

# 센서 네트워크에서의 글로벌 상태 지원 기법의 설계 및 구현<sup>☆</sup>

## Design and Implementation of Global State Management for Sensor Networks

이 근 수\*  
Keunsoo Lee

김 준 영\*\*  
Junyeong Kim

조 기 호\*\*\*  
Kiho Cho

김 두 현\*\*\*\*  
Doohyun Kim

### 요 약

본 논문에서는 센서 네트워크에 글로벌 시간을 실현하고 이를 바탕으로 글로벌 상태를 지원하여 센서들이 효과적으로 협업할 수 있도록 방법에 관하여 논한다. 기존의 방법으로는 RBS, TPSN, FPSN 등이 있으나 시간 동기화만을 지원하고 글로벌 상태를 지원하지 않는다는 단점이 있었다. 본 논문에서는 이러한 단점을 극복하고자 글로벌 상태를 지원하는 SGS(MSimple Global State Management)을 고안하였다. 실험결과 SGS는 재동기화 주기가 1초이고 메시지 송신 주기가 1000ms 일 때 0.935%의 메시지 손실률로 100ms이내의 오차를 갖는 글로벌 상태를 실현 할 수 있다는 것을 실험을 통하여 확인할 수 있었다. 본 논문에서는 센서 네트워크의 글로벌 동기와 글로벌 상태에 대하여 정의하고 SGS 방법을 상세히 소개하며 재동기화 주기와 메시지 손실률 및 동기화 오차간의 상관관계에 관한 실험 결과를 상세히 설명한다.

### ABSTRACT

In this paper, we proposed a mechanism for effective cooperation in sensor network. There are a few mechanism like RBS, TPSN, FPSN for sensor network. However these are supporting synchronization among nodes without global state. Therefore, we proposed SGS(MSimple Global State Management) to maintain global state among sensor nodes. As experimental results, we confirmed loss rate is within 1% as maintaining global state with SGS mechanism. In this paper, we defined global state in sensor network and introduced SGS for improving timing accuracy in sensor environment.

☞ KeyWords : 시간 동기화, 글로벌 스테이트, Time Synchronization, Global State

## 1. 서 론

센서 네트워크의 핵심 기술 요소로는 하드웨어 및 초소형 운영체제 기술, 저전력 네트워크 프로토콜 기술, 시간 동기 및 위치인식 기술, 미들웨어 기술, 네트워크 보안 기술, 센서 네트워크 응용 기술 등으로 분류할 수 있다. 그리고 각각의 기술들이 발달하면서 다양한 분야로 연구 응용되기 시작했다.

일반적으로 센서 네트워크는 센서 노드와 함께 데이터를 수집하는 싱크 노드로 구성된다. 싱크 노드는 인터넷과 같은 일반 네트워크에 연결되어 수집된 데이터를 필요한 사용자나 어플리케이션에 제공한다. 센서 네트워크는 네트워크의 특성상 세

\* 정 회 원 : 건국대학교 대학원 컴퓨터정보통신공학과 박사과정 witholi@konkuk.ac.kr

\*\* 준 회 원 : 건국대학교 대학원 컴퓨터정보통신공학과 석사과정 plaa@konkuk.ac.kr

\*\*\* 정 회 원 : 건국대학교 대학원 컴퓨터정보통신공학과 석사과정 innocent@konkuk.ac.kr

\*\*\*\* 정 회 원 : 건국대학교 인터넷미디어공학과 교수 doohyun@konkuk.ac.kr(교신저자)

[2009/07/15 투고 - 2009/07/24 심사 - 2009/08/24 심사완료]

☆ 본 연구는 2009학년도 건국대학교의 지원과 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 (NIPA-2009-C1090-0902-0026)와 한국과학재단을 통해 교육과학기술부의 WCU 사업(WCU, World Class University)의 지원사업 (과제번호:R33-2008-000-10068-0)의 연구 결과로 수행되었음



한편, 노드  $n$ 이 시간  $t$ 에 행한 동작  $Act_n^t$ 에 대하여 프로그래머가 본래 동작해야 할 것으로 의도한 시간을  $T(Act_n^t)$ 라고 한다면 특정 시간 reference clock  $t$ 와 임의의 노드가 시간  $t$ 에 행한 동작의 시간 오차인 skew[1]  $\delta_i$ 은 다음과 같이 정의 된다.

$$\delta_i = t - T(Act_i^t)$$

그리고 이를 이용하여 특정 시간  $t$ 에서 센서 네트워크 전체 action의 시간 오차  $\delta_{avg}$ 는 다음과 같이 유도 된다.

$$\delta_{avg} = \frac{\sum_{i=A}^n |t - T(Act_n^t)|}{n}$$

본 논문에서는 센서 네트워크 토폴로지에서  $\delta_{avg}$ 가  $\delta_{avg} \leq \delta_{max}$ 의 범위 안에 있다는 조건에 부합이 된다면 이 센서 네트워크는 글로벌 동기화 상태, 혹은 간단히 글로벌 상태(Global State)에 있다고 정의한다. 본 논문의 각종 실험에서는  $\delta_{max}$ 를 100ms로 설정하여 진행하였다.

본 논문은 상기에서 정의한 시간 동기화와 글로벌 동기화 상태를 실현하는 방법을 제시하는데에 그 목적을 두고 있으며, 이를 위하여 본 논문에서는 SGS(Simple Global State Management) 프로토콜을 제안하고 이를 이용하여 시간 동기화와 글로벌 상태를 제어하는 알고리즘을 제안한다.

