

# 무선 센서네트워크의 분산 다중노드를 위한 시간동기화 프로토콜 설계 및 구현

김기현<sup>o</sup> 김희철 홍원기  
 대구대학교 정보통신공학부  
 {khkim\_v<sup>o</sup>, hckim, wkhong}@daegu.ac.kr

## Design and Implementation of Time Synchronization Protocol for Distributed Manifold Nodes in Wireless Sensor Networks

Ki-Hyeon Kim<sup>o</sup> Hie-Cheol Kim Won-Kee Hong  
 Dept. Information and communication engineering, Daegu university.

### 요 약

저전력 연산능력과 무선통신 기능을 포함한 경량의 센서노드들로 구성된 무선 센서네트워크는 자연 환경의 모니터링에서 군사적인 목적에 이르기까지 다양한 어플리케이션을 창출 할 수 있다. 그러나, 다수의 저가 무선 노드를 배치해야 하는 센서네트워크의 특성상 안정성이 높은 고가의 부품을 사용할 수 없어, 이에 대한 보완 방안에 대한 연구가 활발히 진행 중이다. 특히, 무선 노드들 간의 시간동기화는 가용 무선 대역폭이 적은 무선 센서네트워크에서 TDMA 스케줄링등의 기술을 구현하기 위한 필수적인 기반기술로서 네트워크 특성을 고려한 새로운 동기화 기법을 요구하고 있다. 본 논문에서는, 무선센서노드의 특성을 파악하고 시간 동기화 수행 시 발생할 수 있는 오차요인을 분석하여, 넓은 지역에 산발적으로 분포되는 무선 센서네트워크의 특성을 보완할 수 있는 새로운 시간동기화 기법을 제안한다.

### 1. 서 론

초소형 시스템으로 불리는 MEMS(Macro Electro Mechanical System)기술의 발달과 더불어 무선통신 장치와 연산 처리장치의 SoC를 통한 저전력, 초경량 무선 센서노드가 출현함에 따라 그림 1과 같이 무선노드로 구성되는 새로운 형태의 네트워크가 구축되고 있으며, 이를 활용한 다양한 어플리케이션이 개발되고 있다. 그러나, 음파의 전송시간 측정, 저전력 TDMA스케줄링 그리고 암호화기법과 같은 응용에서는 무선 센서노드 간 시간 동기화가 필수적이며, 이를 위한 대책이 필요하다.

무선 센서네트워크 환경에 사용되는 무선 센서노드들은 Mote 계열과 같이 경량의 8비트 CPU가 대부분의 연산을 처리하며, 저가 부품을 채용하여 시스템의 안정성이 떨어지는 단점이 있다. 범용 컴퓨터 시스템 방식과 마찬가지로 무선 센서노드는 발전기에 의해 필요한 주파수를 공급받고 있으나 온도 등 외부환경에 민감한 발전기의 특성상 안정적인 클럭 공급이 어려워 근본적인 노드 간 시간차 유발요소를 내재하고 있다. 이러한 노드 간 시간차를 해결하기 위해 제안된 시간동기화 프로토콜은 [1, 2]에서 제시한 바와 같이 그 수행과정에서 다양한 경로를 통해 새로운 시간오차가 발생할 수 있다.

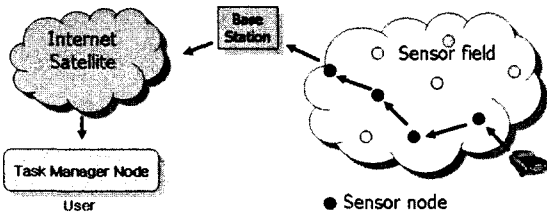


그림 1. 센서네트워크 개념도

임의의 노드에서 다른 노드로 패킷을 전송하는 시간은 기본적으로 송신시간(Send Time), 접근시간(Access Time), 전송시간(Transmission Time), 전파시간(Propagation Time), 청취시간(Reception Time), 수신시간(Receive Time)으로 나눌 수 있다. 송신시간은 메시지를 조립하고 MAC 계층에서 요청을 보내는데 걸리는 시간이며 이는 운영체제의 시스템 콜, 오버헤드, 프로세스 부하에 의존적이며 비결정적인 요소이다. 접근시간은 전송이 시작될 때 전송 채널에 접근하기 위해 걸리는 시간으로 현재 네트워크 트래픽 상황에 의존적이며 비결정적인 요소이

