

웹 기반 개방형 IoT 환경에서 실시간 데이터 전송을 위한 시스템 설계

System Design for Real-Time Data Transmission in Web-based Open IoT System

표경수* · 박진태 · 문일영

한국기술교육대학교 대학원 컴퓨터공학과

Gyung-soo Phyoo* · Jin-tae Park · Il-young Moon

School of Computer Science Engineering, Korea University of Technology and Education, Chungcheongnam-do 31253, Korea

[요 약]

전 세계적으로 인터넷의 발전과 스마트 디바이스의 보급이 급격하게 증가하면서 IoT (internet of things)가 주목받고 있다. IoT가 일상생활에 접목되면서 IoT 시장은 방대해지고 있다. 그래서 전문가들은 앞으로 IoT 디바이스가 10년 내에 약 1조대 이상으로 증가할 것으로 전망하고 있다. IoT와 관련된 기술도 꾸준히 발전되고 있으며, 여러 분야에서 IoT 발전을 위한 연구가 진행되고 있다. 하지만, IoT 서비스를 출시하는 업체에서는 다른 플랫폼과 데이터를 상호작용하지 않아 사일로 현상이라는 장애물에 봉착해 큰 시장으로 성장하는데 제한되고 있다. 이를 해결하기 위해 웹 기술이 주목받고 있다. 웹 기술을 이용하면 플랫폼에 상관없이 데이터를 상호 작용할 수 있고, 그 데이터를 이용하여 다양한 서비스를 개발할 수 있을 뿐만 아니라 개발자 입장에서도 불필요한 비용을 줄일 수 있다. 따라서 본 논문에서는 IoT 플랫폼에 독립적으로 데이터를 실시간 전송할 수 있는 웹 기반의 개방형 IoT 시스템을 연구하였으며, 각 IoT 플랫폼 간 데이터를 전송할 수 있는 시스템을 구현하였다.

[Abstract]

IoT is attracting attention as the development of the Internet and the spread of smart devices are rapidly increasing worldwide. As IoT is integrated into everyday life, the market is getting bigger. So, experts predict that IoT devices will grow to more than one trillion in a decade. Techniques related to IoT are also being developed steadily, and studies are underway to develop IoT in various fields. However, vendors launching IoT services do not interact with data from other platforms. Therefore, it is limited to growing into a big market by facing the obstacle called the silo phenomenon. To solve this problem, web technology attracts attention. Web technology can interact with data regardless of platform, and it can not only develop various services using the data, but also reduce unnecessary costs for developers. In this paper, we have studied a web - based open IoT system that can transmit data independently in real time to the IoT platform.

Key word : Data, Internet of things, Javascript, Web, Web of things.

<https://doi.org/10.12673/jant.2016.20.6.562>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 22 November 2016 Revised 25 November 2016
Accepted (Publication) 28 December 2016 (30 December 2016)

*Corresponding Author; Gyung-soo Phyoo

Tel: +82-10-7706-5260

E-mail: gkgkgy@koreatech.ac.kr

1. 서론

최근 현대 사회에서는 컴퓨터와 컴퓨터의 연결뿐만 아니라 사물과 사물 간에 인터넷을 연결하여, 사물 주변이나 사물에서 일어나는 상황을 관찰하고, 정보를 수집하여 주변 환경을 제어할 수 있는 IoT(internet of things) 즉, 사물 인터넷 기술이 주목받고 있다. 이러한 주목을 바탕으로 IoT 서비스가 폭발적으로 등장하고 있으며, 전문가들은 IoT 서비스의 등장으로 인해 약 10년 이내에 인터넷에 연결된 IoT 디바이스가 약 1조대를 넘어설 것이라고 예측하고 있다. 이러한 예측에도 불구하고, 현재 IoT에서는 여러 가지 문제점을 갖고 있다. 대표적으로, IoT 서비스 업체 간의 사일로 현상(silos effects)이다[1]. 사일로 현상은 곡식을 저장해두는 원통형 모양의 독립된 창고인 ‘사일로’에서 생긴 경제용어로서, 조직끼리 사일로처럼 서로 담을 쌓고, 조직단위로 업무 효율화 측면에서 IT(information technology) 인프라 및 자체 솔루션을 구축하고 사용하여 자기 조직의 이익만 추구하는 현상을 말한다. 이 현상은 네트워크에서 가져올 수 있는 여러 이점을 막을 수 있을 뿐만 아니라, 특히 IoT 서비스를 개발하는 개발자들에게도 괴롭다. 개발자들은 특정 도메인과 플랫폼을 위한 서비스를 만들기 위해 도전해야 하고, 각기 다른 플랫폼에서 어플리케이션을 개발하기 위한 방법을 학습하는 비용이 크다. 따라서 플랫폼 개발자는 IoT 서비스를 위한 개발 비용을 줄이고, 다른 플랫폼과 통합하여 고객의 요구를 이행하고, IoT 시장의 성장을 촉진하기 위하여 상업적인 개방을 위한 방법을 탐색해야 한다. 이러한 방법을 개방형 IoT 플랫폼이라 한다[2].

