

HTTP-CoAP 프록시 서버를 이용한 사물인터넷 플랫폼 설계

이정혁^o, 김상현, 오창세,

김동휘, 오형석, 박현주

한밭대학교

wisoftlab@hanbat.ac.kr, sangmaz@naver.com, atan925@naver.com,

eastray@hanbat.ac.kr, wellstone@hanbat.ac.kr, phj@hanbat.ac.kr

The Design of Internet of Things Platform Using CoAP-HTTP Proxy Server

Lee Jung-Hyuck, Kim Sang-Hyun, Oh Chang-Se, Oh Hyeong-Seock, Kim Dong-Hwi, Park Hyun-Ju

Hanbat National Univ.

요 약

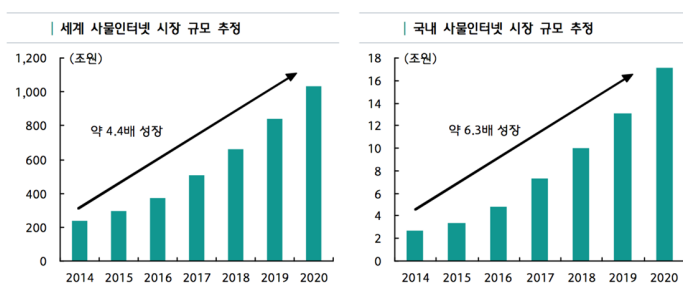
최근 사물인터넷(Internet of Things)을 통하여 여러 가지 서비스들을 제공하려는 시도가 많아지고 있다. 이에 사물과 사물 간 정보를 교환하기 위한 다양한 데이터 전송 방식과 규약들이 소개되고 있다. 현재 웹 상에서 정보를 주고받을 수 있는 HTTP 프로토콜은 가장 보편적인 데이터 전송 방식으로 사용되고 있다. 또한 국제 인터넷 표준화 기구(IETF)에서 정의한 CoAP 프로토콜은 작은 메모리와 낮은 처리 성능을 가진 노드들에 적합한 프로토콜로 oneM2M, ETSI, OMA 등의 표준단체에서 사물인터넷 관련 연구와 개발을 진행하고 있다. 하지만 많은 기업과 단체에서 각자의 표준을 확립하여 사물인터넷 플랫폼을 구축하고 있는 현 상황에서, 사물인터넷과 관련하여 일관된 프로토콜과 데이터 전송 방식에 대한 연구가 이루어지지 않고 있다. 이러한 점을 착안하여 본 논문에서는 HTTP-CoAP Proxy를 구현하여 RESTful API 기반으로 CoAP과 HTTP의 데이터 전송 요청을 처리할 수 있는 사물인터넷 플랫폼을 제안한다.

1. 서 론

최근 다양한 사물이 서로 연결되는 사물인터넷(Internet of Things)에 대한 기대와 관심이 커지고 있다. 세계적인 시장조사기관 가트너에 따르면 2015년 '인터넷 연결 기기(connected things)'의 대수가 올해 보다 30% 증가한 49억 대, 2020년에는 250억 대의 장치들이 무선 인터넷을 기반으로 연결될 것이라고 예상하였다. 장소에 구애받지 않고 언제 어디서나 컴퓨팅 환경에 접속할 수 있는 유비쿼터스 패러다임이 확대되면서 다양한 기기가 인터넷에 연결되어 연동되고 있다. [1]

사물인터넷 기술에 대한 기대와 관심이 고조되고 있는 가운데, 국내·외 많은 단체에서 사물인터넷 플랫폼에 관한 연구를 진행하고 있다. 사물인터넷 플랫폼이란 인터넷에 연결된 모든 기기를 하나의 시스템 안에서 통합·관리할 수 있는 운용체계이다. 사물인터넷 플랫폼은 센싱, 통신 및 네트워크 인프라, 서비스 인터페이스의 주요 기술들을 총체적으로 요구한다. 현재 사물인터넷 관련 기기들이 폭발적으로 늘어나고 있기 때문에 이를 수용하기 위한 사물인터넷 플랫폼 개발에 대한 연구는 필수적이다.