### 3. 관련 연구

#### 3.1 RBS

RBS[2]는 리시버-리시버의 동기화 구조를 가진 대표적인 방식이다. 송신 시간과 액세스 시간에

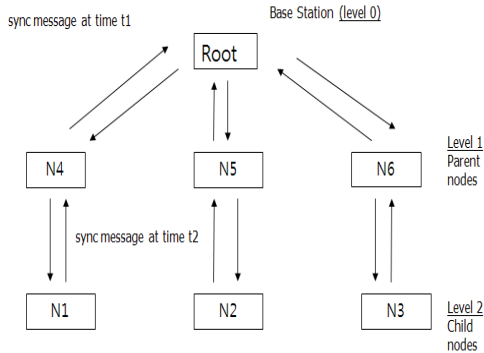
의한 오차를 줄이기 위하여 네트워크로 패킷을 전송하는 시점에서 타임 스탬프를 사용하여 시간 동기를 수행한다. 이 방법은 네트워크를 구성하는 노드 수가 많을수록 패킷 오버헤드가 급격하게 커지고 이웃 노드들의 정보를 가지고 있어야 하기 때문에 많은 메모리를 요구한다. 멀티 홉으로 이루어질 경우 Time-routing 방법을 이용하여 시간 동기를 연결 노드를 통해 변환시켜 맞추어야 한다. 또 RBS는 브로드 캐스트 메시지 교환이 많이 발생하기 때문에 많은 전력을 소모하게 된다.

(표 1) RBS 알고리즘

- |  |
|--|
| <ol style="list-style-type: none"> <li>① 타임 서버(Time Server)는 글로벌 시간을 포함하는 참조 메시지(Reference Message)를 브로드캐스트(broadcast)한다.</li> <li>② 일반 노드들은 참조 메시지를 받는 즉시 자신의 로컬 시간을 기록한 후 글로벌 시간과 기록한 로컬 시간의 오프셋(Offset)를 계산한다.</li> <li>③ 노드들은 계산한 오프셋을 서로 교환하여 각 노드들에 대한 오프셋 테이블을 작성한다.</li> </ol> |
|--|

#### 3.2 TPSN

TPSN[3]은 sender-receiver의 동기화 구조를 가진 대표적인 시간 동기화 알고리즘 방식이다. 하나의 sender가 다음 하나의 receive에 동기를 전달해 주는 전통적인 계층적 방식의 2-way 동기(이 기법은 계층 구조를 생성하기 위해 센서 노드들의 레벨을 결정하는 레벨 탐색 단계와 SNTP(Simple Network Time Protocol)[4]와 유사한 방법으로 시간을 동기화 하는 기간 동기화 단계) 전달 방법을 쓰고 있다. RBS 기법에 비해 이 방식은 속도 측면에서 두 배의 성능 향상을 보인다. 하지만 노드들의 추가 혹은 제거에 따른 네트워크 구성 변화에 동적으로 대응할 수 없는 단점을 갖는다.



(그림 2) TPSN 동기화 과정

(표 2) TPSN 알고리즘

- ① 레벨 0 에 해당하는 루트 노드를 중심으로 네트워크의 각 노드의 계층적 트리 구조를 형성한다.
- ② 하위 레벨의 노드가 상위 레벨의 노드에 시간정보를 전달하여 시간 동기화를 수행한다.
- ③ 모든 노드는 루트 노드에 동기화되며 이로서 네트워크의 전체적인 시간 동기화가 이루어지게 된다.

### 3.3 FTSP

FTSP[5]는 sender-receiver의 동기화 구조를 가지고 있으며 RBS와 유사하지만 더 빠르고 멀티 홉 트리 구조의 지원이 가능하며, TPSN의 단점인 동적 대응문제를 해결한 방법이다. 전역 참조 시간을 브로드캐스트하고 이를 수신하는 노드들은 중복해서 브로드캐스트 메시지를 받음으로써 clock 오차를 선형적인 연산을 통해 줄임으로써 root 노드와 동기화가 이루어지게 된다. TinyOS[6]에 이미 포함되어 있는 이 기법은 주기적인 메시지의 플러딩(flooding)을 이용하여 노드의 결함 및 통신 두절로 네트워크 변화에 동적으로 대응 할 수 있도록 지원한다. 하지만 한 패킷에 여러 개의 타임스탬프를 사용하여 비효율적인 면이 있다.

(표 3) FTSP 알고리즘

- ① 루트 노드 선택 한다. 루트 노드는 글로벌 시간 정보를 규칙적으로 브로드캐스트(broadcast)한다.
- ② 약속된 일정 주기로 시간 정보를 받은 각 노드들은 자신의 시간 정보와 비교하여 시간오차를 보정한다.
- ③ 맥 계층(MAC-layer)에서 타임 스탬핑(Time-stamping)을 한 후에 메시지(Reference Message)를 브로드캐스트(broadcast)한다.

### 3.4 TMO

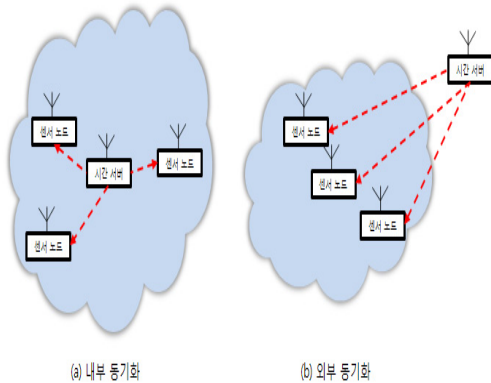
TMO(Time-Triggered Message-Triggered Object)[7]는 분산 실시간 처리를 위해 개발이 되었으며 객체의 상태나 속성 정보 저장을 위한 정보 저장소 ODS(Object Data Store)[7]가 있다. TMO 기술은 이질적인 분산 멀티미디어 환경에서 각각의 장비들에 멀티미디어 데이터를 투명성있게 전송한 후 고감도 고품질을 유지하면서 실시간 멀티미디어를 연출하기 위해 개발된 기술이다. TMO의 핵심 기술 중에 하나인 ODS는 SpM(Spontaneous Method)와 SvM(Service Method)[7]을 이용하여 접근을 할 수 있다.

## 4. SGSM

본 장에서는 센서 네트워크에서노드들을 동기화와 글로벌 상태를 제어하기 위한 방법으로 SGSM(Simple Global State Management) 프로토콜을 제안하고 이를 이용하여 시간 동기화와 글로벌 상태를 제어하는 알고리즘을 제안한다.

