한국산학기술학회논문지
Vol. 9, No. 5, pp. 1289-1295, 2008

센서 네트워크 기반의 Pull 및 Push 서비스 연구

김규리[1], 김도현[2*], 변영철[2]

A Study of Pull and Push Service Based on Sensor Networks

Kyu-Li Kim[1], Do-Hyeun Kim[2*] and Yung-Cheol Byun[2]

요 약 최근에 다양한 유비쿼터스 서비스는 Push와 Pull 서비스 형태로 제공되고 있으며, 이와 같은 Pull/Push 서비스를 제공하기 위한 OpenAPI(Application Program Interface)에 대한 연구가 진행되고 있다. 하지만 아직 Push와 Pull 서비스와 이를 지원하기 위한 OpenAPI에 대한 연구가 아직 미비하다. 이에 본 논문에서는 Pull 서비스와 Push 서비스를 이용하여 센서 네트워크 기반의 응용 인터페이스를 제공하는 Push 서비스와 Pull 서비스 모델을 제시하고, Push 서비스와 Pull 서비스를 제공하는 OpenAPI를 설계하고 구현한다. 이를 위하여 웹 서비스 기반의 Pull 서비스 구조를 제시하고, .Net 프레임워크 기반의 원격 서비스(Remote Service)를 이용하여 센서 네트워크에 수집된 온도, 습도 등의 상황 데이터를 Pull 서비스 형태로 제공하는 OpenAPI를 설계하고 구현한다. 더불어 TCP/IP 소켓 인터페이스를 이용한 Push 서비스 구조를 제안하고, 이 서비스를 제공하는 OpenAPI를 설계하고 구현한다. 이를 통해서 기존의 특정 데이터베이스 중심의 센서 네트워크의 폐쇄적인 응용 인터페이스를 개방적인 표준 인터페이스로 전환하여 사용자가 여러 곳에 널려 있는 많은 상황 데이터를 쉽게 접근하여 확인할 것으로 기대한다.

Abstract Recently, it is progressing a study for supporting various application services using OpenAPI(Application Program Interface). But it is not enough a study related OpenAPI(Open Application Interface) to access many collected context data of sensor networks for ubiquitous application services. Therefore, this paper presents Pull/Push service model based on sensor networks, and implements OpenAPI for Pull/Push application services. And, we design and implement OpenAPI using web service for Pull application services. This Pull OpenAPI supports users the context data of temperature and humidity using the remote service based on .Net framework in sensor networks. And, we design and implement OpenAPI using TCP/IP socket interface for Push application services in sensor network. This Push OpenAPI provides users a state of temperature, humidity collecting in sensor networks. Consequently, user can develop easily various application services as supporting OpenAPI instead of closed application interface of sensor networks based on existed specific database.

Key Words : Sensor Network, Web Service, 개방형 표준 인터페이스

1. 서론

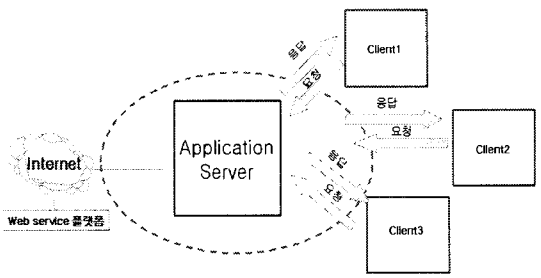
유비쿼터스 컴퓨팅 기술은 빠르게 발전하고 있다. 이에 따라 유비쿼터스 컴퓨팅 기술은 센서 네트워크에 저전력과 신뢰성 있는 데이터 전송 등의 시스템 개발과 더불어 여러 응용 분야에 수집된 상황 데이터를 이용하여 다양한 응용 서비스를 제공하기 위한 연구가 진행되고 있다. 그러나 아직 센서 네트워크 기반의 미들웨어나 응용들은 폐쇄적인 인터페이스를 제공하고 있어 센서 네트워크에서 수집된 많은 상황 데이터에 접근하는 데 한계가 있다[1,2]. 따라서 기존의 특정 데이터베이스 중심의 센서 네트워크의 폐쇄적인 응용 인터페이스를 개방적인 표준 인터페이스로 전환하여 사용자가 여러 곳에 널려 있는 많은 상황 데이터를 쉽게 접근 할 수 있는 서비스가