다. 전송시간은 메시지 전송을 위해 송신단에서 소요되는 시간으로 메시지 길이에 의존적이며 결정적이다. 실제 RF 메시지의 전송속도에 의존적인 전파시간은 송신기에서 수신기로 전송되는 시간으로 보통 300미터 이내에서는 1 마이크로초 이내로 무시할 수 있다. 청취시간은 송신단에서의 전송시간과 같으며, 수신시간은 수신된 메시지를 처리하는 시간으로 송신단에서의 송신시간과 비슷하다.

그 외, [3, 4]에서 제안한 인터럽트 차단에 따른 자연시간인 인터럽트 시간(Interrupt Time), 무선장치 송수신단에서 메시지를 전자기파로 변환하는데 소요되는 시간인 부호화·복호화 시간(Encoding, Decoding Time) 그리고 수신데이터의 정렬에 소요되는 시간인 바이트 정렬시간(Byte Alignment Time)에서 노드 간 새로운 시간오차가 발생할 수 있다.

지금까지 무선 센서네트워크 환경에서의 시간동기화 기법에 대한 연구는 기존의 인터넷에서 사용되는 NTP [5]와 달리 RF의 방송 특성을 기반으로 이루어져 왔다. 그 중 시간동기 패킷을 전달하는 참조노드의 전파가능 영역 내에 위치한 다수의 수신노드 간 시간동기화를 수행하는 RBS [6], 향상된 개념의 적응형 시간동기화 기법 [7]이 제안되었다. 또한, 페어와이즈(Pair-wise) 기법 [8]을 통해 송신노드와 수신노드 간 시간동기화를 수행하는 LTS [9], TPSN [3]이 있으며 MAC 레이어에서의 다중 타임스탬핑을 통해 시간 동기 수행 시 오차를 줄이고자 한 방법으로 FTSP [4]등을 들 수 있다. 그러나, 이들은 다중 홉 단위로 연결되어 광역지역에 분포되는 실제 센서네트워크 환경에 적용 시, 시간동기화를 위해 소비되는 노드의 연산부하 증가와 점유 대역폭의 트래픽 증가로 인해 시간동기화 수행에 수반되는 비용이 적지 않다. 이를 위해 본 논문에서는 다중 노드 간 시간동기화를 신속하고 간결하게 수행하는 새로운 시간동기화 기법을 제안한다. 제안하는 방식은 RF의 방송 특성을 이용하여 노드 간 시간동기를 위해 필요한 참조패킷의 개수를 효과적으로 줄여 다중 홉 상에 배치된 센서노드의 부하와 동기 수행에 소요되는 시간을 줄이는 기법이다.

본 논문의 구성은 다음과 같다. 2장에서는 센서네트워크 환경에서 시간동기화시 고려사항을 포함한 관련 연구를 설명하고, 3장에서는 제안하는 새로운 시간동기화 기법을 소개한다. 4장에서 제안하는 기법의 구현에 대해 설명하고, 제안된 기법의 실험 및 성능평가를 5장에서 다루며, 마지막으로 결론에 대해 언급한다.

## 2. 관련 연구

센서네트워크에서의 시간동기화 시스템은 동기 될 시간 정보가 포함된 패킷을 방송하는 참조노드와 수신된 패킷에 따라 각자의 로컬 시간을 조정하는 수신노드로 구성되며, 이를 활용한 다양한 방법이 제안되고 있다.

먼저, 무선 센서네트워크 환경에서 시간동기화를 수행하기 위해 고려해야 할 사항은 다음과 같다.

첫째, 전역 참조노드(Global Reference Node)의 필요성이다. 이는 전체 네트워크를 전역 시간(Global Time)에 동기화하고자 할 경우, 기준이 되는 시간을 보유한 노드가 있어야 함을 뜻한다. 둘째, 선택적인 시간 동기화의 수행이다. 센서네트워크의 모든 응용이 시간동기화를 필요로 하지는 않으며, 동기화 수행에 따르는 비용 역시 적지 않다. 셋째, 단일 홉 이상의 다중 홉 거리로 구성된 네트워크 특성과 다수의 노드가 분포된 상황에 대한 고려가 필요하다. 이것은 동기화 참조 패킷의 산발적 발생으로 인한 무선 통신채널 트래픽의 부하증가를 야기하여 센서네트워크 본연의 임무 수행에 지장을 초래할 수 있다.

