 통신 유형은 CC1000을 기준으로 두었다. 실험에서 사용한 재동기화 주기는 10초이고, 노드 1이 가지는 동기(BS) 노드와의 클럭 표류는 1초당 +1ms씩 발생한다고 가정하였으며, CDRS 2단계에서 동기화를 유지하기 위한 최대 에러 시간은 ±30ms으로 두고 실험하였다.

<표 1> NS2의 실험 환경 변수

Mote	MICA2 ATmega128
Channel Type	Channel/Wireless Channel
Network Interfaces	Phy/WirelessPhy/802_15_4
Mac Protocol	Mac/802_15_4
Energy Consumption rate (CC1000)	rxPower 0.3
	txPower 0.4
Max error time value for Synchronization	±30 ms
Re-synchronization period	10 sec
Node1_clock_drift	1ms/sec

4.1 클럭 표류에 따른 정확도 비교

(그림 3)은 LTS 이후 시간에 따른 클럭 표류에 의한 시간 오차 발생을 나타낸 것이다. LTS 그래프는 한 번 LTS 수행 이후 더 이상 수행하지 않았을 경우 시간에 비례하게



(그림 3) 클럭 표류로 인한 시간 오차

클럭 표류가 증가함을 알 수 있다. CDRS의 경우 LTS 수행 횟수가 많을 때 더욱 시간 오차 값이 적게 발생하는 것을 알 수 있다. 최대 허용 오차 범위를 10ms 라고 할 때, LTS는 10sec 마다 동기화가 필요하다. 하지만, CDRS를 통해서 클럭 표류율을 구하여 스스로 동기화가 가능하다면 재동기화 횟수를 줄일 수 있다. CDRS에서 동기화 1단계 과정에서의 LTS 횟수가 증가함에 따라 더욱 정확한 시간 동기화가 가능하다.

4.2 시간에 따른 동기화 시간 오차 비교

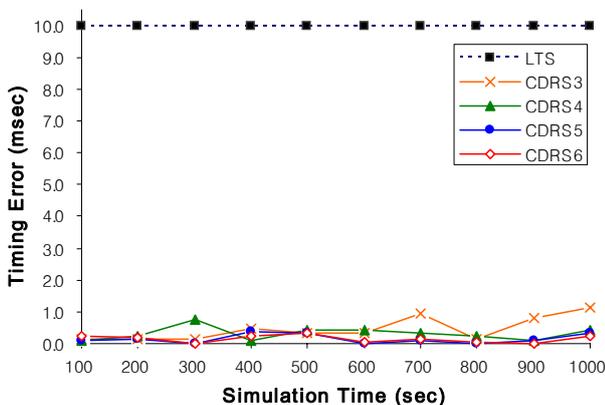
(그림 4)는 매 100초마다 시간 동기화를 수행하기 직전 BS 노드와 Node1 사이의 시간 오차를 측정 한 것이다. LTS의 경우 클럭 표류율로 인해 매 10초마다 10ms씩 지속적인 시간 차이가 발생했다. CDRS는 1단계 과정에서 주고 받는 LTS 횟수를 3번에서 6번으로 나누어 실험한 결과 횟수가 많아질수록 시간 오차는 줄어들고 실제 시간에 더욱 근접하는 것을 확인할 수 있다. 동기화가 유지되는 2단계에서는 비동기 노드는 주기적으로 수신한 동기화 신호를 기록한 누적 시간 테이블을 이용함으로써 지속적으로 시간 오차가 벌어지지 않고 일정하게 유지 됨을 확인할 수 있다. 즉 비동기 노드는 자신의 로컬 시간에 클럭 표류율과 오프셋 시간을 적용하여 시간 정확도를 높일 수 있다.

4.3 시간당 메시지 발생 수 비교

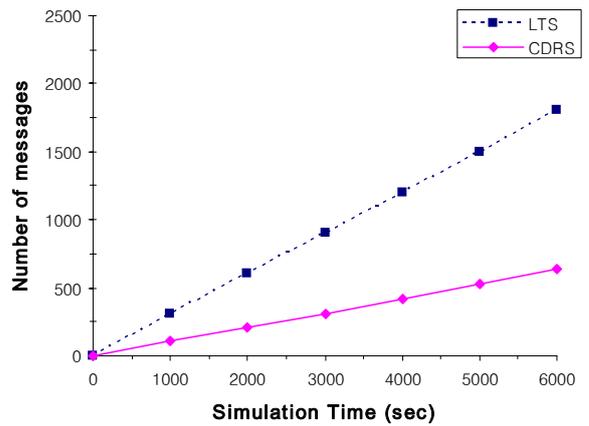
(그림 5)는 LTS와 CDRS에서의 시간당 메시지 발생량 평균을 나타낸 것이다. CDRS 기법을 사용하면 LTS 대비 약 35%의 메시지 발생량을 가지는 것을 확인할 수 있다.