본 논문은 2007년도 산업자원부 지방기술혁신사업(지자체주도 연구개발 사업)의 지원과 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업((IITA-2008-C1090-0801-0040)의 연구결과로 수행되었음.
[1]경희대학교 산학협력단(기술연구원)
[2]제주대학교 통신컴퓨터공학부
*교신저자: 김도현(kimdh@cheju.ac.kr)
접수일 08년 09월 04일 수정일 08년 10월 10일 게재확정일 08년 10월 16일

필요하다. 이를 위하여 본 논문에서는 센서 네트워크 기반의 Pull/Push 서비스 모델을 제시하고, 웹 서비스와 TCP/IP 소켓 인터페이스를 이용하여 Pull/Push 서비스를 지원하는 OpenAPI(Application Program Interface)를 설계하고 구현한다. 이를 위해 기본적인 센서 네트워크 기반의 Pull 서비스 모델과 웹 서비스를 이용한 Pull 서비스 구조를 제시하고, 더불어 기본적인 센서 네트워크 기반의 Push 서비스 모델과 TCP/IP 소켓 인터페이스를 이용한 Push 서비스 구조를 제안한다. 그리고 센서 네트워크에 수집된 온도, 습도 등의 상황 데이터를 요청에 따라 제공하는 Pull 서비스를 지원하기 위해 .Net 프레임워크 기반의 원격 서비스(Remote Service)를 이용하여 웹 서비스 기반의 OpenAPI를 개발한다. 또한, 서버가 실시간으로 클라이언트에게 상황 데이터를 전달하는 Push 서비스를 제공하기 위해 TCP/IP 소켓 인터페이스 기반의 OpenAPI를 구현하여 제시한 Pull/Push 모델과 구조를 검증한다. 본 논문의 구성은 2장 Pull과 Push 서비스 개념과 기존의 센서 네트워크 기반의 응용 서비스를 고찰하고, 3장에서는 센서 네트워크 기반의 Pull/Push 서비스 모델을 살펴보고, 4장에서는 Pull/Push 서비스를 제공하는 OpenAPI를 구현한다. 마지막으로 5장에서는 결론을 맺는다.

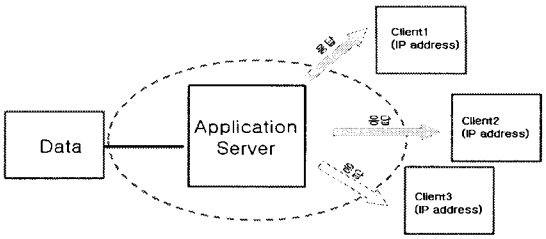
2. 관련연구

PULL 서비스는 일반적인 서버에서 클라이언트 모델에서는 클라이언트의 요청을 서버가 응답하는 형태로 서비스가 제공되고, 클라이언트가 웹을 통해 콘텐츠를 요청하고 서버를 부터 받는다. 그림 1은 Pull 서비스의 구조를 나타내고 있다.



[그림 1] 일반적인 Pull 서비스의 구조

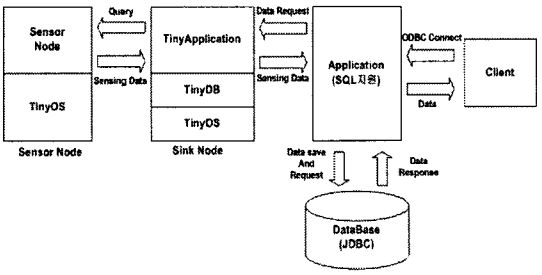
Push 서비스는 클라이언트의 특정한 요청 없이 서버 측에서 클라이언트로 콘텐츠를 전송하는 방식으로 정의한다. 그림 2에서는 Push 서비스의 구조를 나타내고 있다. Push 서비스에서는 서버에서 클라이언트에서 접속을 할 경우, 클라이언트 요청 없이 여러 개의 클라이언트 주소로 데이터의 정보를 보내게 된다.



[그림 2] 일반적인 Push 서비스의 구조

.