개방형 IoT 플랫폼을 위한 여러 가지 방법 중에서 웹 기술이 주목받고 있다. 웹 기술은 그 자체에서 많은 장점을 갖고 있지만, IoT 측면에서 바라보자면, 브라우저가 가능하고, low-level의 네트워크에 상관없이 HTTP(hyper text transfer protocol) 프로토콜을 이용하여 통신할 수 있다. 또한, 웹 기반 IoT 서비스에 검색기능을 추가할 수 있고, 설치되어 있는 방화벽을 넘나들 수 있고, 문제가 생겼을 경우 네트워크를 우회할 수 있다는 여러 장점이 있지만, 그 중에서도 특히 웹 개발 언어(HTML; hyper text markup language, javascript, PHP 등)는 최근 개발자들이 쉽고 빠르게 배울 수 있는 친숙한 언어 중에 하나이다. 따라서 웹 기술을 사용하여 IoT 플랫폼에 상관없이 디바이스에 접근하고, 제어할 수 있다면 언제 어디서든지 IoT 서비스를 이용할 수 있는 개방형 초연결사회가 만들어질 것이다[3].

따라서, 본 논문에서는 앞서 언급한 현재 IoT의 사일로 현상을 극복하기 위해 웹 기술을 이용하여 디바이스에 접근하고, 제어할 수 있는 웹 기반의 개방형 IoT 플랫폼 즉, WoT(web of things) 시스템을 제안하고자 한다.

II. 개방형 IoT 연구 동향

본 절에서는 기존 개방형 IoT 시스템의 연구 동향에 대해서 알아보하고자 한다.

빠르게 발전하는 IoT 시장에 맞춰서 국내외에서는 활발하게 연구가 진행 중이다. IoT와 관련된 표준화 단체에는 ISO(international organization for standardization), ITU(international telecommunication union)와 같은 국제 표준화 기구, ETSI(european telecommunications standards institute), 3GPP(the 3rd generation partnership project), IETF(internet engineering task force), IEEE(institute of electrical and electronics engineers) 등이 있다.

현재 IoT 기술 표준은 유럽을 중심으로 다양한 연구와 기술 개발이 진행 중이고, 국내에서는 한국정보통신기술협회(TTA)를 중심으로 표준화가 진행 중이다.

앞서 나열한 표준화 단체 중에서 3GPP는 이동통신 관련 단체들 간의 공동 연구 프로젝트로 IoT에서 이동통신 기반의 MTC(machine type communication)기술 개발을 진행 중이다. IEEE는 전 세계 전기전자공학 전문가들의 국제조직으로서, 무선통신 기술을 개발하고 있다. 그리고 IETF는 국제 인터넷 표준화 기구로서, IoT에서 사용되는 IPv6에 관련된 네트워크 스택의 표준화를 진행하고 있다 [4],[5].

국외뿐만 아니라 국내에서도 표준화가 진행되고 있다. 한국정보통신기술협회(TTA)에서는 사물인터넷 특별 기술위원회(STC1)를 개설하여 IoT 기술 분야(서비스, 네트워크 등)에 대한 표준화를 진행하고 있고, 해외 표준화 단체와 협력하여 국내 IoT 환경을 개선하기 위해 연구하고 있다.

그리고 IoT의 표준화를 위한 컨소시엄으로 그림 1과 같이 AllSeen Alliance, IIT(industrial internet consortium), OIC(open interconnect consortium), Thread가 있다.

