

WoT 환경에서 웹과 IoT 플랫폼 사이의 데이터 통신 연구

A Study of Data Communication Between the Web and the IoT Platform in WoT

표경수* · 박진태 · 문일영
한국기술교육대학교 대학원 컴퓨터공학과

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

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

[요 약]

최근 사람과 사물, 사물과 사람을 연결하는 IoT (internet of things)가 주목받고 있으며, 빠른 속도로 발전하고 있다. IoT가 발전하면서 전문가들은 10년 내에 인터넷에 연결된 디바이스는 약 1000억 대를 돌파할 것으로 예상하고 있다. 하지만, 현재 IoT 업체들 사이에서는 서로 다른 업체의 IoT 플랫폼 간에 데이터를 상호작용 하기 어려운 상태인 사일로 현상(ssilos effects)을 겪고 있다. 이러한 사일로 현상은 IoT가 더욱 큰 시장으로 성장하는 것을 제한할 것이다. 따라서 IoT 서비스를 공급하는 업체에서는 각 IoT 플랫폼 간 데이터를 상호작용 할 수 있는 개방형 IoT 플랫폼을 개발해야 한다. 현재 개방형 IoT 플랫폼을 위해 웹 기술이 급부상하고 있다. 개발자들에게 친숙하고 배우기 쉬운 웹 기술을 이용하면 IoT 디바이스를 발견하고, 수집된 데이터를 다양한 서비스에 활용할 수 있으며, 네트워크에 문제가 발생했을 경우 우회할 수 있다. 따라서 본 논문에서는 이러한 사일로 현상을 극복하고자 하는 WoT 환경에서 사용될 수 있는 데이터 통신 기술에 대해 알아보고 비교분석하였다. 그 결과, IoT의 정보를 단순하게 주고받을 수 있는 폴링 방식에서 빠른 전송 속도를 얻을 수 있었다.

[Abstract]

Recently, IoT(Internet of Things) to connect the objects and people has attracted attention. And it is rapidly developing. IoT is to develop, experts are predicting that the device is connected to the Internet to break through around 100 billion within 10 years. However, the current IoT companies are having a difficult state to the other each other IoT platform for interaction between data silos effects. These silos effects will limit the IoT grow to a larger market. Thus, the IoT service supplier shall develop an open platform for interaction IoT data between each IoT platform. These web technologies for current open IoT platform has emerged. If developers are using familiar, easy-to-learn Web technologies, developers can take advantage of a variety of services for the collected data, and found the IoT devices can be bypassed if there is a problem on the network. In this paper, we investigate the data communications technology that can be used in WoT environment to overcome these silos effects. As a result, the polling method that can simply send the IoT information the fastest.

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

<http://dx.doi.org/10.12673/jant.2016.20.4.374>



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 July 2016; Revised 27 July 2016
Accepted (Publication) 23 August 2016 (30 August 2016)

*Corresponding Author; Gyung-soo Phyo

Tel: +82-10-7706-5260
E-mail: gkgkgy@koreatech.ac.kr

I. 서론

각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술을 사물인터넷 (IoT)이라고 한다. 최근, 인터넷과 스마트 디바이스의 발전을 통해서 약 수십 억 개의 사물이 인터넷에 연결되고, 이러한 IoT를 기반으로 한 서비스가 폭발적인 성장을 하고 있다. 전문가들은 앞으로 10년 이내에 인터넷에 연결된 IoT 디바이스가 약 1000억 대를 돌파할 것으로 예상하고 있다[1].

