

# 유비쿼터스 센서 네트워크를 위한 효율적인 시간 동기화 프로토콜 연구

신문선<sup>1</sup>, 정경자<sup>2\*</sup>, 이명진<sup>3</sup>

<sup>1</sup>안양대학교 교양학부, <sup>2</sup>충청대학교 디지털마케팅학과, <sup>3</sup>가림정보기술(주)

## A Study on Time Synchronization Protocol to Cover Efficient Power Management in Ubiquitous Sensor Network

Moon-Sun Shin<sup>1</sup>, Kyeong-Ja Jeong<sup>2\*</sup> and Myong-Jin Lee<sup>3</sup>

<sup>1</sup>Div. of Liberal Arts, AnYang Univ., <sup>2</sup>Dept. of Digital Marketing, ChungChung Univ., <sup>3</sup>GalimIT

**요약** 유비쿼터스 센서 네트워크 환경은 다양한 이기종의 센서와 센서들 상호간의 통신을 통해 데이터를 수집하고 제공하여 물리공간의 지능화된 환경을 제공한다. 이러한 상태감지를 위한 센서노드들은 싱크노드와 센서노드로 구성되는데 이기종 센서 노드들간의 시간동기화를 고려하지 않고 전체 센서네트워크의 시간 동기화가 불가능 하게 된다는 문제점이 발생한다. 이러한 이기종 센서노드들간의 시간동기화 문제를 해결하기 위해서 본 논문에서는 싱크노드 아래의 센서노드들 중 싱크노드와 클럭소스가 같은 센서노드를 시간동기 마스터로 설정하고, 싱크노드와 다른 클럭소스를 가지는 센서노드를 마스터 아래에 속하는 시간동기 슬레이브로 설정하여 시간동기 마스터가 동작을 개시할 때에만 시간동기 슬레이브 노드들이 동작하도록 하는 마스터 슬레이브 시간동기화 기법을 제안한다. 제안하는 마스터-슬레이브 토폴로지 기반 시간 동기화 기법은 센서의 설치가 용이하지 않은 USN환경에서 최대 슬립타임을 유지함으로써 센서의 전력소모를 최소화 할 수 있다.

**Abstract** The sensor networks can be used attractively for various application areas. Time synchronization is important for any Ubiquitous Sensor Networks (USN) systems. USN makes extensive use of synchronized time in many contexts for data fusion. However existing time synchronization protocols are available only for homogeneous sensor nodes of USN. It needs to be extended or redesigned in order to apply to the USN with heterogeneous sensor nodes. Because heterogeneous sensor nodes have different clock sources with the SinkNode of USN, it is impossible to be synchronized global time. In addition, energy efficiency is one of the most significant factors to influence the design of sensor networks, as sensor nodes are limited in power, computational capacity, and memory. In this paper, we propose specific time synchronization based on master-slave topology for the global time synchronization of USN with heterogeneous sensor nodes. The time synchronization master nodes are always able to be synchronized with the SinkNode. Then time synchronization master nodes enable time synchronization slave nodes to be synchronized sleep periods. The proposed master-slave time synchronization for heterogeneous sensor nodes of USN is also helpful for power saving by maintaining maximum sleep time.

**Key Words** : Time Synchronization, Ubiquitous Sensor Network, Heterogeneous Sensor nodes, Master Slave Topology

### 1. 서론

유비쿼터스 센서 네트워크(USN) 환경은 다양한 이기

종의 센서와 센서들 상호간의 통신을 통해 데이터를 수집하고 제공하여 물리공간의 지능화된 환경을 제공한다. 이러한 상태감지를 위한 센서들은 싱크노드와 센서노드

\*교신저자 : 정경자(kjeong@ok.ac.kr)

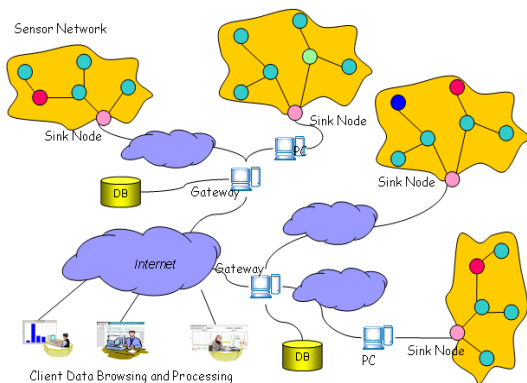
접수일 09년 07월 20일

수정일 (1차 09년 09월 11일, 2차 09년 11월 13일)

게재확정일 10년 03월 18일

로 구성되는데 센서들은 유비쿼터스 센서 망을 통하여 상호 연결되며 중앙의 관제센터와의 연결을 위하여 인터넷에 연결되어 있는 싱크노드에 접속된다. 그림 1은 많은 수의 센서들과 저전력 ad-hoc 네트워크를 형성하는 센서 네트워크의 특징을 보여준다. 그러나 센서네트워크는 노드의 이동성이 거의 없고 센서노드의 크기가 극히 소형으로 제한적인 컴퓨팅 능력과 한정된 배터리를 가지고 센서 필드에 배치되며 일단 설치되고 나면 추가적인 에너지 공급에 제약이 따른다. 따라서 유비쿼터스 센서 네트워크시스템에서 전력소모최소화는 전체 센서네트워크의 수명을 좌우하는 중요한 요소이다[1].

센서의 데이터는 센서들간의 라우팅을 통하여 싱크노드까지 전달되며 이후 인터넷으로 중앙의 관제센터까지 전달된다. USN 환경에서 센서 노드들 간에는 무선통신을 위해서 저전력 무선 센서 노드의 개발과 저전력 데이터 전송 라우팅 알고리즘의 개발이 매우 중요하다.



[그림 1] 이기종의 센서노드들이 혼재하는 USN

대부분의 USN시스템에서 모든 센서 노드들이 동종인 경우가 대부분이지만 향후 광역 복수의USN환경에서는 이기종의 센서노드들이 혼재하게 되며 이기종 센서노드들 간의 시간동기화를 통한 액티브 슬립 타임 제어가 요구된다.

향후 전개될 다양한 이기종의 센서들이 혼재하여 구성되는 유비쿼터스 센서 네트워크 환경에서 동종의 센서들은 클럭소스가 같으므로 기존의 시간동기화 알고리즘을 이용하여 동기화가 가능하다[7,9,14]. 클럭소스 자체는 하드웨어 의존적인 것으로 이기종의 센서노드들이란 싱크노드와 다른 클럭소스를 가지는 센서노드로 일반적인 시간동기화 알고리즘으로 시간 동기화가 불가능 하다.

