

# 유비쿼터스 센서 네트워크에서의 데이터 전송시간 분석 시스템의 구현 사례

## A System for Analyzing Data Transmission Time in Ubiquitous Sensor Network

정기원(Chong Ki Won)\*, 김재철(Kim Jae Cheol)\*\*, 김주일(Kim Ju Il)\*\*\*,  
이우진(Lee Woo Jin)\*\*\*\*

### 초 록

센서 네트워크에서는 데이터의 실시간 처리가 중요한 요소 중의 하나이다. 각 노드들이 감지한 데이터를 정해진 시간 내에 전달해야 필요한 시기에 적합한 처리가 가능하다. 따라서 노드들이 데이터를 상해된 시간 내에 제대로 전달하고 있는가를 점검하는 것은 매우 중요하다. 이에 따라 본 논문에서는 데이터 전송시간에 대한 모니터링을 통하여 센서 네트워크에 존재하는 노드들이 허용시간 범위 내에서 서버로 데이터를 전송하고 있는가를 확인하기 위한 데이터 전송시간 분석 시스템의 구현 사례를 제안한다. 이를 위하여 데이터 전송시간 분석을 위한 절차를 제시하고, 제시한 절차에 따라 전송시간을 분석하기 위해 필요한 시간차 분석 방법, 데이터 송수신 시간 수집 방법 및 데이터 전송시간 계산 방법을 제시한다. 또한 제시한 방법을 바탕으로 데이터 전송시간을 모니터링하고 분석하기 위한 시스템을 구현하고, 사례 연구를 수행한 결과를 보인다.

### ABSTRACT

In a ubiquitous sensor network (USN) with several nodes, real time data processing is one of important factors. In order to process data appropriately, all the nodes should transmit sensor data in time and the transmission between nodes and their server should be managed very systematically. For the purpose of systematic management of transmission in a USN, this paper proposes a system for analyzing transmission time of sensor data. To implement the proposed system, an analyzing process of data transmission time, an analyzing method of clock drift, a collecting method of data send/receive times, and calculating formulas of data transmission duration are proposed. According to the proposed process and methods, this paper presents a system for monitoring and analyzing data transmission duration, and it also shows the results of a sample case.

키워드 : 유비쿼터스 센서 네트워크, 데이터 전송시간, 실시간 모니터링  
Ubiquitous Sensor Network, Data Transmission Time, Real time Monitoring

---

본 연구는 송실대학교 교내연구비 지원으로 이루어졌음.

- \* 송실대학교 컴퓨터학부 교수
- \*\* 송실대학교 대학원 컴퓨터학과 박사과정
- \*\*\* 송실대학교 대학원 컴퓨터학과 석사과정
- \*\*\*\* 송실대학교 컴퓨터학부 시간강사

## 1. 서 론

무선통신과 전자기술의 발달로 인해 소형이며, 단거리 통신이 가능한 저비용, 저전력, 다기능의 센서 노드가 개발되어왔다. 이러한 초소형의 센서 노드는 감지, 데이터처리, 통신 컴포넌트로 구성되어, 노드들 간의 네트워크를 형성할 수 있다[1]. 유비쿼터스 센서 네트워크[2]는 컴퓨팅 구현을 위한 기반 네트워크로 초장량, 저전력의 많은 센서들로 구성된 무선 네트워크이다. 유비쿼터스 센서 네트워크에서는 센서 노드들이 도처에 편재하여 센싱을 통해 사용자는 언제 어디서나 원하는 서비스를 제공 받을 수 있게 된다. 이러한 센서 네트워크는 국방 분야, 환경 감시 분야, 의료 분야 등 다양한 곳에서 필요한 실시간 응용 시스템을 개발하는데 사용되고 있다. 실시간 시스템에서 가장 중요한 것은 데이터의 실시간 수집 및 처리로, 센서 네트워크를 이용한 응용 시스템에서 데이터의 실시간 수집 및 처리가 올바르게 이루어지기 위해서는 센서 네트워크 상에서 데이터 전송이 허용시간 범위 내에서 이루어져야 한다. 특히 산불 및 가스 모니터링 등과 같이 사람의 생명과 직접 관련되는 기능을 수행하기 위한 시스템에서는 데이터 전송시간이 0에 가까울수록 큰 사고를 미리 방지할 수 있기 때문에 센서 네트워크 상에서 데이터의 실시간 처리가 매우 중요하다. 센서 네트워크는 수많은 센서들로 구성되기 때문에 어느 구간에서 데이터 전송이 지연되는가를 알아내기가 쉽지 않다. 따라서 서버로의 데이터 전송시간이 요구한 시간 내에 전송되는가를 알아내는 것 뿐만이 아니라 어느 구간에서 문제가 발생하

여 지연되고 있는가를 알아내는 것 또한 필요하다.

지금까지 실시간 시스템에서 데이터의 수집 및 처리가 허용시간 내에서 이루어지고 있는가를 모니터링하고 분석하기 위한 여러 기법들[3~6]이 연구되어왔다. 대부분의 실시간 시스템은 이벤트가 발생하면, 발생한 이벤트에 대응하여 행동을 취하도록 설계되어 있다. 따라서 실시간 시스템에서의 데이터 수집 및 처리에 대한 모니터링을 수행하는 기법들[3~6]은 이벤트를 감지하고, 이를 처리하는 과정이 허용시간 내에 이루어지고 있는가를 모니터링하고 분석한다. 그러나 센서 네트워크에서는 이벤트가 발생할 경우에 데이터를 수집하거나 처리하는 것이 아니라, 주기적으로 센서 노드가 주변의 데이터를 감지하고 이를 서버로 전송하게 된다. 따라서 기존에 실시간 시스템에서 사용하는 방법으로 센서 네트워크에서의 데이터 수집, 전송 및 처리 시간에 대한 모니터링을 수행하는 것은 어려운 일이다.

이에 따라 본 논문에서는 센서 네트워크를 이용한 응용 시스템에서 허용시간 내에 실시간으로 데이터가 처리되고 있는가를 모니터링하고 분석하기 위한 모니터링 시스템의 구축 사례를 보인다. 센서 네트워크에서의 데이터 전송시간에 대한 모니터링을 통하여 센서 네트워크에 존재하는 노드들이 허용시간 범위 내에서 서버로 데이터를 전송하고 있는가를 확인하고, 데이터 전송이 지연되는 구간을 발견할 수 있는 데이터 전송시간 분석 방법 및 시스템을 제시한다.

