
RIA 를 적용한 무선 센서네트워크의 실시간 메시지 시스템

A Real-Time Message System of Wireless Sensor Network Using Rich Internet Application

김훈*, 전영준**, 신승호***

요약 최근 웹을 기반으로 하는 소프트웨어가 발전하면서, 웹을 통한 서비스가 점차 증가하고 있다. 이것 때문에 많은 연구기관과 보고서들이 웹을 소프트웨어의 플랫폼이라고 부르고 있다. 이렇게 웹 기술이 발전하는 동안 Wireless Sensor Network(WSN)의 미들웨어와 응용 서비스는 데스크탑의 운영체제 기반에서 개발되었다. 그래서 WSN의 기술은 데스크탑 어플리케이션의 한계점을 가지고 있다. 그리고 센서 노드의 하드웨어와 밀접한 관련이 있는 강 결합(Tightly-Coupled) 구조 때문에 WSN의 통합, 응용 서비스 배포, 유지보수 관리 등에 어려움이 있다. 이와 다르게 Web은 약 결합(Loosely-Coupled)이고 개방된 시스템이기 때문에 Web Service나 Rich Internet Application(RIA)는 이러한 한계점을 해결할 수 있도록 도와준다. 특히 RIA는 웹 어플리케이션이지만 데스크탑 어플리케이션과 비슷한 사용자환경과 다양한 통신방법을 제공하기 때문에 WSN 응용서비스의 플랫폼이 될 수 있다. 본 연구에서 WSN의 연결성과 접근성을 인터넷으로 확장시키고 RIA를 적용하여 실시간 메시지 시스템을 제안한다.

Abstract As web based software has developed recently, service via web has increased gradually. For this, many research organizations and reports call web is the platform of software. While web based technologies have developed, middleware of WSN and its application service has been developed from desktop based operation system. So WSN's technology reached uppermost limit of desktop application. There's difficulty in integrating, distributing and maintaining & repairing for WSN due to a tightly-coupled structure which's related closely to the hardware of sensor nod. Unlike this, web has a loosely-coupled structure and an opened system, so web service or Rich internet application (RIA) is helpful to solve the above limits.

Especially, RIA is web application but can be the platform of WSN' application because it gives us various methods to communicate with user interface similar to desktop application. In this study, I suggest Message System for WSN Using RIA, expanding WSN's interconnectedness and accessibility to internet.

핵심어: *Wireless Sensor Network, Rich Internet Application, Web Platform, Real-Time Message System*

본 논문은 2008년 Brain Korea21 Project 학술 연구비 지원에 의하여 연구되었음.

본 논문은 산업자원부의 인천대학교 동북아 전자물류 연구센터 학술 연구비 지원에 의하여 연구되었음.