전송된 내용을 기반으로 진행된 시간동기화 기법에 대한 연구는 크게 수신노드-수신노드 간 시간동기 기법과 송신노드-수신노드 간 시간동기 기법으로 크게 나눌 수 있다. 그중, 수신노드-수신노드 간 시간동기 기법으로 대표적인 RBS [6]는 방송되는 RF 메시지가 전파지역이 극히 적은 수신기들의 집합에 전달된다는 관찰에 기초하여, 참조노드가 방송한 동기패킷을 수신한 노드들이 각각의 로컬 클럭에 따라 수신시간을 저장하고, 저장된 시간을 이웃노드와 교환하여 인접 노드들 간의 시간 오프셋 값을 산출하는 방식이다.

또한, [7]은 RBS를 기반으로 하여 시간동기 오차율을 줄이고 RBS와 달리 참조노드를 두지 않는 기법을 제안함으로써 별도의 참조노드가 필요한 RBS의 다중 홉 문제를 효율적으로 해결하고자 하였다. 앞선 두 기법들은 동기화하고자 하는 전역 시간을 두어 그 시간에 모든 노드를 동기화하는 것이 아닌 인접 노드 간 시간 차이를 계산하여 이벤트 발생시간 차이를 산출하는 기법이다.

이에 반해, 송신노드-수신노드 간 시간동기 기법에 속하는 LTS [9]는 노드의 동작 실패와 이동성을 고려하여 다중 홉 상의 노드들을 효과적으로 해결하고자 전체 노드를 스패닝 트리(Spanning tree)로 구성하고 노드쌍 간 페어와이즈를 통해 시간동기화를 수행하였다. 또한, TPSN [3]은 네트워크 내 노드들의 계층구성 단계, 페어와이즈를 통한 시간동기 단계를 통해 두 노드 간 시간동기화를 수행하였으며, FTSP [4]는 RF메시지의 방송특성을 기반으로, 전송된 시간동기 오차요인을 효과적으로 해결하기 위해 다중 타임스탬프를 작성하여, 예측 불가능 요소에 대한 실질적인 소요 시간을 산출함으로써 해당오차요소를 상쇄하였다. 이 기법들은 모든 노드가 동기화하고자 하는 전역 시간을 송신노드의 시간에 맞추어 방송함으로써, 수신하는 모든 노드가 최초 송신노드의 시간에 모두 동기화 되는 기법이다.

앞서 언급한 다양한 시간동기 기법들은 본 절의 서두에 소개한 시간오차 발생요인들 중 비결정적인 요소를 효과적으로

해결하고, 결정적인 요소에 대해 동기화 정확도를 높이는 데만 집중하였다. 이는 넓은 지역에 다중 홑의 거리로 분포되는 센서네트워크의 실제 환경에 적용 시, 동기패킷의 중복 방송에 따른 송신노드 및 수신노드들의 부하 증가를 초래하며, 현재 ISM 밴드의 협대역을 사용하는 무선센서노드의 통신 채널 트래픽 증가를 야기하게 된다. 이에 본 논문에서는 기존의 연구들과 비슷한 정확도를 유지하며, 다수의 센서노드가 배치된 환경에 적합한 시간동기화 프로토콜을 제안한다.

3. 다중 노드 시간동기화 프로토콜

전송된 바와 같이 무선센서네트워크 환경에서 노드 간 시간동기화 시, 동기 정확도외에 고려되어야할 사항으로 동기 패킷과 네트워크 트래픽을 감소시킬 수 있는 경량성과 전체 노드의 동기화를 신속하게 처리할 수 있는 신속성을 들 수 있다. 이는 무선 채널을 통해 수집된 정보를 실시간으로 처리하여야 하는 센서네트워크 본래의 태스크를 원활히 수행하는데 필수적인 요소이다.