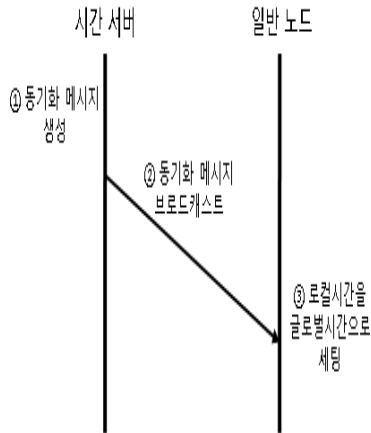
### 4.1 SGSM 프로토콜

각기 다른 로컬 시간을 갖는 노드들을 동기화하기 위해서는 기준이 되는 시간, 즉 글로벌 시간이 필요하다. 이 때 글로벌 시간을 제공하는 노드를 시간 서버라 하고, 시간 서버의 위치에 따라 그림 3과 같이 내부 동기화 또는 외부 동기화로 분류가 된다.



(그림 3) 시간 서버의 위치에 따른 동기화 분류

SGSM 프로토콜은 시간 서버가 센서 네트워크를 구성하는 노드들 중 하나인 내부 동기화 방법이다. SGSM 프로토콜은 노드들을 동기화 하기 위하여 그림 4와 같은 과정으로 진행이 된다.



(그림 4) 초기 시간 동기화 과정

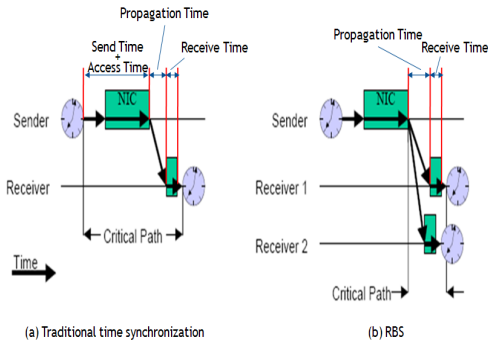
시간 서버는 글로벌 시간이 되는 자신의 로컬 시간을 타임 스탬프로 만들어 동기화 메시지를 생성한다. 그리고 동기화 메시지를 일반 노드에게 브로드캐스트하면 일반 노드는 동기화 메시지를 받는 즉시 타임 스탬프를 확인하여 자신의 로컬 시간을 세팅하면 시간 서버의 로컬 시간과 일반 노드의 로컬 시간은 같게 된다.

그러나 여기에 의문점이 있다. 일반 노드가 동기화 메시지를 수신하여 타임 스탬프를 확인하기 까지 메시지 전송 지연 시간이 발생하여 타임 스탬프의 시간과 글로벌 시간에 차이가 있을 수 있기 때문이다.

이 문제는 SGSM 프로토콜이 RBS의 특징을 사용함으로써 해결이 가능하다. 메시지를 전송하는 동안 발생할 수 있는 메시지 지연 요소는 표 4와 같이 크게 전송 시간, 접근 시간, 전달 시간, 수신 시간이 있다. SGSM 프로토콜은 그림 5와 같이 Critical Path에서 전송 시간과 접근 시간을 제거 하였으며 무선 통신에서 메시지의 전송 속도는 빛의 이동 속도와 비슷하다고 할 수 있기 때문에 전달 시간은 거의 0과 같다고 할 수 있다. 수신 시간의 경우 메시지를 받는 순간 인터럽트가 발생하여 전기적인 신호로 처리되기 때문에 지연 시간은 무시할 정도의 지연이 발생되므로 0으로 가정을 할 수 있다.

(표 4) 메시지 전송 지연 발생 요인

요인	설명
전송 시간 (Send Time)	송신자가 메시지를 생성하는 동안 소비되는 시간
접근 시간 (Access Time)	송신자가 메시지를 전송하기 위해서 통신 채널에 접속할 수 있는 권한을 획득하기 위해 기다리는 시간
전달 시간 (Propagation Time)	메시지가 송신자를 떠나 통신 채널을 통하여 목적지인 수신자에게 도착할 때까지의 시간
수신 시간 (Receive Time)	통신 채널을 통하여 수신자의 네트워크 인터페이스에 수신된 메시지를 호스트가 알아차리고 처리할 때까지의 시간



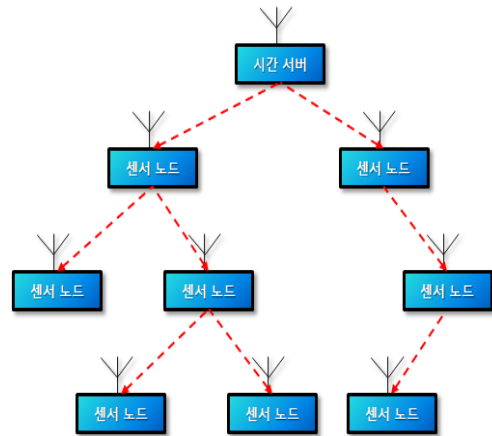
(그림 5) 전통적인 시간 동기화와 RBS와의 비교(2)

## 4.2 재동기화

실제 세계의 시계들은 표준 시간과 같게 설정하여도 어느 정도의 시간이 흐르면 모든 각각의 다른 시간을 갖는다. 이는 시계의 시간을 측정하는 무브먼트의 종류가 같다고 할지라도 각각의 특성과 환경적인 영향으로 1초를 측정하는데 있어 각기 다른 오차를 갖고 있기 때문이다. 이는 센서 노드들도 마찬가지다. 모든 노드들이 동기화가 되어 같은 시계의 무브먼트와 같이 노드의 로컬 클락이 1초를 측정하는 것이 서로 다르기 때문에 시간이 흐르다 보면 동기화는 깨지게 된다. 로컬 클락이 정확하지 않은 이유는 하드웨어적인 특성도 있지만, 중력과 자력등의 환경적인 요인에 의해서 많은 영향을 받기 때문이다. 그리고 무엇보다도 실제 인터럽트를 처리하는 과정에서 부정확한 지연이 발생하는 등 커널에서의 원인이 가장 크다.

이를 방지하기 위한 작업이 재동기화 과정이다. 노드들을 동기화 한 후 보장하고 싶은 오차가 발생하기 전에 다시 동기화를 하는 과정, 즉 재동기화를 거치면 센서 네트워크는 보장된 오차 범위에서 동기화 할 수 있다.