센서 네트워크는 센서 노드와 계산, 센싱, 통신을 하는 네트워크로써, 특정 지역에서 센서 노드들 설치하여 주변 정보 또는 상황 정보를 획득하고, 수집된 상황 정보를 활용한다. 대표적인 센서 네트워크에 사용되는 하드웨어로 Mote 시리즈가 있으며, 이 하드웨어에 미국 버클리 대학의 TinyOS 운영체제와 TinyDB가 개발되어 이용되고 있다. TinyDB에서는 SQL과 같은 질의를 위한 인터페이스를 이용하여 센서 네트워크으로 부터 데이터를 수집하여 싱크노드와 서버로 전송 작업을 수행한다.



[그림 3] 기존의 센서 네트워크 기반의 응용 서비스 구조

그림 3은 TinyDB를 이용하여 센서 네트워크 기반의 실시간 정보 서비스의 구성을 나타내고 있다. 센서 노드는 센싱 데이터 질의를 통해 상황 데이터를 수집하여 싱크노드와 데이터베이스 서버로 전달된다. 서버측 응용프로그램에서는 실시간 상황 데이터를 JDBC와 SQL 명령을 통해 데이터베이스에 저장된다. 클라이언트 응용프로그램은 ODBC를 이용하여 SQL 데이터베이스에 실시간으로 저장되는 센싱 데이터를 수신하여 사용자에게 보여준다. 이 프로그램은 센서 노드들에 의해 데이터가 저장된 데이터베이스 서버로부터 온도, 시간 정보, 습도 등의 가장 최근의 값을 제공한다. 위의 응용 서비스 구조에서는 클라이언트가 필요할 때마다 저장된 데이터베이스에 접근하여 상황 데이터를 읽어가는 방식이다. 이와같은 방

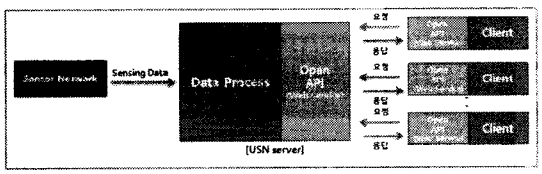
식으로는 다수의 사용자에게 실시간 상황 데이터를 전달하기가 어려우며, 사용자 응용 프로그램에서 구간 질의, 실시간 이벤트 처리 등의 다양한 응용 서비스 유형을 제공하기 힘들다[2].

3. 센서 네트워크 기반의 Pull/Push 서비스 모델 및 구조

본 절에서는 센서 네트워크 기반의 기본적인 Pull/Push 서비스 모델을 제시하고, 웹 서비스를 이용한 Pull 서비스 구조를 제시하고, TCP/IP 소켓 인터페이스를 이용한 Push 서비스 구조를 제안한다.

3.1 센서 네트워크 기반의 Pull 서비스 모델

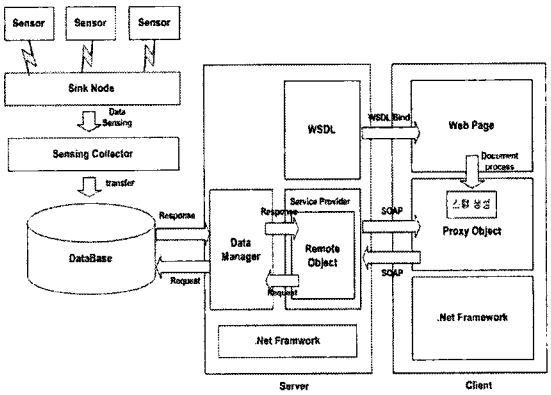
센서 네트워크에서 다수의 클라이언트가 여러 곳에 널려 있는 많은 상황 데이터를 쉽게 접근할 수 있도록 하는 Pull 서비스 모델을 제시한다. Pull 서비스 모델은 그림 4와 같이 센서 네트워크, USN 서버, 클라이언트 등의 세 부분으로 구분된다. 센서 네트워크에서는 온도, 습도 등의 상황 데이터를 수집하여 이를 데이터베이스로 전송하고, 서버에서는 온도, 습도 등의 여러 데이터를 데이터베이스나 파일에 저장한다. 클라이언트가 데이터를 요청할 경우 OpenAPI를 이용하여 클라이언트에게 요청에 대한 상황 데이터를 전달한다.