IoT 디바이스와 서비스가 폭발적으로 등장하고 있는 현재 상황에서의 문제점은 각 IoT 업체 별로 사일로 현상을 겪고 있다는 점이다. 사일로 현상은 곡식을 저장해두는 원통형 모양의 독립된 창고인 ‘사일로’에서 생긴 경제용어이다. 조직끼리 사일로처럼 담을 쌓고, 조직 단위로 인프라와 자체 솔루션을 구축하고, 자기 조직의 이익만을 추구하는 현상을 말한다. 이러한 사일로 현상은 폭발적으로 성장 중인 IoT 환경을 제한할 것이다. 또한 이는 IoT 개발자와 사용자 측면에서도 많은 단점이 존재한다. IoT 서비스를 개발해야 하는 개발자는 특정 플랫폼에 맞는 서비스를 개발하기 위해서 해당 플랫폼을 학습해야 한다. 또한 IoT 서비스를 이용해야 하는 사용자는 자신이 원하는 IoT 서비스들이 다른 업체에서 개발되었다면 서비스를 이용할 때 발생하는 비용이 증가하게 될 것이다[2]. 이를 방지하기 위해 개방된 표준을 기반으로 하는 개방형 IoT 플랫폼이 연구되어야 할 것이다. 현재 연구되어지고 있는 개방형 IoT 플랫폼에서 주목 받고 있는 것은 웹 기술이다. 웹 기술은 그 자체에서 강점을 많이 갖고 있다. IoT 측면에서 바라보았을 때, 웹 기술을 이용하여 IoT 디바이스를 검색할 수 있고, HTTP프로토콜을 이용하여 통신이 가능하다. 그리고 웹 기술을 기반으로 하여 다른 위치에 설치되어 있는 방화벽을 넘나들 수 있다. IoT 환경에서 가장 문제되고 있는 보안 측면에서도 네트워크에 문제가 발생했을 경우 우회하여 보완할 수 있다. 또한 웹 기술은 개발자 측면에서 친숙하고, 배우기 쉬운 언어이기 때문에 개발비용을 절약할 수 있다. 그리고 사용자 측면에서는 평소 웹을 가장 많이 사용하여 익숙하기 때문에 거부감을 덜어준다[3].

따라서 본 논문에서는 위에서 언급했던 IoT 환경에서의 문제점인 사일로 현상을 극복하기 위해 웹 기술을 활용한 개방형 IoT 플랫폼인 WoT 환경에서 사용될 수 있는 데이터 통신 기술에 대해 알아보려고 한다.

II. WoT 연구 동향

본 장에서는 웹 기반의 개방형 IoT 시스템인 WoT의 개요와 WoT의 연구 동향과 관련 기술에 대해서 알아보려고 한다.

2-1 WoT 개요

최근 IT가 발전함에 따라서 사용자들은 언제 어디서든지

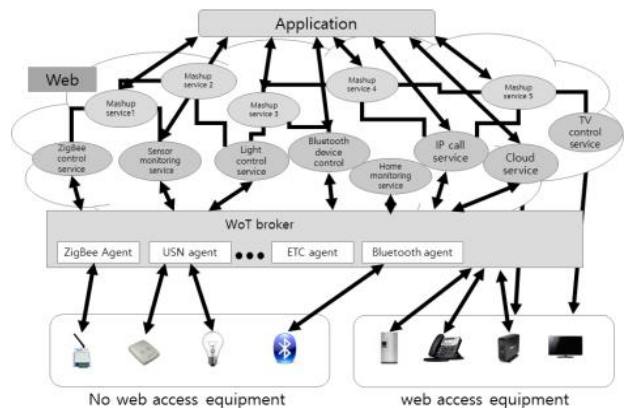
인터넷에 접속하는 것을 요구하였고, 클라우드와 같은 서비스가 등장하였다. IoT기술에 대해서도 사용자는 언제 어디서든지 사용자의 집에 설치되어 있는 스마트 디바이스나 회사 건물에 설치되어있는 디바이스에 접속하여 제어하는 것을 요구한다. 그래서 웹 기술을 활용한 IoT 서비스인 WoT가 연구되었다. WoT는 사물에 URI (uniform resource identifiers)를 부여하고, 웹 프로토콜인 HTTP를 이용하여 IoT 디바이스를 탐색하고, 사물에서 수집된 정보를 가져와 서비스 하는 것을 말한다. WoT를 위해서는 오픈 웹 API 기반의 스마트 디바이스를 이용한다. 이를 통해 사용자는 스마트폰이나 컴퓨터에서 웹 브라우저를 이용하여 자신이 소유하고 있는 IoT 디바이스에 접근하고, 해당 디바이스를 제어할 수 있다[4].