본 논문에서는 경량성과 신속성을 효과적으로 향상시킬 수 있는 새로운 시간동기화 기법인 MNTP(Multi Node Time synchronization Protocol)를 제안한다. 제안하는 기법은 시간동기 패킷을 방송하는 참조노드의 단일 홑 내 모든 노드들이 전역 시간에 각자의 시간을 동기화하는 송신노드-수신노드 간 시간동기화 방법으로 일반적인 센서 네트워크에서와 같이 모든 노드는 유일한 노드 ID와 각자의 로컬 클럭을 가진다고 가정한다.

제안하는 MNTP에서는 네트워크상에 분포된 모든 노드에 대해 참조노드와 다중 수신노드 간 시간동기화를 수행한다. 이때, 참조노드로 선정된 노드들은 자신이 동기화 된 후, 단 두개의 타임스탬핑 시간동기 메시지만을 방송하여, 단일 홑내의 다수 수신노드들에게 시간동기 기준점을 제공함으로써, 전체 노드들의 시간동기 수행에 따른 통신채널의 부하를 효과적으로 줄여 줄 수 있다. 또한, 수신노드는 인접노드와의 로컬시

간 차이값 교환이 필수적인 RBS, TPSN과 송수신노드 양단에서 다중 스탬핑을 수행해야 하는 FTSP와 달리 송수신 양단에서 단 두번의 타임스탬핑을 통해 수신노드별 독자적인 시간차이값 산출이 가능하므로 동기화 진행시간을 크게 단축시킬 수 있다.

그림 2에 MNTP에서의 타임스탬핑 및 동작절차를 나타내었다. 모든 노드에서의 타임스탬핑은 MAC 레이어 타임스탬핑을 통해 많은 오차요소를 제거하였다.

동작절차는 다음과 같다. 참조노드  $D_i$ 가 첫 번째 동기 패킷  $P1_{SYNCl}$ 에 전역시간  $T1_{Di}$ 을 타임스탬핑하여 방송하면, 이를 수신한 수신노드  $U_x$ 들은 각자의 로컬 타임에 따라 첫 번째 패킷의 수신시간  $T2_{Ux}$ 를 타임스탬핑하여 저장하고, 다시 전역시간  $T3_{Di}$ 에 타임스탬핑 된 두 번째 동기 패킷  $P2_{SYNCl}$ 의 수신시간  $T4_{Ux}$ 를 저장하여 참조노드  $D_i$ 와 수신노드  $U_x$ 간 두개의 시간차이 값  $CD1_{DiUx}$ 와  $CD2_{DiUx}$ 를 각자 계산한 후, 산출된 평균값으로 자신의 지역시간을 조정한다. 이때, 전역시간과의 평균 시간차이 값  $CDavg_{DiUx}$ 는 다음의 식들을 통해 계산할 수 있다.

먼저, 참조노드  $D_i$ 에서의 전역시간  $T1_{Di}$ 와  $T3_{Di}$ 는 다음의 식 (1), (2)

$$T1_{Di} = T2_{Ux} + CD1_{DiUx} + PD1_i \quad (1)$$

$$T3_{Di} = T4_{Ux} + CD2_{DiUx} + PD2_i \quad (2)$$

로 나타낼 수 있으며, 각 기준점에서 2개의 시간차이 값은 식 (1), (2)를 통해 다음의 식 (3), (4)

$$CD1_{DiUx} = T1_{Di} - T2_{Ux} - PD1_i \quad (3)$$

$$CD2_{DiUx} = T3_{Di} - T4_{Ux} - PD2_i \quad (4)$$

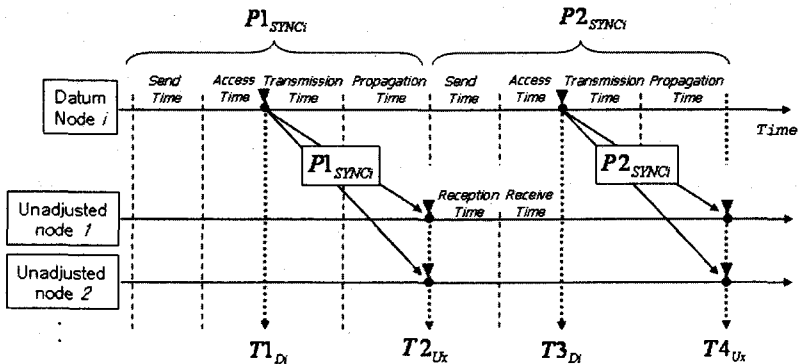


그림 2. MNTP 타임스탬핑 및 동작절차

로 유도할 수 있으며, 이들 시간차이 값의 평균은 식 (3), (4)를 통해 다음의 식 (5)