현재 공개된 사물인터넷 플랫폼들은 각자의 표준을 확립하여 생태계를 구축하기 위해 노력하고 있으나 일관된 표준과 프로토콜을 사용하고 있지 않고 있다. 이것은 실질적인 특성들이 발생하게 만들며 수많은 사물들을 식별하고 인지하여 상호작용하는데 어려움이 따르게 한다. 따라서 제 3자 입장에서 자신의 디바이스를 특정 사물인터넷 플랫폼에 연결하고 싶을 경우, 특정 표준을 준수해야 한다는 제약이 발생할 수 있다. 또한 플랫폼들 간 서로 다른 생태계를 구축했을 경우, 종합적인 지능화서비



[그림 1] 사물인터넷 시장 규모 추정

스에 어려움이 있을 것으로 예상된다.

본 논문에서는 HTTP-CoAP Proxy를 구현하여 RESTful API 기반으로 CoAP과 HTTP의 데이터 전송 요청을 처리할 수 있는 사물인터넷 플랫폼을 제안한다. 또한 수집 데이터에 대한 특정 이벤트를 처리하기 위해 웹 소켓을 활용한 이벤트 처리 방법에 대해 논한다. 제안한 플랫폼은 현재 가장 보편적으로 사용되는 HTTP와 작은 메모리와 낮은 처리 성능을 가진 노드들에 적합한 CoAP 기반의 데이터 전송 방법을 선택적으로 사용하여 사물인터넷 서비스를 운영할 수 있다.

2. 관련 연구

2.1 CoAP(Constrained Application Protocol)

최근 M2M(Machine-to-Machine) 및 IoT, WoT(Web of Things) 시대로 접어들음에 따라 센서 및 각종 디바이스 간에 상호 통신할 수 있는 공통 프로토콜이 필요하게 되었다. IoT 프로토콜로 IETF(Internet Engineering Task Force) CoAP(Constrained Application Protocol) 프로토콜이 핵심 역할을 하고 있으며, 차세대 센서 접속 프로토콜로 IETF CoAP이 많이 사용될 것으로 예상된다.[2] CoAP은 HTTP보다 가벼우면서 Publish/Subscribe 통신 모델까지 커버하는 응용 프로토콜이다. 특별히 저전력 센서, 스위치 등과 같이 작은 메모리와 낮은 처리 성능을 가진 기기들을 표준적인 인터넷 환경에서 제어하기 위한 목적으로 만들어졌다.

2.2 REST(REpresentational State Transfer) API

REST(Representational State Transfer)는 분산 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처의 한 형식이며 Roy Fielding의 2000년 논문에 의해서 소개되었다. 현재의 아키텍처가 웹의 본래 설계의 우수성을 많이 사용하지 못하고 있다고 판단했기 때문에, 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처를 소개했는데 그것이 바로 REST이다.

RESTful 웹 서비스는 자원 중심의 표현, 전달, 접근 방식의 특성으로 인해 자원 기반의 아키텍처(ROA:Resource Oriented Architecture)라고 한다. ROA는 서비스 중심의 SOA에 대응되는 개념으로 일컬어지고 있다.

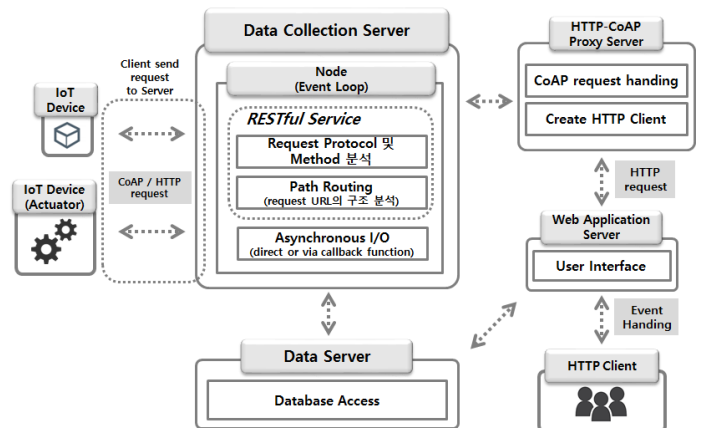
REST는 자원지향구조로 웹 사이트의 콘텐츠(Text, 이미지, 동영상)와 DB의 내용 등을 자원으로 파악하여 각 자원에 고유한 URI를 부여하고, 해당 자원을 웹의 기본 요청유형인 POST, GET, PUT, DELETE를 통해서 처리한다.

