

센서 네트워크를 이용한 교량 실시간관리를 위한 미들웨어 연구

유춘근*, 임철우*, 김종근**, 강병욱**

*영남대학교 컴퓨터공학과

**영남대학교 전자정보공학부 컴퓨터공학전공

e-mail:tonko96@nate.com

A Study on Middleware for Realtime Bridge Control with Sensor-Network

Chun-Gun Yu*, Chul-Woo Rhim*, Chong-Gun Kim**, Byung-Wook Kang**

*Dept of Computer Science, Yeungnam University

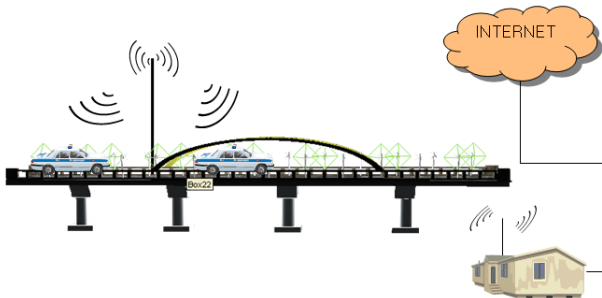
** School of Electronic Engineering, Communication Engineering &
Computer Science, Yeungnam University

요 약

센서 네트워크를 이용하여 교량을 실시간으로 관리하기 위하여 미들웨어를 사용한다. 그러나 현재 미들웨어는 단순히 센서 네트워크에서 수집한 자료를 처리하는 단순한 수준에 불과하다. 본 논문에서 실시간으로 센서로부터의 정보를 수집/가공하고 센서 노드들과 전용 애플리케이션 및 웹 브라우저 간의 중재 역할을 하는 미들웨어를 제안한다. 또 데이터베이스 고장이 발생하였을 경우 파일로 수집정보를 저장 하며 데이터베이스가 복구되었을 때 데이터베이스 부하를 줄이기 위하여 데이터베이스의 사용률이 적을 때 파일로 수집된 정보를 데이터베이스에 저장하는 기능을 제안한다. 그리고 실시간으로 교량 정보등을 감시하여 이상이 발생하였을 때 경고 메시지를 발생시켜 관리자가 교량의 비정상적인 상태에 대하여 빠르고 쉽게 인지할 수 있는 메커니즘을 제안한다.

1. 서론

최근 외부에서 발생하는 이벤트를 감지하여 센서를 이용해 전달하는 센스 네트워크 기술이 활발히 연구되고 있는 추세이다[1]. 도로 유지관리 측면에서도 센서 네트워크 기술이 적용되고 있으며 지능형 교통정보 시스템 등 일부 분야에서도 제한적으로 사용되고 있다. 교량이나 도로 상태를 현장에서 직접 파악하고 대처하는 현재의 방법은 효율성, 신뢰성, 신속성 및 안정성 등에서 여러 문제점을 가지고 있다. 센스 네트워크를 이용하면 그림 1에서 보는 것과 같이 전용 애플리케이션 또는 웹 브라우저를 통하여 원격으로 교량의 모든 지점의 상태를 실시간으로 파악하고 문제가 발생하였을 경우 신속하게 대처할 수 있다.



(그림 1) 센서네트워크를 이용한 교량관리 시스템

기존 논문들은 센서 네트워크를 이용하여 교량을 관리할 수 있다는 점을 제안하고는 있지만 구체적인 미들웨어의 기능과 표준 구조에 대하여 고찰하고 있지 못하다. 대부분의 연구들은 교량 관리 시스템을 구현할 때에 필수적인 기능과 효율적인 구조에 대하여 제안하고 있지 않기 때문에 교량 관리 시스템을 구현하고자 할 경우에는 충분한 자료가 되지 못하고 있다.

본 연구에서는 위에서 제시한 문제점을 해결하고자 센서 네트워크를 이용하여 실시간 관리가 가능한 미들웨어의 구조와 기능을 제안함으로써 센서 네트워크를 이용한 교량 관리 시스템의 신뢰성, 효율성, 안정성 및 정확성을 제고하도록 한다.