## 4.3 멀티 홉과 SGSM 프로토콜

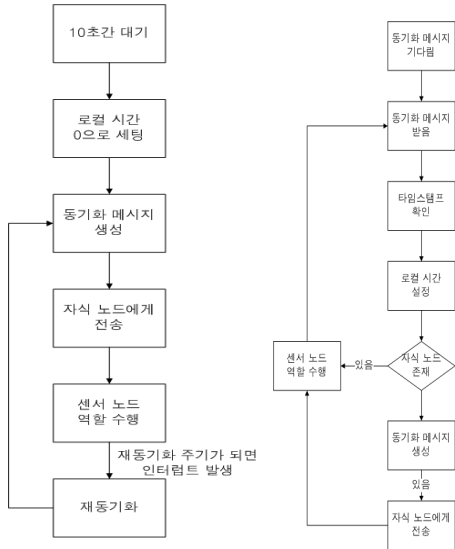


(그림 6) 트리형태의 센서 네트워크

SGSM 프로토콜은 RBS 프로토콜과 마찬가지로 동기화 메시지를 브로드캐스트 할 때 지연시간을 0으로 간주 할 수 있을 만큼 작다는 아이디어에서 시작하였다. 그러나 센서 네트워크는 그림 6과 같이 멀티 홉 형태의 네트워크로 구성이 된다. 이런 경우 0만큼 작은 지연 시간이라 할지라도 홉의 단계가 커지면 커질수록 지연 시간은 증가하여 결국 동기화 할 수 없게 된다.

SGSM 프로토콜은 이러한 문제를 해결하기 위하여 트리형태로 구성이 된 센서 네트워크에서 다음과 같은 방법으로 멀티 홉의 시간 동기화를 지원을 한다. 싱글 홉과 마찬가지로 시간 서버의 로컬 시간을 글로벌 시간으로 생각을 하고 자신의 로컬 시간을 이용한 동기화 메시지를 생성하여 자식 노드들에게 전송한다. 이후 동기화 메시지를 받은 자식 노드들은 자신의 로컬 시간을 글로벌 시간으로 설정하고 자식 노드가 있는 노드의 경우 자신의 자식노드에 대한 시간 서버 역할을 수행 할 수 있다. 이러한 방법으로 트리 형태로 구성이 된 센서 네트워크 속에 있는 모든 노드를 동기화 시킬 수 있다.

멀티 홉에서 SGSM 프로토콜의 동기화 알고리즘을 정리하면 그림 7과 같다.



(a) 시간 서버 흐름도 (b) 일반 노드 흐름도  
(그림 7) SGSM 프로토콜의 동기화 과정 흐름도

#### 4.4 오피셜 릴리즈 타임(ORT)

ORT란 Official Release Time의 약자로 글로벌 스테이트(Global State)의 데이터를 업데이트 하는 경우 ORT를 설정하면 해당 시간 이후에 업데이트 되는 것을 말한다. ORT가 적용된 글로벌 스테이트 큐는 일반적인 글로벌 스테이트 큐와 비슷하다. 그러나 결정적인 차이는 타임스탬프가 아닌 ORT를 기준으로 데이터가 정리되어 있다는 것이다. 타임 스탬프를 사용하는 경우 현재 시간과 관계 없이 가장 늦은 타임 스탬프의 자료를 선택하면 되나, ORT는 현재 시간보다 늦은 미래의 시간을 갖을 수도 있기 때문에 현재 시간보다 미래의 시간을 갖는 데이터를 제외한 데이터들 중 선택을 해야한다.

### 5. SGSM 구현

본 절에서는 앞서 설명을 한 SGSM 프로토콜을 구현을 하고 SGSM 프로토콜의 APIs에 대하여 설명을 한다.

### 5.1 NanoQplus

NanoQplus[8]는 한국전자통신연구원(Electronics and Telecommunication Research Institute, ETRI)에서 순수 국내 기술로 개발한 센서 운영체제로, 선점형 스케줄러를 가진 멀티 스레드 기반의 모델이다.

NanoQplus는 트리 형태의 센서 네트워크 개발이 쉽고, 표준 C문법 지원, 공개 소스라는 특징을 가지고 있다. 이와 같은 NanoQplus의 특징 때문에 본 논문에서 제안한 SGSM 프로토콜을 NanoQplus에 모듈화 하여 직접 구현을 하였다.

### 5.2 SGSM

본 논문에서 제안을 한 SGSM 프로토콜을 구현하기 위하여 NanoQplus에서 제공하는 메시지 프레임의 수정과 동기화/재동기화, Global State Queue에 대하여 구현을 하였다.

#### 5.2.1 메시지 프레임

SGSM 프로토콜에서는 동기화, 재동기화, 일반 업데이트, ORT 메시지와 같은 다섯 종류의 메시지가 있다. 동기화 메시지와 재동기화 메시지는 글로벌 시간의 타임 스탬프가 필요하고 일반 메시지와 업데이트 메시지의 경우 메시지를 생성한 시점을 의미하는 타임 스탬프가 필요하다. 그리고 ORT 메시지는 데이터가 업데이트 되기를 희망하는 시간의 타임 스탬프가 필요하다. 이런 공통점을 반영하여 기존의 NanoQplus의 메시지 프레임 그림 8을 그림 9와 같이 수정을 하였다.

Frame Control Field	Data Sequence Number	Destination PAN ID	Destination Address	Source Address	Source ID	MAC layer payload	Frame Check Sequence
---------------------	----------------------	--------------------	---------------------	----------------	-----------	-------------------	----------------------

(그림 8) NanoQplus 메시지 프레임

Flag	Time stamp	Frame Control Field	Data Sequence Number	Destination PAN ID	Destination Address	Source Address	Source ID	MAC layer payload	Frame Check Sequence
------	------------	---------------------	----------------------	--------------------	---------------------	----------------	-----------	-------------------	----------------------

(그림 9) 수정된 NanoQplus 메시지 프레임

Flag 필드는 SGSM의 메시지 종류를 의미하고 타임 스탬프 필드는 메시지의 종류에 따라 표 5와 같은 의미의 시간을 나타낸다. Data 필드는 일반 메시지, 업데이트 메시지, ORT 메시지에서 전송될 데이터를 갖는다.