2-2 WoT 연구 동향

WoT는 2007년에 처음 등장하기 시작하였다. 그 이후 2009년에 SAP Research에서 ‘Web of Things Whitepaper’를 작성함으로써 WoT의 정의가 완성되었다. 이때부터 WoT에 대한 연구가 활발하게 진행되었고, 유럽을 중심으로 현재까지 연구가 진행 중이다.

1) 국외 연구 동향

국외에서는 ITU-T에서 가장 활발한 연구가 진행되고 있다. ITU-T에서는 주로 프레임 워크와 서비스 모델, 구조를 개발하고 있다. ITU-T는 개발 중인 WoT 프레임 워크를 활용할 수 있는 장비를 두 가지로 구분하고 있다. 첫 번째로는 ZigBee와 RFID, 블루투스를 사용하는 웹 접근이 불가능한 장비로 웹 프로토콜을 이용할 수 있는 WoT 브로커를 통해서 웹 서버에 디바이스의 데이터들이 저장된다. 두 번째는 HTTP 프로토콜과 WWW의 기능을 갖춘 웹 접근이 가능한 장비로 직접적으로 웹 프로토콜을 이용하여 웹 서버에 접속하여 디바이스의 정보를 저장한다. 그림 1은 ITU-T에서 개발 중인 WoT 프레임워크의 구조를 나타낸다.



* 출처 : ETRI

그림 1. ITU-T의 WoT Framework 구조

Fig. 1. The number of devices connected to each field in the internet of things.

W3C (World Wide Web Consortium)에서는 WoT의 interest group과 community group을 신설하여 WoT에 대한 연구를 진행하고 있다[5].

W3C의 interest group은 현재 IoT 플랫폼의 문제점인 플랫폼 간의 상호운용성을 해결하기 위한 연구가 진행 중이다. 그리고 그룹 내의 멤버들과 함께 WoT를 사용하면서 사용자가 겪을 수 있는 여러 가지 경우에 대해 인지하기 위하여 기술적인 논의가 진행되고 있다. IoT와 웹 데이터를 혼합하기 위한 웹 기술의 역할을 정하고, 이 역할을 기반으로 하여 응용 프로그램과 서비스를 연구하고 있다. Interest group은 WoT의 응용 프로그램을 개발하는 개발자를 위하여 독립적인 API의 플랫폼과 다른 플랫폼과 상호작용 할 수 있는 플랫폼을 개발하는 것을 목표로 하고 있다.

W3C의 community group에서는 WoT의 표준에 대한 논의와 연구가 진행되고 있으며, WoT 시스템을 작동시킬 수 있는 여러 가지 프로토타입이 연구되고 있다. Community group의 연구 목표는 다음과 같다.

- 요구 사항들을 인지하기 위한 기준으로 사용 사례를 수집
- WoT를 위한 구조를 설명하는 데이터 발견
- 기존 IoT의 표준의 적용 가능성 검토
- 새로운 표준이 적용되어야 하는 공간 식별
- 새로운 표준 개발
- WoT를 폭 넓게 인식시키기 위한 기회 발견
- WoT를 구현하는 개발자 커뮤니티에 적극 참여

W3C에서는 interest, community group 이외에 working group을 2016년에 신설할 계획이다[6].

2) 국내 연구 동향

국내에서는 2010년에 처음으로 ‘Smart IT를 통한 Smart KOREA 구현방향’ 보고서에서 처음으로 WoT에 대한 기술이 소개되었다. 하지만 이후, 국내에서는 해외만큼의 활발한 연구가 진행되고 있지 않다. 국내에서는 한국전자통신연구원(ETRI)에서 ITU-T와 IETF 표준 개발에 참여하고 있다. 그리고 한국정보통신기술협회(TTA) 산하의 Web project group에서 국내 표준에 대한 연구가 진행되고 있다.