현재 센서 네트워크 분야에서는 전력과 시간 동기화 문제를 주된 이슈로 다루고 있다

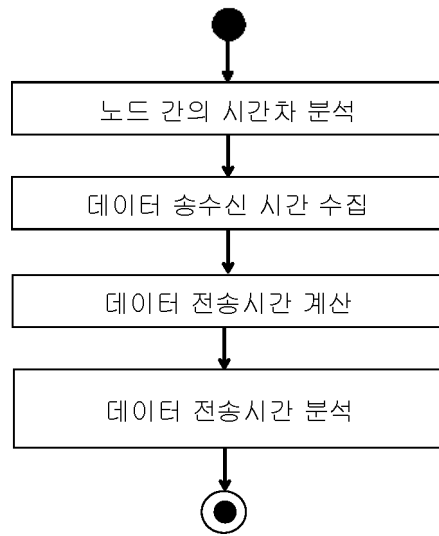
[7]. 센서 노드들은 제한된 전력을 이용하여 데이터를 수집, 처리, 전송하므로, 센서 네트워크에서 데이터의 실시간 처리가 지속적으로 올바르게 수행되기 위해서는 전력의 소비를 최소화하기 위한 방안이 연구되어야 하며, 데이터 전송시간에 대한 모니터링을 위해서는 시간 동기화가 이루어져야 한다. 따라서 지금까지는 센서 네트워크에서의 실시간 처리를 위한 진행 연구가 주로 이루어지고 있는 것이다. 본 논문에서는 이러한 선행 연구들을 바탕으로 하이 센서 네트워크를 이용한 응용 시스템에서 가장 중요한 사항 중의 하나인 데이터의 실시간 처리에 대한 연구를 수행하고자 센서 네트워크에서 데이터 전송시간을 분석하기 위한 방법과 이를 구현한 시스템을 제시하고자 한다.

제 2장에서는 데이터 전송시간에 분석을 위한 절차를 설명하고, 제 3장에서는 각 절차에 따른 방법을 제시한다. 전송시간 분석을 위해 필요한 노드 간의 시간차 계산 방법, 데이터 송수신 시간 수집 방법, 데이터 전송시간 계산을 위한 공식을 제시한다. 제 4장에서는 논문에서 제시하는 방법을 지원하는 시스템을 구현한 사례를 보이고, 제 5장에서는 본 논문에 대한 요약과 앞으로 해결해야 하는 과제에 대하여 설명한다.

## 2. 데이터 전송시간 분석 프로세스

센서 네트워크에는 수많은 노드들이 존재하므로, 서너로 데이터가 전달되기까지는 많

은 노드들이 서로 데이터를 주고받아야 한다. 그러므로 전송시간을 측정하기 위해서는 하나의 데이터에 대하여 각 노드에서의 송수신 시간을 수집해야 한다. 데이터를 수집한 후에는, 하나의 데이터에 대하여 전체 전송시간을 계산한다. 데이터를 감지한 센서 노드로부터 시피에 오기까지 각 노드에서의 송수신 시간을 이용하여 전체 전송시간을 계산한다. 마지막으로, 측정된 전송시간과 기대한 전송시간을 비교하여 허용시간을 준수하고 있는가를 확인한다. <그림 1>은 데이터 전송시간 분석 프로세스를 보여준다.



<그림 1> 데이터 전송시간 분석 프로세스

노드 간의 시간차 분석 단계에서는 센서 네트워크에 존재하는 각 노드 간의 시간차를 분석한다. 센서 네트워크를 구성하는 노드들은 한정된 자원을 사용하여 오퍼레이션을 수행하므로 자원을 절약하기 위하여 활동하지 않는 동안에는 휴면 상태에 있게 되고, 필요

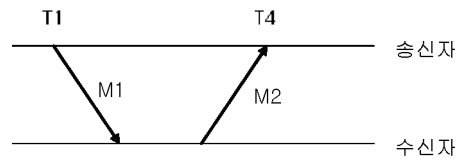
한 경우에 깨어나게 된다. 휴면 상태에 있는 경우에는 각 노드의 타이머가 정지되었다가 깨어날 경우에 다시 동작이 되기 때문에 각 노드 간에는 시간의 편차가 생길 수 밖에 없다. 따라서 데이터 전송시간을 분석하기 위해서는 가장 먼저 노드 간의 시간 편차를 구해야 한다. 노드 간의 시간 편차는 싱크 노드를 기준으로 하여 분석한다. 데이터 송수신 시간 수집 단계에서는 센서 네트워크를 구성하는 각 노드에서의 데이터에 대한 송수신 시간을 수집한다. 센서 네트워크에서 데이터의 전송시간을 알아내기 위해서는 데이터가 센서 노드로부터 싱크 노드로 전송되는 시간을 알아야 한다. 이를 위해서는 데이터 전달 경로에 존재하는 각 노드에서의 데이터 송수신 시간을 수집해야 한다. 데이터 전송시간 계산 단계에서는 이전 단계에서 분석한 노드 간의 시간 편차와 각 노드에서의 데이터에 대한 송수신 시간을 바탕으로 데이터 전송시간을 계산한다. 데이터 전달 경로에 존재하는 각 노드에서의 데이터 송수신 시간과 시간 편차를 알면 노드 간의 데이터 전송시간을 계산할 수 있으며, 센서 노드로부터 싱크 노드까지의 노드 간의 데이터 전송시간을 합하면 데이터의 전체 전송시간을 구할 수 있다. 데이터 전송시간 분석 단계에서는 이전 단계에서 측정된 각 데이터에 대한 전송시간이 사용자가 기대하고 있는 전송시간의 허용범위 내에 있는가를 확인한다. 측정된 전송시간이 허용범위에 있지 않다면, 데이터 전송 경로상에 존재하는 노드 간의 데이터 전송시간을 분석하여 어느 구간에서 전송시간이 지연되었는가를 찾아내고 문제점을 분석한다.

### 3. 데이터 전송시간 분석 방법

본 장에서는 제 2장에서 설명한 프로세스에 따라 논문에서 제안하는 시스템이 센서 네트워크에서의 데이터 전송시간을 분석하기 위하여 노드 간의 시간차 분석, 데이터 송수신 시간 수집 및 데이터 전송시간을 계산하는 방법을 설명한다.