(표 5) Flag에 따른 메시지의 종류와 Timestamp의 의미

Flag	메시지 종류	Timestamp 의미
0	동기화/재동기화	글로벌 시간
1	일반 메시지	메시지가 생성된 시간
2	업데이트 메시지	메시지가 생성된 시간
3	ORT 메시지	ORT(미래의 시간)

### 5.2.2 동기화/재동기화

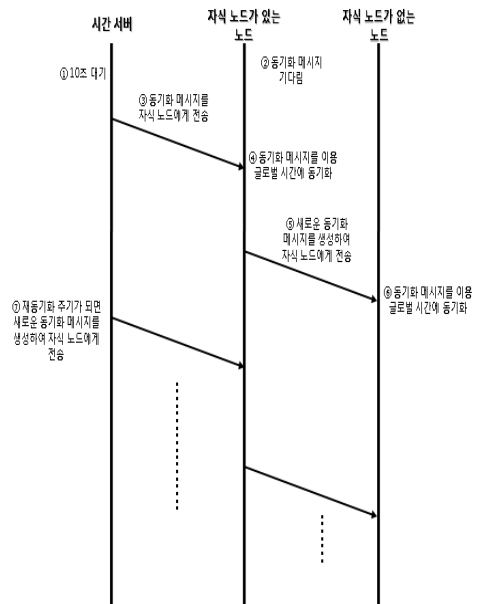
NanoQplus에서 로컬 시간은 최소 2초 단위로 갱신이 되도록 제한이 되어 있다. 이러한 제한적으로 로컬 시간을 관리하는 경우 각각 노드들의 로컬 시간의 오차를 1초보다 더 정확하게 측정이 불가능하기 때문에 센서 네트워크가 동기화된 상태를 분석하기가 힘들고, 더 정밀한 시간으로 센서를 동기화 하기가 어렵다. 이러한 문제점을 해결하기 위하여 센서 모드에서 로컬 시간을 100ms 단위로 갱신을 할 수 있도록 수정을 하였다.

동기화와 재동기화는 최초의 동기화인가 그 이후의 동기화인가에 따른 구분일 뿐 과정은 같기 때문에 같은 형태의 메시지를 사용한다. SGSM 프로토콜은 동기화 및 재동기화를 위하여 표 6과 같은 APIs를 제공한다.

(표 6) SGSM 동기화/재동기화 관련 APIs

함수 원형	기능
BOOL sgsm_sync_msg (BOOL init, SGSM_TX_INFO sgsm_data)	동기화/재동기화 메시지를 만들고 자식 노드에게 전송함
void sgsm_local_clock_setting (TimeType time)	로컬 시간 설정
void sgsm_resync_period (TimeType period)	재동기화 주기를 설정

SGSM 프로토콜에서 제공을 하는 APIs를 이용하여 그림 10과 같은 과정의 알고리즘을 이용하여 동기화/재동기화를 이루게 된다. 재동기화의 주기는 사용자가 SGSM 프로토콜에서 제공을 하는 재동기화 주기를 설정할 수 있는 API를 통하여 재동기화 주기를 설정할 수 있고 SGSM 프로토콜은 사용자가 설정을 한 재동기화 주기로 센서 네트워크의 센서 노드를 재동기화 하게 된다.



(그림 10) SGSM 프로토콜을 이용한 동기화/재동기화 과정



### 5.2.3 SGSM 글로벌 스테이트 큐

SGSM 글로벌 스테이트 큐(이하 글로벌 큐라 함)의 목적은 항상 최신의 데이터를 유지하다가 노드가 데이터를 요청하는 경우 최신의 데이터를 제공하는 것이다. 이를 위하여 모든 센서 노드들은 글로벌 큐를 갖고 있으며 항상 동일한 상태로 관리 된다. 업데이트 메시지와 ORT 메시지는 타임 스탬프 필드가 메시지를 생성한 시간 또는 메시지가 업데이트 되기를 희망하는 시간을 의미하는 차이만 있을 뿐 그 형태는 같기 때문에 동일한 방법으로 글로벌 큐에 의해 관리된다. 업데이트 메시지나 ORT 메시지를 받은 경우에는 글로벌 큐를 검사하여 빈 슬롯에 메시지를 저장한다. 빈 슬롯이 없는 경우에는 저장되어 있는 메시지들의 타임 스탬프를 확인하여 글로벌 시간 이전의 메시지 중에서 최신 데이터를 제외한 임의의 슬롯에 삽입을 한다. ORT 메시지들은 아직 업데이트 시간이 되지 않아 미래의 시간을 갖고 있을 수도 있기 때문에 최신의 데이터는 글로벌 시간과 비교하여 바로 이전의 시간을 갖는 데이터를 의미한다. 반대로 노드가 데이터를 요청하는 경우에는 글로벌 큐를 확인하여 타임 스탬프가 글로벌 시간 바로 이전의 시간을 갖는 메시지의 데이터를 제공한다. 이유는 삽입을 하는 경우 마찬가지로 ORT 메시지가기 때문이다. ORT 메시지는 특정한 시간 이후로 업데이트 되기 희망하는 데이터이기 때문에 타임 스탬프가 글로벌 시간과 비교하여 미래의 시간인 경우에는 아직 업데이트 되어서는 안 되기 때문이다. SGSM에서는 글로벌 스테이트 큐와 관련하여 표 7과 같은 APIs를 제공한다.

(표 7) SGSM 글로벌 스테이트 큐 관련 APIs

함수 원형	기능
SGSM_QUEUE* queue_init()	글로벌 큐의 초기화
UINT8 search_SGSM_queue (SGSM_DATA* sgsm_data)	글로벌 큐에서 삽입 가능한 슬롯을 검색
void put_sgsm_queue (SGSM_DATA* sgsm_data)	글로벌 큐 업데이트
SGSM_DATA* get_sgsm_queue()	글로벌 큐에서 최신 데이터를 획득

### 5.2.4 업데이트 메시지와 ORT 메시지

앞 절에서 언급을 한 업데이트 메시지와 ORT 메시지의 기본 형태는 같다. 그 이유는 메시지 프레임의 수정을 통하여 메시지의 종류와 타임 스탬프를 넣었기 때문에 사용자가 응용프로그램에서 각 기능을 사용하기 위한 API만 다를 뿐 기본적인 구성 요소는 같다. SGSM 프로토콜에서는 업데이트 메시지와 ORT 메시지를 사용하기 위하여 표 8과 같은 APIs를 제공한다.