*주저자 : 인천대학교 컴퓨터공학과 석사과정 e-mail: kimhoon@incheon.ac.kr

**공동저자 : 인천대학교 컴퓨터공학과 박사과정 e-mail: 0961144@incheon.ac.kr

***교신저자 : 인천대학교 컴퓨터공학과 교수; e-mail: shin0354@incheon.ac.kr

1. 서론

1999 년 Pennine Group(Durham 대학, Keele 대학, Manchester 대학의 소프트웨어 엔지니어링 연구자 컨소시엄)은 객체지향과 컴포넌트 구조적 기반 설계 스타일의 개발방법은 소프트웨어 산업현장을 발전시키지 못할 것이라고 단언하였다[1]. 그들은 급변하는 소프트웨어 개발 패러다임에 집중하였고, 소프트웨어는 하나의 서비스 형태로 전달될 것이라고 했다. 동적인 서비스들로 구성된 서비스 모델의 소프트웨어는 전통적인 소프트웨어의 사용방법, 배포, 발전의 많은 제약을 극복하게 해준다. 이 개념을 Software as a Service(SaaS)라고 한다[1]. 그리고 Web Service 를 사용하여 이러한 개념을 구현하고 있다. 이렇게 소프트웨어가 발전하고 있는 동안에 유비쿼터스 패러다임의 대표적인 기술인 Wireless Sensor Network(WSN)의 응용서비스는 미들웨어에 의존하고 있다. 하지만 아직 상업적인 목적으로 대량 생산되어 사용되는 응용 사례를 찾아보기 힘들다. 그러나 앞으로 정보사회의 기반이 되는 표준 기술로 발전하면 다양한 응용서비스들의 출현으로 인한 대량의 센서 데이터 발생과 다양한 미들웨어가 등장할 것이다. 완전한 미들웨어는 다양한 어플리케이션들을 지원하고 조정하는 실행환경을 포함해야 한다. 그리고 데이터 통합, 제어, 관리, 효율적인 시스템 자원 사용과 같은 표준화된 시스템 서비스를 지원하여야 한다[2]. 즉, 미들웨어는 실세계 정보를 수집하고 제공하기 위하여 응용 프로그램과 하위 계층을 연결해 주는 기본 기능과 다양한 응용서비스를 제공할 수 있어야 한다. 또한 언제, 어디서, 어떤 데이터라도 통신가능하고 기존의 시스템과 통합 가능하여야 한다. 현재 다양한 응용서비스와 정보를 수집하기 위해 연구된 미들웨어는 Courgar, TinyDB, SINA, DSWare, Milan, Impala 등이 있지만 각 미들웨어 시스템의 한계점을 해결하고 통합관리 하는 것은 어려운 일이다. 네트워크 내부에 있는 미들웨어는 하드웨어와 강 결합(Tightly-Coupled)이기 때문이다. 일반적으로 센서 데이터 관리는 각종 미들웨어에서 하고 있으나, 다양한 센서 플랫폼과 대량의 센서 데이터는 표준화 되고 개방된 시스템을 기반으로 통합 관리하는 것이 필요하다. 즉, 유비쿼터스 장치에서 생성된 정보는 표준 Web 플랫폼을 사용한 응용서비스에서 처리되어야 한다. Web 은 개방적이고 국제표준이기 때문에 위 소프트웨어의 한계점을 해결할 수 있다. 현재 Web 기술은 더 발전된 형태인 Web Service 와 Rich Internet Application 등이 있다. 특히 Rich Internet Application 은 데스크탑 어플리케이션과 비슷한 유저 인터페이스를 제공해주고 비동기 통신도 제공해준다. 이러한 RIA 의 장점을 WSN 미들웨어에 적용하면 현재 개발된 미들웨어의 한계점을 해결할 수 있다. 본 연구에서

데이터 서버에 장착된 베이스 노드를 통하여 WSN 데이터를 클라이언트에게 실시간으로 제공하고, 센서 노드와 양방향통신이 가능한 RIA 기반 플랫폼을 제안하여 한계점을 해결한다. 결과적으로 운영체제와 단말장치, 시간, 장소의 제약을 해결하여 웹 브라우저 만으로 센서 데이터를 실시간으로 확인하고, 센서 노드를 제어하는 것을 목적으로 한다.