[그림 4] 센서 네트워크 기반의 Pull 서비스 모델

3.2 웹 서비스를 이용한 Pull 서비스 구조

Pull 서비스를 제공하기 위해 .Net 프레임워크를 이용해서 센서 네트워크 기반의 Pull 서비스 구조를 보여주고 있다. Pull 서비스의 구조는 센서 네트워크, 데이터베이스 및 서버, 클라이언트로 구성되며, 센서 네트워크는 온도, 습도 등의 상황 데이터를 수집한다. 데이터베이스에서는 수집된 온도, 습도 등의 센싱된 데이터를 저장한다. 서버는 .Net 프레임워크 기반으로 응용을 이용하여 상황 데이터를 처리하고 웹 서비스를 제공하기 위해 데이터 관리자(data manager), 서비스 제공자(service provider), WSDL로 이루어져있다. 데이터 관리자에서는 센서 네트워크에서 수집된 상황 데이터를 서비스 제공자와 클라이언트가 요청할 경우 서버와 클라이언트 간의 데이터를 접근 할 수 있도록 관리한다. 서비스 제공자내에 원격 오브젝트(remote object)는 원격 서비스를 하기 위해서 사용되며, 클라이언트와 통신하기 위해서 서버 채널(server channel)을 제공한다. 서버 채널에서는 전송되는 메시지들은 적절한 부호화를 거쳐 통신상에서 전송될 데이터로 변환된다. 이때, 서비스 제공자 내의 SOAP 변환기(Fommater)에서는 부호화를 담당하고, 상황 데이터를 XML 형식의 SOAP 메시지로 변환한다.



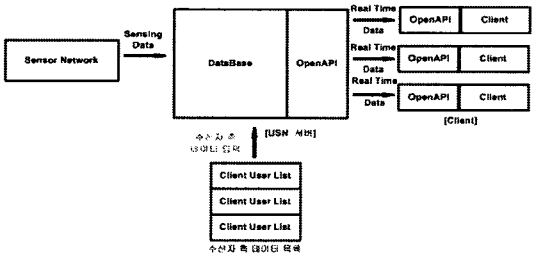
[그림 5] 웹 서비스 기반의 Pull 서비스 구조

그림 5는 센서 네트워크 기반의 Pull 서비스 모델을 지원하기 위한 .Net 프레임워크을 이용한 웹 서비스 기반의 Pull 서비스 구조를 보여주고 있다. 그리고 클라이언트에서는 서버에서 WSDL로 부호화 된 XML 상황 데이터를 보고 동적으로 프록시를 생성하여 XML 상황 데이터를 웹으로 배포된다. 이때 클라이언트에서는 원격 오브젝트를 참조할 수 있는 프록시 오브젝트(proxy object)가 필요하다. WSDL 문서에서 생성된 코드는 원격 오브젝트에 바인딩하여 SOAP 메시지를 생성하고, 전송시키는 스텁 역할을 맡게 된다. 원격 오브젝트에서는 클라이언트와 서버 사이를 넘나드는 참조 값을 이용하게 되는데, 클라이언트에서는 원격 서비스를 제공하기 위해 프록시(proxy)를 통하여 원격 참조를 수행하고, 클라이언트 채널을 가진다. 즉, 서버는 SOAP 변환기를 통해서 부호화된 XML 형식의 상황 데이터를 클라이언트 채널을 통해 사용자에게 웹 형식으로 제공한다[3-5].

3.3 센서 네트워크 기반의 Push 서비스 모델

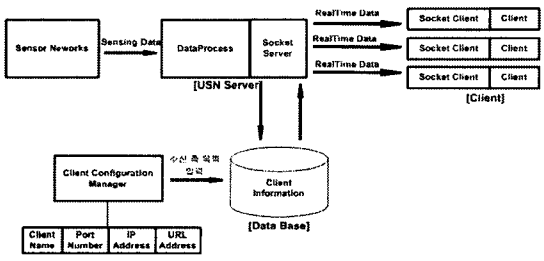
Push 서비스 모델의 구성은 센서 네트워크, USN 서버, 수신자 측 데이터 목록과 클라이언트로 이루어진다. 그림 6은 Push 서비스 모델을 나타내고 있다. Push 서비스 모델에서는 클라이언트가 요청이 없는 긴급한 환경에서도

사용자에게 상황 데이터를 전달하기 위하여 미리 사용자 정보를 서버에 저장한다. 만약 센서 네트워크로부터 위험하거나 긴급을 요하는 상황 데이터를 서버가 수신할 경우 사용자 정보를 이용하여 클라이언트에 데이터를 전달한다.