현재 국내에서는 IoT를 이용한 연구가 활발하게 진행되면서 상용화된 IoT 제품이 등장하고 있다. 그리고 REST API를 이용한 웹 서비스에 대한 관심도 증가하고 있지만, IoT에 웹 기술을 적용시키는 연구는 활발하게 진행되고 있지 않다. 더욱 방대해지는 IoT 환경을 발전시키기 위한 방법을 국내에서도 연구해야 할 필요가 있다.

III. 통신 방법 연구

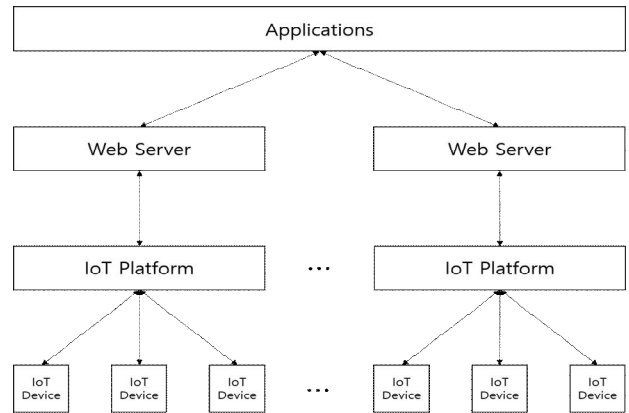


그림 2. WoT의 시스템 구성도
Fig. 2. System configuration of WoT.

본 장에서는 그림 2와 같은 WoT환경에서 IoT 플랫폼에 설치된 웹 서버와 어플리케이션 간의 데이터 통신을 위한 여러 가지 기법에 대해 알아보고, 비교 분석하고자 한다.

3-1 폴링 (Polling)

폴링 방식에는 2가지가 존재한다. 첫 번째로 주기적인 폴링 방식이다. 주기적인 폴링 방식은 클라이언트, 서버 모두 구현이 단순하다. 요청에 대한 서버 부담이 크지 않거나 실시간 메시지 전달이 크게 중요하지 않은 서비스에 적합하다. 실시간으로 전송되는 메시지의 중요성에 따라 요청 주기를 조절할 수 있지만, 요청 주기가 짧으면 서버에 무리를 줄 수 있기 때문에 주의해야 한다.

그리고 메시지 전달을 고려해서 요청 주기를 짧게 설정하여도 서버의 상태가 자주 변경되지 않으면 불필요한 트래픽이 발생한다.

표 1. getJSON, Ajax 방식 주요 코드

Table 1. the main code of getJSON, Ajax.

Method	Example
getJSON	<pre>\$.getJSON(' URL ', function (data) { Data process });</pre>
Ajax	<pre>\$.ajax({ type:'get', url: 'URL', contentType: "application/json", success: function(data){ setTimeout(function name, 2000); }, error:function(err){ console.log(err); } });</pre>

두 번째로 롱 폴링(Long polling)은 기존의 폴링 방식과는 조금 다르다. 기존의 폴링 방식이 주기적으로 클라이언트가 서버에 요청해서 응답을 받는 방식이라면, 롱 폴링 방식은 클라이언트의 요청을 서버에서 가지고 있다가 이벤트가 발생하면 응답하는 방식이다. 롱 폴링 방식은 주기적인 폴링 방식에 비해 불필요한 요청/응답 트래픽이 덜 발생한다. 롱 폴링 방식은 서버의 응답을 무한정 기다리는 것이 아닌 특정 시간이 지나면 새로운 요청을 하는 방식을 사용하고 있기 때문이다.

1) Ajax 폴링

Ajax (asynchronous Javascript And XML)는 대화식 웹 어플리케이션의 제작을 위해 서버 측 스크립트와 통신하는 XMLHttpRequest 객체를 사용하는 것을 말한다. Ajax를 통해 서버 측으로 다양한 형식 (JSON, XML, HTML 및 일반 텍스트 등)의 정보를 주고받을 수 있다. Ajax는 페이지 이동 없이 고속으로 화면을 전환할 수 있고, 처리를 기다리지 않는 비동기 요청이 가능하며, 수신하는 데이터양을 줄일 수 있어서 클라이언트에게 처리를 위임할 수 있다.