4.4 동기화 주기에 따른 메시지 발생량 비교

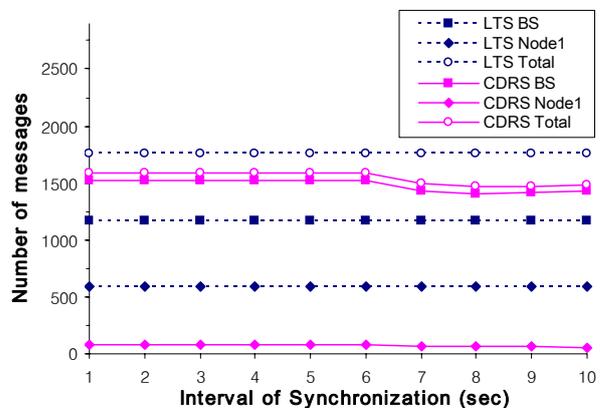
(그림 6)은 동일한 에너지 레벨을 갖는 BS와 Node1에서 각 동기화 주기 별 메시지 발생량을 나타낸 것이다. CDRS가 LTS보다 동일한 에너지를 가지고도 약 115% 더 많은 메시지를 발생시킬 수 있음을 확인할 수 있다. 특히, CDRS에서는 비동기 노드인 Node1에서 발생하는 메시지 수가 TPSN 대비 8배 이상 감소하는 것을 알 수 있다. 일반적으로



(그림 4) 시간에 따른 동기화 시간 오차 비교



(그림 5) 시간당 메시지 발생량 비교

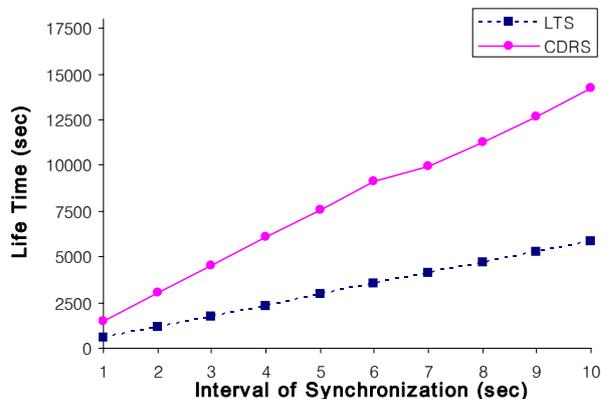


(그림 6) 동기화 주기에 따른 메시지 발생량 비교

로 센서 네트워크에서의 시간 동기화는 낮은 홉 수를 가지는 동기 노드가 인접한 높은 홉 수를 가지는 비동기 노드를 동기화 한다. 즉, 동기화 과정에서 발생하는 메시지 발생 부하를 동기 노드에게 집중 시킴으로써 네트워크 내에서 발생하는 충돌이나 간섭 현상을 최소화할 수 있다.

4.5 노드의 수명 비교

(그림 7)은 동일한 에너지 레벨에서의 동기화 주기에 따



(그림 7) 노드의 수명 비교

른 노드의 수명으로 동기화 주기가 길어질수록 수명이 오래 간다. 앞서 본 CDRS의 메시지 발생량의 감소로 인해 LTS 대비 시간 동기화를 위한 에너지 소비가 약 2.5배 감소한다.

5. 결론 및 향후 과제

무선 센서 네트워크에서의 시간은 프로토콜이나 여러 응용 기법에 매우 중요한 역할을 수행한다. 특히 시간 동기 기술은 동기 기반 통신 프로토콜 개발뿐만 아니라, 암호화 기술에서의 타임 스탬프, 타 노드들로부터의 같은 이벤트 중복 감지 인식, 기록된 이벤트의 발생 순서 구분 등 다양한 응용을 위해 필수적이다. 또한 메시지 충돌을 감소시키고 전력 소모를 줄이기 위하여 TDMA 스케줄링을 수행하기 위해서는 정확한 시간 동기화 기술이 요구된다.