$$CD_{avg\ D_{ik}} = \frac{CD_{D_{ik}} + CD_{D_{ik}}}{2} \quad (5)$$

로 이끌어 낼 수 있다. 이때, 전파지연 값인  $PD_i$ ,  $PD_i$ 는 실험을 통한 측정치의 평균을 상수값으로 가진다. 단일 홉 및 다중 홉에서의 MNTP 성능은 5절에서 Mote 계열의 플랫폼상의 실제 구현을 통해 RBS, TPSN과 비교평가 하였다. 실험 결과에서 제안하는 MNTP는 대표적인 시간동기화 기법인 RBS, TPSN과의 시간동기화 오차를 평가에서 단일 홉과 다중 홉 모두 향상되거나 비슷한 성능을 나타내었다.

전술된 방식에 따라 최초 참조노드에서 말단의 수신노드까지의 동기화를 수행하는 다중 홉 환경에서 MNTP는 더욱 향상된 성능을 보인다. 제안된 MNTP와 같이 RF의 방송특성을 이용하는 RBS는 다수의 노드들에게 동기 메시지를 전달 할 수 있으나, 수신노드-수신노드간 시간동기화를 수행하므로 두 수신노드 간 시간차이 값 계산을 위해 추가적인 메시지 교환이 필요하며, 송신노드-수신노드간 동기화를 위해 페어와이즈 기법을 사용하는 TPSN의 경우, 노드의 배치밀도에 따라 홉 간 동기화 수행시간이 불규칙적이며 메시지 발생량 역시 높아 통신채널의 부하를 가중시킨다.

그러나, 정적인 환경에서 방송 메시지의 전송반경 가장 자리에 위치한 노드들을 차후 참조노드로 선정하는 MNTP는 동기화에 소요되는 시간과 메시지 발생량을 줄여 통신채널의 부하를 확실히 감소시킨다. 방송되는 메시지  $M_D$ 가 단지 두개의 MNTP에서 요구되는 총 메시지 발생량  $M_{MNTP}$ 는 네트워크에 분포된 참조노드 개수  $i$ 에 따라 다음의 식 (6), (7)

$$M_{MNTP} = M_{D1} + M_{D2} + M_{D3} + \dots + M_{Di} \quad (6)$$

$$M_{MNTP} = 2 * N_{Total\ Datum} \quad (7)$$

로 나타낼 수 있으며, 이는 네트워크내의 전체 참조노드 수에 비례적으로 증가함을 알 수 있다. MNTP의 동기화 수행시간은,

$$T_{MNTP} = (T_D + T_{PD} + T_U) * N_{Hop} \quad (10)$$

로 참조노드에서의 수행시간  $T_D$ 와 전파지연 시간  $T_{PD}$  그리고 전체 수신노드 수행시간  $T_U$ 의 합산시간에 곱해질 네트워크 전체 홉 수  $N_{Hop}$ 에 따라 결정되어진다. 이때, 참조노드의 수행시간  $T_D$ 는 MAC 충돌을 고려해 약 1.7ms의 지연시간이 추가되었으며, 홉 수를 곱하는 이유는 홉 마다 선정된 기준노드들이 병렬적으로 동기화를 수행하기 때문이다. 실제 동기화

수행시간은 각 노드 운영체제의 부하량과 통신채널 상태에 따른 전파지연 등 변동 요소가 많아 일정한 소요시간 산출이 어렵다. 이에 본 논문에서는 MNTP와 비교대상인 TPSN의 동기화 수행시간 산출을 위해 실험을 통한 표본 동기화 수행시간을 측정하였다. 표본시간측정 실험에 대한 내용은 지면의 제약상 생략하며, 측정된 두 프로토콜 간 수행시간은 MNTP가 33msec, TPSN이 25.5msec로 나타났다.