3. 본 론

3.1 연구 개요

본 논문에서는 “보편적으로 사용되고 있는 웹 프로토콜인 HTTP”와 “작은 메모리와 낮은 처리 성능을 가진 기기들을 표준적인 인터넷 환경에서 제어하기 위한 CoAP”을 지원하는 사물인터넷 플랫폼을 제안한다.

제안한 플랫폼은 그림 2와 같은 구조를 가진다. 이 구조는 크게 5가지로, 센싱 정보를 전송하고 명령을 수행하기 위한 IoT Device(Sensing or Actuator), 수집된 데이터를 분석하는 Data Collection Server, 수집된 데이터를 저장하는 Data Server, HTTP-CoAP Proxy Server, 클라이언트용 Web Application Server로 구성된다.



[그림 2] 사물인터넷 플랫폼 구성도

3.2 시스템 흐름

IoT Device로부터의 정보 수집은 크게 HTTP와 CoAP 기반의 정보 요청으로 구분된다. HTTP 기반의 정보 요청이 CoAP 기반의 정보 요청과 가지는 차이점은 단지 HTTP-CoAP Proxy 서버를 거치지 않고 데이터의 흐름이 진행된다는 점이다.

IoT Device들이 센서 데이터를 센싱을 하게 되면, 이를 데이터 수집 서버에게 전달한다. Data Collection Server는 CoAP, HTTP 기반의 다양한 요청을 수용할 수 있다. 또한 각 요청에 대해 REST API를 이용하여 RESTful Service를 제공한다. 수집된 센서 정보는 Database Server에 저장되고, 클라이언트는 이 정보를 웹 어플리케이션을 통해 다양한 UI 형태로 확인할 수 있다. 또한, 사용자가 웹 어플리케이션을 통해 설정한 이벤트(예 : CO2 값이 1500ppm 이상일 때)가 발생하였을 시에는 Data Collection Server에서 이를 감지하여 센서 정보와 이벤트 정보를 HTTP-CoAP Proxy 서버에 전송한다. Proxy 서버는 Data Collection Server로부터 받은 정보를 해석한 뒤 사용자에게 이 정보에 대해 이벤트 처리를 한다.

위에서 언급한 RESTful Service를 사용하는 이유는 확

장성, 인터페이스의 단순화, 컴포넌트 수정의 용이함, 프로그램 코드와 데이터의 컴포넌트화로 높은 이식성, 시스템 실패에 대한 강력한 저항성과 같은 장점 때문이다. REST의 가장 큰 특징 중 하나로는 모든 자원을 요청 URL로 표현한다는 것인데, HTTP와 CoAP의 경우에는 RESTful API를 적극적으로 사용하는 프로토콜로 알려져 있다.

3.3 프로토타입의 사물인터넷 환경 구성

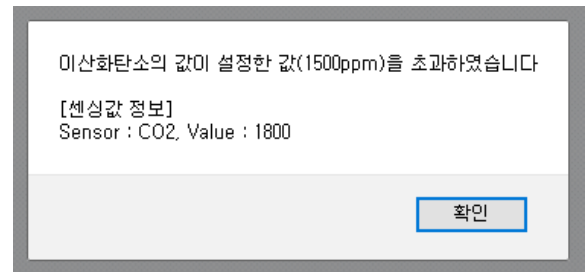
본 논문에서는 제안한 사물인터넷 플랫폼을 구현하기에 앞서 시스템 흐름에 맞게 프로토타입의 사물인터넷 환경을 구성해보았다. 가상의 CO2 센서 노드가 REST API에 맞게 request하는 상황을 위해 Chrome(HTTP request), Firefox Copper(CoAP request)를 사용하여 특정 값을 전송하게 하였고, 이벤트가 발생하였을 시에는 HTTP-CoAP Proxy Server를 거쳐서 사용자에게 알림 메시지를 주는 시스템을 구현해 보았다.