2. 관련연구

2.1 WSN 과 미들웨어

WSN 에서 각 노드의 통신을 위한 하드웨어는 Zigbee 를 구현한 RF 모듈을 사용한다. WSN 은 무선통신이 가능한 많은 센서 노드들로 구성되며, 현실세계의 아날로그 신호를 디지털 신호로 변환하여 정보를 수집한다. 이러한 센서 노드의 하드웨어는 전력을 공급하는 배터리, 아날로그 신호를 감지하는 Sensor 및 Actuator, Micro Controller, 무선통신모듈 등으로 구성되어있다. 하드웨어 모델은 Crossbow 의 Mica series 와 Intel 의 iMote, Moteiv 의 Telos 등이 있다. 마이크로 컨트롤러는 ATmega128 과 MSP430, ARM7TDMI 등을 사용하고, RF Module 은 Zigbee 통신과 Bluetooth 를 지원하는 제품이 사용된다. 그리고 하드웨어를 제어하고 운영하기 위해 작은 운영체제 프로그램을 설치한다. 대표적인 운영체제는 TinyOS, TRON 프로젝트의 T-Engine, MANTIS, Nano Qplus 등이 있다. TinyOS 는 컴포넌트 기반 운영체제이며 각각의 컴포넌트를 조립하여 응용프로그램을 작성할 수 있다[3]. T-Engine 은 약한 표준을 가진 TRON 프로젝트와 반대로 강한 표준을 가진 프로젝트이다[4]. 오픈 아키텍처로 누구든지 사용가능하며 T-Engine 에서 사용하는 임베디드 운영체제를 T-kernel 이라고 한다. T-kernel 중 nT-kernel(nano)은 센서 노드를 운영할 수 있는 최소한의 인터페이스를 가지고 있다. MANTIS 는 C 언어 방식의 API 를 지원하고 선점 스케줄링 기법, 멀티 쓰레딩, 하드웨어 추상화 등의 특징을 가진다[5]. 센서 네트워크 미들웨어는 상위 응용시스템과 센서 네트워크의 중간 부분에 위치하며, 둘 사이의 유연한 통합을 지원하는 기능을 한다. 성공적인 미들웨어가 가져야할 기능은 다음과 같다[2]. 센서 노드의 에너지, 컴퓨팅 파워, 메모리, 통신 대역폭등의 자원은 제한되어있기 때문에 효율적으로 자원을 사용하여 기본적인 센싱, 데이터 프로세싱, 통신 메커니즘을 제공해야한다. 또한 센서노드를 이동하거나 추가할 때 네트워크 성능에 영향을 주지 않는 동적인 네트워크 토폴로지 구성이 가능해야한다. 그리고 공통적인 시간과 공간의 척도를 설정하여 수집된 데이터의 정확성과 실시간성을 보장해야하고 다양한 센서 노드 하드웨어와

네트워크의 인터페이스 메커니즘을 제공해야한다. 마지막으로, 다양한 데이터 형태를 통합할 수 있어야하고 어플리케이션 특성과 네트워크의 QoS 를 지원해야한다. 이것은 센서 노드의 측정과 배치, 네트워크 적용범위, 노드의 생명주기 등이 포함된다. 응용서비스에 따라 개인정보 및 중요한 정보가 노출되지 않도록 보안 기능도 가져야 한다.

현재 연구 개발된 미들웨어는 다음과 같다[2]. 데이터베이스 기반의 TinyDB 는 센서 네트워크를 가상 데이터 베이스로 간주하고 SQL-like 언어를 사용한다[6]. 그러나 스페닝트리를 이용한 네트워크 확장을 유지해야 하기 때문에 모든 노드에게 쿼리를 전송해야한다. 또한 새로운 기능 추가시 모든 노드의 질의처리기 수정을 요구하기 때문에 개방성은 오직 부분적으로 지원한다[2]. 가상머신 기반의 MATE 는 TinyOS 를 사용하고, Byte Code Interpreter 를 가지고 있다[7]. 그리고 "Capsule"이라 불리는 프로그램이 센서 노드에 탑재되어 하나의 작업을 수행한다. MATE 는 Duty-Cycle 이 낮은 어플리케이션에 적합하다. 그러므로 복잡한 어플리케이션의 경우, 명령어 해석의 과부하 때문에 성능을 낭비한다. 또한 사용하기 힘든 아키텍처와 바이트코드로 구성된다. 그래서 어플리케이션 개발을 위해 고수준의 언어와 프로그래밍 모델이 필요하다[2]. DSWare 는 이벤트 신청, 이벤트 탐지, 데이터 저장, 그룹관리 등의 여러 서비스들을 응용계층에서 추상화했다[8]. 그래서 저수준의 복잡한 기술과 상관없이 표준 SQL 을 사용하여 데이터를 얻을 수 있다. 그러나 DSWare 는 이기종과 이동성을 위한 해결방법을 제공하지 않는다[2].

