

CoAP 프로토콜을 이용한 Node.js 기반의 IoT 노드 제어 서비스 설계

김보경, 김문권, 김도현*

*제주대학교 컴퓨터공학과

qhru5725@hanmail.net, pluskmk12@live.com, kimdh@jejunu.ac.kr

Design of IoT Node Control Service Based on Node.js Using CoAP Protocol

Bo-Kyung Kim, Wen-Quan Jin, Do-Hyeun Kim*

Department of Computer Engineering, Jeju National University

요 약

최근 사물인터넷 분야의 성장과 관심이 급격하게 일어나면서 인터넷에 접속하는 사물이 급속히 증가하고 있다. IETF(Internet Engineering Task Force) CoRE(Constrained RESTful Environment) 워킹그룹에서 IoT 프로토콜로 CoAP(Constrained Application Protocol)을 표준으로 채택하고 있다. 본 논문에서는 인터넷 환경에서 CoAP 프로토콜을 이용하고 Node.js 기반의 IoT 임베디드 노드를 등록하고 제어하는 서비스를 설계한다. 이를 통해 논 블로킹 I/O 모델을 지원하고, 가볍고 효율적으로 IoT 노드를 개발할 수 있다.

I. 서 론

최근 사물인터넷 분야의 성장과 관심이 급격하게 일어나면서 인터넷에 접속하는 사물이 증가하고 있다. 따라서 효율적인 사물 제어 서비스에 대한 관심이 점점 더 커지고 있다. 이에 사물을 제어하기 위한 통신 프로토콜로 MQTT(Message Queue Telemetry Transport), 웹소켓(WebSocket), CoAP(Constrained Application Protocol) 등이 등장하고 있다.

CoAP 프로토콜은 저성능 CPU, 작은 저장장치, 저전력 등의 IoT 노드 제약 조건과, 높은 데이터 손실과 느린 데이터 전송 등의 네트워크 제약 환경에서 데이터를 전송하는 응용 프로토콜이다. 현재 oneM2M, LWM2M, OIC IoTivity 등 IoT 플랫폼에 CoAP 프로토콜이 적용되고 있으며, 제한된 하드웨어 및 네트워크 자원을 갖고 있는 IoT 노드 등을 지원하고 있다[1,2]. 또한 인터넷 환경에서 IoT 노드를 이용하여 상황을 인지하고 사물을 제어하기 위해 IETF의 표준 CoAP 프로토콜이 활용되고 있다.

논 블로킹 I/O 모델을 사용하는 이벤트 기반의 Node.js가 등장하면서 소형 싱글보드 컴퓨터의 효율적인 제어가 가능해졌다. Node.js는 Chrome V8 JavaScript 엔진으로 개발된 JavaScript 런타임이다. Node.js는 이벤트 기반과, 논 블로킹 I/O 모델을 사용해 가볍고 효율적이다[3]. Node.js는 가볍고 효율적이고, 사물인터넷 환경에서 사용하기에 적합하여 개발에 활용되고 있다. 현재 개발된 다양한 Node.js 기반의 소형 싱글 보드 컴퓨터 제어 프레임워크가 오픈소스로 배포되고 있다.

본 논문에서는 사물인터넷에서 사용되고 있는 여러 프로토콜 중 제한된 통신 환경에 최적화하여 개발된 REST 기반의 경량 메시지 전송 프로토콜인 CoAP과 Node.js를 이용해 사물을 등록하고 제어하는 서비스를 설계한다. 이를 통해 IoT 노드의 작은 용량의 메모리와 저전력 등 제한된 환경에서 센서나 구동체 노드와 이동 단말 간의 통신을 지원하고, 상황 데이터를 획득할 수 있을 것으로 예상된다.

하고 환경을 제어하는 IoT 서버 기반의 서비스 구조를 그림 1과 같이 제안한다.

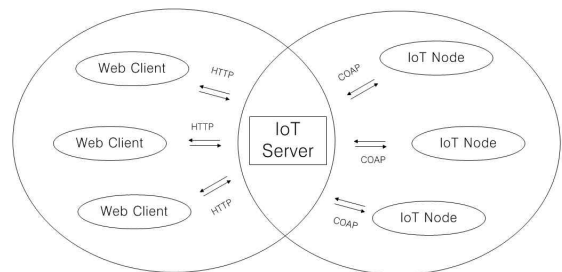


그림 1. IoT 서버 기반의 사물 제어 서비스 구조

제한된 Node.js 기반의 사물 제어 서비스는 IoT 서버, 클라이언트, IoT 노드 이 3가지 요소로 구성된다. IoT 서버는 클라이언트에게 Restful 서비스를 제공하고, IoT 노드와 CoAP 프로토콜을 이용하여 데이터를 전송할 수 있다. 클라이언트는 웹 애플리케이션을 통해 IoT 서버에 사물 제어 서비스를 요청하고, IoT 서버에게 서비스를 제공 받는다. IoT 서버 기반의 사물 제어 서비스 구조는 그림 1과 같다.