실험 결과에서 MNTP가 TPSN보다 더 많은 시간이 소비된 것으로 볼 수 있으나, 이것은 네트워크내 전체 노드들을 고려하지 않은 것으로 각 프로토콜간 전체 동기화 수행시간은 식 (10)을 참조하여 다음의 식 (11)로 나타낼 수 있다.

$$T_{MNTP} = (33 * N_{Hop}) msec \quad (11)$$

#### 4. Mote 플랫폼상의 MNTP 구현

본 절에서는 제안된 MNTP의 구현에 대해 대표적인 센서네트워크 플랫폼 Mote계열을 대상으로 설명한다. 사용된 센서노드는 Mib510과 같은 별도의 프로그래밍 보드 없이 노드와 직렬통신선을 구축 할 수 있는 Mica2 기반의 Cricket 노드를 선택하였다. 센서노드의 운영체제는 확장 모듈을 C언어로 손쉽게 작성가능하며, 동적 메모리 할당이 가능한 SOS[10]를 선택하였다. 현재, SOS는 Mica2를 기반으로 작성되고 있으나, 하드웨어 컴퍼넌트 부분의 소스코드를 수정하면 간단히 Cricket 보드상에 이식할 수 있다. 그림 4에 실제 구현한 MNTP와 SOS, 그리고 Cricket 플랫폼의 주요 컴퍼넌트 구조를 다이어그램으로 나타내었다.

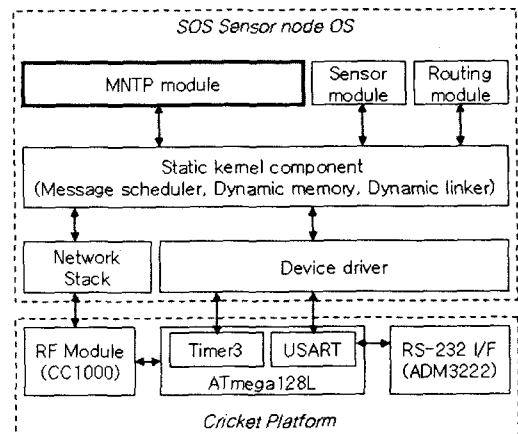


그림 3. Cricket 플랫폼과 SOS

MNTP의 구현에서 가장 중요한 시스템 타이머는 7.3728MHz의 크리스탈로 동작하는 타이머 3를 사용하였다. 시스템 시간은 SOS에서 제공하는 API인 `ker_systime32()`를

사용하여, ATmega128의 타이머 3 레지스터인 TCNT3의 값과, 타이머 오버플로우 횟수를 나타내는 변수 값으로 획득할 수 있다. 또한, 노드 디버깅을 위해 Cricket의 ADM3222 칩을 하드웨어 초기화 단계에서 작동 가능하도록 수정하였다.

제안된 프로토콜은 전송된 알고리즘을 기초로 모듈의 형태로 작성하여 노드상에 이식하였다.

5. 실험 및 성능 평가

본 논문에서는 크게 두가지 항목으로 나누어 제안하는 MNTP의 성능평가를 수행하였다. 먼저, 시간동기 프로토콜의 기본성능인 동기화 정확도를 단일 홉과 다중홉으로 나누어 RBS, TPSN과 비교평가 하였으며, 다중 노드 분포에 따른 동기 수행시간과 요구패킷량의 비교를 위해 20m x 20m의 환경에 가변량의 노드를 임의로 배치하여 시뮬레이션을 수행하였다.

5.1 단일·다중 홉에서의 시간동기 정확도 측정

본 연구에서는 다양한 동기방식과의 객관적인 비교평가를 위해, 각 방식을 대표하는 TPSN과 RBS를 구현하여 실험하였다. 실험방법은 단일 홉 거리에 노드가 배치된 환경에서 5초 간격으로 각각의 프로토콜에 따라 동기화를 수행하고, 샘플링 노드에 의해 송신된 샘플링 패킷의 수신시간을 각 노드의 지역 시간에 따라 타임스탬핑 하여 PC로 전송하는 방식을 취하였다.