2.2 Rich Internet Application

Web Application Server(WAS)로 사용된 Tomcat 은 단순히 JSP 나 HTTP 문서를 제공하는 것 이상의 기능을 한다. 웹은 처음 HTML 형태로 포맷된 정적 링크 페이지로서 연구자들이 문서를 공유하는데 사용됐다. 그 후 웹 페이지는 복잡한 구조의 텍스트와 그림, 오디오, 비디오 스트림을 재생할 수 있는 프로그램을 플러그인하는 형태로 발전하였다[9]. 그리고 스크립트기능과 사용자 상호작용이 개선되는 동안 클라이언트가 웹서버에게 웹페이지를 요청하고 응답을 받는 동기식 기반 구조는 변하지 않았다. 그러나 최근에, 웹 개발자들은 Rich Internet Application 이라는 새로운 모델을 발전시켜왔다. Rich Internet Application(RIA)은 전통적인 데스크탑 어플리케이션의 모양과 기능을 가진 웹 어플리케이션이다[10]. "Rich Internet Application"이라는 용어는 2002 년 3 월 Macromedia 에서 발표된 백서에서 소개되었다. 전통적인 웹 어플리케이션은 Thin-Client 와 Server 에서 작동했었기 때문에 이 시스템의 모든 프로세싱은 Server 에서 처리했다. 즉, 클라이언트가 모든 데이터를 Server 에게 전송하고 처리된 데이터를 다시 응답받는 구조였다. RIA 는 클라이언트 엔진이라 불리는 것을 이용해

동기식 통신방식뿐만 아니라 비동기식 통신도 가능하게 한다. 클라이언트 엔진을 사용한 웹 어플리케이션의 비동기식 응답은 Foreground 와 Background 두 가지 과정으로 이루어진다. 브라우저에 플러그인 형태로 설치되는 ActiveX, AJAX 에 사용되는 XmlHttpRequest, Java runtime, Adobe 의 Flash, MS 의 Sliverlight 등이 클라이언트 엔진에 속한다. 이처럼 RIA 는 데스크탑 어플리케이션의 장점을 가지고 있는 웹 어플리케이션이다. 그래서 데스크탑 어플리케이션처럼 설치할 필요가 없고, 업그레이드 및 업데이트가 자동적이다. 가장 매력적인 장점은 인터넷 접속만 가능하다면 컴퓨터와 운영체제에 독립적이다. 그리고 기존 HTML 에서 얻을 수 없는 뛰어난 유저인터페이스를 제공한다. 또한 전통적인 웹의 동기식 통신보다 비동기식 통신을 사용하여 네트워크 효율성과 응답성을 높인다. 이것은 항상 페이지 전체를 서버와 주고받는 것이 아니라 필요한 데이터만 주고받을 수 있게 한다. 그래서 네트워크 트래픽을 줄이고 사용자에게 빠른 응답성을 제공한다.

3. WSN 을 위한 RIA 실행환경과 실시간 메시지 시스템

본 연구에서 사용한 센서 노드의 모델은 Micaz 이다. 실험은 인터페이스 기능을 가진 Adapter 와 데이터서버인 Web Server, 그리고 응용 서비스인 RIA 로 나뉘어 진다. RIA 를 구현하기 위하여 Adobe Flex 를 사용하였다. 쉬운 이해를 위해 제안한 시스템을 컴퓨터로 묘사하였다. Adapter 는 센서 노드 모델에 따라 다른 연결성을 유지하기 위해 인터페이스 기능이 있고 키보드와 같은 입력장치 역할을 한다. 그리고 Web Server 에게 WSN 데이터를 HTTP 로 송수신한다. Web Server 는 저장 장치와 통신제어 역할을 한다. 또한 RIA 를 실행할 수 있게 해준다. 마지막으로 RIA 는 센서 데이터의 연산과 출력을 담당한다. 컴퓨팅 장치처럼 입력, 저장, 연산, 출력으로 구성된다. 데이터가 이동하는 Ethernet, LAN 과 Serial Port 를 데이터 버스로 가정하였다. 그리고 이 연구에서 플랫폼은 WSN 을 위한 웹 어플리케이션의 실행환경이라고 정의한다.