본 논문의 구성은 다음과 같다. 2장 관련연구에서는 미들웨어와 센서 네트워크에 대해 서술한다. 3장에서는 제한된 시스템 즉 미들웨어의 설계 및 구현에 대하여 기술하고 4장에서는 본 논문에 제안된 센서 네트워크를 이용한 미들웨어 구현에 관하여 서술하고 마지막 5장에서는 결론 및 향후 연구에 대하여 기술한다.

2. 관련연구

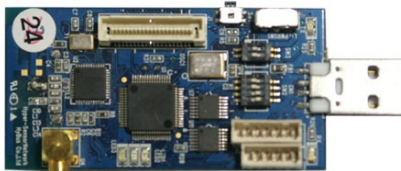
2.1 미들웨어

미들웨어는 컴퓨터와 컴퓨터간의 통신을 쉽고 안전하게 할 수 있도록 해주고 이에 대한 관리를 도와주는 소프트

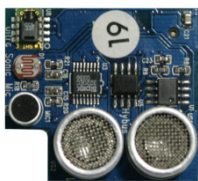
웨어로 정의할 수 있다. 즉, 서로 다른 프로토콜이나 시스템 운영체제, 데이터베이스와 애플리케이션 간에 통신을 지원해 주는 소프트웨어를 의미하며, 애플리케이션이 어떤 정보시스템 환경에서도 작동할 수 있도록 지원해주는 역할을 한다. 따라서 미들웨어는 클라이언트와 서버사이에 (혹은 컴퓨터와 컴퓨터 사이) 통신 기능을 제공해 줄 수 있는 모든 제품들을 의미하며, 이 기능을 제공한다면 아무리 간단하더라도 미들웨어라고 부를 수 있다.

2.2 센서 네트워크

센서 네트워크란 그림 2, 그림 3과 같은 센서가 부착되어 있어서 정보 센싱이 가능하고, 수집된 정보를 가공할 수 있는 프로세서가 달려있으며 이를 전송할 수 있는 무선 송수신기를 갖춘 소형장치, 즉 센서노드로 구성된 네트워크를 의미한다. 센서 네트워크의 일반적인 특징은 그 성향이 비슷한 애드혹(Ad-hoc) 네트워크에 비해 상대적으로 규모가 큰 센서 노드들이 조밀하게 배치될 수 있다는 점이다. 센서 노드는 열악한 환경 속에서 동작하므로 쉽게 그 기능을 수행하지 못할 수 있고 네트워크 토폴로지가 매우 빈번하게 변경될 수 있다. 기존의 네트워크는 일대일 통신이 고려 대상이었지만 센서 네트워크에서는 브로드캐스팅 패러다임이 주로 이용된다. 센서 노드들은 전원, 메모리를 비롯한 하드웨어 자원뿐만 아니라 계산능력 또한 매우 제한된다. 또한 센서 노드들은 하나의 동일한 대상에 다량으로 배치되는 경우가 많기 때문에 IP 주소와 같은 국제적인 식별자를 갖는 것이 어렵다[3].



(그림2)MSP430MoteType2



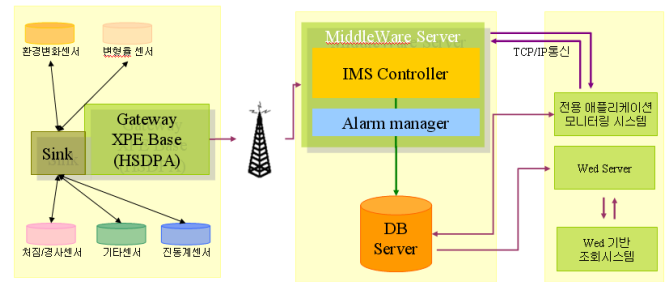
(그림3)US/MicSensors

3. 미들웨어 설계 및 제한

3.1 시스템 개요

미들웨어 전체 시스템은 그림 4와 같이 크게 센서 단, 미들웨어 단, 애플리케이션 단으로 이루어져있다. 센서 단에서는 환경변화 센서, 변형률 센서, 처짐/경사 센서, 진동 센서 등에서 수집한 데이터를 싱크노드로 전송한다. 싱크 노드에 전송된 데이터는 시리얼 통신으로 Gateway인 XPE(Windows XP Embedded)에 전송되고 XPE로 전송된