본 논문에서는 센서 노드가 가지는 클럭 오류 증상에 의해 만들어지는 시간 차이를 노드 스스로 보정 가능한 신뢰성 있는 시간 동기 알고리즘을 제안하였다. 두 개의 노드를 이용하여 LTS 기법을 반복 사용함으로써 시간 보정 값을 구하고, 환경 변화나 지속적인 동기화 상태 확인을 위해 어떤 데이터도 신지 않은 기준 신호를 이용하여 주기적으로 동기화를 검증하였다. 이 방법을 통해 기존 LTS 대비 메시지 발생량을 반으로 감소시킬 수 있었고, 시간 동기화 과정에서 발생하는 에너지 소비는 2.5배 감소하는 것을 알 수 있었다. 1단계 동기화 이후에는 주기적으로 신호를 받음으로써, 추후 토폴로지의 변화에도 대응하면서 LTS 대비 항상 시간 정확도를 높게 유지할 수 있었다.

향후에는 이 알고리즘을 이용하여 멀티 홉에서도 동작 가능한 시간 동기 프로토콜을 연구할 예정이다. 또한 이를 바탕으로 실제 무선 센서 네트워크 환경에서 TDMA 스케줄링을 활용하는 라우팅 프로토콜을 구현할 예정이다.

참 고 문 헌

[1] Jamal N. Al-Karaki, Ahmed E. Kamal "Routing techniques in wireless sensor networks: a survey", IEEE Wireless Communications, pp.6-28, December, 2004.
 [2] J.V. Greunen, J. Rabaey, "Lightweight time synchronization for sensor networks", Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, pp.11-19, 2004.
 [3] J. Elsonm L. Girod, and D. Estrin. "Fine-Grained Network Time Synchronization using Reference Broadcasts", In the proceedings of the fifth symposium on Operating System Design and Implementation (OSDI 2002), December, 2002.
 [4] J. Polastre, R. Szewczyk, C. Sharp and D. Culler, "The Mote Revolution: Low Power Wireless Sensor Network Devices", In proceedings. Hot Chips 16: A Symposium on High Performance Chips, August, 2004.

[5] W. Ye, J. Heideman, and D. Estrin. "An energy-efficient mac protocol for wireless sensor networks", In proceeding of the 21st International Annual Joint Conference of IEEE Computer and Communications Societies (INFOCOM 2002), June, 2002.
 [6] Soyoung Hwang, Yunju baek, "Fault Toerant Clock Management schme in Sensor Networks", 한국통신학회 논문지 Vol.31 No.9A, pp.868-877, 9, 2006.
 [7] S. Ganerawal, R. Kumar, M. Srivastava, "Timing-sync Protocol for sensor networks", ACM SenSys 2003. 03.
 [8] Jinhong No, Youngsik Hong, "Time Synchronization using the Accumulative Time Information in Wireless Sensor Networks", 한국 정보과학회 추계 논문집 Vol.33, No. 2(D), pp.44-49, 10, 2006.
 [9] Chongmyung Park, Dongsun Lim, Inbum Jung, "Reference Interpolation Protocol for Reducing the Synchronization Messages in Wireless Sensor Network", 한국 정보과학회논문지 제 34권 제 6호, pp.446-457, 12, 2007.
 [10] Robert Akl, Yanos Saravanos, "Hybrid energy-aware synchronization algorithm in wireless sensor networks", Proceedings of IEEE PIMRC: 18th Annual IEEE International Symposium on Personal, IEEE International Conference on Signal Processing and Communication, November 2007.
 [11] <http://www.isi.edu/nsnam/ns>



김 현 수

e-mail : kim_hyounsoo@nate.com
 2007년 충북대학교 컴퓨터공학과(학사)
 2007년~현 재 충북대학교 전자계산학과
 석사과정
 관심분야: 임베디드 시스템, 센서 네트워크



전 중 남

e-mail : joongnam@cbu.ac.kr
 1981년 연세대학교 전자공학과(학사)
 1985년 연세대학교 전자공학과(공학석사)
 1990년 연세대학교 전자공학과(공학박사)
 1990년~현 재 충북대학교 전기전자 컴퓨터
 공학부 교수

관심분야: 임베디드 시스템, 컴퓨터 구조, 센서 네트워크