[그림 6] 센서 네트워크 기반의 Push 서비스 모델

센서 네트워크 기반의 Push 서비스를 제공하기 위하여 그림 6과 같이 TCP/IP 소켓 인터페이스를 이용한 Push 서비스 구조를 제시한다. 여기서 Push 서비스를 제공할 사용자 정보를 미리 저장한다. Push 서비스 구조는 센서 네트워크, USN 서버, 클라이언트 구성 관리자(Client Configuration Manager), 클라이언트 정보를 저장하는 데이터베이스(클라이언트 목록), 클라이언트 등으로 구성된다. 클라이언트 구성 관리자에서는 사용자로부터 클라이언트의 이름, 포트번호, IP 주소, URL 주소를 입력받고, 데이터베이스에 저장한다.



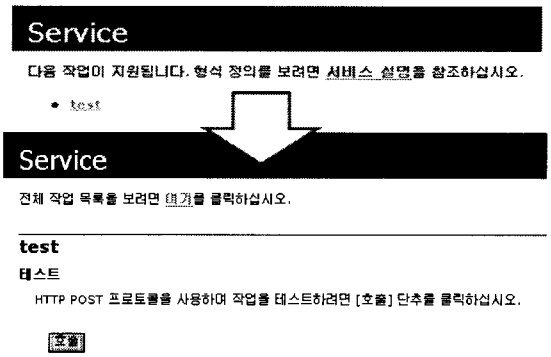
[그림 7] TCP/IP 소켓 인터페이스를 이용한 Push 서비스 구조

그리고 데이터베이스에서는 클라이언트 이름, 포트 번호, IP 주소, URL 주소의 정보를 갖고 있다가 서버로부터 사용자 정보 요청이 있을 경우, 관련 정보를 전달한다. 긴급한 상황에 해당하는 이벤트가 발생할 경우 서버에서는 클라이언트의 IP 주소, 포트번호를 확인하고, TCP/IP 소켓 인터페이스를 이용하여 클라이언트에게 상황 데이터를 전송한다.

4. Pull/Push 서비스를 위한 OpenAPI 구현

4.1 웹 서비스를 이용한 Pull 서비스의 OpenAPI

대표적인 웹 서비스를 제공하고 있는 마이크로소프트웨어 사의 .Net 프레임워크의 원격 서비스를 이용하여 Pull 서비스의 OpenAPI를 C# 컴퓨터 프로그램 언어를 사용하여 구현한다[6]. 그림 8은 웹 서비스의 메서드 호출 동작을 확인하기 위해 실험한 결과이며, 웹 서비스를 위한 메서드를 호출하기 위해 .Net 프레임워크에서 제공하는 ADO.NET의 데이터 셋(data set)과 데이터 어댑터를 이용하여 원격 데이터베이스 질의와 연결(connection)을 실험한다.



[그림 8] .Net 프레임워크에서의 웹 서비스의 원격 메서드 호출 실험

여기서 데이터 어댑터에서는 데이터베이스의 원격 연결과 데이터 셋을 생성하는 역할을 담당하고 있다. 데이터 어댑터에서는 SQL과 연결하여 메서드를 호출하기 위해 연결(connection) 객체를 생성하고, 데이터베이스에 연결하여 SQL 질의를 수행한다. 그리고 데이터 어댑터에서는 데이터 테이블을 생성한 데이터 셋으로부터 데이터를 받기 위해 원격 메서드를 호출하여 데이터 셋에 입력된 레코드의 정보와 연결 객체의 상태를 확인한다.

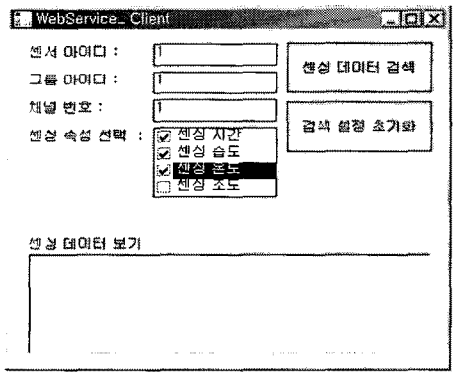
```
<xs:element name="SD_SI_Code" type="xs:long" minOccurs="0" />
<xs:element name="SD_Time" type="xs:dateTime" minOccurs="0" />
<xs:element name="SD_Humidity" type="xs:double" minOccurs="0" />
<xs:element name="SD_Temperature" type="xs:double" minOccurs="0" />
<xs:element name="SD_Intensity" type="xs:double" minOccurs="0" />
<xs:element name="SD_AirPressure" type="xs:double" minOccurs="0" />
<xs:element name="SD_WindVelocity" type="xs:double" minOccurs="0" />
<xs:element name="SD_RainFall" type="xs:double" minOccurs="0" />
<xs:element name="SD_SnowFall" type="xs:double" minOccurs="0" />
<xs:element name="SD_Voltage" type="xs:double" minOccurs="0" />
```