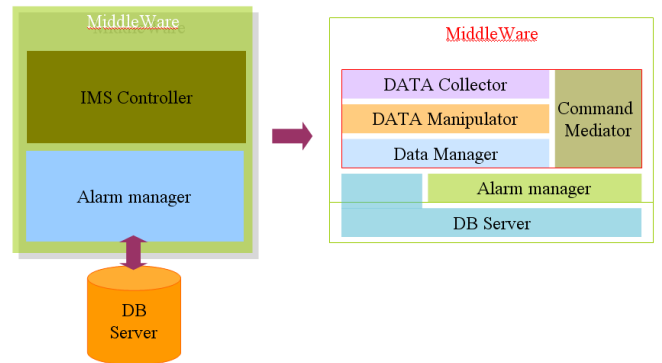
데이터는 HSDPA(high speed downlink packet access)를 통해 휴대폰 기지국 망을 이용하여 TCP/IP 통신으로 미들웨어로 전송되어진다. 미들웨어로 전송된 Raw Data는 IMS Controller를 통해 가공된다. 가공된 데이터는 DB에 저장되고 일부 관심 데이터들은 알람 매니저에 전달되어 알람발생 여부를 실시간으로 판단한다. 실시간 데이터들은 IMS Controller를 통해서 전용 애플리케이션이나 웹 서버로 전송되며 과거 데이터 이력들은 DB서버에서 애플리케이션이나 웹 서버로 제공된다. 또한 애플리케이션 단에서 센서 노드에 명령을 전송할 경우 미들웨어의 IMS Controller가 명령들을 센서 단으로 중계한다.



(그림4) 전체 시스템 흐름도

3.2 미들웨어 아키텍처

미들웨어 아키텍처는 각각의 센서에서 가공되지 않은 데이터를 받아서 실제 단위로 환산하고, 환산된 데이터를 애플리케이션 단에 중계하며, 데이터베이스에 저장하는 역할을 담당한다. 그리고 전용 애플리케이션에서 센서노드에 전송할 명령을 미들웨어가 중계하여 각각의 노드 센서에 전송한다.

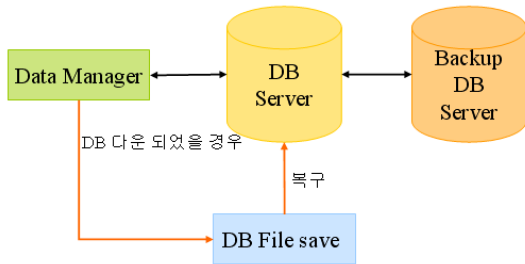


(그림5) 미들웨어 아키텍처

미들웨어는 그림 5와 같이 여러 단계를 거쳐 데이터를 처리한다. 각각의 기능은 다음과 같은 역할을 한다.

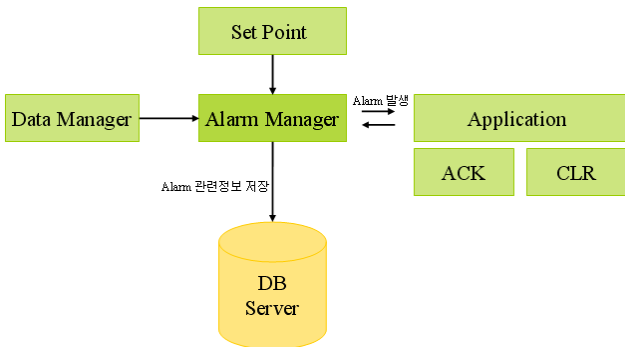
- Data Collector - 각각의 센서 노드에서 Raw Data를 수집한다.
- Data Manipulator - Data Collector에서 수집한 Raw Data를 공학적 물리량으로 환산하는 역할을 한다.