#### 3.1 노드 간의 시간차 분석

센서 네트워크에 존재하는 노드 간의 시간차는 기존에 제시되어 있는 기법들[8, 9]을 이용하여 분석할 수 있다. 본 논문에서는 기본적으로 송신자 - 수신자 동기화 기법[10]을 이용하여 각 노드 간의 시간차를 분석한다.



<그림 2> 두 노드 간의 데이터 교환

<그림 2>는 송신자와 수신자인 두 노드 간의 데이터 교환을 나타낸다. T1과 T4는 송신자 측의 타이머를 이용한 송수신 시간이며, T2와 T3은 수신자 측의 타이머를 이용한 송수신 시간이다. M1은 시간 동기화를 위하여 송신자가 보낸 메시지이며, M2는 수신자의 메시지를 처리한 후 답신으로 송신자가 보내는 메시지이다. T1, T2, T3, T4를 이용하여 두 노드 간의 시간편차와 메시지 전파시간을 알아낼 수 있다. 시간편차는 송

신 노드의 시간을 기준으로 했을 때 수신 노드와 송신 노드의 시간 차이를 의미하며, 진파시간은 2번의 데이터 전송에 대한 평균 진파시간을 의미한다.

$$T2 = T1 + \text{시간편차} + (\text{평균})\text{진파시간} \quad (1)$$

$$T4 - T3 = \text{시간편차} + (\text{평균})\text{진파시간} \quad (2)$$

$$\text{시간편차} = (T2 - T1) - (T4 - T3) / 2 \quad (3)$$

$$(\text{평균})\text{진파시간} = ((T2 - T1) + (T4 - T3)) / 2 \quad (4)$$

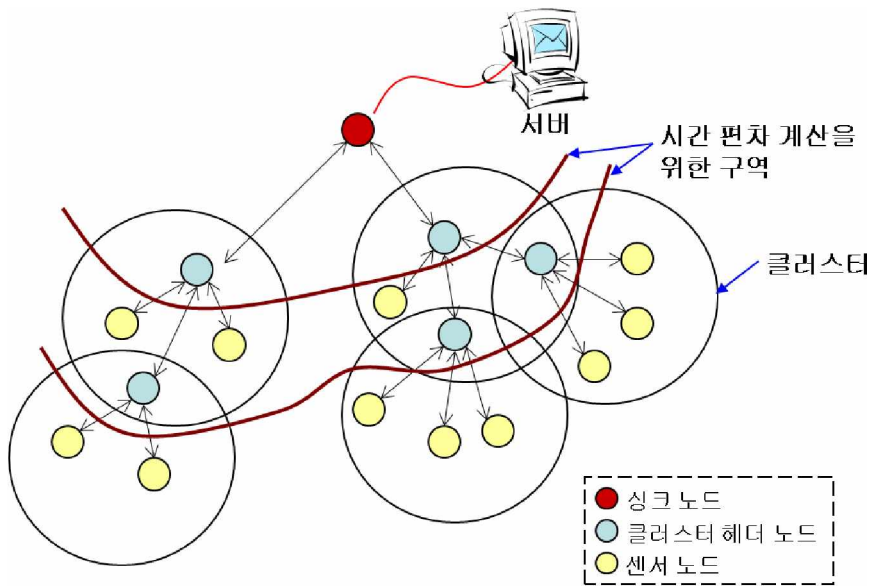
두 노드 간의 시간편차와 평균 데이터 전송시간은 식 (1)과 식 (2)로부터 도출할 수 있다. 식 (3)과 식 (4)는 식 (1)과 식 (2)로부터 도출한 두 노드 간의 시간편차와 평균 데이터 전송시간을 구하는 식이다.

이러한 방식을 이용하여 서버의 시간을 기준으로 각 노드 간의 시간차와 전송시간을 도출하고, 서버에 저장하였다가 전송시간을

계산할 때에 사용한다. 본 논문에서는 노드 간의 시간편차를 구할 때 다음과 같은 가정을 한다.

- 각각의 노드들은 자신의 시계를 가지고 시간을 측정한다.
- 센서 네트워크에는 클러스터 헤드가 존재한다.
- 싱크 노드는 시비와 연결되어 있다.
- 센서 네트워크에서의 표준 시간은 시비와 연결된 싱크 노드의 시간이다.

<그림 3>은 논문에서 제시한 시스템이 센서 네트워크에서 노드 간의 시간 편차를 도출해 내는 방법을 보여준다. 노드 간의 시간 편차는 두 단계로 이루어진다. 첫째는 싱크 노드와 클러스터 헤더 노드 간의 편차를 구하는 것이고, 둘째는 클러스터 헤더 노드와 클러스터에 속해 있는 다른 노드들 간의



<그림 3> 노드 간의 시간편차 도출 개념

편차를 구하는 것이다. 싱크 노드는 시버와 연결되어 있어 표준 시간이 되므로 두 단계를 통해서 편차를 구하게 되면 각 노드의 시간과 표준 시간과의 편차를 구할 수 있다.

### 3.2 데이터 송수신 시간 수집

센서 네트워크에는 무수히 많은 센서 노드들이 존재하므로, 감지한 데이터가 서버로 오기까지는 많은 노드들을 거쳐야 한다. 따라서 데이터가 직결한 시간 안에 서버로 전송되는가를 측정하기 위해서는 처음 데이터를 감지한 노드로부터 시버로 오기까지 걸린

시간을 계산해야 한다. 이를 위하여 데이터 전송시간 시스템은 각 노드에서 전송되는 패킷으로부터 각 노드들이 데이터를 송수신한 시간을 모두 수집할 필요가 있다.