시스템 구성	사용 소프트웨어
IoT Device (HTTP, CoAP request)	Chrome v45.0.2454.101 Firefox Copper 0.18.4.1
Data Collection Server	Node.js v0.12.7
Database Server	MariaDB v10.1.8
HTTP-CoAP Proxy Server	Node.js v0.12.7
Web Application Server	Tomcat 8.0, Node.js v0.12.7, socket.io v0.9
Client	Chrome v45.0.2454.101

[표 1] 프로토타입의 사물인터넷 플랫폼 구성

Data Collection Server와 HTTP-CoAP Proxy Server는 Node.js를 사용하였으며 'coap' 모듈을 분석하여 구현하였다.[3] IoT Sensor Device는 REST API를 준수하여 URL 요청을 통해 자원에 접근한다. 이벤트 발생 시 Data Collection Server는 CoAP 클라이언트를 생성하여 Proxy Server에 전송하고, Proxy Server는 HTTP Client를 생성하여 HTTP Server에 이벤트가 정보를 전달한다. 여기서 Data Collection Server는 런타임 시 HTTP-CoAP Proxy Server의 정보를 주입 받기 때문에 Data Collection Server는 단지 HTTP-CoAP Proxy Server를 호출하기만 할 뿐 HTTP-CoAP Proxy Server가 무엇인지 관심을 가지지 않아도 된다. HTTP Server는 Node.js의 socket.io를

통해 실제 클라이언트가 이벤트 알림 페이지를 요청할 때마다 소켓 연결을 시도하고, 이벤트가 발생되면 해당 클라이언트에게 이벤트를 emit한다. 아래 그림 4는 CO2가 1500ppm을 초과하였을 때 사용자에게 전송되는 알림 메시지이다.



[그림 4] 사용자가 받은 알림 메시지

4. 결 론

본 논문에서는 "보편적으로 사용되고 있는 웹 프로토콜인 HTTP 프로토콜"과 "작은 메모리와 낮은 처리 성능을 가진 기기들을 표준적인 인터넷 환경에서 제어하기 위한 CoAP 프로토콜"을 지원하는 사물인터넷 플랫폼을 설계 및 구현하였다. 제안한 플랫폼은 기존의 HTTP 프로토콜을 수용하고 진보된 CoAP 프로토콜을 통합적으로 지원함으로써 사물인터넷 연구자로 하여금 편리하게 프로토콜을 선택하여 사물인터넷 시스템을 구축할 수 있다. 또한, 이 플랫폼은 RESTful Service를 지원하기 때문에, 클라이언트는 이 플랫폼의 REST API 사용사례만 파악하면 간단하게 사물인터넷 시스템을 운용할 수 있다.

향후 연구에서는 사물인터넷에 적합한 프로토콜 중 하나인 MQTT 프로토콜을 이용한 데이터 수집 솔루션과 효율적인 데이터 저장 기술에 대한 연구를 진행하여 현재 진행 중인 "IoT 플랫폼 개발을 위한 선택적 프로토콜 기반의 실시간 데이터 수집 솔루션 및 저장 기술에 관한 연구"에 적용할 계획이다. 또한 제안한 플랫폼의 성능과 안정성을 향상시킬 방안에 대해 검토한다.

참 고 문 헌

- [1] 전종홍, 차흥기, 이원석, 김형준, 오픈소스 사물인터넷 (OSIoT) 동향 및 전망, 한국통신학회지, 제32권 제5호, 2015.04, 23-30
- [2] 고석갑, 박일균, 손승철, 이병탁, IETF CoAP 기반 센서 접속 프로토콜 기술 동향, Electronics and Telecommunications Trends
- [3] node-coap, <https://github.com/mcollina/node-coap>