[그림 9] 웹 서비스의 XML 테이블

```
- <Table diffgr:id="Table1" msdata:rowOrder="0">
    <SD_SI_Code>0</SD_SI_Code>
    <SD_Time>2007-10-28T12:15:17.14+09:00</SD_Time>
    <SD_Humidity>52.262802</SD_Humidity>
    <SD_Temperature>24</SD_Temperature>
    <SD_Intensity>43.21289</SD_Intensity>
    <SD_AirPressure>0</SD_AirPressure>
    <SD_WindVelocity>0</SD_WindVelocity>
    <SD_RainFall>0</SD_RainFall>
    <SD_SnowFall>0</SD_SnowFall>
    <SD_Voltage>2.9992676</SD_Voltage>
  </Table>
```

[그림 10] 웹 서비스의 XML 테이블 데이터 속성

웹 서비스를 제공하기 위해 .Net 프레임워크에서 XML 인코드 된 상황 데이터를 전달하기 위해 정의한 XML 테이블을 그림 9에서 보여주고 있다. 그리고 이 XML 테이블을 SOAP을 이용하여 호출할 경우HTTP POST 프로토콜을 사용하여 그림 10과 같이 원격에서 웹 서비스의 상황 데이터를 획득할 수 있다. 예를 들어 제공되는 정보는 센서 코드(sensor code), 시간(time), 습도(humidity), 온도(temperature), 밀도(intensity), 기압(air pressure), 풍속(wind velocity), 강우량(rain fall), 강설량(snow fall), 전압(voltage) 등이 될 수 있으며, 이들 정보와 더불어 데이터 속성 값을 제공한다.

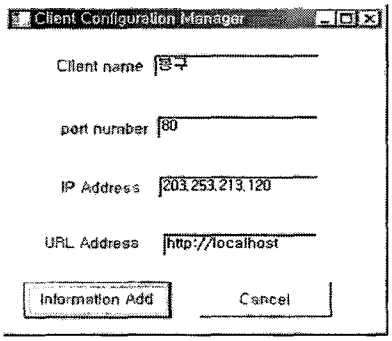


[그림 11] 웹 서비스 기반의 Pull 서비스의 클라이언트 화면

그림 11은 웹 서비스 기반의 Pull 서비스를 제공받는 클라이언트 화면을 보여주고 있다. 여기서 센서 식별자, 그룹 식별자, 채널번호를 선택할 수 있도록 설정 환경을 제공하고, 원하는 센싱 속성을 선택할 수 있다. 그리고 선택한 센싱 속성에 따라 검색하여 사용자에게 상황 데이터를 제공한다. 여기서 센싱 속성으로 센싱 시간, 센싱 습도, 센싱 온도, 센싱 조도를 제공한다.