따라서 본 논문에서는 유비쿼터스 센서 네트워크 환경에서 향후 그 수가 급격히 증가하여 혼재하게 될 이기종의 센서노드들 간의 시간동기화 기법을 제안한다. 제안하는 기법은 센서노드들 간에 마스터-슬레이브(Master-

Slave) 개념을 적용하여 동종의 센서 노드들을 시간동기 마스터(TimeSync Master)로 두고 이기종의 센서노드들은 각각의 마스터에 속한 시간동기 슬레이브(TimeSync Slave)로 두어 마스터노드와 동기화를 함으로써 전체 시간동기화가 가능하도록 하였으며 이는 슬레이브노드들이 최소의 액티브타임을 가지게 되어 최대 슬립 타임을 유지하므로 전력소모의 최소화를 보장하게 된다.

논문의 구성은 다음과 같다. 2장에서는 관련연구로서 기존의 시간동기화 기법들과 센서 네트워크 환경에서의 시간동기화 기법을 분석하고 문제점들을 제시한다. 3장에서는 마스터 슬레이브 토폴로지를 적용한 이기종 센서노드들 간의 시간동기화 기법을 제안하며 응용애플리케이션을 구현하기 위한 task 스케줄링과 알람기능 등에 대하여 기술한다. 4장에서 실제 적용한 실험결과를 기술하고 마지막으로 결론을 맺는다.

## 2. 관련 연구

네트워크 프로토콜로는 Zigbee Alliance에서 네트워크 계층을 표준화하고 있고, Flooding/Gossip, SPIN, LEACH, TEEN, MECN, GAF 등과 같이 전용 프로토콜이 많이 연구되고 있으나 주도적인 프로토콜은 여전 부재 상태이다[1,4]. 대부분의 연구가 노드간 통신을 위한 네트워킹 기술에 중점을 맞추고 있고 복수 이기종의 센서 네트워크 연동을 위한 공통 인터페이스 기술에 대한 연구는 미비하다[11,12].

국내에서도 센서 데이터 관리, 위치 인식, 동기화, TinySec과 같은 정보보호 및 응용 서비스 개발을 위한 미들웨어 개발 등에 대한 연구를 진행하고 있으나 네트워크의 진화에 따른 이기종, 광역, 복수 센서 네트워크를 대상으로 진행되는 연구는 초기단계에 머무르고 있다. 따라서 센서 네트워크의 진화에 부응하는 다양한 USN 응용에 필요한 공통기능을 제공하는 범용 USN미들웨어 기술 개발에 대한 요구사항이 커지고 있으며 이러한 범용 USN 미들웨어의 요소기술로서 센서 데이터 관리, 동기화 및 정보보호 기술 개발 등이 필수적이다.

특히 센서노드간의 동기화와 정보보호는 이기종 센서 네트워크부터 수집된 생생 데이터의 의미적, 형태적 복잡성문제를 표준화된 센서 정보 표현 및 교환기술개발로 해결하기 전에 우선적으로 선행 개발되어야 할 필수적인 요소 기술 들이다.

센서네트워크에서 센서 노드는 기본적으로 배터리의 소모를 줄이기 위해서 노드가 통신을 하지 않을 때에는 최소한의 전력만을 사용하는 슬립(Sleep) 모드로 전환하

는 저전력 기능이 요구된다[15-17]. 싱크 노드는 저전력 기능 및 형상 관리 기능뿐만 아니라 센서 네트워크에 뿌려져 있는 센서 노드들로부터의 센싱 데이터를 수집하여 센서 데이터를 관리하는 중앙관계 시스템으로 보내주는 역할을 수행한다[8]. 중앙관계 시스템은 싱크 노드로부터 수집된 센서 데이터를 분석하여 실시간으로 환경모니터링 시스템에게 전송한 후 센서 데이터를 센서의 타입 및 종류와 시간대별 측정된 센서 값을 분류하여 저장하는 역할을 한다[10].

싱크노드와 센서노드들간의 무선 통신을 위하여 많은 수의 센서노드를 가지고 있는 유비쿼터스 센서 네트워크에서는 센서 노드들간에 시간동기화가 이루어져야 한다 [13]. 무선 센서네트워크의 한 표준으로서 ZigBee 프로토콜[4]은 IEEE802.15.4 프로토콜 기반으로 저전력의 무선 통신 프로토콜로서 저가격, 저전력 소모, 양방향, 무선 통신 표준을 목표로 만들어졌다. 이 표준은 가전제품, 홈 및 빌딩 자동화 산업제어, PC 주변기기, 의료용 센서 응용, 장난감 및 게임에 이르는 매우 폭넓은 응용분야를 지원한다. ZigBee는 주로 저속의 데이터 전송속도를 요구하는 원격 제어나 자동화 등에 적합하다. 특히 Zigbee 토폴로지에 의한 시간동기화 기법의 경우 팬코디네이터에서 주기적으로 비콘메시지를 전송하고 비콘 프레임 안에 있는 정보를 통하여 액티브타임 및 액티브 타임내에서의 타임슬롯을 구성하고 이 타임 슬롯내에서 각 센서노드들이 데이터를 전송한다. Zigbee네트워크는 서로 다른 클럭 소스를 사용하는 디바이스들로 구성되지만 기본적으로 비콘메시지를 통하여 동기화가 가능하게 된다. 그러나 실질적으로 중간에서 비콘프레임을 중계해주어야 하는 코디네이터나 라우터에는 항상 전원공급이 이루어져야 하므로 기존의 파워 온/오프에 의한 액티브타임과 슬립타임의 설정방식의 시간동기화만으로는 전력소모의 최소화를 이룰 수 없다. 뿐만 아니라 비콘프레임의 한계는 5분 정도의 주기로 동작을 한다는 것이다. 센서네트워크에서 시간동기화가 필요한 시간주기는 응용도메인에 따라서 다양하다. 1분, 하루, 일주일, 때로는 한 달이 될 수도 있기 때문에 Zigbee 토폴로지를 일률적으로 적용하는 것에는 많은 문제점이 있다. 이러한 문제점들을 해결하기 위해 본 논문에서는 이기종 센서노드를 포함하는 광역의 USN환경에서 센서노드들의 전력소모 최소화를 위하여 이기종 센서노드들의 시간동기화를 위한 마스터-슬레이브 토폴로지 기반의 시간동기화 프로토콜을 제안한다.

### 3. MSTSP 프로토콜

USN 환경에서는 각 센서노드들은 배터리로 전원을 공급받아 동작하는 것이 일반적이다. 따라서, 한정된 자원인 배터리 전원을 최소한으로 사용하는 것이 중요한 요구사항이 되고 있으며, 이를 구현하기 위한 다양한 방법이 있을 수 있다[2,5,6].최소한의 전원을 가지고 센싱과 무선통신기능을 수행해야 하는 USN 시스템에서는 센서 네트워크의 모든 노드들이 Active/Sleep 시간을 번갈아가며 동작하도록 구현되며, Sleep 시간에는 센서노드는 최소한의 전원으로 동작한다. 이는 USN환경에서 센서노드들이 아이들 라디오 리스닝 동안에 가장 많은 전력을 낭비하는 것을 방지하기 위한 것이다. 따라서 모든 노드들이 유기적으로 동작하기 위해서는 모든 노드가 주기적으로 동시에 온/오프하고 액티브타임과 슬립타임을 가지기 위해 멀티홉 시간동기화가 이루어져야 하며 이를 위해서는 시간동기화와 타스크 스케줄링이 선행되어야 한다. 즉 모든 센서노드는 동시에 Active/Sleep 시간을 반복해야 하며, 이 동시성을 만족하기 위해서는 센서노드들간 시간 동기화가 이루어져야 한다.