<그림 4>는 데이터를 감지한 센서 노드가 라우터 노드나 싱크 노드에게 송신하는 패킷의 구조를 보여준다. 하나의 센서 노드가 가스, 조도, 온도, 습도와 같이 여러 가지 데이터를 감지할 수 있도록 하기 위하여 여러 개의 센서를 장착할 수 있으므로 노드에 몇 개의 센서가 장착되어 있는가를 알아야 할 필요가 있으므로 Data Length 필드가 필요하다. 센서가 여러 개 장착된 노드는 센서

Node ID	Packet Type	Data Length	Sensor Type	ADC High	ADC Low	...	Send Time	Padding
---------	-------------	-------------	-------------	----------	---------	-----	-----------	---------

- Node ID : 송신하는 노드 ID(1 byte)
- Packet Type : 송신하는 패킷의 타입(1 byte)
- Data Length : 송신하는 패킷의 센서 데이터 개수(1 byte)
- Sensor Type : 센서의 타입(1 byte)
- ADC High : 센서 데이터 값의 상위 영역(4 byte)
- ADC Low : 센서 데이터 값의 하위 영역(4 byte)
- ... : Data Length - 1 만큼의 {Sensor Type, ADC High, ADC Low} 값(가변적)
- Send Time : 송신하는 시간 (6 byte)
- Padding : 공백 값(1 byte)

<그림 4> 센서 노드의 패킷 구조

Node ID	Packet Type	Dest Node ID	Source Node ID	Hop Count	Router Node ID	...	Send Time	Recv Time	Org Data
---------	-------------	--------------	----------------	-----------	----------------	-----	-----------	-----------	----------

- Node ID : 송신하는 노드 ID(1byte)
- Packet Type : 송신하는 패킷의 타입(1 byte)
- Dest Node ID : 목적지 노드 ID(1 byte)
- Source Node ID : 보내는 노드 ID(1 byte)
- Hop Count : 거쳐온 노드 개수(1 byte)
- Router Node ID : 라우터 노드 ID(1 byte)
- ... : Hop Count - 1 만큼의 {Router Node ID} 값(1 byte)
- Send Time : 송신하는 시간(6 byte)
- Recv Time : 수신한 시간(6 byte)
- Org Data : 수신한 패킷(가변적)

<그림 5> 라우터 노드의 패킷 구조

의 개수만큼 Sensor Type, ADC High, ADC Low 필드를 패킷에 실어 보낸다.

<그림 5>는 센시 노드로부터 데이터를 전달 받은 라우터 노드가 다시 다른 라우터 노드나 싱크 노드로 송신하는 패킷의 구조를 보여준다. Dest Node ID는 데이터가 도달해야 하는 최종 노드의 식별 번호로, 다른 라우터 노드나 싱크 노드의 식별 번호가 될 수 있다. Hop Count는 센시 노드로부터 감지된 데이터가 각 라우터 노드로 전달될 때마다 증가한다. 즉, Hop Count를 통해서 몇 개의 라우터 노드를 거쳐서 데이터가 전달되었는가를 알 수 있다. Router Node ID는 이전에 기척온 라우터 노드의 식별 번호로, Hop Count의 값이 1이면 Router Node ID는 Node ID의 값과 같다. Router Node ID 필드는 데이터가 라우터 노드로 전달될 때마다 추가된다.

<그림 6>은 센시 네트워크에서 서버의 역할을 수행하는 싱크 노드의 패킷 구조를 보여준다. 싱크 노드는 센시로부터 감지된 데이터의 최종 목적지이므로, 다른 패킷에 비하여 구조가 단순하다. Packet Type은 싱크 노드가 다른 노드로 전송하는 데이터 패킷의 종류를 나타낸다.

센시 네트워크에서는 이러한 구조들을 가지는 데이터 패킷을 통하여 각 노드가 데이터를 송수신한 시각을 수집할 수 있고, 각 노드가 데이터를 송수신한 시각은 서버의 데이터베이스에 저장된다. 서버에서는 데이터베이스에 저장된 정보를 바탕으로 데이터 전송 시간을 계산한다.

### 3.3 데이터 전송시간 계산

서버로 전달된 데이터의 전송시간을 계산하기 위해서는 각 노드로부터 전송된 패킷을 분석해야 한다. 이를 위해서 서버에서는 각 노드로부터 전송된 패킷으로부터 필요한 정보를 뽑아내어 데이터베이스에 저장한다. 서버는 저장된 정보로부터 각 데이터에 대한 전송시간을 계산한다. 전송시간을 계산하기 위해 필요한 정보를 저장하는 데이터베이스의 테이블에는 다음과 같은 내용이 저장된다.

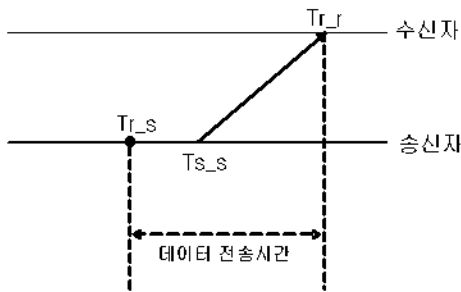
- 노드 ID
- 시간편차
- 전달 데이터
- 송신 시각
- 수신 시각
- 데이터 수신 순서

Node ID	Packet Type	Send Time	Recv Time	Original Data
---------	-------------	-----------	-----------	---------------

- Node ID : 노드의 ID(1 byte)
- Packet Type : 패킷의 타입(1 byte)
- Send Time : 메시지 전송 시각(6 byte)
- Recv Time : 메시지 수신 시각(6 byte)
- Original Data : 수신한 패킷(가변적)

<그림 6> 싱크 노드의 패킷 구조

노드 ID는 각 노드의 식별자이며, 기준시간과의 시간편차는 서로 연결되어 있는 두 노드 간의 시간편차를 말한다. 전달 데이터는 센서 노드로부터 감지된 데이터가 무엇인가를 말한다. 센서 네트워크에 존재하는 각 노드들은 수많은 데이터를 송수신하므로, 데이터 전송시간은 각각의 데이터에 대하여 계산해야 한다. 송신 시각은 노드가 해당 데이터를 송신한 시각이며, 수신 시각은 해당 데이터를 수신한 시각이다. 센서 노드에서는 데이터를 감지한 시각이 수신 시각으로 지정된다. 송신 시각과 수신 시각은 각 노드의 타이머를 이용하여 측정된다. 데이터 수신 순서는 노드들이 해당 데이터를 전달받은 순서를 의미한다. 해당 데이터의 전송시간은 데이터 수신 순서대로 계산하여 합산한다.