AllSeen Alliance는 ‘AllJoyn’이라는 프레임워크를 기반으로 IoT를 개발하는 단체이다. AllJoyn은 웹킷이 공개한 오픈소스 프로젝트로 가전기기, 자동차, 컴퓨터가 커뮤니케이션 할 수 있는 프레임워크이다.

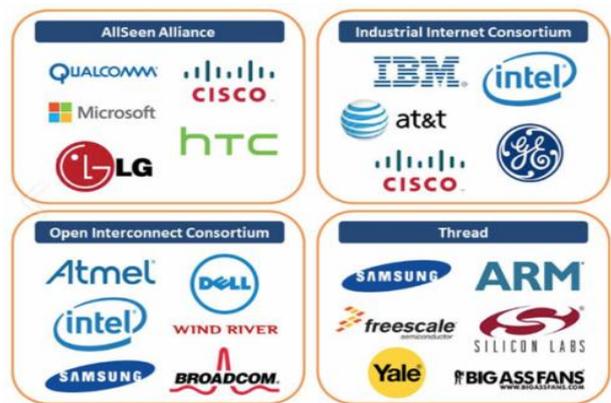


그림 1. 표준화 컨소시엄
Fig. 1. Standardization consortium.

AllSeen Alliance는 다양한 분야의 기업들이 참여하고 있으며, AllJoyn을 채택한 상용화 제품이 출시되었다.

IIT는 IoT를 위한 엔지니어링 표준화 작업을 진행하고 있다. 즉, IoT 활성화를 위해 어디서든지 네트워크, 디바이스를 연결할 수 있는 정보 처리 망을 구축하는 것이다. IIT의 최종 목표는 IoT 비즈니스를 창출하기 위한 플랫폼을 다양화하고, 활용하는 것이고, IoT 최대의 문제점인 보안 문제를 해결하는 것이다.

OIC는 IoT 네트워크상에서 OS와 플랫폼을 사용하는 경량화 장치를 연결하는데 필요한 요구사항을 정의하고, 장치 간 상호 운용성을 보장하는 것을 목적으로 기업들을 선도하기 위해 설립된 조직이고, 약 90여 개의 기업이 참여하고 있다.

OIC는 개발자와 업체가 오픈소스 프로젝트를 이용하여 보안과 연결성을 보장하는 것을 목표로 하고 있다.

Thread는 사물마다 상이한 프로토콜과 데이터 전송 거리, 배터리 제약 문제를 해결하는 것을 목표로 삼고 있는 단체로써, 가정용 기기들을 연결하고 제어하는 최선의 방법을 찾기 위해 설립되었다.

따라서 Thread에서는 쓰기 쉽고, 안전하며, 전력 효율이 좋은 IPv6 기반의 공개 프로토콜과 허브가 없는 메시 네트워크, Zigbee 기반의 프로토콜을 개발하고 있다[6].

III. 시스템 설계 및 분석

3-1 시스템 구성

본 논문에서 제안하는 웹 기반의 개방형 IoT 플랫폼은 그림 2와 같은 구성으로 설계되었다. 다양한 IoT 디바이스에서 수집되는 정보를 각 IoT 플랫폼에서 수집하고, 이때 사용되는 프로토콜은 MQTT(mq telemetry transport), CoAP(constrained environments application protocol) 등 각 업체들에서 사용하고 있는 IoT 프로토콜을 이용한다.

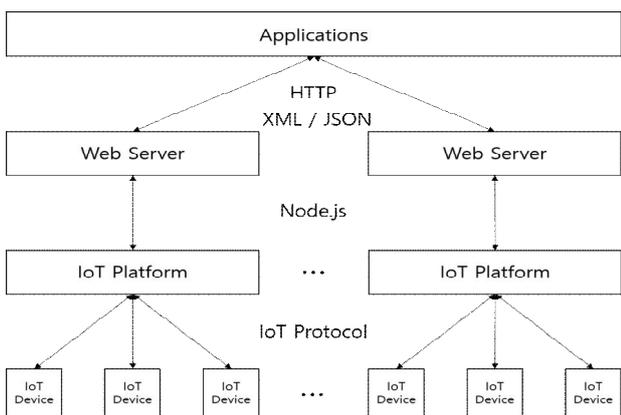


그림 2. 시스템 구성도
Fig. 2. System configuration.