#### 3.1 Ad-hoc, Multi-hop 라우팅

본 연구에서 개발되는 이기종 센서노드의 시간 동기화 프로토콜인 MSTSP는 실제 구현된 USN 시스템에 적용될 것으로 실제 USN 시스템에서는 시간동기화 알고리즘을 사용하여 각 센서 노드간 시간동기화를 이루고, 이것을 이용하여 센서노드가 Sleep/Active Mode를 동시에 반복할 수 있는 응용을 구현한다.

[표 1] 라우팅 테이블

Symbol	type	Meaning
id	uint16	Neighbor address
parent	uint16	Neighbor 의 부모 주소
missed	uint16	missed된 메시지 개수
received	uint16	받은 메시지 개수
lastSeqno	int16	받은 마지막 일련번호
flags	uint8	Entry의 상태, (Valid, New, Init)
hop	uint8	Neighbor e로부터 sink까지 hop의 개수
receiveEst	uint8	Neighbor 로 부터 내 자신까지 추정값(Link 상태)
sendEst	uint8	나로부터 Neighbor 까지 추정값(Link 상태)

또한, 내부 crystal clock 속도가 다른 이기종 센서노드들이 동시에 Sleep/Active Mode를 반복할 수 있도록 Master-Slave 개념을 적용하며 이러한 개념의 도입으로, 네트워크 라우팅 경로 설정의 방법에 변형이 필요할 것이다. 또한 동시에 Active된 센서노드들이 동시에 같은

일을 수행함으로써 발생할 수 있는 네트워크 충돌을 방지하기 위하여, 랜덤 휴식시간을 포함한, 각 센서노드들이 수행하는 Task에 대한 스케줄링을 재정의 한다. WSN(Wireless Sensor Network)의 각 노드들은 무선통신을 통하여 센싱데이터 및 제어 데이터를 송수신한다. 무선통신은 전파의 전송범위가 제한되어있기 때문에 (IEEE 802.15.4인 경우 10m ~ 100m) 하나의 센서노드에서 직접적으로 (Direct) 싱크노드와 통신할 수 없다. 따라서, 대부분의 센서네트워크에서는 센서노드에서 싱크노드로 데이터를 전송하기 위해서는 여러 센서노드를 거쳐서 데이터를 전송하게 된다. 또한, 데이터를 전송하는 경로에 있는 센서노드 중 하나의 센서노드가 이상이 생긴 경우 (배터리 소진, 장애물에 의해 무선 전송 불가능 등...)에는 전송 경로가 자동으로 재 설정되어서, 데이터 전송에 문제가 없어야 한다. 이러한 이유로 인하여 센서네트워크의 모든 센서노드들은 주위의 센서노드와 지속적인 정보 교환을 통하여 자신의 데이터가 싱크노드로 전달될 수 있는 전송 경로를 설정하고 유지하며, 네트워크의 변화에 따라서, 경로를 재설정할 수 있는 기능을 가져야 한다.

- Shortest Path First Algorithm : 싱크노드까지 가는 가장 작은 hop count(싱크노드까지 데이터를 보낼 때 거쳐가는 노드의 숫자)를 가진 노드를 부모(Parent)노드로 선정한다. 부모노드가 선정되면 이후의 모든 데이터는 부모노드에게 전달한다.

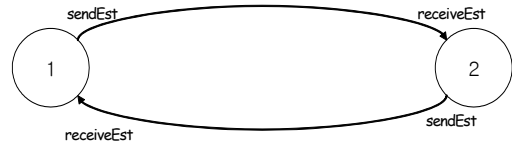
- Distributed Algorithm : 라우팅 경로를 설정하기 위해서 수행되는 알고리즘은 중앙집중(Centralized)되어 하나의 노드에서 제어를 받는 것이 아니라, 각 센서노드에서 독립적으로 수행되는 알고리즘이다.

- Table Driven Algorithm : 주위의 센서노드들 중에 하나의 부모노드를 선정하기 위하여, 주위의 센서노드들에 대한 라우팅 정보를 메모리에 테이블 형태로 저장한다. (센서노드는 하드웨어 제약(Hardware Constraint)이 있는 경우가 많기 때문에, 메모리의 크기가 충분하지 않다. (2 kbyte, 4 kbyte) 따라서, 라우팅 테이블의 크기에 제약을 가질 수 밖에 없고, 본 시스템에서는 라우팅 테이블에 유지하는 주의 센서노드의 개수를 8개 또는 16개로 제한했다)

- Link Estimation 방법 : 부모노드로 선정가능한 주의 노드들이 여러 개인 경우 (hop count가동일한 후보 부모노드가 있는 경우), 각 노드과 자기자신과의 Link Quality를 계산하여, Link Quality가 가장 좋은 노드를 부모노드로 선정한다. Link Quality를 계산하는 방법은 수신하는 패킷의Sequence Number를 가지고, missed 패킷의 수와 received 패킷의 수를 계산하고 missed 패킷의 수가 적은 Link는 Quality가 높다고 판단한다.

- 라우팅 테이블 : 하나의 센서노드에 저장되는 라우팅테

이블의 내용은 다음 표 1과 같으며 이 내용이 1개의 주위 센서노드당 1 Row씩 저장된다.



[그림 2] 두 노드간 ReceiveEst와 SendEst 값의 관계

[표 2] 라우팅 테이블 업데이트 수행 알고리즘

```

1. 현재 라우팅테이블에서 해당 노드의 Entry가 있는지 검사
1.1 없으면, 새로운 Entry생성
    1.1.1 라우팅테이블의 남은 공간이 없는 경우, sendEst값이 제일 작은 것을 대체
    1.1.2 Tbl.received++, Tbl.lastSeqno = seqno
1.2 있으면,
    1.2.1 중간에 잃어버린 패킷수(delta)
        = 현재 받은 seqno 테이블에 기록된 마지막 seqno - 1) 값 계산
    1.2.1 delta >= 0 일때,
        Tbl.missed += delta, Tbl.received++, Tbl.lastSeqno = seqno
    1.2.2 delta 0 < 이면서, delta < -20 (ACCEPTABLE_MISSED) :
        새로운 Entry로 등록
    1.2.3 return FALSE (메시지가 중복되었다는 의미)
2. Table 값 설정 : Tbl.parent = parent, Tbl.hop=hopcount
3. 받은 estEntry중에 내 노드의 정보가 있으면
   보면 노드의 Tbl 정보에서: Tbl.sendEst = receiveEst, Tbl.liveliness++
    
```