<그림 7> 각 노드 간의 데이터 전송시간

<그림 7>은 각 노드에서의 데이터 전송시간을 측정하는 방법을 보여준다. 그림에서 Tr은 데이터를 수신한 시각을, Ts는 데이터를 송신한 시각을 나타낸다. 센서 노드의 경우에는 Tr이 데이터를 감지한 시각이 된다. 본 논문에서는 각 노드에서의 데이터 전송시간을 데이터를 수신한 시각으로부터 데이터를 처리하여 타겟 노드로 송신하고, 타겟 노

드가 데이터를 수신한 시간까지로 간주한다.

<그림 6>에 나타난 Tr과 Ts는 각 노드의 타이머를 사용한다. 따라서 실제의 전송시간을 계산하기 위해서는 시간 동기화를 통해 구한 기준시간과의 편차를 이용해야 한다. 각 노드 간의 데이터 전송시간을 계산하는 공식은 다음과 같다.

- 송신자의 표준시간과의 시간편차 : Ds
- 수신자의 표준시간과의 시간편차 : Dr
- 송신자의 데이터 수신 시각 : Tr\_s
- 송신자의 데이터 송신 시각 : Ts\_s
- 수신자의 데이터 수신 시각 : Tr\_r
- 데이터 전송시간 : T

$$T = (Ts_s - Tr_s) + \{(Tr_r - Dr) - (Ts_s - Ds)\} \quad (4)$$

식 (4)와 같은 방법으로 데이터베이스에 저장한 정보를 바탕으로 해당 데이터에 대하여 각 노드 간의 데이터 전송시간을 계산하여 합하면 센서 노드로부터 서버로까지의 데이터 전송시간을 구할 수 있다. 송신자가 센서 노드이면 (Ts\_s - Tr\_s)의 값은 0이 된다. 센서 노드로부터 n번째 노드까지의 전체 데이터 전송시간을 식 (5)를 이용하여 구할 수 있다.

- 데이터 수신 시각 : Tr
- 데이터 송신 시각 : Ts
- 시간 편차 : D

$$T_{tot} = \sum_{k=2}^N [(Ts(k-1) - Tr(k-1)) + \{(Tr(k) - D(k)) - (Ts(k-1) - D(k-1))\}] \quad (5)$$



이렇게 센서 노드로부터 시버로까지의 데이터 전송시간을 분석하면, 실제 데이터 전송 시간이 요구사항으로 주어진 제약시간을 만족시키지 못한 경우에는 어느 부분에서 지연이 많이 되는가를 알아낼 수 있다. 따라서 지연이 생기는 부분에 대한 원인을 알아내어 해결하면, 센서 네트워크에서의 전송시간 데이터 처리를 올바르게 할 수 있다.

### 3.4 데이터 전송시간 분석

3.3절에서의 방법으로 각 데이터에 대한 전송시간을 구하고 나면, 센서 네트워크에서 허용되는 전송 제약시간과 실제로 데이터를 전송하는데 걸리는 시간을 비교하여 확인한다. 3.3절을 통하여 각 데이터 전송에 있어서 다음과 같은 정보를 알 수 있다.

- 데이터 전송 횟수
- 각 노드간의 데이터 전송 시간
- 각 노드에서의 데이터 처리 시간
- 센서 노드로부터 서버로의 전송 시간

데이터 전송 횟수는 각 데이터를 시버로 전송하기까지 몇 개의 노드를 기렸는가를 알려주며, 각 노드간의 데이터 전송 시간은 3.3 절에서 설명한 전송 시간을 말한다. 각 노드에서의 데이터 처리 시간은 각 노드에서 데이터를 수신하여 다른 노드로 송신하기 전까지 걸린 시간을 말한다. 이것은 식 (4)에서 ( $T_{s,s} - T_{r,s}$ ) 부분을 말한다. 그리고 센서 노드로부터 시버로의 전송 시간은 각 데이터에 대한 전체 전송 시간을 말한다.

## 4. 데이터 전송시간 모니터링 및 분석 시스템

이 장에서는 제 2장과 제 3장에서 설명한 절차와 방법을 바탕으로 센서 네트워크에서의 데이터 전송시간을 분석하기 위한 데이터 전송시간 모니터링 및 분석 시스템을 구현하고, 사례 연구를 수행한 결과를 설명한다.

### 4.1 시스템의 구조

<그림 8>은 본문에서 제안하는 전송시간 분석 방법을 지원하기 위하여 구현한 데이터 전송시간 모니터링 및 분석 시스템의 구조를 보여준다. 이 시스템은 이클립스 플랫폼의 플러그인으로 개발되었으며, 사용자 인터페이스, 접근 제어기, 시간 수집기, 편차 분석기, 시간 분석기, 정보 저장소로 구성된다.



<그림 8> 데이터 전송시간 모니터링 및 분석 시스템의 구조

시스템을 구성하는 각 모듈의 역할은 다음과 같다.

- 사용자 인터페이스 : 사용자가 시스템에 접근하여 센서 네트워크에서의 데이터 전송시간을 모니터링하고 분석할 수 있도록 하는 인터페이스를 제공한다.
- 접근 제어기 : 외부로부터 시스템으로의 접근을 제어하는 모듈이다. 접근 기기를 통해서 승인된 사용자만이 시스템에 접근할 수 있다.
- 시간 수집기 : 센서 노드로부터 싱크 노드로 전달된 데이터에 대한 전송시간을 수집하는 모듈이다. 전송시간을 수집하기 위하여 각 노드로부터 싱크 노드로 전달된 데이터 패킷으로부터 데이터 전송시간을 추출한다. 각 노드에서 데이터의 전송시간을 수집하기 위해서 사용된 패킷의 구조는 3.2절에서 설명한 패킷의 구조를 따른다.
- 편차 분석기 : 싱크 노드와 다른 노드 사이의 시간 편차를 분석하기 위한 모듈이다. <그림 3>의 개념을 바탕으로 노드간에 주고받은 패킷을 분석하여 시간 편차를 구하고, 그로부터 싱크 노드와 각 노드 사이의 시간 편차를 분석한다.
- 시간 분석기 : 데이터의 전송시간을 분석하기 위한 모듈이다. 시간 수집기가 수집한 데이터 전송시간과 편차 분석기가 계산한 노드 간의 시간 편차를 바탕으로 데이터가 센서 노드로부터 싱크 노드까지 전달된 전체 전송시간을 계산한다.
- 정보 저장소 : 센서 노드로부터 전달된 데이터와 데이터의 전송시간 및 노드 간의 시간 편차와 같이 데이터 전송시

간을 분석하기 위해 필요한 모든 정보를 저장하는 데이터베이스이다.