3.1 Gateway for WSN

WSN 에서 각 노드의 통신을 위한 하드웨어는 Zigbee 를 구현한 RF 모듈을 사용한다. 그리고 Intel 의 iMote 는 Bluetooth 를 사용하기도 한다. 노드들의 무선통신을 통하여 수집된 데이터는 싱크 노드 또는 베이스 노드가 자원이 풍부한 상위 프로그램이나 장치에게 보낸다. 베이스, 싱크노드가 상위 컴퓨팅 장치에게 수집된 데이터를 보낼 때 Mica Series, Telos 등의 모델은 RS-232 통신이나 USB 통신으로 데이터를 보낸다. 또한 임베디드 보드를 통하게 되면 LAN 이 이용되기도 한다. 이렇게 상위 컴퓨팅 장치에게 수집된 데이터를 송신할 때 여러 가지 장치의

데이터를 통합 하기 위한 인터페이스가 필요하다. RS-232 통신의 경우 각각의 WSN 모델마다 Baud Rate, Parity Bit, Error Bit 설정이나 센서 노드에 설치한 프로그램에 따라 메시지구조가 다르다. 그리고 센서 노드의 운영체제에 따라 메시지 구조가 다르다. 실험에서는 RS-232 통신으로 받아온 바이트 스트림을 Web Server 에게 Request 한다. Jakarta Project Open Source 의 HTTPClient 컴포넌트를 사용하여 Web Server 에게 센서 데이터를 Request 한다. Post 방식으로 XML 문서를 전송할 수 있고, RS-232 에서 입력된 16 진수 데이터 스트림을 그대로 Request 할 수 있다. 또한 Web Server 와 양방향 통신을 하기 위해 HTTP Server 를 구현하였다. HTTP Server 는 ServerSocket 클래스를 사용하여 HTTP Request 의 바이트 스트림을 받아온다[3]. 바이트 스트림은 RS-232 커넥터를 통해 베이스 노드에게 송신된다. Adapter 는 센서 노드의 하드웨어와 네트워크에 대한 인터페이스 메커니즘을 구현한다. 결과적으로 센서 노드의 데이터를 인터넷에 접속시키는 Gateway 역할을 한다.

3.2 Web Application Server

Web Application Server(WAS)로 사용된 Tomcat 은 단순히 JSP 나 HTTP 문서를 제공하는 것 이상의 기능을 한다. My-SQL 을 이용한 Data Server 와 클라이언트에게 Rich Internet Application 응용서비스를 제공한다. 그리고 Server-Side Data Push 기능을 가진다. 마지막으로 Adapter 의 Request 에 대한 Response 를 처리한다. Request 에 대한 연산은 크게 두 가지이다. Message Service 를 사용하여 현재 접속된 모든 클라이언트에게 데이터를 전송하는 것과 데이터베이스에 저장하는 것이다. Tomcat 은 JSP Container 이다. 그러나 Adobe 는 Tomcat 에서 Flex 를 실행할 수 있도록 SDK 를 제공한다. 그 중 Message Service 는 센서 데이터를 Server-Side Data Push 하는데 사용된다. Flex 는 JMS(Java Message Service)를 기반으로 더 쉽고 확장된 Message Service 를 제공한다. JMS 는 Queue 와 Topic 방법이 있다[11]. 실험에 사용된 Topic 방법은 데이터를 송신하는 Publisher 와 데이터를 수신하는 Subscriber 로 구성된다. 그리고 연결과 관련된 Connection Factory 가 있다. Message 는 Header, Property, Body 로 구성된다. Body 에는 어떠한 Data Type 도 넣을 수 있다. Message 의 종류는 TextMessage, BytesMessage, StreamMessage, ObjectMessage 등이 있다. Flex 의 Message Service 를 위해 Message-config 에서 Message 의 Destination 과 Channel 을 설정한다. Java 는 JMS 의 JNDI 를 이용하여 Connection Factory, Queue, Topic 의 환경을 설정하지만 Flex 는 XML 문서 수정으로 설정한다. Destination 은 클라이언트 프로그램이 어디로, 어떠한 방법으로 통신할 것인지 결정한다. 그리고 Channel 은 통신에 사용할 프로토콜을 결정한다. 프로토콜은 Adobe 에서 개발한 RTMP(Real-Time Messaging Protocol)와 AMF(Action Message Format)가 사용된다. 위 과정은 Rich Internet