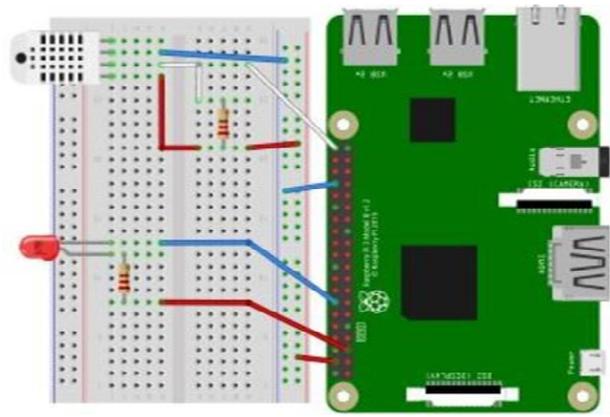


그림 3. 라즈베리파이 구성
Fig. 3. Configuration of Raspberry Pi.

각 IoT 플랫폼에 수집된 정보는 웹 서버에 의해서 관리된다. 어플리케이션에 의해서 요청을 받았을 때, 웹 서버는 IoT 플랫폼에서 수집된 데이터를 XML이나 JSON형식으로 가공하여 어플리케이션으로 전송한다.

반대로 어플리케이션에서 XML이나 JSON 형식으로 보낸 데이터를 해석하여 IoT 플랫폼으로 전송할 수 있다. IoT 플랫폼은 웹 서버에 의해 받은 데이터를 이용하여 IoT 디바이스를 제어한다.

따라서 이러한 시스템 구성을 통해서 한 IoT 플랫폼에서 나온 데이터를 이용하여 다른 IoT 플랫폼의 센서를 제어할 수 있다. 시스템의 각 구성에 대한 자세한 설명은 하드웨어 구성과 웹 서버, 어플리케이션에서 설명한다.

3-2 하드웨어 구현

본 논문에서 제안한 웹 기반의 IoT 시스템을 구현하기 위해 먼저, 하드웨어를 구성해보았다. IoT 플랫폼에 상관없이 데이터를 상호교환하기 위해서 기본적으로 다른 OS의 두 가지 IoT 플랫폼을 준비했다. 첫 번째 IoT 플랫폼은 라즈베리파이 3을 이용하였다.

라즈베리파이에는 1개의 온도 센서와 LED를 구성하였다. 그리고 라즈베리파이에서 센서 데이터를 수집하기 위해 Node.js를 사용하였다. Node.js의 Gpio 모듈을 이용하여 LED를 제어하고, node-dht-sensor 모듈을 이용하여 온도센서를 제어하였다. 그림 3은 라즈베리파이와 센서를 연결한 회로도이다.

라즈베리파이에서 센서 값을 수집하고 제어하기 위해 'node-dht-sensor' 모듈을 이용하여 온도 센서의 값을 실시간으로 가져왔다. LED는 GPIO의 'onoff' 모듈을 이용하여 제어하였다.

두 번째 IoT 플랫폼은 SK텔레콤에서 제공해주는 IoT 서비스 플랫폼으로 국제 IoT 표준인 OneM2M API 기반의 ThingPlug를 이용하였다.

ThingPlug는 OneM2M 기반의 가상의 디바이스를 이용한 IoT 플랫폼을 개발하고자 하는 개발자들에게 개발 환경을 제공

해주는 플랫폼이다.

ThingPlug의 가상 디바이스에서 센서 값을 가져오는 방식은 IoT 프로토콜 중 MQTT 방식을 이용하여 값을 가져오고, Node.js의 'mqtt' 모듈을 이용하여 수집한다.

ThingPlug을 이용하면 가상의 디바이스를 통해서 온도 데이터를 간단하게 수집할 수 있기 때문에, 다른 플랫폼의 회로를 직접 구성하지 않고 연구를 진행하였다.

위에 설명한 하드웨어를 바탕으로 본 논문에서는 ThingPlug에서 수집되는 온도가 일정 온도가 되면 라즈베리파이의 LED에서 불이 들어오는 방식을 구현하고자 한다.