## 4.2 시스템의 구현 및 사례연구

논문에서 제안하는 방법에 따라 센서 네트워크에서의 데이터 전송시간을 분석하기 위하여 4.1절에서와 같은 구조를 가지는 시스템을 구현하였다. 전송시간을 분석하기 위하여 각각의 노드들은 3.2절에서 설명한 구조를 가지는 패킷을 서버로 전송하였으며, 이를 바탕으로 서버는 제 3장에서 제시한 알고리즘에 따라서 데이터 전송시간을 분석하도록 하였다. 실제로 구현을 해보니, 노드에서 패킷을 전송할 때 인터럽트가 발생하여 각 노드의 타이머들이 잠시 멈추는 현상이 발생하여 각 노드간의 시간편차는 일정하지가 않았다. 따라서 주기적으로 노드간의 시간편차를 계산해야 하는 문제가 발생하였다. 시간 편차를 계산하기 위해서는 노드간에 데이터 패킷을 송수신해야 하므로 오버헤드가 발생하게 되고, 시간편차를 자주 계산할수록 오버헤드가 증가하게 되므로 시간편차의 계산 주기도 센서 네트워크의 상황에 따라 적절히 결정할 필요가 있다. 이에 따라 구현한 시스템에서는 사용자가 시간편차 계산주기를 설정하면, 설정한 주기마다 노드간의 시간편차를 계산하도록 하였다.

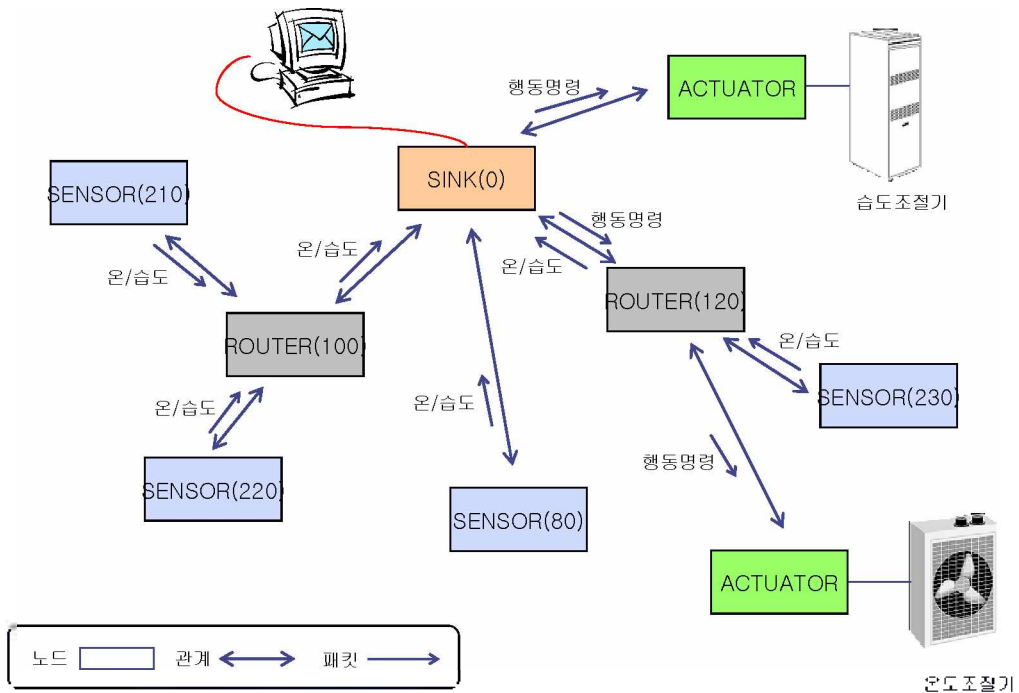
구현한 시스템을 이용하여 센서 네트워크에서의 데이터 전송시간을 분석한 사례 연구로 실내 온/습도 최적화 유지 시스템을 설계하여 적용하였다. 실내 온/습도 최적화 시스템에서는 센서 노드로부터 감지된 온/습도 값이 라우터를 거쳐, 싱크 노드로 전달이 되고, 일정한

한계(threshold) 값의 범위를 벗어나게 되면 액츄에이터 노드가 작동되어 온/습도를 조절한다. 습도와 관련된 범위 값이 상한 값을 벗어나게 되면 습도 액츄에이터는 습도조절기의 작동을 멈추고, 하한 값을 벗어나면 습도 조절기가 작동하여 실내의 습도를 높인다. 온도의 경우도 습도 조절의 경우와 같이 상한 값과 하한 값에 따라 온도조절기의 작동을 조절하여 최적의 실내 온도를 유지한다. <그림 9>는 이러한 실내 온/습도 최적화 유지 시스템의 구조를 보여준다. 실내 온/습도 최적화 유지 시스템에서는 감지한 온/습도의 서버로의 전송 허용시간을 3초로 설정하였다.