4.2 TCP/IP 소켓 인터페이스를 이용한 Push 서비스의 OpenAPI

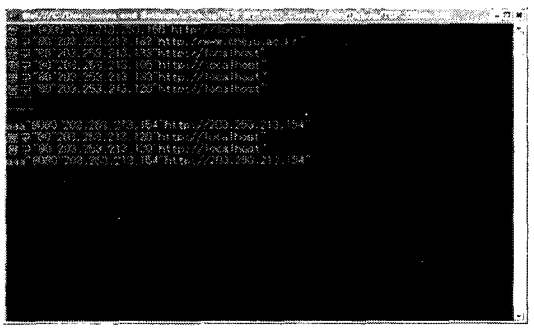
TCP/IP 소켓을 이용하여 Push 서비스를 제공하기 위해 클라이언트 구성 관리자, 데이터베이스, 서버와 클라이언트를 구현한다. 그림 12에서는 클라이언트 구성 관리자를 보여주고 있으며, 클라이언트 정보를 확인하기 위해서 클라이언트 이름, 포트 번호, IP 주소, URL 주소를 입력 받고, 데이터베이스에 저장한다. 클라이언트 구성 관리자는 서버에서 수집된 상황 데이터를 사용자에게 전달하기 위해서 클라이언트 정보를 알아야 한다. 그림 13은 클라이언트 구성 관리자에 입력된 클라이언트 이름, 포트번호, IP 주소, URL 주소의 데이터를 보여주고 있다. 예를 들어 그림 12에서 클라이언트 이름은 봉구, 포트번호를 80, IP 주소를 203.253.213.120, URL 주소를 http://localhost로 입력할 경우 그림 13에서와 같이 데이터베이스에 클라이언트 정보 목차 테이블에 삽입되는 것을 확인 할 수 있다.



[그림 12] 클라이언트 구성 관리자 입력 화면

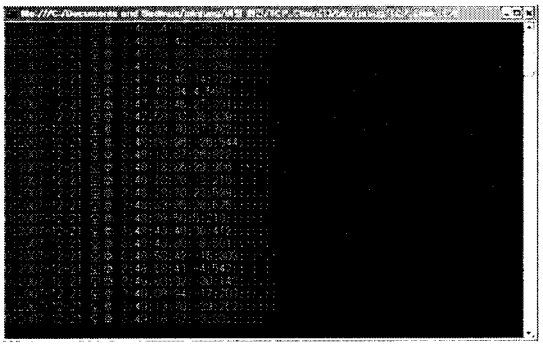
Clientname	Port_number	IP_address	URL_address
광구	80	203.253.213.133	http://local
영구	80	203.253.213.182	http://www.chej...
랜구	80	203.253.213.133	http://localhost
랭구	80	203.253.213.105	http://localhost
봉구	80	203.253.213.133	http://localhost
봉구	80	203.253.213.120	http://localhost

[그림 13] 클라이언트 정보 목차 테이블



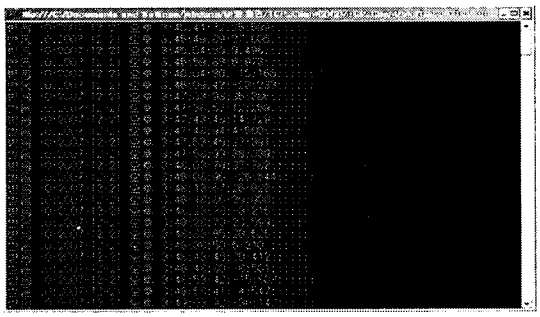
[그림 14] 클라이언트 정보 목록 화면

그림 14는 서버가 클라이언트 구성 관리자로부터 수신 측 클라이언트 정보 목록을 보여주고 있다. 여기서 목록에는 서버에서 전달 받은 센서 노드 번호, 데이터를 받은 날짜 및 시간 그리고 센서의 온도, 습도, 조도 등이 있다.



[그림 15] 상황 데이터를 클라이언트로 전송하는 서버 화면

서버에 수집된 상황 데이터를 클라이언트로 전송하는 화면을 그림 15에서 보여주고 있다. 이 화면은 센서 노드 번호와 데이터를 받은 날짜, 시간, 온도, 습도, 조도 등의 센서 데이터를 나타내고 있다. 소켓 인터페이스를 이용한 Push 서비스를 제공하기 위해 서버는 클라이언트 구성 관리자로부터 클라이언트 정보 목록을 가져와서 IP 주소와 클라이언트를 이용하여 온도, 습도, 조도 등의 상황 데이터를 사용자 요청 없이 전달하고 있다. 여기서 서버에서 이용되는 클라이언트 정보는 클라이언트의 이름, 포트 번호, IP 주소, URL 주소 등이다.



[그림 16] 클라이언트에서 상황 데이터 수신 화면

그림 16은 클라이언트에서 서버로 부터 상황 데이터를 수신하는 화면을 보여주고 있다. 여기서는 센서 노드 번호, 데이터를 받은 날짜, 온도, 습도, 조도를 수신하고 있다. 그리고 클라이언트가 서버에 요청을 하지 않아도 서버에서 클라이언트 정보에서 수신 목록을 받아 서버에서 상황 데이터를 사용자에게 전달하고 있다. 따라서 서버에서는 클라이언트로부터 데이터 요청이 없는 경우에도 센서 노드 번호와 데이터를 받은 날짜, 온도, 습도 등의 상황 데이터를 전송하고, 클라이언트에서는 긴급하거나 위급한 상태에서 상황 데이터를 신속하게 확인할 수 있다.