$$newAve = (255 * received) / (received + missed) \quad missed \text{가 증가될 수록 } newAve \text{는 } 255 \text{보다 작아짐}$$

$$초기 \quad receiveEst =: newAve;$$

$$2회이후 \sim receiveEst =: ((2 * receiveEst) + newAve * 6) / 8$$

- Link Quality 값 계산 : 본 라우팅 알고리즘에서 부모노드를 선정하는 방법은 싱크노드까지 hop count가 가장 작은 노드를 선정하는 Shortest Path First방식이다. 그러나 부모노드의 후보가 여러 개인 경우 가장Link Quality가 큰 노드 중에 부모노드를 선정하게 된다. Link Quality는 ReceiveEst (Receive Quality Estimation)과 SendEst (Send Quality Estimation)을 가지고 계산하게 되는데 두 노드(1번 노드, 2번 노드)간의 ReceiveEst와 SendEst 값의 관계는 다음 표 22와 같다.

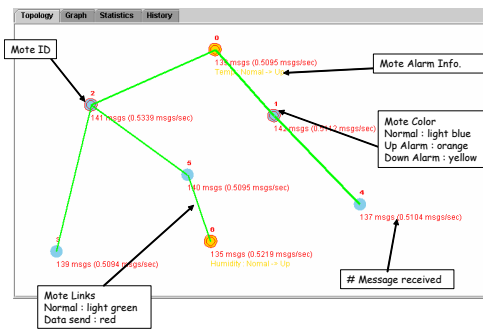
1번 노드에서 2.receiveEst (2번 노드로부터의 ReceiveEst 값)를 계산하여 (5번 주기로) 결과 값을 2번

노드로 보내주면, 2번 노드에서 받아 1.sendEst(1번 노드의 SendEst 값)에 같은 값으로 설정한다. 모든 노드에서는 주위 노드로 부터의 ReceiveEst 값만 계산하면 되고, 주위 노드로의 SendEst 값은 해당 노드로부터 ReceiveEst값을 받아 설정하게 된다. 다음은 ReceiveEst 값을 계산하는 방법이다.

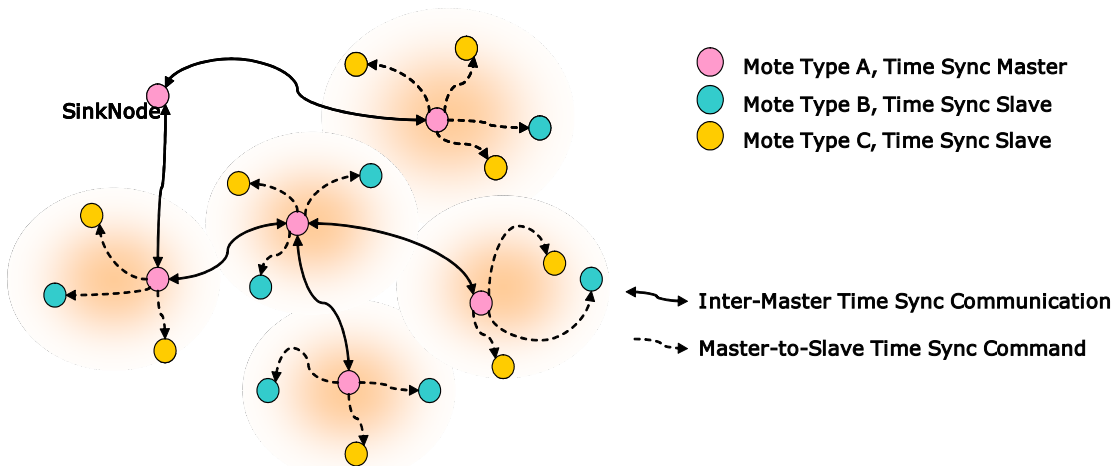
새로운 newAve값을 그대로 receiveEst값에 설정하지 않고, 기존의 receiveEst값을 반영하여 설정함 (급격하게 증가되거나, 감소하는 것을 방지함)

• 알고리즘 수행과정

알고리즘 수행 과정은 크게 주위노드로부터 RouteMsg를 받았을 때 수행하는 라우팅 테이블 Update 과정과 Timer를 통하여 주기적으로 수행되는 과정 (ReceiveEst값을 계산하고, RouteMsg를 전송하며, 부모를 설정하는 과정)으로 나뉜다. 그림 3은 라이팅 테이블 업데이트 알고리즘 수행과정을 나타낸 것이며 그림 3은 위의 라우팅알고리즘 수행 결과 예를 보여준다.



[그림 3] 라우팅 알고리즘 수행 예



[그림 4] 이기종 센서노드를 위한 마스터-슬레이브 토폴로지

3.2 이기종 센서노드 시간 동기화

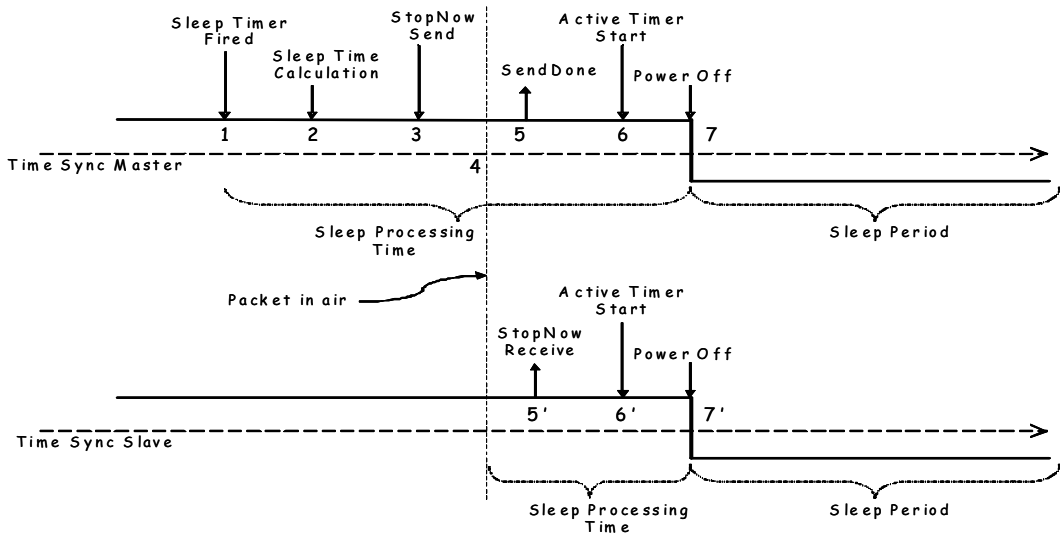
이기종 센서 노드들간에 시간 동기화를 위해서는 Master-Slave 개념을 적용한다. 즉 싱크노드와 동종인 센서 노드들을 Time-Synchronize Master로 두고 이기종의 센서노드들은 각각의 Master 에 속한 Time-Synchronize Slave 로 두어 마스터노드와 동기화를 함으로써 전체 시간동기화가 가능하게 된다. 이렇게 함으로써 슬레이브노드들이 최소의 active time을 가지게 되어 전력소모의 최소화를 보장하게 된다. 그림 4는 Master Slave Topology 개념을 적용하여 이기종 센서노드 들간의 시간동기화를 도식화 한 것이다.