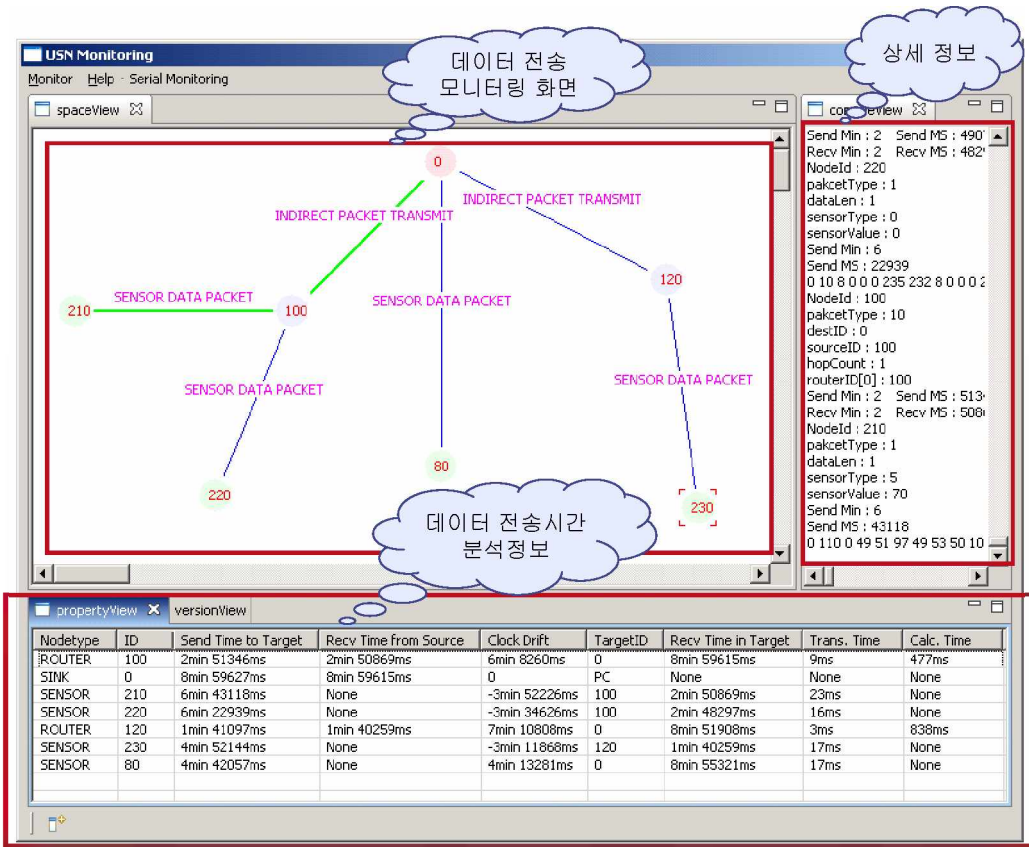
<그림 10>은 실내 온/습도 최적화 유지 시스템에서 설정한 전송 허용시간인 3초 내에 감지한 온/습도에 대한 데이터가 서버로

전송되고 있는가를 모니터링하고 분석하기 위한 데이터 전송시간 모니터링 및 분석 시스템의 실행 화면을 보여준다. 화면의 윗부분은 센서 노드들이 데이터를 전송하는 과정을 실시간으로 미주얼하게 보여주는 부분이고, 그 오른쪽에는 상세한 정보가 표시된다. 그리고 화면의 아랫 부분은 데이터 전송시간을 분석하기 위한 정보를 보여준다.

데이터 전송시간을 분석하는 부분을 좀 더 자세히 살펴보면 <그림 11>과 같다. 각 노드가 데이터를 전달해야 하는 목표 노드는 TargetID를 보면 알 수 있고, Clock Drift는 목표 노드와의 시간 편차를 의미한다. Trans. Time은 전파시간으로 노드 간의 시간편차를 고려하여 이미 계산한 값이며, Calc. Time은 각 노드에서의 작업 처리시간을 의미한다.



<그림 9> 실내 온/습도 최적화 유지 시스템의 구조도



〈그림 10〉 데이터 전송시간 모니터링 및 분석 시스템 화면

Nodetype	ID	Send Time to Target	Recv Time from Source	Clock Drift	TargetID	Recv Time in Target	Trans. Time	Calc. Time
ROUTER	100	2min 51346ms	2min 50869ms	6min 8260ms	0	8min 59615ms	9ms	477ms
SINK	0	8min 59627ms	8min 59615ms	0	PC	None	None	None
SENSOR	210	6min 43118ms	None	-3min 52226ms	100	2min 50869ms	23ms	None
SENSOR	220	6min 22939ms	None	-3min 34626ms	100	2min 48297ms	16ms	None
ROUTER	120	1min 41097ms	1min 40259ms	7min 10808ms	0	8min 51908ms	3ms	838ms
SENSOR	230	4min 52144ms	None	-3min 11868ms	120	1min 40259ms	17ms	None
SENSOR	80	4min 42057ms	None	4min 13281ms	0	8min 55321ms	17ms	None

〈그림 11〉 데이터 전송시간 분석

센서로부터 감지된 데이터가 서버로 전달되는 시간을 구하기 위해서는 각 노드에서의 전송시간, 작업 처리시간을 합하면 된다. <그림 11>의 결과를 바탕으로 210번 센서 노드가 감지한 데이터가 서버로 전송되는 시간을 분석해보면, 210번 센서 노드는 감지한

데이터를 100번 라우터 노드로 전달하고, 100번 라우터 노드는 전달받은 데이터를 싱크 노드로 다시 전송한다. 따라서 210번 센서 노드가 감지한 데이터가 서버로 전송되는 시간은 210번 노드에서의 Trans. Time(23ms), 100번 노드에서의 Calc. Time(477ms)와 Trans.

Time(9ms)를 모두 합한 결과인 509ms이다. 이렇게 계산한 509ms는 실내 온/습도 최적화 유지 시스템의 설계 시에 설정한 전송 허용시간인 3초보다 작으므로, 210번 센서 노드에서 감지한 실내 온/습도 값은 원하는 시간 내에 서버로 잘 전송되고 있음을 확인할 수 있다.

## 5. 결론 및 향후 연구

센서 네트워크는 국방 분야, 환경 감시 분야, 의료 분야 등 다양한 곳에서 필요한 실시간 응용 시스템을 개발하는데 사용되고 있다. 실시간 시스템에서 가장 중요한 것은 데이터의 실시간 수집 및 처리로, 센서 네트워크를 이용한 응용 시스템에서 데이터의 실시간 수집 및 처리가 올바르게 이루어지기 위해서는 센서 네트워크 상에서 데이터 전송이 허용시간 범위 내에서 이루어져야 한다. 센서 네트워크는 수많은 센서들로 구성되기 때문에 어느 구간에서 데이터 전송이 지연되는가를 알아내기가 쉽지 않다. 따라서 서버로의 데이터 전송시간이 요구한 시간 내에 전송되는가를 알아내는 것뿐만 아니라 어느 구간에서 문제가 발생하여 지연되고 있는가를 알아내는 것 또한 필요하다.