Ajax 폴링은 기존의 전통적인 폴링 방식으로 사용된 getJSON 방식에 비해 클라이언트 측면에서 예외처리와 속성을 제어하는 방법에서 더욱 안정적으로 데이터를 전송할 수 있다. 그리고 대화식 웹 어플리케이션 제작에 사용되기 때문에, 짧은 단순한 센서 데이터들을 송수신하는데 장점을 갖고 있다.

표 1에서 보는 것과 같이 getJSON은 간단한 방법을 통해 웹 Server에 저장되어 있는 센서의 데이터를 수집할 수 있지만, 예외처리를 서버 측에서 해결해야 하고, Ajax에서는 다양한 속성을 통해 예외처리를 할 수 있다.

3-2 웹 소켓

웹 소켓은 컴퓨터 네트워크용 통신 규약의 하나이다. 웹 소

표 2. 웹 소켓 방식의 주요 코드
Table 2. The main code of Web socket.

Method	Example
Web Socket	<pre> var socket = new WebSocket('ws://URL'); socket.onopen = function (event) { " Connected " }; socket.onmessage = function (event) { " Data transfer " }; socket.onerror = function (error) { " Error " }; socket.onclose = function (error) { " Disconnected " }; </pre>

켓은 HTTP의 단점을 보완하기 위해 등장했다. HTTP는 클라이언트-서버 간 접속을 유지하지 않고 한 번에 한 방향으로 통신이 가능한 반이중 통신을 사용했다.

이 방법은 인터넷에서 클라이언트-서버 간에 주고받는 데이터양이 많아지면서 성능이 저하되었다. 또한 HTTP는 지나치게 많은 헤더 데이터를 갖고 있다. 이러한 HTTP의 단점을 보완하고자 HTML5에 웹 소켓이 포함되었다. 웹 소켓을 사용하면 더 이상 ActiveX를 사용하지 않고 TCP/IP를 구현할 수 있으며, 네트워크의 과부하를 줄일 수 있다. 헤더 문제에 있어서도 웹 소켓은 바이트 수준으로 압축이 가능하다. 웹 소켓은 실시간 양방향 통신이 가능하고, 수많은 동시 접속자를 수용할 수 있다. 이러한 특징을 이용하여 웹 소켓은 다양한 분야에서 활용될 수 있다.

표 2에서 보는 것과 같이 웹 소켓은 http://가 아닌 ws://를 사용한다. 그래서 웹 서버에서도 ws 프로토콜을 이용해야 한다. 데이터를 전송하기 전에 연결을 확립해야 하기 때문에 onopen, onclose 함수가 존재한다. 따라서 데이터를 전송하기 위한 다양한 경우에 대비하여 처리할 수 있다.

3-3 WebRTC

WebRTC (web realtime communication)는 Google, Mozilla 등에서 홍보하는 오픈 소스 프로젝트이다. 자바스크립트 API를 통해서 별도의 플러그인을 설치하지 않아도 실시간으로 커뮤니케이션이 가능하다. 웹 RTC는 음성이나 영상, 파일뿐만 아니라 데이터를 공유할 때 사용하며 웹 어플리케이션에서 구현된다.

WebRTC는 P2P (peer to peer)통신 방식을 이용하고 있으며, NAT 환경에서 브라우저 간 통신을 위하여 STUN (session traversal utilities for NAT)서버를 이용하여 P2P 방식을 구성한다. 멀티미디어 스트리밍 방식 역시 TURN (traversal using relays around NAT)서버를 이용하여 P2P 방식을 구성해야 한다. WebRTC에서는 MediaStream 혹은 getUserMedia를 이용하여 카메라 및 마이크의 접근이 가능하고, RTCPeerConnection을 이용하여 오디오 및 비디오 연결이 가능하다. 또한 RTCDataChannel을 이용하여 데이터 P2P 통신이 가능하다. 주요 소스 코드는 WebRTC의 Github (<https://github.com/webrtc>)에서 확인할 수 있다.