5. 결론

센서 네트워크 기반의 다양한 미들웨어나 응용 서비스는 폐쇄적인 인터페이스를 제공하고 있어 여려 센서 네트워크에서 수집된 많은 상황 데이터를 상호 접근하는데 어렵다. 이를 위하여 본 논문에서는 센서 네트워크 기반의 Pull/Push 서비스 모델을 제시하고, 웹 서비스와 소켓 인터페이스를 이용하여 Pull/Push 서비스를 지원하는 구조를 제안한다. 그리고 센서 네트워크에서 수집된 온도, 습도 등의 상황 데이터를 요청에 따라 제공하는 Pull 서비스를 지원하기 위해 .Net 프레임워크 기반의 원격 서비스를 이용하여 웹 서비스 기반의 OpenAPI를 설계하고 구현한다. 그리고 TCP/IP 소켓 인터페이스를 이용하여 Pull 서비스를 위한 OpenAPI를 설계하고 구현한다. Pull/Push 서비스를 위한 OpenAPI에 대한 구현과 실험을 통해 제시한 Pull/Push 서비스 모델과 구조를 검증하였다. 본 Pull/Push 서비스와 OpenAPI 연구를 통해 개방적인 응용 인터페이스를 이용하여 여러 유비쿼터스 응용 서비스에 쉽게 개발할 수 있으며, 다양한 실시간 상황 정보를 신속하게 사용자에게 제공할 수 있다.

참고문헌

[1] 임성호, 김주만, "웹 서비스 기반 URC 로봇 원격 모니터링 기술의 설계 및 구현", 한국콘텐츠학회논문지, 제 6권 제 11호, 2006

[2] 강경옥, 김용우, 권훈, 김부림, 김도현, "Tiny-DB와 MySQL을 이용한 유비쿼터스 센서 네트워크 기반의 실시간 정보 서비스 설계 및 구현", 한국산학기술학회논문지, 제 7권, 제 2호, 2006, pp. 175 ~ 181

[3] 박유미, 최영일, 이병선, "웹 서비스 기반의 개방형 서비스 기술", 한국통신학회지, 제22권, 제5호, 2005, pp. 28 ~ 42

[4] 한창환, 한연희, 길준민, 최장원, 윤준원, "Korea@Home 시스템에서 웹 서비스 Open API활용을 위한 설계", 한국정보기술학회 하계학술대회 논문집, 2007, pp. 464 ~ 469

[5] 이원석, "웹 서비스(Web Service)", 디지털 행정, 제 25권, 제 2호, 2002, pp. 82 ~ 92
[6] 최영관, "소설같은 C# Novel C#", 도서출판 자북, 서울, 2003

김 규 리(Kyu-Li Kim) [정회원]

- 2006년 2월 : 제주대학교 컴퓨터공학과 (컴퓨터공학 학사)
- 2008년 2월 : 제주대학교 컴퓨터공학과 (컴퓨터공학 석사)
- 2008년 6월 ~ 현재 : 경희대학교 산학협력단 (기술연구원)

<관심분야>
웹 서비스, 센서 네트워크

김 도 현(Do-Hyeun Kim) [종신회원]

- 2000년 8월 : 경북대학교 전자공학과 정보통신전공(공학박사)
- 1990년~1995년 국방과학연구소 연구원
- 1999년~2004년 천안대학교 정보통신학부 조교수
- 2004년~현재 제주대학교 통신컴퓨터학부 부교수

<관심분야>
센서 네트워크, 이동성 관리, WBAN, WPAN, 텔레매틱스

변 영 철(Yung-Cheol Byun) [정회원]

- 2001년 8월 연세대학교 컴퓨터공학과(공학박사)
- 2001년 ~ 2002년 한국전자통신연구원 선임연구원
- 2002년 ~ 현재 제주대학교 통신컴퓨터공학부 교수

<관심분야>
패턴인식, 시맨틱 웹, 지능형 컴퓨팅, 상황인식, 유비쿼터스 미들웨어