- Data Manager - Data Manipulator에서 환산한 정보를 DB에 저장하고 애플리케이션 단에서 현재 모니터링하고 있는 데이터를 애플리케이션 단으로 실시간으로 보내기도 하고 특정한 데이터를 감시하도록 Alarm Manager에게 보낸다. Data Manager는 DB 고장 여부를 감시하며 DB 고장에 대하여 대처한다. 만약 DB 서버에 고장이 발생하면 중요도가 높은 특정 데이터를 아래 그림 6과 같이 파일로 저장한다. 데이터들을 파일로 저장하면서 데이터베이스가 복구될 때까지 Data Manager는 일정한 간격을 두고 데이터베이스를 감시한다. 데이터베이스가 복구되면 데이터를 바로 복구 하지 않고 데이터베이스의 부하가 걸리지 않도록 천천히 데이터들을 DB로 저장한다.



(그림6) 데이터베이스 복구과정

- Alarm Manager - IMS Controller와 데이터베이스 사이에 위치하며 중요도가 높고 지속 모니터링이 요구되는 데이터를 관찰하며 비정상 상태가 되면 알람 메시지를 발생한다. 그림 7은 알람발생 처리과정을 보여주고 있다. Alarm Manager는 데이터베이스에 실시간으로 감시하도록 설정된 데이터들을 Data Manager에서 전달받는다. 알람은 4가지로 LL, L, H, HH로 분류 된다. 관심 데이터에 정상적인 값의 범위를 설정해 두고 데이터가 낮은 범위에 근접하며 L, 낮은 범위를 벗어나면 LL, 높은 범위에 근접하면 H, 높은 범위를 벗어나면 HH 알람을 발생시킨다. 알람이 발생되면 청각적 시각적 경고메세지가 관리자가 전달되어 관리자가 쉽게 알람을 인지할 수 있도록 한다. 관리자가 알람을 확인하면 알람 메시지를 ACK 상태로 만들고 적절한 조치를 취한다. 조치된 모든 알람을 삭제하기 위하여 CLR명령이 이용된다.



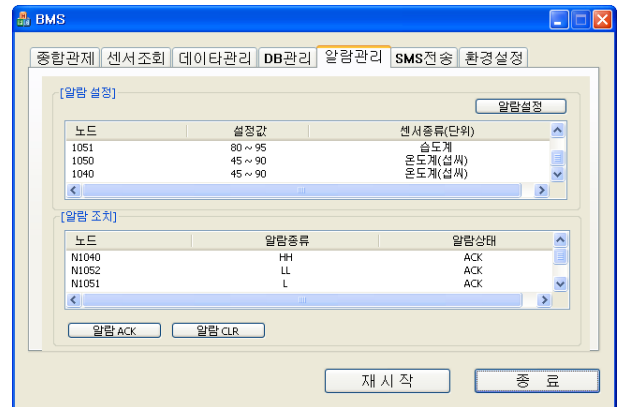
(그림7) 알람발생 처리과정

- Command Mediator - 전용 애플리케이션에서 센서 단에 보내는 명령어를 전달 받아 중계하는 역할을 한다. 발생된 명령어는 DB에 저장하여 명령 이력과 조치 이력을 유지한다.

- DB Server - Data Manager, Alarm Manager, Command Mediator에서 보내오는 데이터를 저장한다. 전용 애플리케이션 또는 웹 브라우저에서 과거 데이터가 필요한 경우 이들이 요구에 따라 미들웨어를 거치지 않고 데이터를 제공한다. 애플리케이션 단에서 미들웨어를 거치지 않고 데이터베이스와 바로 통신하는 이유는 빈번한 데이터 요구가 발생하면 미들웨어에 부하가 증가하므로 이를 해결하기 위함이다.

4. 미들웨어 구현

그림 8은 본 논문에서 제시한 아키텍처 구조를 이용하여 구현한 미들웨어 프로그램을 보이고 있다. 미들웨어 프로그램은 종합관제, 데이터 환산, 데이터관리, DB관리, 알람관리, SMS관리 및 환경설정 등의 기능으로 구성되어있다.