(표 8) SGSM 업데이트 메시지와 ORT 메시지 관련 APIs

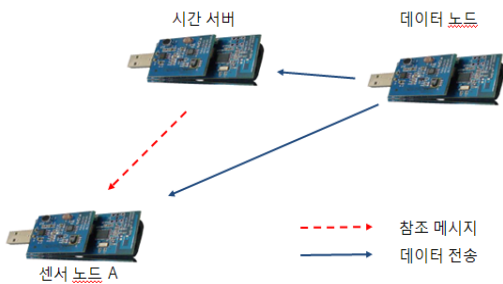
함수 원형	기능
void sgsm_update_msg (SGSM_TX_INFO* sgsm_data)	업데이트 메시지 생성 및 전송
void sgsm_ort_msg (SGSM_TX_INFO* sgsm_data)	ORT 메시지 생성 및 전송

## 6. 실험 및 평가

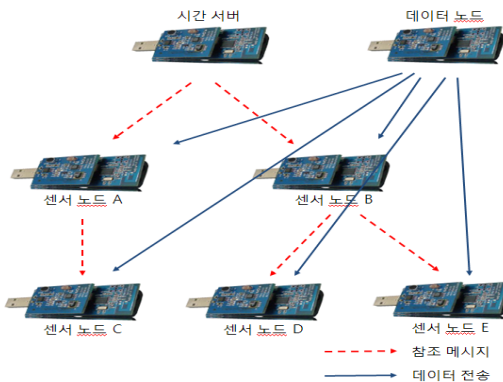
본 절에서는 그림 10과 같이 시간 서버 T와 T 노드를 부모노드로 하는 하나의 자식 노드 S로 구성된 센서 네트워크를 구성한 후 동기화 이후 시간의 흐름에 따른 글로벌 시간과 로컬 시간의 차이가 어떻게 변화하는지 실험하였다. 두 개의 노드가 동일한 로컬 클록을 갖는다면 시간의 차이는 0ms이거나 항상 일정한 차이를 보여야 한다. 그러나 센서 모드는 각기 다른 환경적인 요인이 각각 모드의 성능에 많은 영향을 미치기 때문에 T와 S사이의 시간적 오차는 증가할 것이다. 본 절에서는 이러한 문제점을 실험을 통하여 증명을 하고 본 논문에서 제시한 SGSM 프로토콜을 이용하였을 경우에 이러한 문제점에 대한 해결책이 됨을 증명을 하고 이를 바탕으로 본 논문에서 제시한 글로벌 상태의 성능에 대하여 증명을 하고자 한다.

### 6.1 실험 구성

싱글 홉 환경과 멀티 홉 환경의 센서 네트워크를 정하여 글로벌 시간과 각 노드들의 로컬 시간 차이를 확인하여 동기화 상태를 점검 하였다. 실험에 사용할 모드는 NanoQplus에서 공식적으로 지원하는 Hmote2420[9]를 사용하고, 실험 방법은 시간 서버의 시간을 글로벌 시간으로 설정을 하고 센서 네트워크에 있는 노드를 동기화 시킨다. 그 후 데이터 노드에서 임의의 정수를 브로드캐스팅하게 되고 이 데이터를 시간 서버와 다른 노드들이 이 메시지를 받는 즉시 각 노드의 로컬 시간을 시리얼 통신을 통하여 PC에 기록을 한다. 싱글 홉과 멀티 홉 환경의 센서 네트워크의 구성은 각각 그림 11과 그림 12와 같다.



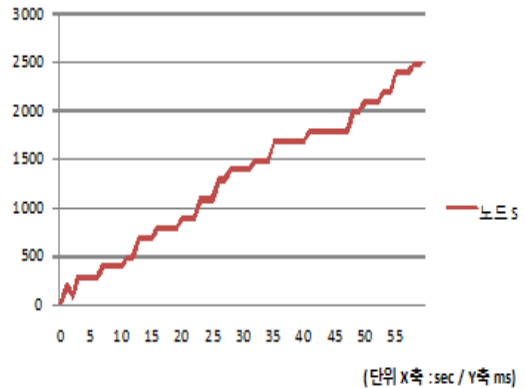
(그림 11) 싱글 홉 센서 네트워크 구성



(그림 12) 멀티 홉 센서 네트워크 구성

### 6.2 재동기화가 없는 센서 네트워크의 상태 분석

싱글 홉에서 실험 방법은 모든 노드가 부팅되면 최초의 동기화를 수행한 후 데이터 노드는 1초마다 임의의 정수가 포함된 메시지를 A와 시간 서버 노드에 브로드캐스팅을 하게 된다. 데이터 노드로부터 수신한 데이터를 수신한 시간을 시간 서버와 A 노드에서는 자신의 로컬 클럭을 PC에 시리얼 통신을 통하여 기록을 하게 된다. 기록된 실험 결과를 정리하면 그림 13과 같은 그래프를 얻게 된다. 시간 서버와 A 노드의 로컬의 시간의 차이는 시간이 지날수록 증가하는 것을 확인 할 수 있다.



(그림 13) 재동기화가 없는 경우의 상태 변화

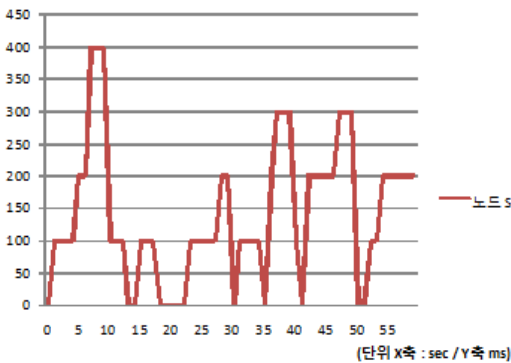
싱글 홉에서 그림 13과 같은 결과를 얻었으므로 자식 노드에 대하여 동기화 과정을 한번 더 수행을 하는 멀티 홉 환경에서도 싱글 홉 환경에서와 비슷한 시간이 지날수록 증가할 것이라고 유추를 할 수 있다.