실험횟수는 노드 간 동기화를 한 회당 150번씩 3회 실시하여, 총 450회의 실험을 실시하였다. 비교대상 별 동기오차 분포 결과를 표 1에 정리하였다. 결과에서 송신노드-수신노드 방식의 TPSN과는 비슷한 평균 오차를 나타내었으며, 수신노드-수신노드 방식의 RBS와 비교하여 43.4msec 정도의 향상된 동기정확도를 나타내었다. 모든 방식에서 최소오차는 0으로 정확한 시간동기를 수행할 때도 있었으나, 최대오차의 경우 MNTP가 TPSN과 RBS와 비교하여 낮은 분포를 가지는 것을 확인할 수 있었다.

다중 홉 환경에서의 실험은 실제 건물 내 복도에 센서노드를 3.5미터 간격으로 배치하여 다중 홉 환경을 구성하고 그 정확도를 측정하였다. 측정 방식은 단일 홉 방식과 유사하게 진

행되며, 하나의 수신노드가 자신의 동기화를 수행한 후 다음 홉의 수신노드를 동기화하여 다중 홉을 수행하도록 하였다. 다중 홉 실험에서는 앞선 실험에서 비슷한 성능을 보인 TPSN과 그 성능을 비교평가 하였다. 실험은 5초 간격의 측정주기로 한 회당 150번씩 2회를 실시하였으며, 결과는 표 2에 나타내었다. 결과에서 MNTP와 TPSN은 다중 홉 환경에서도 대체로 비슷한 성능을 보였으며, 실험 시 주위환경 및 노드의 부하에 따라 성능의 차이가 발생함을 확인할 수 있다. 특히, 평균 전파지연 시간을 상수값으로 가지는 MNTP는 TPSN과 비교하여 환경변화에 강한 성능을 보였다.

구분	회차	1 Hop	2 Hop
MNTP	1	16.12	17.18
	2	15.56	15.65
	Avg	15.84	16.41
TPSN	1	16.65	17.98
	2	14.23	15.50
	Avg	15.44	16.74

표 2. MNTP과 TPSN 다중 홉 시간동기 오차율 비교

5.2 다중노드, 다중 홉 환경에서의 동기화 메시지 요구량 및 수행시간 측정

본 절에서는 다중 노드 분포에 따른 동기 수행시간과 요구 패킷량의 비교를 위해 그림 4와 같이 20m x 20m의 네트워크 환경을 구성하고 노드를 임의로 배치하여 성능을 TPSN과 비교평가 하였다. 실험에서는 단일 홉 내에 모든 노드가 배치되는 상황을 회피하고, 4홉 거리의 다중 홉 환경을 구성하기 위해 좌표 (0,0)에 위치한 기준노드를 기점으로 한 홉 거리 5m의 가장자리에 최소의 비조정노드 5개를 우선적으로 배치하였다.

실험조건에서 노드 간 무선 전송거리는 5m로, 배치거리는 1m로 제한하였다. 실험방법은 그림 6과 같은 네트워크 환경에 노드의 개수를 70회에 걸쳐 증가시키며 임의로 노드를 배치하여 실제 센서네트워크와 유사한 환경을 구성하여, MNTP와 TPSN의 프로토콜에 따라 동기화 수행 시 필요한 동기화 메시

비교항목 \ 대상	MNTP(S-R)	TPSN	MNTP(R-R)	RBS
평균 오차 (clock ticks)	16.53	16.31	17.18	21.95
최소 오차 (clock ticks)	0	0	0	0
최대 오차 (clock ticks)	42	53	46	84

표 1. MNTP, TPSN, RBS의 단일 홉 시간동기 성능평가

지량과 수행시간을 시뮬레이션 하였다. 무선 통신채널 사용에 따른 MAC 지연은 MNTP와 TPSN 모두 공통적으로 무시하였다.

실험결과는 표 3에 정리된 것과 같이 MNTP는 동기화 메시지량에서 최대 690개의 메시지 절감효과를 보였으며, 처리시간의 경우 최대 8.8초 향상된 수행시간을 나타내어, 다중노드, 다중 홉 환경에 효율적임을 보였다.