3-3 웹 서버

본 논문에서 제안한 시스템에서 각 IoT 플랫폼에 의해 수집된 데이터를 어플리케이션까지 보내고, 어플리케이션에서 요청한 정보를 IoT 플랫폼에 전송하기 위한 웹 서버를 구현하였다.

Node.js를 이용하여 웹 서버를 구축하고, IoT 플랫폼에서 수집한 데이터를 어플리케이션에 전송한다. ThingPlug의 웹 서버에서는 센서의 데이터를 XML 형식으로 보내고, 라즈베리파이에서는 JSON 형식으로 보낸다.

라즈베리파이에서는 Node.js의 http 모듈을 이용하여 서버를 구축하였다. 어플리케이션에서 데이터를 요청할 때 센서에서 측정된 데이터를 JSON으로 가공해서 전송한다.

하지만, JSON 방식으로 데이터를 전송할 때 크로스 도메인 문제점이 발생한다. 크로스 도메인 (Cross Domain)이란, 서로 다른 도메인에서 Ajax 통신을 시도할 때 발생하는 이슈로 자바스크립트의 보안 정책인 같은 도메인 내에서만 작동하는 것을 원칙으로 하는 동일 근원 정책(Same-Origin Policy) 때문이다. 이를 해결하기 위해서 전송하는 데이터의 타입으로 JSONP를 이용하였다. GET 방식만 사용할 수 있기 때문에, 데이터양에 한계가 있지만, 센서 값만을 전송하는 IoT 환경에서는 문제가 되지 않는다. JSONP를 호출하기 위해서는 'callback'이라는 매개변수를 붙여서 호출해야만 하며, 서버 측 응답에서도 주어진 'callback' 매개변수로 응답해야 정상적인 통신이 이루어진다.

Thingplug 웹 서버도 Node.js를 이용하여 구축하였으며, 'express' 모듈을 이용하여 사용자가 원하는 요청을 처리했다. ThingPlug 웹 서버에서는 IoT 플랫폼에 의해 수집된 데이터를 XML 형식으로 어플리케이션에게 전송한다.

3-4 클라이언트

본 논문에서 제안한 시스템을 위한 어플리케이션을 구현해 보았다. 어플리케이션에서는 ThingPlug와 라즈베리파이에 주기적으로 온도 값을 요청하여 웹 브라우저에 그래프 형식으로 나타낸다. 그리고 받아온 온도 값을 지속적으로 확인하여 일정 온도로 내려가면 라즈베리파이로 LED를 켜야 한다는 요청을 전송한다.

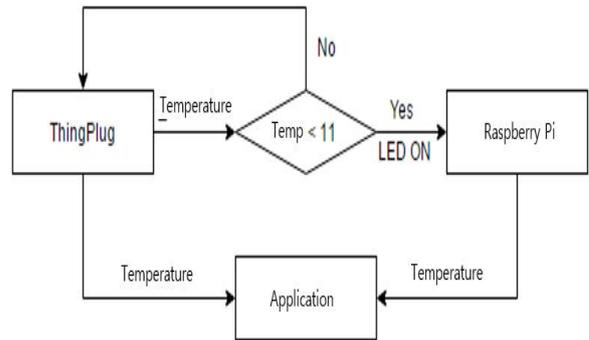


그림 4. 어플리케이션의 데이터 흐름도
Fig. 4. Application data flow diagram.

그림 4는 어플리케이션 내에서 이동하는 데이터의 흐름도를 나타낸다.

어플리케이션은 웹이나 모바일 등 언제 어디서든 접근이 가능하기 위해 반응형 웹으로 제작하였다.

어플리케이션은 보편적으로 사용되고 있는 HTML5를 이용하였기 때문에 디자인적인 구현방법보다는 어플리케이션과 웹 서버 간 요청에 대해 설명하고자 한다.

어플리케이션에서 웹 서버로 데이터를 요청할 때, ThingPlug와 라즈베리파이 모두 GET 방식을 이용하였다.

Thingplug에서는 Node.js의 express를 이용한 GET 방식을 이용했고, 라즈베리파이에서는 Ajax 통신을 이용하였다.

그림 5는 ThingPlug에서 측정된 온도 값에 따라서 어플리케이션에 표시되는 그래프이다. 그래프에 표시되는 값이 11도 이하일때마다 라즈베리파이에 설치되어있는 LED에 불을 켜야 한다고 요청할 것이다.