### 6.3 재동기화가 있는 센서 네트워크의 상태 분석

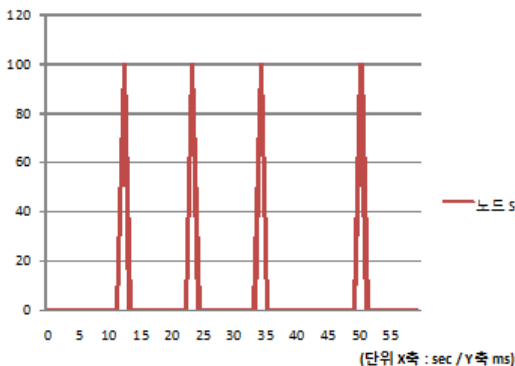
재동기화가 없는 센서 네트워크의 상태 분석을 통하여 노드들은 동기화 한 후 시간이 지날수록 글로벌 시간과 로컬 시간의 오차가 증가하여 센서 네트워크의 동기화가 깨지는 것을 확인하였다. 이러한 문제를 예방하기 위한 방법이 재동기화이다. 본 절에서는 재동기화 주기에 따른 동기화 상

태 변화를 체크하기 위한 실험을 진행 하였다. 이를 위하여 싱글 홉 센서 네트워크에서 재동기화 주기를 각각 10초와 1초로 설정한 경우 노드들의 동기화 상태를 체크 하였다. 10초와 1초의 경우 센서 네트워크 상태 변화는 그림 14와 그림 15와 같다.

재동기화 주기가 10인 경우에는 최대 400ms의 오차가 발생하게 되고 평균 0 - 200ms 정도의 오차가 발생하게 된다. 재동기화 주기가 1초인 경우에는 동기화 상태가 잘 유지되고 가끔 100ms의 오차가 발생하는 것을 확인 할 수 있었다. 재동기화를 하지 않았을 경우와 비교하여 재동기화 주기를 1초로 하였을 경우 비교적 정밀한 상태의 동기화를 유지하는 것을 확인 할 수 있다.



(그림 14) 재동기화 주기가 10초인 경우



(그림 15) 재동기화 주기가 1초인 경우

#### 6.4 재동기화가 있는 센서 네트워크에서의 글로벌 상태 분석

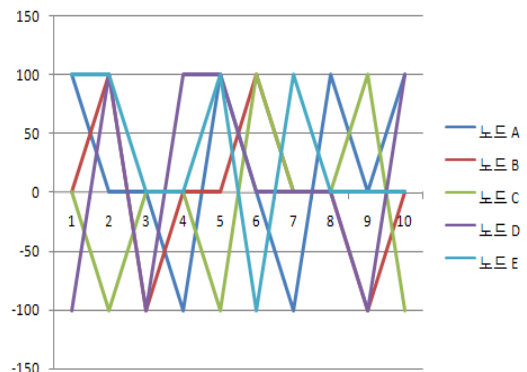
글로벌 상태 분석을 위한 실험 방법은 다음과

같다. 그림 11, 그림 12와 같은 싱글 홉과 멀티 홉 센서 네트워크에서 데이터 노드는 ORT 값을 생성하여 시간 서버와 자기 자신을 제외한 센서 네트워크에 속하여 있는 모든 노드들에게 데이터를 전송을 하고 이러한 데이터를 받은 센서 노드는 데이터를 글로벌 스테이트 큐에 저장을 하고 ORT 값에 따라 정해진 시간에 따라 이를 실행을 하고 이 때 모든 센서 노드들의 시간을 측정하여 시간 서버와의 시간 오차를 측정하는 방법을 사용하였다.



(그림 16) 싱글 홉에서 글로벌 상태 오차 측정

그림 16는 그림11과 같은 센서 네트워크에서의 글로벌 상태 오차 측정을 한 결과 값이다. 본 논문에서 제시를 한 글로벌 상태에서의 오차 범위에 속하기 때문에 싱글 홉에서는 글로벌 상태가 잘 유지 된다고 볼 수 있다.



(그림 17) 멀티 홉에서 글로벌 상태 오차 측정

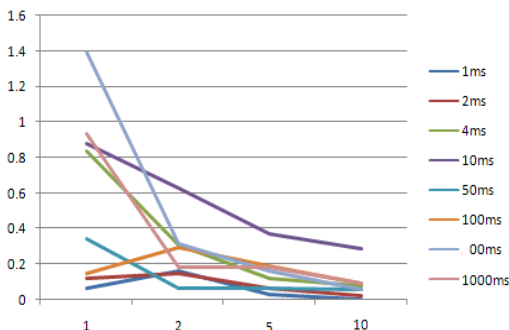
그림 17은 그림 12와 같은 멀티 홉 환경에서 글로벌 상태 오차 측정을 한 결과 값이다. 1 ~ 10초까지의 글로벌 상태의 오차의 결과값이 본 논문 2.2절에서 정의를 한 글로벌 상태의 오차 범위  $\delta_{avg}$  에 잘 부합이 됨으로 글로벌 상태가 잘 유지된다고 할 수 있다.

### 6.5 재동기화가 있는 센서 네트워크의 분석

본 절에서는 재동기화 주기와 센서 네트워크의 부하에 대한 상관 관계를 증명하기 위하여 다음과 같은 실험을 하였다. 그림 12와 같이 멀티 홉 환경에서의 재동기화 주기를 1, 2, 5, 10초 단위로 시간 서버에서 재동기화 메시지를 보내고, 사용자 메시지를 1, 2, 5, 10, 50, 100, 500, 1000ms 주기로 데이터 노드에서 메시지를 20,000번을 보내어 센서 노드 E에서 수신을 하지 못한 메시지의 개수의 비율을 손실률이라고 정의를 하고 실험을 진행하여 손실률을 측정하였다.