마스터 노드들은 싱크노드와 같은 종류의 센서노드들로서 싱크노드와 시간동기화에 기존의 알고리즘을 적용하는데 문제가 없으며 각 슬레이브들은 마스터를 부모 노드로 선택하여 마스터노드의 Active/Sleep time 제어를 받아 전체 시간동기화가 가능해지는 것이다.

Time-Synchronize Master 노드와 Time-Synchronize Slave 노드에서 수행하는 기능에 대한 설명은 다음과 같다.

[표 3] Master/Slave 노드의 기능

기능	단일형	혼합형		비고
		Master	Slave	
Time Sync Protocol	Send/Receive	Send/Receive	X	
Time Sync Command	X	Send	Receive	
Routing Protocol	Send/Receive	Send/Receive	Send/Receive	
Parent Enable?	O	O	X	



[그림 5] Master와 Slave간의 제어명령에 따른 수행 Sequence

노드에서 수행하는 기능에 대한 설명은 다음과 같다. 위의 표 3에서 단일형이란 이기종 센서네트워크가 아닌, 하나의 기종만으로 이루어진 센서네트워크로서, 이러한 경우에는 모든 센서노드가 위에서 설명한 시간동기화 알고리즘 수행을 통하여 시간동기화를 이루기 때문에 Master/Slave개념이 없고, 네트워크 라우팅 기능도 부모 노드의 설정에 제약이 없다.

혼합형이란, 이기종 센서노드들로 이루어진 센서네트워크로서, Master노드인 경우 위에서 설명한 시간동기화 알고리즘의 수행을 통하여 시간동기화를 이루며, 자신의 Slave 노드에게 Active/Sleep Mode 제어 명령을 전송한다. 또한, Master 노드인 경우 네트워크 라우팅 기능에서도 부모노드가 될 수 있다. Slave 노드인 경우에는 위에서 설명한 시간동기화 알고리즘을 수행하지 않으며, 단지 Master노드로부터 Active/Sleep에 대한 명령을 받아 그것에 따라 Active/Sleep Mode를 반복하게 된다. 또한, Slave 노드는 네트워크 라우팅의 부모 노드가 될 수 없다.

Master노드가 Slave에 대한 Active/Sleep Mode의 변경을 제어하는 것으로서, Master 노드는 계산된 Global Time에 맞추어 Sleep Mode로 들어가기 전에, Slave 노드들에게 "Stop Now" 메시지를 전송(브로드캐스트)한다. 이 메시지에는 Slave들이 얼마만큼의 시간동안 Sleep Mode에 있다가 다시 Active Mode로 들어올 것인가에 대한 시간정보가 포함되어 있다.

"Stop Now" 메시지를 수신한 Slave 노드들은 수신 즉시 메시지에 지정된 시간만큼 Sleep Mode로 들어간다.

다음 그림 5는 Master와 Slave간의 제어명령에 따른 수행 Sequence를 보여준다.

1. [Master] Sleep Timer Fired : Master 노드가 Sleep Mode로 변경되어야 함을 소프트웨어 모듈에게알린다. 센서노드가 Active 상태로 들어가면서, Sleep Time만큼 Timer를 설정해 놓으면, 해당 시간이 경과되면 Timer에서 Interrupt가 발생하여 소프트웨어 모듈이 인식하게 되는 것이다.
2. [Master] Sleep Time Calculation : 애플리케이션에서 정의된 Sleep 시간을 만족하기 위해서는 지금부터 몇 밀리초를 Sleep 상태로 있어야 하는 지를 계산한다. (위의 "Sleep시간을 계산하는 방법은 다음과 같다."를 참조). 이것은 매 Active/Sleep주기마다 시간동기화를 수행하여 Global time를 설정하기 때문에, 매 주기마다 각 노드가 Sleep 해야 할 시간이 다를 수 있고, 그렇기 때문에 매 주기마다 다시 계산해 주어야 한다. 여기서 계산된 시간값을 TS 라 한다.
3. [Master] "Stop Now" Send : 주위의 Slave 노드들에게 전달될 제어 패킷을 Radio를 통하여 전송하는 명령을 수행한다. 이 패킷에는 계산된 TS 을 포함된다. 이렇게 센서노드의 Radio 전송명령을 수행하게 되면, 해당 소프트웨어 및 드라이버 모듈이 Radio 칩에 데이터를 전달하며, 해당 Radio Chip은 데이터 버퍼가 준비되는 대로, 해당 패킷을 전파로 전송한다.
4. [Packet in air] 실제 패킷이 전파를 통하여 공기 중

에서 전달되는 과정이다. Master에서 송신되는 주파수는 주위의 모든 노드들에게 전달된다.

5. [Master] Send Done : Radio 칩에서 데이터를 전파를 통하여 전송하고 나면, Radio 칩에서는 데이터 전송이 끝났음을 알리는 Interrupt가 발생하고, 끝났음을 소프트웨어 모듈에게 알린다.

5'. [Slave] "Stop Now" Receive : Radio 칩의 수신 모듈에서 전파를 통하여 데이터를 수신하면, Radio 칩은 데이터의 수신을 알리는 Interrupt가 발생하고, 수신이 되었음을 소프트웨어 모듈에게 알린다.

6. [Master] Active Timer Start : 계산된 TS 시간만큼 후에 Interrupt가 발생하는 Active Timer를 설정해 놓는다.

6' [Slave] Active Timer Start : Master에서 받은 TS 시간만큼 후에 Interrupt가 발생하는 Active Timer를 설정해 놓는다.

7 과 7'. [Master][Slave] Power Off : Sleep 모드가 진행되는 동안 센서노드의 전원사용을 최대한으로 줄이기 위하여, 센서노드의 대부분의 컴포넌트를 Power Off하게 된다. 이곳에서 Power Off되는 컴포넌트들은 Sensor H/W, Radio Chip, Clock (Active Timer Fired를 위한 Clock은 제외)등이다. 이 단계가 지나면 센서노드는 하나의 Clock Source만을 제외한 나머지 컴포넌트가 Off가 되며 최대 절전 모드에 있게 된다.

이기중 센서노드간의 시간동기화를 이루도록 하기 위한 알고리즘은 다음과 같다.