3-4 데이터 전송속도 비교

앞에서 설명한 폴링, Ajax, 웹 소켓, WebRTC의 각 특징을 표 3에 설명하였다. 각 통신 기법의 데이터 전송속도를 비교하기 위한 실험 과정은 다음과 같다. 첫 번째, 라즈베리파이에 온도센서를 연결하여 IoT 플랫폼을 준비하였다.

표 3. 데이터 전송 방법 비교

Table 3. Comparing the data transfer method.

Method	Feature
Polling	<ul style="list-style-type: none"> - Simple Code - Simple implementation both client and server - Error processing on the server side - It may cause a strain on the server
Ajax	<ul style="list-style-type: none"> - Error processing on the client side - Interactive communication - A simple data transfer - Browser security policy - Need cross-domain
Web Socket	<ul style="list-style-type: none"> - Full-duplex - A large number of connections - Scalable SOA - Not needed in transmitting a short data
WebRTC	<ul style="list-style-type: none"> - Various data processing (voice, video, data) - Limitations of browser support - Firewall problem(TURN, STUN server)

그림 3은 라즈베리파이를 이용하여 구성된 회로도이다. 두 번째, Node.js를 이용하여 라즈베리파이에 연결된 온도 센서의 값을 읽어왔다. 세 번째, 어플리케이션에서 IoT 플랫폼에 온도 데이터를 요청한다. 폴링과 Ajax 방식은 URL (uniform resource locator)을 이용하여 요청하였고, 웹 소켓과 WebRTC는 웹 서버와 연결을 확립한 후, 데이터를 요청하였다. 데이터 전송속도를 비교하기 위한 조건으로 상호 간 연결이 확립되는 시간은 배제하였다. 전송 속도는 웹 브라우저에서 서버 프로세싱, DNS (domain Name Service), TCP (transmission control protocol), 브라우저 프로세싱 시간을 나타내는 TTFB (time to first byte)를 이용하여 측정했다. 네트워크 환경에 따라 데이터 전송 속도는 다르지만, 총 50회의 데이터를 요청하여 평균 속도를 측정하였다.

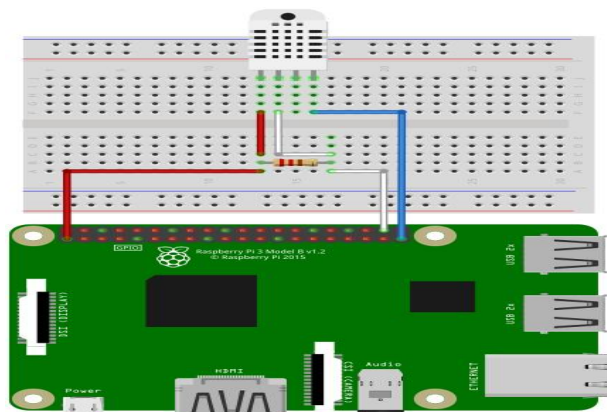


그림 3. 라즈베리파이 온도센서 회로

Fig. 3. Raspberry Pi temperature sensor circuit.

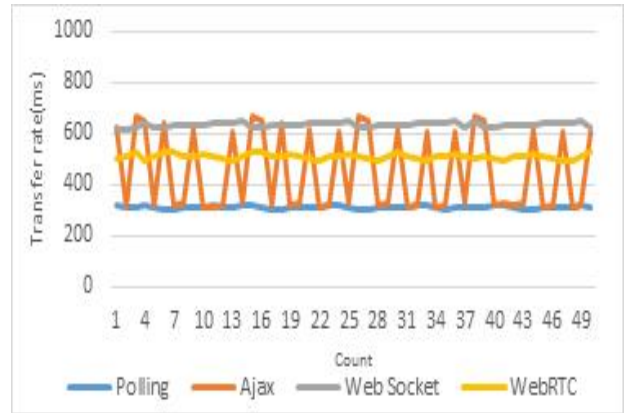


그림 4. 데이터 전송 속도 그래프

Fig. 4. Data transfer rate graph.

표 4. 데이터 평균 전송 속도

Table 4. The average data transfer rate.