(표 9) 동기화 주기와 메시지 손실률 관계 표

재동기화 주기 메시지 주기	1sec	2sec	5sec	10sec
1ms	0.065	0.16	0.025	0
2ms	0.12	0.15	0.07	0.025
5ms	0.835	0.305	0.12	0.075
10ms	0.875	0.625	0.37	0.29
50ms	0.345	0.065	0.06	0.055
100ms	0.145	0.295	0.19	0.09
500ms	1.39	0.315	0.16	0.055
1000ms	0.935	0.185	0.185	0.095



(그림 18) 멀티 홉에서 네트워크 손실률 측정

위의 실험을 바탕으로 표 9와 그림 18과 같은 결과를 도출하였다. 이 자료를 분석을 하여 보면 재동기화 주기와 네트워크의 손실률간의 도출하여 낼 수가 있다. 메시지 주기가 같을 경우 재동기화 주기가 길어질수록 메시지의 손실률 부분이 작아지는 경향을 보이는 것으로 나타났다. 이는 재동기화 주기가 길어질수록 센서 네트워크에 걸리는 부하가 작아지고, 반대로 재동기화 주기가 짧아질수록 센서 네트워크에 걸리는 부하가 커지는 것을 의미하며, 이는 재동기화 주기와 센서 네트워크 부하가 서로 반비례 관계를 갖는다는 일반적인 현상이 본 방법에도 존재함으로 의미한다. 따라서 글로벌 상태라는 기능을 활용함에 있어서 야기되는 네트워크 부하가 목표 응용 시스템이 요구하는 범주를 초과하지 않도록 재동기화 주기를 선택하여 사용하여야 할 것이다.

## 7. 결론

본 논문에서는 센서네트워크의 글로벌 동기화 글로벌 상태에 대하여 정의하고 이를 바탕으로 글로벌 상태를 지원하는 SGSM 방법을 설명하였다. 아울러 재동기화 주기와 메시지 손실률 및 동기화 오차간의 상관관계에 관한 실험 결과를 설명하였다. 실험결과 본 논문에서 고안한 방법은 재동기화 주기가 1초이고 메시지 송신 주기가 1000ms 일 때 0.935%의 메시지 손실률로 100ms이 내의 오차 갖는 글로벌 상태를 실현 할 수 있다는 것을 실험을 통하여 확인할 수 있었다.

실제로 실험에 사용한 NanoQplus에서의 로컬 시간은 일정 시간 간격으로 tick이 발생하고 일정 수의 tick이 발생 하였을 경우 100ms의 시간이 지났다고 판단하여 로컬 시간을 갱신을 한다. 하지만 이러한 로컬 시간의 갱신 방식은 정확한 100ms의 측정이 불가능하다는 단점이 있고 환경적인 이유로 인하여 실제 하드웨어 tick이 불규칙하다는 문제점도 있다. 이러한 문제점을 해결하기 위하여 재동기화 주기를 짧게 할 경우 빈번한 재

동기화로 인하여 제한적인 전력을 갖는 센서 노드들의 효율성이 감소하는 문제점이 있고 센서 네트워크에 부하가 증가하는 측면이 있지만, 센서 네트워크를 구성하는 모든 노드들이 약 100ms 이내의 오차를 갖는 글로벌 시간을 갖을 수 있도록 동기화 할 수 있다는 점에서 의의를 찾을 수 있을 것이다.

### 참 고 문 헌

- [1] Hermann Kopetz, "REAL-TIME SYSTEMS: Design Principles for Distributed Embedded Applications", Kluwer Academic Publishers.
- [2] Jeremy Elson, Lewis Gridod and Deborah Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts", ACM SIGOPS Operating System Review, Vol. 36, pp. 147 - 163, December 2002.
- [3] Saurabh Ganeriwal, Ram Kumar, Mani B. Srivastava, "Timing-sync protocol for sensor networks" ACM SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems, November 2003.
- [4] D. Mills, "Simple Network Time Protocol(SNTP)", RFC Editor Simple Network Time Protocol(SNTP) Version 4 for IPv4, IPv6 and OSI, October 1996.
- [5] Miklós Maróti, Branislav Kusy, Gyula Simon, Ákos Lédeczi, "The flooding time synchronization protocol", ACM SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems, November 2004.
- [6] <http://www.tinyos.net>
- [7] Kane Kim, Chansik Im and Prasad Atheya, "Realization of a Distributed OS Component for Internal Clock Synchronization in a LAN Environment", Proc. ISORC 2002(IEEE CS 5th International Symposium on Object-Oriented Real-Time Distributed Computing), April 2002.
- [8] <http://www.qplus.or.kr>
- [9] <http://www.hybus.co.kr>

● 저 자 소개 ●



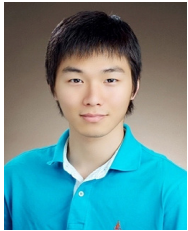
**이 근 수**

2000년 건국대학교 인터넷미디어공학부(공학사)  
2009년 건국대학교 대학원 컴퓨터정보통신공학과(공학석사)  
2009년~현재 건국대학교 대학원 컴퓨터정보통신공학과 박사과정  
관심분야 : 임베디드 소프트웨어, Time Synchronization, OFP, etc.  
E-mail : withholi@konkuk.ac.kr



**김 준 영**

2009년 건국대학교 인터넷미디어공학부(공학사)  
2009년~현재 건국대학교 대학원 컴퓨터정보통신공학과 석사과정  
관심분야 : 임베디드 소프트웨어, Time Synchronization, Fault Tolerance, etc.  
E-mail : plaa@konkuk.ac.kr



**조 기 호**

2009년 건국대학교 인터넷미디어공학부(공학사)  
2009년~현재 건국대학교 대학원 컴퓨터정보통신공학과 석사과정  
관심분야 : 임베디드 소프트웨어, Time Synchronization, Video Conference, etc.  
E-mail : innocent@konkuk.ac.kr



**김 두 현**

1985년 서울대학교 컴퓨터공학과(공학사)  
1987년 한국과학기술원 전산학과(공학석사)  
2003년 한국과학기술원 전산학전공(공학박사)  
2004~현재 건국대학교 인터넷미디어공학과 교수  
관심분야 : 임베디드 소프트웨어, 실시간분산컴퓨팅, 공개소스엔지니어링 etc.  
E-mail : doohyun@konkuk.ac.kr