▪ Distributed Algorithm : 본 알고리즘은 중앙 집중(Centralized)되어 하나의 노드에서 제어를 받아 시간동기화를 이루는 것이 아니라, 각 센서노드에서 독립적으로 수행되어 참여하고 있는 모든 센서노드들의 시간을 동기화하는 알고리즘이다.

▪ Root Selection : 본 알고리즘이 수행되는 모든 센서노드들은 서로 시간동기화 정보를 담은 패킷을 송수신하면서, ID값이 가장 작은 센서노드를 시간동기화 Root 노드로 선정한다. (ID값이 제일 작은 센서노드는 보통 ID값이 0이며, 이 노드는 게이트웨이에 연결되는 싱크노드가 되는 것이 일반적이다. 또한, 싱크노드는 배터리로 동작하는 것이 아니고, 고정된 전원을 공급받는 것이 일반적이다.) 선정된 Root 노드의 Internal Clock을 기반으로 전체 센서네트워크의 Global Time이 결정된다.

▪ Message Broadcast : 동기화된 시간을 유지하기 위하여 각 센서노드들은 시간동기화에 관련된 정보가 포함되는 패킷을 주기적으로 주위노드들에게 Broadcast하고, 이 패킷을 받은 센서노드는 해당 정보를 가지고 내부의 시간동기화 계산을 수행하며, 다시 그 정보를 Broadcast한다.

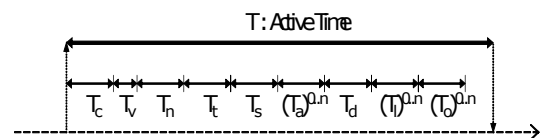
▪ Time Stamping : 하나의 노드에서 다른 노드로의 패킷전송은 실제로 많은 과정을 거쳐서 이루어지며, 각 과정에서 소프트웨어 프로그램의 동작 시간을 결정하는 것은 일반적인 Task Scheduling OS에서는 불가능하다. 본 알고리즘에서는 패킷전송의 과정 중에서 Radio Chip의 안테나를 통하여 실제 패킷을 전송하는 시점과, Radio Chip의 안테나를 통하여 실제 패킷을 수신하는 시점에서, 그 패킷의 특정 필드에 Time Stamping을 함으로서, 시간 동기화의 오차를 최소한으로 줄인다. 물론, 본 알고리즘이 동작하기 위해서는 Radio Chip에서, 실제 패킷을 전송/수신하는 시점에 Interrupt를 발생시켜 Interrupt Handler에서 송수신되는 패킷을 제어할 수 있어야 한다.

▪ Multi-hop : 전체 센서네트워크에 있는 센서노드 중 Root로 선정된 센서노드와 그 외 센서노드들은 Radio 전송 범위 안에 있을 수 있지만, 그렇지 않은 센서노드들이 있을 수 있다. 이렇게Root의 Radio 전송범위 밖에 있는 센서노드들은 Root에서 전송하는 Global Time 메시지를 직접 받지 못하고, 중간에 있는 센서노드들의 중계를 통해서 메시지를 받는다. 이렇게 중계를 통해서 받은Global Time 메시지는 직접 받은 Global Time 메시지와 같은 처리과정을 거쳐서, 시간동기화를 이룬다. 일반적으로 이렇게 multi-hop으로 시간동기화가 이루어지는 경우, 1개의 hop당 약 + 2jiffes ( 1 jiffes는 1/32k secs)가 생길 것으로 예측된다.

### 3.3 Task Scheduling

동시에 Active/Sleep Mode를 반복하는 센서네트워크 내의 모든 센서노드들이 Active Mode에서 수행해야할 Task들의 순서를 정의한 것이다.

Task 스케줄링이 필요한 이유는 정확한 시간동기화를 통하여 센서네트워크 내의 모든 센서노드들이 동시에 Active Mode에 들어가게 되는데, 이러한 Active Mode에서 수행되는 일들이 인접노드간의 간섭을 일으키지 않고, 응용 애플리케이션에서 요구하는 작업을 수행할 수 있도록 하기 위함이다. Active Mode에서 수행되어야 할 Task를 스케줄링한 것은 다음 그림 6과 같다. 이러한 스케줄링의 결과로 나타나는 모든 Task의 실행 시간 합은 전체 Active Time보다 작아야 한다. 각 Task에 대한 설명은 다음과 같다.



[그림 6] Task 스케줄링

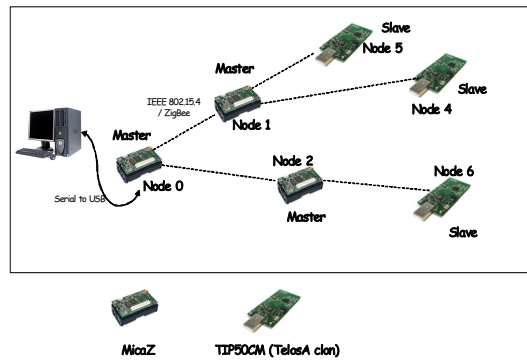
- Tc : 관리자 또는 Gateway에서 센서노드로 전달되는 Command를 처리하는 Task이다. 이 Task를 Active Time의 맨 처음에 둔 이유는 Command는 매우 적게 발생하는 행위이며, 시간동기화에 의해 발생할 수 있는 미세한 시간의 차이는 각 센서노드에서 Command를 기다리는 시간에 의해 해결될 수 있기 때문이다. 이 Task에서 각 센서노드는 자기 자신으로 전달되는 Command를 기다리게 되며, 만약에 전달되는 Command가 있는 경우 해당 Command를 처리하거나 또는 다른 센서노드로 Command를 전달하게 된다.
- Tv : Variable Time. 각 센서노드마다 랜덤하게 시간을 지정하고 해당시간만큼 아무것도 하지 않는 시간이다. 실제로는 각 센서노드마다 다른 Tc의 연장이라고 볼 수 있다. 각 센서노드마다 랜덤하게 이러한 시간을 둔 이유는 이 후 다른 Task들이 서로 다른 시간에 수행되도록 하기 위함이다. 만약에, 이후 다른 Task가 동시에 동작한다면, 메시지를 동시에 전송하게 되고, 이것은 네트워크 채널의 경쟁을 발생시켜서, 메시지의 전송 실패를 발생시키게 된다.
- Tn : Network Routing Time. 주위노드에서 전송한 네트워크 라우팅 메시지를 수신하여 처리한다.
- (Tt) : Time Sync. Time. 주위노드로부터 전달되는 시간동기화 메시지를 처리한다. Time Sync. Slave 인 경우 이 Task는 수행하지 않고 다음 Task로 즉시 넘어간다.
- Ts 0..n : Sensing Time. 기 설정된 센서를 통하여, 센싱을 수행한다.
- (Ta)0..n : Actuator Time. 구동기(Actuator)를 동작해야 하는 상황이면 동작시킨다. 구동기를 구동할 상황이 아니면 즉시 다음 Task로 넘어간다.
- (Td) : Sensing Data Sending Time. 수집된 센싱 값을 싱크노드로 전달한다. 전달하는 경로는 네트워크 라우팅 기능에 의해서 선정된 부모노드로 전달한다.
- (Tl) 0..n : Alarm Message Sending Time 알람 상황이 발생하면, 해당 메시지를 싱크노드로 전달.
- To : Operation Info. Sending Time. 게이트웨이에서 Command로 Operation Info.를 요청한 경우 해당 메시지를 싱크노드로 전달한다.