이에 따라 본 논문에서는 데이터 전송시간에 대한 모니터링을 통하여 센서 네트워크에 존재하는 노드들이 허용시간 범위 내에서 서버로 데이터를 전송하고 있는가를 확인하기 위한 데이터 전송시간 분석 시스템을 제시하였다. 이를 위하여 시스템이 데이터 전송시간을 분석하기 위한 절차를 제시하고, 제시

한 절차에 따라 전송시간을 분석하기 위해 필요한 시간차 분석 방법, 데이터 송수신 시간 수집 방법 및 데이터 전송시간 계산을 위한 공식을 제시하였으며, 제시한 방법을 바탕으로 전송시간 모니터링 및 분석 시스템을 구현하고, 사례 연구를 수행하였다. 논문에서 제시한 방법 및 시스템을 통하여 데이터 전송시간을 분석하였을 경우의 장점은 다음과 같다.

- 센싱된 데이터의 서버로의 전송시간 측정 : 제시한 방법을 이용하면 센서 네트워크에서 각 노드가 데이터를 전송하는 시간을 알아낼 수 있으며, 이를 통하여 센싱된 데이터의 종류에 따라 서버로 전송되는 시간을 알아낼 수 있다.
- 데이터 전송 지연 구간의 발견 : 제시한 시스템을 통하여 각 노드에서의 데이터 송수신 시간을 측정할 수 있으므로, 전송이 지연될 경우에 어느 구간에서 지연되는가를 알아낼 수 있다.
- 데이터 전송시간 요구사항에의 부합성 검사 : 센싱된 데이터의 서버로의 전송시간을 측정하여 전송시간 요구사항과 비교함으로써, 측정된 전송시간이 데이터 전송시간 요구사항에 부합하는가를 확인할 수 있다.
- 기능 중지된 센서 노드의 발견 : 데이터 전송이 지연되는 구간을 발견함으로써 문제가 발생한 센서 노드를 찾아낼 수 있으며, 문제가 발생한 센서 노드를 테스트해 봄으로써 센서 노드의 기능이 중지되었는지를 알아낼 수 있다.

이와 같은 장점을 가지는 시스템을 통하여 지연이 생기는 부분에 대한 원인을 알아내어

해결하면, 센서 네트워크에서의 실시간 데이터 처리를 올바르게 할 수 있다.

향후에는 제안하는 방법 및 시스템을 발전시켜, 센서 네트워크에서의 데이터 전송시간 모니터링 및 분석을 통하여 발견한 문제점을 해결하기 위한 디버깅 방법 및 지원 도구에 대한 연구를 수행하고자 한다.

---

### 참 고 문 헌

---

[1] I. F. Akyildiz, W. Su et al., "A Survey on Sensor Networks," *IEEE Communications Magazine*, August 2002.

[2] 채동현, 한규호, 임정수, 안순신, "센서 네트워크의 개요 및 기술동향", *한국정보과학회 학회지*, 제22권, 제12호, 2004.

[3] A. K. Mok, and G. Liu, "Efficient Run-Time Monitoring of Timing Constraints," *Third IEEE Real Time Technology and Applications Symposium*, 1997.

[4] J. J. P. Tsai, and S. J. H. Yang, eds, *Monitoring and Debugging of Distributed Real-Time Systems*, IEEE CS Press, 1995.

[5] S. C. V. Raju, R. Rajkumar, and E. Jahanian, "Monitoring Timing Const

raints in Distributed Real Time Systems," *Real Time Systems Symposium*, 1992.

[6] Chan-Gun Lee, Aloysius K. Mok, and Prabhudev Konana, "Monitoring of Timing Constraints with Confidence Threshold Requirements," *24th IEEE International Real Time Systems Symposium*, 2003.

[7] Niels Aakvaag, Mogens Mathiesen, and Gilles Thonet, "Timing and Power Issues in Wireless Sensor Networks—An Industrial Test Case," *2005 International Conference on Parallel Processing Workshops*, 2005.

[8] Ganciwai, S., Kumar, R., and Srivastava, M. B., "Timing-Sync Protocol for Sensor Networks," *The First ACM Conference on Embedded Networked Sensor System (SenSys)*, 2003.

[9] 노진홍, 홍영식, "무선 임베디드 환경에서의 시간 동기화", *정보과학회논문지 : 정보통신*, 제32권, 제6호, 2005, pp. 668-675.

[10] D. L. Mills, "Internet Time Synchronization : The Network Time Protocol," *IEEE TRANSACTIONS ON COMMUNICATIONS*, Vol. 39, No. 10, 1991, pp. 1482-1493.

저 자 소 개



정기원  
1967년  
1981년  
1983년  
1990년~현재  
관심분야

(E-mail : chong@ssu.ac.kr)  
서울대학교 공과대학 전기공학과 졸업  
미국 알라바마 주립대학(현츠빙) 신산학과 (공학 석사)  
미국 텍사스주립대학(알링턴) 신산학과 (공학 박사)  
숭실대학교 컴퓨터학부 교수  
소프트웨어 개발 프로세스, 방법론, 모델링, 실시간 응용,  
전자거래, 정보시스템 개발 및 평가



김제철  
1994년  
2004년  
2005년~현재  
1995년~현재  
관심분야

(E-mail : fbjckim@yahoo.co.kr)  
경원대학교 독어독문학과 졸업  
숭실대학교 정보과학대학원 (공학 석사)  
숭실대학교 대학원 컴퓨터학과 박사과정 재학  
한양(주) 정보지원팀 팀장  
임베디드 소프트웨어, 소프트웨어 공학, 소프트웨어 개발  
방법론, 엔터프라이즈 아키텍치, 프로젝트 관리



김주연  
2004년  
2006년  
2006년~현재  
관심분야

(E-mail : sespop@empal.com)  
한말대학교 컴퓨터공학과 졸업  
숭실대학교 대학원 컴퓨터학과 (공학 석사)  
숭실대학교 대학원 컴퓨터학과 박사과정 재학  
유비쿼터스 컴퓨팅, 임베디드 시스템, 웹 서비스, 실시간  
컴퓨팅, 소프트웨어공학



이우진  
2000년  
2002년  
2007년  
2008년~현재  
관심분야

(E-mail : bluewj@empal.com)  
숭실대학교 컴퓨터학부 졸업  
숭실대학교 대학원 컴퓨터학과 (공학 석사)  
숭실대학교 대학원 컴퓨터학과 (공학 박사)  
숭실대학교 컴퓨터학부 시간강사  
유비쿼터스 컴퓨팅, 임베디드 시스템, 웹 서비스, 모바일  
컴퓨팅, 소프트웨어공학