Application 의 통신 설정이다. Gateway 의 Request 가 올 때마다 센서 노드의 데이터 스트림을 StreamMessage 로 전송한다. Message 는 Message Oriented Middleware(MOM)를 통하여 클라이언트에게 전송된다. 그리고 각각의 센서 네트워크에서 보낸 데이터 스트림은 서로 다른 목적지로 전송된다. 결과적으로 서로 다른 센서 네트워크의 통신경로가 생성된다. Gateway 의 Request 는 센서 데이터가 발생했다는 Event 와 같다. 이것은 센서 데이터의 실시간성을 제공한다. Request 가 발생할 때 Message Producer 는 응용서비스에게 Data 를 전송한다. Figure 5 는 Message Service 를 나타낸다. 이 때 데이터의 통합 및 관리를 위해 표준 SQL 을 사용하는 데이터베이스에 저장된다. 저장되는 데이터는 가공하지 않은 Byte Stream 과 Meta Data 이다. 이 데이터는 Rich Internet Application 의 응용서비스에서 연산이 이루어진다. <Figure 1>는 강한 결합(Tightly-Coupled)과 약한 결합(Loosely-Coupled)으로 분리된 컴포넌트 구성을 보여준다.

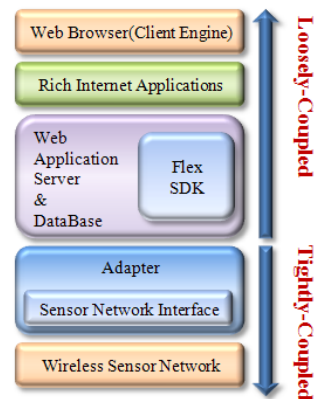


Figure 1. 강한 결합구조에서 약한 결합구조로 구성

3.3 WSN 을 위한 실시간 메시지 RIA

FLEX 는 웹에서 비동기 통신을 제공하기 위하여 JMS 기술 기반 메시지 서비스를 제공한다. 관련연구에서 JMS 는 2 가지 통신방식을 가지고 있다고 설명하였다. WSN 을 위한 실시간 메시지 서비스는 센서 네트워크 계층에서 발생한 데이터를 하나의 메시지로 변환하여 실시간으로 전송해 준다. <Figure 2>는 실시간 메시지 서비스를 실행한 웹 브라우저 화면이다.