라즈베리파이의 웹 서버에서 온도 센서의 값이 수신되면 어플리케이션에서는 온도 값을 그래프로 표시한다. 표시된 온도 그래프는 그림 6과 같다.

그림 7은 ThingPlug의 온도가 11도 이하일 경우 LED가 켜졌을 때와 꺼졌을 때의 모습을 나타낸다.

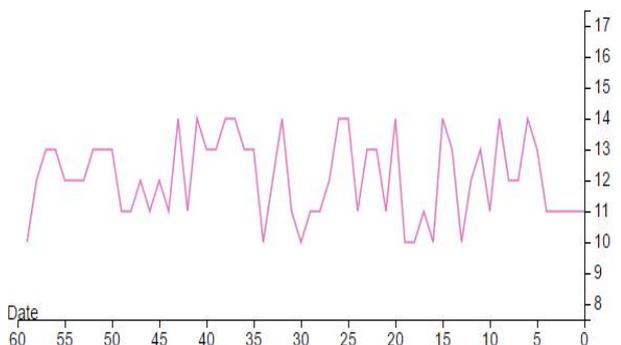


그림 5. ThingPlug의 온도 그래프
Fig. 5. Temperature graph of ThingPlug.

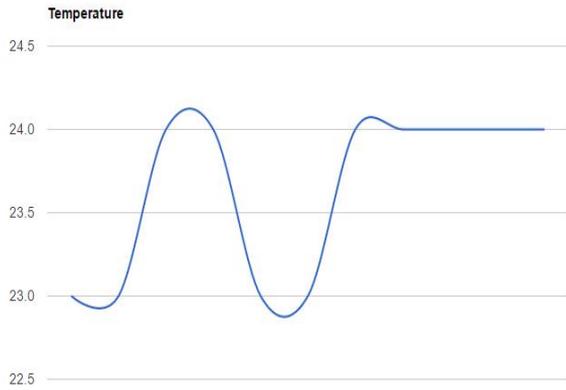


그림 6. 라즈베리파이의 온도 그래프
 Fig. 6. Temperature graph of Raspberry Pi.

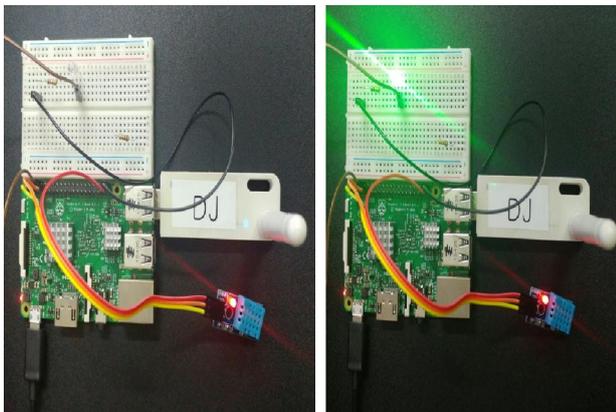


그림 7. 요청에 따른 라즈베리파이 LED
 Fig. 7. LED Raspberry Pie on request.

3-5 전송 속도 측정

본 시스템의 성능 평가를 위해서 IoT 디바이스에서 어플리케이션까지 센서의 데이터가 전송되는 시간을 측정하였다. 전체 시스템에 구간이 많아 데이터 전송 시간을 한 번에 측정할 수 없기 때문에, 시스템의 구간 별 데이터 전송 시간을 측정하여 각 구간의 합을 이용하였다.

그리고 다른 브라우저에 의해 네트워크가 방해 받는 것을 예방하기 위해서 모든 브라우저를 종료하였고, 시스템을 구동함과 동시에 측정을 시작하였다. 측정 방법은 시스템 각 구간의 데이터 전송 부분에 시간을 출력하는 함수를 직접 입력하여 측정하였다.

같은 조건에서 두 시스템을 측정하기 위해 제안 시스템을 먼저 측정하고, 모든 프로세스를 초기화한 후, 비교 시스템의 전송 시간을 측정하였다.

측정 범위는 시스템 구동 시작 3시간, 12시간, 24시간, 48시간이다.

표 1은 평균 데이터 전송 시간 결과를 나타낸 것이다.