어떠한 Task를 수행한다고 하더라도 Active Time안에 자기 자신의 센서노드로 전달되는 어떠한 메시지도 정상적으로 처리된다.

#### 4. 실험

실험 환경을 구성하여 서로 다른 센서 노드들로 가상

의 센서 네트워크를 구성하고 시간동기화와 데이터 전송을 시뮬레이션하여 본 논문에서 제안하는 이기종 센서노드들의 시간동기화 프로토콜을 검증하였다.. 이기종 광역 센서네트워크를 위한 효율적인 시간동기화 프로토콜로서 MSTSP의 시간동기화 유효성검증을 위해 이기종의 센서노드들로 USN환경을 구성하여 시간동기화와 최소 전력소모를 보장하기 위한 최대 sleep time 을 실험하였다. 그림 8은 간단한 실험환경을 구성해본 것이다 논문에서 제안한 마스터-슬레브 토폴로지를 적용한 이기종 센서노드들의 시간 동기화의 검증을 위해서 싱크노드와 동종의 센서노드와 이기종의 센서노드로 구성된 실험환경은 Time Sync. Master 3개, Slave 3개로 그림 7과 같이 구성하였다. 각 노드에서 센싱하고 게이트웨이로 전달한 데이터가 도착한 시간을 측정함으로써 이기종 센서노드들의 시간동기화를 검증할 수 있었다. 표 4는 실험에 사용된 노드들의 사양을 보여준다.



[그림 7] 이기종의 Mote로 구성된 실험 환경

[표 4] 실험에 사용된 센서 노드 사양

구분	Master	Slave
Name	MicaZ	TIP50CM (TelosA clone)
Processor	8bit RISC 16MHz	16bit RISC 8MHz
Radio Chip	CC2420	CC2420
Clock	921.6 khz	32khz
Range	10m ~ 100m	10m ~ 100m
Sensor	Temperature, Photo	Temperature, Photo, Humidity
Battery	AA 2개	AA 2개
Power	2.7 ~ 3.3 V	2.7 ~ 3.3 V

실험은 이기종의 Mote들로 구성된 가상의 센서 네트



워크 환경에 마스터와 슬레이브노드들에서 센싱하고 게이트웨이로 전송된 데이터 도착시간을 측정하여 논문에서 제안한 것처럼 시간동기화 알고리즘이 최소의 active time을 보장하여 최소의 전력소비만으로 시간동기화가 이루어지는지 확인을 하는 것으로 수행되었다.

총 실험시간은 15분이며 Active Time을 5초로, Sleep Time을 30초로 주었다 시간은 각 노드의 패킷이 Gateway에 도착한 시각이며 Cycle Num은 각 노드에서 전송하는 패킷의 일련번호이다. 실험 결과 이기종의 센서 노드들의 시간동기화가 이루어지고 있음을 확인할 수 있었다. 표 5는 실험 결과를 보여주고 있으며 슬레이브 노드들과 마스터 노드들 간에 시간동기화가 이루어지고 있음을 알 수 있었다.

[표 5] 실험 결과

횟수	Node 1		Node 2		Node 4		Node 5		Node 6	
	시간	Cycle Num	시간	Cycle Num	시간	Cycle Num	시간	Cycle Num	시간	Cycle Num
1	00:00:30	704	00:00:30	719	00:00:30	584	00:00:30	684	00:00:30	665
2	00:01:05	705	00:01:05	720	00:01:05	585	00:01:05	685	00:01:05	666
3	00:01:40	706	00:01:40	721	00:01:40	586	00:01:40	686	00:01:40	667
4	00:02:14	707	00:02:14	722	00:02:15	587	00:02:15	687	00:02:15	668
5	00:02:49	708	00:02:49	723	00:02:49	588	00:02:49	688	00:02:49	669
6	00:03:24	709	00:03:24	724	00:03:24	589	00:03:24	689	00:03:24	670
7	00:03:59	710	00:03:59	725	00:03:59	590	00:03:59	690	00:03:59	671
8	00:04:34	711	00:04:33	726	00:04:34	591	00:04:34	691	00:04:34	672
9	00:05:08	712	00:05:08	727	00:05:08	592	00:05:09	692	00:05:09	673
10	00:05:43	713	00:05:43	728	00:05:43	593	00:05:43	693	00:05:43	674
11	00:06:18	714	00:06:18	729	00:06:18	594	00:06:18	694	00:06:18	675
12	00:06:53	715	00:06:53	730	00:06:53	595	00:06:53	695	00:06:53	676
13	00:07:28	716	00:07:27	731	00:07:28	596	00:07:28	696	00:07:28	677
14	00:08:02	717	00:08:02	732	00:08:02	597	00:08:03	697	00:08:03	678
15	00:08:37	718	00:08:37	733	00:08:37	598	00:08:37	698	00:08:37	679
16	00:09:12	719	00:09:12	734	00:09:12	599	00:09:12	699	00:09:12	680
17	00:09:47	720	00:09:47	735	00:09:47	600	00:09:47	700	00:09:47	681
18	00:10:21	721	00:10:21	736	00:10:22	601	00:10:22	701	00:10:22	682
19	00:10:56	722	00:10:56	737	00:10:56	602	00:10:56	702	00:10:56	683
20	00:11:31	723	00:11:31	738	00:11:31	603	00:11:31	703	00:11:31	684
21	00:12:06	724	00:12:06	739	00:12:06	604	00:12:06	704	00:12:06	685
22	00:12:41	725	00:12:40	740	00:12:41	605	00:12:41	705	00:12:41	686
23	00:13:15	726	00:13:15	741	00:13:15	606	00:13:16	706	00:13:16	687
24	00:13:50	727	00:13:50	742	00:13:50	607	00:13:50	707	00:13:50	688
25	00:14:25	728	00:14:25	743	00:14:25	608	00:14:25	708	00:14:25	689
26	00:15:00	729	00:15:00	744	00:15:00	609	00:15:00	709	00:15:00	690

결과적으로 이기종 센서노드가 시간동기화를 통한 데이터 수집이 정상적으로 이루어짐을 확인할 수 있었다. 실제 센서네트워크상에서 실험을 하는 것은 불가능하여 가상의 센서네트워크환경을 만들어서 실험을 하였으나 향후 하나의 마스터 노드 아래에 둘 수 있는 슬레이브의 적절한 개수에 대한 실험 등이 필요하다. 하나의 마스터에 너무 많은 슬레이브가 종속될 경우 병목현상이 발생할 수도 있으므로 실제 USN시스템에서는 슬레이브가 마스터를 parent로 선택할 수 있도록 하는 방법도 고려되어야 할 것이다. 그러나 필요에 따라서 그룹을 적당한 크기로 나누어주는 일을 수행해야 할 필요성도 있다.