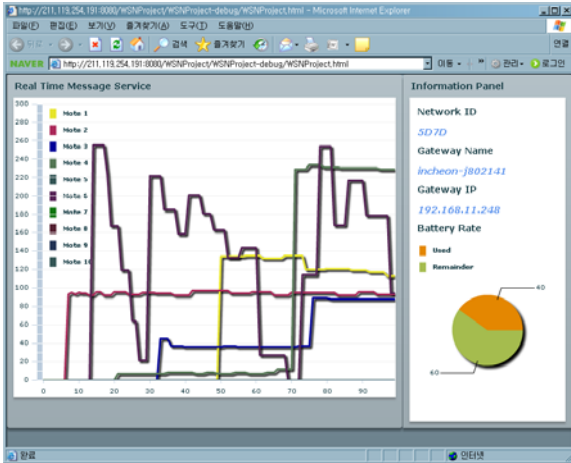


Figure 2. 실시간 메시지 RIA

Gateway 에서 HTTP 로 요청된 바이트 스트림을 WAS 에서 JMS 기반 메시지 서비스를 통하여 클라이언트 엔드까지 Topic 방식으로 전송된다. Topic 방식은 메시지 발행자와 구독자방식을 사용한다. 발행자는 구독자에게만 메시지를 전송하는데 이 방법은 멀티캐스트와 유사하다. 즉 WAS 에 접속되어 실시간 메시지 서비스 RIA 를 실행한 웹 브라우저에게만 메시지가 전송된다. 발행자를 만들기 위해 Producer 라는 객체를 사용하고 구독자를 만들기 위해 Consumer 객체를 사용한다. Consumer 가 Producer 의 메시지를 수신하기 위해서 Producer 의 채널 정보만 필요하다. 이 기술을 정확히 Server-Push 또는 Data-Push 라고 한다. 기존의 웹 서버와 웹 브라우저는 동기통신을 하지만 이 기술은 비동기 통신을 제공해 주기 때문에 <Figure 2>처럼 실시간 그래프 표현이 가능한 것이다. 그래프의 값은 특정 ID 를 가진 센서 노드의 센싱 데이터를 나타낸다. WAS 에 접속된 모든 클라이언트에게 같은 메시지가 전송되기 때문에 <Figure3>처럼 같은 그래프화면을 볼 수 있다.

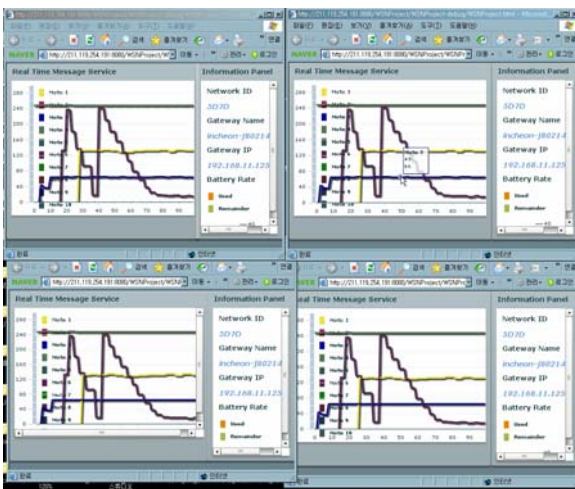


Figure 3. 다중 실시간 메시지 RIA

3.4 실험 결과

TinyDB, Cougar 등은 센서 노드 자체를 하나의 데이터베이스 속성으로 가정한다. 그래서 센서의 데이터를 조회할 때 매번 노드에게 데이터를 요청해야 한다.