표 1. 데이터 전송속도 측정

Table 1. Measure the data transfer rate.

Hour	ThingPlug	Raspberry Pi
3	23.808ms	54.523ms
12	23.526ms	55.234ms
24	23.268ms	55.356ms
48	22.795ms	55.324ms

ThingPlug와 라즈베리파이의 전송 시간은 유선과 무선 네트워크로 인해 차이가 발생했다.

IV. 결 론

본 논문에서는 방대해지고 있는 IoT 환경에서 각 업체 간에 발생하는 사일로 현상을 극복하기 위한 기술로 웹 기반의 개방형 IoT 시스템 즉, WoT 시스템을 제안하고, 구현하였다. IoT 디바이스에서 IoT 플랫폼을 거쳐 웹 서버를 통해 어플리케이션까지 데이터가 전송되고, 어플리케이션으로 전송된 데이터를 이용하여 다른 IoT 플랫폼에 연결되어 있는 IoT 디바이스를 제어한다. WoT에서는 여러 가지 웹 기술을 활용하여 시스템을 구축할 수 있지만, 본 논문에서는 주로 간단한 데이터를 빠르게 전송할 수 있는 Ajax 통신과 파싱이 빠르고 간단한 구조를 지니고 있는 JSON을 이용하여 각 IoT 플랫폼 간에 데이터를 상호 교환하였다. 그리고 각 IoT 플랫폼에 웹 서버를 구축함으로써, 실시간으로 IoT 디바이스의 센서를 빠르게 수집할 수 있었다. 추후 기존 WoT 시스템과 비교하여 성능 비교분석을 한 후, 전송속도 향상 방안에 대해 연구하고자 한다.

본 논문에서 제안하는 개방형 IoT 시스템을 이용하여 현재 또는 앞으로 이슈가 되는 IoT 환경에서 각 업체 간의 사일로 현상을 해결하고, 앞으로 IoT가 표준화되어 기술이 발전하고, 사용자에게 편리함을 줄 수 있는 기술이 등장할 수 있기를 기대한다.

참고 문헌

[1] S. H. Kim, "Internet of things technology", *The Institute of Electronics and Information Engineers*, Vol. 43, No. 3, pp. 64-71, March. 2016.

[2] J. S. Song, S. C. Choi, Martin Ting Miao, H. C. Park, and J. H. Kim, oneM2M application developer guide standardization, *Telecommunications Technology Association*, 2015.

[3] M. K. In, K. C. Lee, and S. Y. Lee, Web of things(WoT) standardization trends, *Telecommunications Technology Association*, pp.79-84, 2011.

[4] D. Raggett, "The web: the next 25 years", *The IEEE Computer Society*, Vol. 48, No. 5, pp.26~32, May. 2015.
[5] D. Zeng, S. Guo, and Z. Cheng, "The web of things: a survey", *Journal of Communications*, Vol. 6, No. 6, pp.

424~438, September. 2011.
[6] L. Yao, Q. Z. Sheng, and S. Dustdar, "Web-based management of the internet of things", *IEEE Internet Computing*, Vol. 19, No. 4, pp.60~67, June. 2015.



표 경 수 (Gyung-Soo Phyo)

2015년 8월 : 한국기술교육대학교 컴퓨터공학과 졸업 (공학사)
2015년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과
※ 관심분야 : IoT, 웹, WoT



박 진 태 (Jin-Tae Park)

2013년 8월 : 한국기술교육대학교 컴퓨터공학과 졸업 (공학사)
2015년 8월 : 한국기술교육대학교 대학원 컴퓨터공학과 졸업 (공학석사)
2015년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과
※ 관심분야 : 웹OS, HTML5, 자바스크립트



문 일 영 (Il-Young Moon)

2000년 2월 : 한국항공대학교 항공통신정보공학과 졸업 (공학사)
2002년 2월 : 한국항공대학교 대학원 항공통신정보공학부 졸업 (공학석사)
2005년 2월 : 한국항공대학교 대학원 정보통신공학과 졸업 (공학박사)
2004년 ~ 2005년 : 한국정보문화진흥원 선임연구원
2005년 3월 ~ 현재 : 한국기술교육대학교 컴퓨터공학부 교수
※ 관심분야 : 무선 인터넷 응용, 무선 인터넷, 모바일 IP