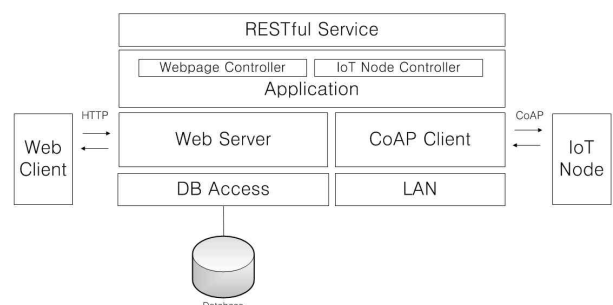


그림 2. IoT 서버 구조

II. Node.js 기반의 사물 제어 서비스 구조

CoAP을 이용한 사물인터넷 환경에서의 사물로부터 상황 정보를 수집

IoT 서버는 클라이언트에게 웹 서비스를 제공하고, IoT 노드와 CoAP 프로토콜을 이용하여 IoT 노드로부터 상황 데이터를 수집하여 데이터베이스에 저장한다. 데이터베이스에 저장된 IoT 노드에 데이터를 바탕으로 클라이언트에게 사물 제어 서비스를 제공한다. 그림 2는 설계한 IoT 서버의 구조를 보여주고 있다.

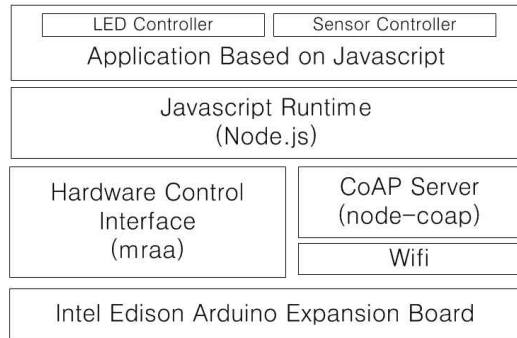


그림 3. IoT 노드 구조

IoT 노드는 Node.js 기반으로 사물의 등록하고 제어하는 서비스를 설계한다. 이때 IoT 노드 내에서는 CoAP을 이용하여 데이터를 송수신하고, 메시지를 해석해 제어 신호로 출력할 수 있다. 설계한 IoT 노드의 구조는 그림 3과 같다.

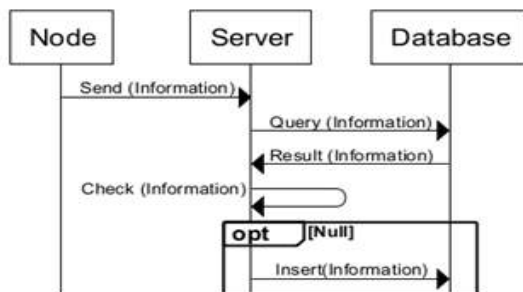


그림 4. IoT 노드 등록 시퀀스 다이어그램

IoT 노드에서는 CoAP 프로토콜을 이용하여 IoT 서버에 등록을 요청한다. IoT 노드 정보가 없을 경우 IoT 서버에 등록이 된다. 그리고 IoT 노드는 센서와 구동체를 제어하거나 상황 데이터를 수집할 수 있다. IoT 노드 등록 과정은 그림 4의 IoT 노드 등록 시퀀스 다이어그램을 보여주고 있다.

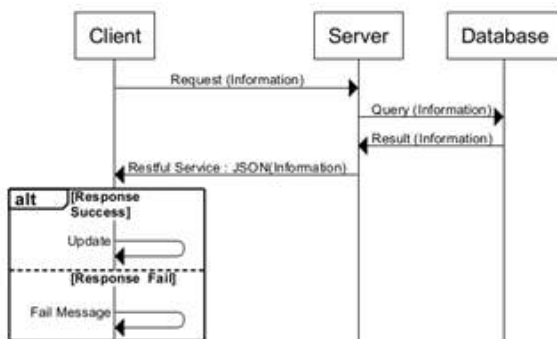


그림 5. IoT 서버와 클라이언트 데이터 전송 시퀀스 다이어그램

IoT 서버는 클라이언트의 요청에 따라 데이터베이스에서 데이터를 얻고 클라이언트에게 보낸다. 클라이언트가 데이터를 성공 여부 메시지를 보여준다. 이와 같은 과정은 그림 5의 IoT 서버와 클라이언트 데이터 전송 시퀀스 다이어그램을 보여주고 있다.

III. 결론

본 논문에서는 인터넷 환경에서 CoAP 프로토콜과 Node.js 기반의 IoT 노드 제어 서비스를 설계한다. 이를 통해 논 블로킹 I/O 모델을 지원하고, 가볍고 효율적으로 IoT 노드를 개발할 수 있다. 그리고 인터넷 상에서 실 내 IoT 노드를 통해 환경을 감시하고 제어할 수 있다. 향후 이 서비스는 다양한 도시 공간에서 지능적인 IoT 서비스를 개발하는 데 기여할 것이다.

ACKNOWLEDGMENT

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원(R0101-16-0129, 개방형 고성능 표준 IoT 디바이스 및 지능형 SW 개발)과 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음. (IITP-2016-H8501-16-1017).

참 고 문 헌

- [1] Z. Shelby, B. Frank, D. Sturek, "Constrained Application Protocol (CoAP)", RFC 7252, June, 2014.
- [2] 김문권, 김도현, "상호 호환성 검증을 위한 IoT 기반의 CoAP 프로토콜 구현 및 실험", The Journal of the Institute of Internet, Broadcasting and Communication, Vol. 14 No. 4, 2014
- [3] <https://nodejs.org/ko>
- [4] Wen-Quan JIN, Do-Hyeun Kim "Implementation and Experiment of CoAP Protocol Based on IoT for Verification of Interoperability", The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC), Vol. 14 No. 4, August 2014.
- [5] Jian Wang, June SaKong, Ho-Young Kwak, Do-Hyeun Kim. "Design and Implementation of IoT Middleware Using Data Refinement Scheme based on IETF CoAP", The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC), Vol. 15 No. 6, Dec. 2015.