(그림 8) 미들웨어 구현

- 종합관제 - 미들웨어를 구성하는 센서조회, 데이터관리, DB관리, 알람관리, SMS관리 등으로 이루어져있으며, 이러한 정보들을 종합적으로 모니터링 해야 하는 핵심 요소들을 모아 놓았다.

- 데이터 환산 - Data Manipulator에 대한 인터페이스를 제공하며 각 노드마다 데이터 환산을 어떻게 할 것인지를 여기서 설정하도록 한다.

- 데이터 관리 - Data Collector에 대한 인터페이스를 제공하며 센서 단에서 수집되는 데이터를 감시하는 기능을 포함한다.

- DB관리 - Data manager에 해당되는 인터페이스 부분이며 주로 DB 고장 여부를 감시하여 DB 고장에 대하여 대처한다. 만약 DB 서버에 고장이 발생하면 중요도가 높은 특정 데이터를 파일로 저장하도록 설정하는 부분이 있다.

- 알람관리 - 중요도가 높고 지속적인 모니터링이 요구되는 데이터를 등록하고 알람의 발생 범위를 설정하는 인터

페이스 부분과 발생한 알람을 처리할 수 있는 부분으로 구성되어 있다.

- SMS관리 - 알람 발생 시 관리자 또는 등록된 엔지니어에게 SMS를 발송하도록 설정하는 인터페이스로 이루어진다.
- 환경설정 - 미들웨어의 프로그램의 기본적인 옵션을 설정할 수 있는 인터페이스들이 들어간다.

5. 결론 및 향후 연구과제

본 논문에서는 센서 네트워크를 이용한 교량 감시 시스템에서 미들웨어의 핵심적인 기능과 효율적인 구조에 대하여 고찰하였고 기본적으로 표준적인 미들웨어 아키텍처를 제안하였다. 미들웨어가 센서단과 어플리케이션 단 사이에서 수행하여야 할 가장 기본적인 기능들을 계층으로 정의하였고 계층사이의 데이터의 흐름을 제안하였다. 결과적으로 제안된 아키텍처를 기반으로 센서 네트워크를 이용한 교량 관리 시스템에서의 신뢰성, 효율성, 안정성 및 정확성에 대하여 고찰할 수 있는 기반을 마련하였다.

향후 연구과제는 본 논문에서 제안된 미들웨어 아키텍처를 실제 교량 시스템에 적용하여, 제안된 구조에 의한 미들웨어의 성능에 대하여 검증하여야 할 것이고, 이에 의해 발생하는 성능상의 문제나 단점을 보완하는 연구가 필요하다.

참고문헌

- [1] D.Culler, D. Estrin, M. Srivastava "Overview of Sensor Networks" IEEE Computer Society pp.41-49 August 2004.
- [2] Akyildiz, I.F., W. Su, Y.Shankarasubramaniam, E. Cayirci" A Survey On Sensor Networks" IEEE Communications Magazine pp.102~114 August 2002.
- [3] 장정환, 김완중, 안호현, 이세호, 정태영 "센서기반 교량 유지관리 시스템" 한국소음진동공학회 pp.602~607 2003.
- [4] 이종원, 김재동 "작용응력 측정을 위한 교량 상시감시 기법" 한국구조물진단학회 pp31~36, 1999.
- [5] C. Shen, C. Shisathapornphat, and C. JaiKeeo, "Sensor Information Networking Architecture and Application." IEEE Personal Communications." IEEE Personal Communications, Vol. 8, No. 4, pp.52~59, Aug. 2003.
- [6] 이찬경, "CBD 이해와 활용" 삼성 SDS IT Review 200.6.
- [7] T. Nieberg, S. Dulman, P.Havinga, L.v. Hoesel, and j.Wu. Collaborative algorithms for communication in wireless sensor network. In T. Basten, M. Geilen, and H. de Groot, editors, Ambient Intelligence:Impact on Embedded Systems Design, pp271~294, 2003.