5. 결론

무선센서 네트워크는 배터리로 동작하는 다수의 센서 노드들로 구성되며 센서노드들은 배터리 동작을 최소화 하기 위해 주로 액티브타임과 슬립타임을 반복하면 동작 하게 된다. 즉 액티브타임일 때에는 센서를 통한 센싱 및 데이터 처리와 무선통신 등의 기능을 수행하게 되며 최대한의 슬립 타임을 유지하게됨으로써 센서 노드의 전원 소모를 최소화한다. 데이터는 싱크노드를 통해서 인터넷 과 연결되면 서버로 데이터를 전송하는 것이 가능한데 무선전송의 범위가 제한되므로 중간의 다른 노드들을 경유 하여 전송되게 된다. 따라서 모든 센서노드들이 동시에 액티브 및 슬립모드를 수행해야 하며 이를 위해서 global time 동기화가 필수적이다.

본 논문에서는 이기종의 센서노드들이 혼재하게 되는 센서네트워크에서 전력소모 최소화를 위하여 마스터-슬레이브 토폴로지를 적용한 이기종 센서노드들의 시간동기화 기법을 제안하였다. 제안된 기법은 동종의 센서노드들을 시간동기마스터로 설정하여 시간동기화 알고리즘을 이용하여 싱크노드와 시간동기화가 가능하며 이기종의 센서노드들은 시간동기 마스터 아래에 시간동기 슬레이브로 두어 마스터가 슬레이브에게 신호를 보내서 마스터-슬레이브간 시간 동기화가 이루어지므로 모든 센서노드들의 시간동기화가 가능해질 뿐만 아니라 계속해서 파워 온을 유지해야할 센서노드가 필요치 않으므로 기존의 방식보다 전력소모를 줄일 수 있게 된다.

그러나 하나의 마스터에 너무 많은 슬레이브가 종속될 경우 병목현상이 발생할 수 있다는 문제점이 있으므로 이를 해결하기 위해서 슬레이브가 마스터를 선택 할 수도 있게 하는 해결방안도 고려할 수 있으며 혹은 필요에 따라 그룹을 적당한 크기로 나누어주는 일을 수행해야 할 필요성도 있다.

향후 시뮬레이션을 통해서 제안된 마스터-슬레이브 기반 시간동기화 프로토콜 (MSTSP)과Zigbee를 비교하여 MSTSP가 전력소모 가 더 적음을 검증하는 것이 필요하다.

참고문헌

[1] G.Asada, M.Dong, T. Lin, F.Newberg, G.Pottie, W.Kaiser and H.O.Marcy, "Wireless integrated network sensors:low power systems on a chip," in Proc. Of the 24th European Solid-State Circuits Conference, The Hague, Netherlands, Sept, 1998.  
 [2] M. Lemmon, J. Ganguly, and L. Xia, "Model-based clock synchronization in networks with drifting clocks", in Proc. Of the 2000 Pacific Rim

International Symposium on Dependable Computing, Los Angeles, CA, Dec, 2002.

[3] R. Ostrovsky and B. Patt-Shamir, "Optimal and efficient clock synchronization under drifting clocks", in Proc. of the 18th ACM Symposium on Principles of Distributed Computing(PODC1999), 1999.

[4] <http://www.zigbee.org/>

[5] D. L. Milis, "Internet time synchronization: the network time protocol", IEEE Trans. Communications, vol. 39, no. 10, pp.1482-1493. Oct, 1991.

[6] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks," in Proc. Of the 2001 International Parallel and Distributed Processing Symposium(IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, Apr, 2001.

[7] K. Romer, "Time synchronization in ad hoc networks," in Proc. Of ACM Mobihoc, Long Beach, CA, 2001.

[8] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", published by Elsevier Science, Dec, 2001.

[9] J. Heidemann, F. Silva, C. Intanagonwiwat, "Building efficient wireless sensor networks with low-level naming ," in Proc. Of the Operating Systems Principles, Banff, Canada, 2001.

[10] C. Intanagonwiwat, R. Govindan, D, Estrin, "Directed diffusion : a scalable and robust communication paradigm for sensor networks in Proc. of the ACM MobiCom'00, Boston, Ma, pp.55-67, 2000.

[11] C. Jaikaeo, C. Srisathapornphat, C. Shen, "Diagnosis of sensor networks", IEEE International conference on Communications ICC'01, Helsinki, Finland, Jun, 2001.

[12] Yang.H and Sikdar. B "A Protocol for tracking mobile targets using sensor network," IEEE Workshop on Sensor Network Protocols and Applications, May 2003

[13] M.Sichitiu, C.Veerarittiphan, " Simple, Accurate Time Synchronization for Wireless Sensor Networks"

[14] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts" in UCLA Technical Report 020008, Feb, 2002.

[15] Generiwal, S., Kumar, R. and Srivastada,M. "Timing Sync Protocol for Sensor Networks", The first ACM conference

[16] M.Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol", In the proc. Of Sensys'04, Baltimore, Maryland, 2004.

[17] W. Ye, J. Heidemann and D. Estrin. "An energy

efficient mac protocol for wireless sensor networks", In IEEE INFOCOM, 2002.

**신 문 선(Moon-Sun Shin)**

[정회원]



- 2004년 8월 : 충북대학교대학원 전자계산학과(이학박사)
- 2005년 8월 ~ 2008년 12월 : 건국대학교 컴퓨터시스템 강의교수
- 2010년 3월 ~ 현재 : 안양대학교 교양학부 조교수

<관심분야>

데이터베이스, 정보보안, USN, RFID 보안

**정 경 자(Kyeong-Ja Jeong)**

[정회원]



- 1996년 2월 : 충북대학교 대학원 전자계산학과(이학박사)
- 1995년 2월 ~ 현재 : 충청대학교 디지털마케팅학과 교수

<관심분야>

데이터베이스 질의 최적화 , 유비쿼터스 컴퓨팅, e-러닝, RFID 보안

**이 명 진(Myong-Jin Lee)**

[정회원]



- 1986년 2월 : 숭실대학교 대학원 전자계산학과(이학석사)
- 2005년 8월 : 충북대학교 대학원 전자계산학과(이학박사)
- 1989년 1월 ~ 2001년 12월 : 케이티인포텍 망관리사업팀 망관리사업부 근무
- 2002년 2월 ~ 현재 : 가림정보기술(주) 대표이사

<관심분야>

NMS, USN, U-헬스, USN/RFID 미들웨어