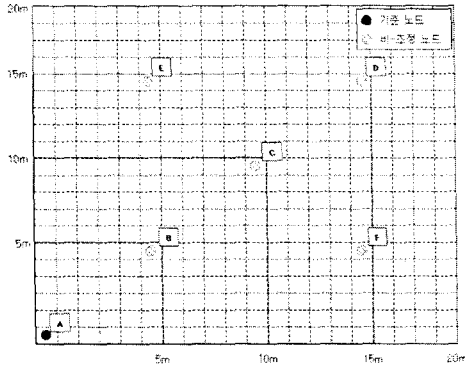


그림 4. 20m x 20m 환경의 기본 네트워크 구성

항목	비교대상		MNTP	TPSN
	최대	최소		
동기화 메시지량	최대	12	702	
	최소	8	12	
동기화 처리시간 (msec)	최대	132	8950.5	
	최소	132	135	

표 3. MNTP과 TPSN 동기화 메시지량, 처리시간 비교

### 6. 결론

본 논문에서는 단일 홉 이상의 다중 홉 영역에 다수의 노드가 배치되는 센서네트워크 환경에 적합한 시간동기화 프로토콜 MNTP를 제안하였다. 기존의 연구들이 동기화 정확도에만 집중된 반면, 제안된 프로토콜은 실제 센서네트워크 환경을 고려하여 기존의 연구들과 비슷한 정확도를 유지하며 다수의 노드들을 신속하고 간결하게 동기화 시키는데 초점을 두었다. 이를 위해, MNTP에서는 RF의 방송특성과 MAC 단계에서의 타임스탬핑을 적절히 조합하여 노드 간 시간동기화를 수행하였다. 실험결과에서 동기노드 간 정확도는 대표적인 수신노드-수신노드간 동기화 방식인 RBS와 비교하여 약 43.4ms 정도 향상된 성능을 보였으며, 대표적인 송신노드-수신노드간 동기방식인 TPSN과는 비슷한 동기정확도를 나타내었다. 그러나, 동기화시 요구되는 메시지량과 수행시간의 실험에서

MNTP의 성능은 TPSN과 비교하여 요구 패킷량과 처리 시간에서 최대 690개의 패킷 감소와 8.8초의 빠른 동기화 수행시간을 보여 다중노드가 분산 배치된 환경에 효율적임을 나타내었다.

### 참고 문헌

- [1] Kopetz, H., and Ochsenreiter, W. Clock Synchronization in Distributed Real-Time Systems. IEEE Transactions on Computers, C-36(8), pp. 933-939, August 1987.
- [2] Kopetz, H., and Schwabl, W. Global time in distributed realtime systems. Technical Report 15/89, Technische Universitat Wien, 1989.
- [3] S. Ganeriwal, R. Kumar, and M.B. Srivastava, Timing-Sync Protocol for Sensor Networks, SenSys'03, pp. 138-149, October 2003.
- [4] Miklos Maroti, Branislav Kusy, Gyula Simon, Akos Ledecz, The Flooding Time Synchronization Protocol, ACM SenSys, '04, pp. 39-49, November 2004.
- [5] Mills, D.L. Internet time synchronization : the Network Time Protocol. IEEE Trans. Communications COM-39, pp. 1482-1493, 10 October 1991.
- [6] J. Elson, L. Girod, and D. Estrin, Fine-Grained Network Time Synchronization using Reference Broadcasts, OSDI 2002, pp. 147-163, December 2002.
- [7] S. Palchauthuri, A. K. Saha and D. B. Johnson, Adaptive clock synchronization in sensor networks, IPSN '04, pp. 340-348, April 2004.
- [8] Saurabh Ganeriwal, Ram Kumar, Sachin Adlakh and Mani Srivastava, "Network-wide Time Synchronization in Sensor Networks," Technical Report UCLA, April 2002.
- [9] J.V. Greunen and J. Rabaey, Lightweight Time Synchronization for Sensor Networks, WSNA'03, pp. 11-19, September 2003.
- [10] Chih-Chieh Han, Ram Kumar Rengaswamy, Roy Shea, Eddie Kohler and Mani Srivastava. SOS: A dynamic operating system for sensor networks, Mobisys'05, pp. 163-176, June 2005