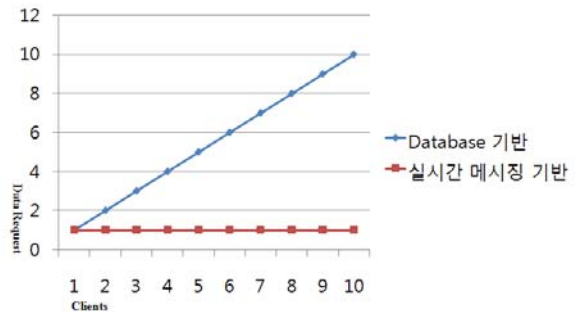


Figure 4. 다중 실시간 메시지 RIA

그리고 데이터를 요구한 클라이언트가 많으면 조회 쿼리 수행은 더 늘어나게 된다. 제안한 시스템은 실시간 메시지 시스템으로 한번 입력된 데이터는 접속된 클라이언트에게 모두 전송되며 Server-Side 데이터베이스에 저장된다. 그러므로 <Figure 4>처럼 클라이언트가 늘어나도 센서의 메시지는 한번만 요청하게 된다. 클라이언트가 데이터베이스 기반 시스템에서 센서 네트워크의 데이터를 조회하면 각 클라이언트마다 SQL-like 명령을 실행하게 된다. 그러므로 클라이언트요청의 증가는 센서 네트워크 접근빈도나 데이터베이스 접근빈도의 정비례를 뜻한다. 그리고 수집된 데이터는 센서 네트워크 계층에 다시 접근하지 않고 Server-Side 데이터베이스에서 요청하여 분석할 수 있다. 결과적으로 데이터의 분석은 RIA 로 개발된 응용서비스에서 실행된다.

4. 결론

이 연구는 미들웨어의 기본 기능인 WSN 계층과 응용서비스를 연결하기 위한 실행환경을 제안하여 실시간 메시지 시스템을 구축하였다. 그러나 기존의 데스크탑 어플리케이션 미들웨어가 아닌 RIA 기반 플랫폼과 응용서비스를 제안하였다. 이것은 운영체제와 하드웨어에 의존하지 않고 웹을 사용한 실행환경을 제공한다. RIA 기반 WSN 의 응용 서비스는 웹 어플리케이션이다. 그래서 다음과 같은 장점이 있다. 웹 브라우저를 사용하여 언제 어디서라도 WSN 에 접속 가능하다. Gateway 통하여 각각의 센서 노드와 양방향 통신을 할 수 있기 때문에 웹 브라우저로 센서 노드를 제어할 수 있고 실시간 모니터링이 가능하다. 이것은 WSN 의 응용 서비스가 약한 결합(Loosely-Coupled) 시스템에서 실행되기 때문이다. 그러나 향후 실시간 메시지의 응답시간을 측정하여 데이터베이스 기반 시스템 보다 개선 되고 자원 사용 부분에서 효율적임을 나타내는 연구가 필요하다.

참고문헌

- [1] M. Turner, D. Budgen and O.P. Brereton, "Turning Software into a Service", IEEE Computer, 36(10) October 2003, pp 38-44.
- [2] S. Hadim and N. Mohamed, "Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks," in IEEE Distributed Systems Online, vol. 7, no. 3, 2006.
- [3] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for wireless sensor networks", Ambient Intelligence, Springer-Verlag, 2004.
- [4] Ken Sakamura, "Challenges in the age of ubiquitous computing: a case study of T-Engine, an open development platform for embedded systems", Proceedings of the 28th international conference on Software engineering, Shanghai, 2006, pp.713-720
- [5] SHAH BHATTI, JAMES CARLSON, "MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms", Mobile Networks and Applications volume10, Kluwer Academic Publishers, Netherlands, 2005, pp.563-579
- [6] S. R. Madden, M. J. Franklin, and J. M. Hellerstein, "TinyDB: An Acquisitional Query Processing System for Sensor System Networks," ACM Trans. Database Systems, Vol. 30, No. 1, 2005, pp. 122-173.
- [7] P. Levis and D. Culler, "Mate: A Tiny Virtual Machine for Sensor Networks," Proc. 10th Int'l Conf. Architectural Support for Programming Languages and Operating System(ASPLOS-X), ACM Press, 2002, pp.85-95.
- [8] S. Li, S. Son, and J. Stankovic, "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks," Proc. 2nd Int'l Workshop Information Processing in Sensor Networks(ISPN 03), LNCS 2634, 2003, pp. 502-517.
- [9] Chris Loosley, Rich Internet Applications: Design, Measurement, and Management Challenges, Keynote Systems, 2006
- [10] Budi Kurniawan, Paul Deck, How Tomcat Works :a Guide to Developing Your Own Java Servlet Container, BrainySoftware, 2004 .04
- [11] Yakov Fain, Dr.Victor Rasputnis, Anatole Tartakovsky, Rich Internet Application with Adobe Flex & Java, Sys-Con Media, USA, 2007