Method	Average rate
Polling	311.82 ms
Ajax	451.94 ms
Web Socket	635.74 ms
WebRTC	511.6 ms

그림 4는 횟수마다 각 방식에서 측정된 데이터 전송속도를 나타낸다. 전통적인 폴링과, 웹 소켓, WebRTC는 비교적 안정적이고 비슷한 전송 속도를 꾸준히 보였지만, Ajax의 경우에는 약 33%의 확률로 TTFB가 높게 측정되었다.

표 4는 각 통신 기법의 평균 데이터 전송 속도이다. 전통적인 폴링의 경우에는 311.82 ms, Ajax는 451.94 ms, 웹 소켓은 635.74 ms, WebRTC는 511.6 ms의 평균 속도가 측정되었다. 통신 방식 중에서 전통적인 폴링 방식이 가장 빠른 속도를 나타내었고, 웹 소켓이 가장 느리게 측정되었다.

IV. 결 론

본 논문에서는 현재 IoT 플랫폼의 문제인 사일로 현상을 해결하기 위한 해결책인 WoT에서 사용될 수 있는 데이터 통신 기술에 대해 알아보았고, 비교 분석하였다. 네트워크 환경이나 IoT 플랫폼, 웹 서버 구현 방법에 따라 결과는 다르겠지만, 간단한 온도 센서의 데이터 전송의 성능을 비교 한 결과 전통적인 폴링의 데이터 전송 속도가 다른 통신 기법보다는 빠르게 나타났다. WoT는 IoT 디바이스와 IoT 플랫폼, 웹 서버, 어플리케이션의 모든 과정에서 데이터들이 중요하게 다루어져야 하고, 특히 전송되는 데이터들의 보안이 중요하다. 현재까지는 IoT 디바이스에서 수집되는 데이터의 양이 적고, 클라이언트와 서버 간의 데이터 요청 수도 적지만, IoT 발전에 따라서 WoT에서 다

뤄야하는 데이터양은 증가할 것이다. 데이터양이 다른 경우에 따라 다른 웹 데이터 전송 기법이 사용될 수 있다. 데이터가 전송되는 목적에 따라 다양하게 데이터를 처리할 수 있는 연구가 진행되어야 한다.

현재 IoT는 빠르게 발전하고 있으며, 앞으로 5G, SDN (software defined network)으로 인해 더욱 빠르게 발전할 것이다. 하지만, 이에 맞춰서 개방된 표준을 설립하지 않으면, IoT의 발전 속도는 제한될 것이다. 개방된 IoT 시스템을 위해서 다양한 나라의 연구단체에서 연구가 진행되고 있지만, 아직까지 국내에서는 미비하다. 전 세계적으로 IoT 기술이 부상하고 있다. 국내에서는 당장 앞에 보이는 IoT 기술을 연구하는 것보다는 미래의 IoT 기술의 발전에 대비하여 충분한 연구와 그에 맞는 지원이 이루어져야 한다.

참고 문헌

[1] D. Raggett, "The web of things: challenges and opportunities." *The IEEE Computer Society*, Vol. 48, No. 5,

pp. 26-32, 2015.
 [2] M. K. In, K. C. Lee, and S. Y. Lee, "Web of things (WoT) standard trend," *TTA Journal*, Vol. 138, pp. 79-84, 2011.
 [3] D. S. Kim, "WoT need for a platform for Secure Communication (Web of Things)," *The Proceedings of the Korea Electromagneto Engineering Society*, Vol. 24, No. 4, pp. 45-51, July 2013.
 [4] D. H. Kim, M. K. Kim, and Y. G. Hong, "IoT applied communication technology," *The Korean Institute of Communications and information Sciences Information & Communication Magazine - Open Lecture Series*, Vol. 32, No. 12, pp. 3-11, Nov 2015.
 [5] M. S. Kong, H. J. Chae, and B. H. Yoo, "Internet of things (IoT) technology trends and prospects," *Journal of the Korean Society of Mechanical Engineers*, Vol. 56, No. 2, pp. 32-36, Feb 2016.
 [6] Web of Things Interest Group: [Internet]. Available: <https://www.w3.org/WoT/IG/>



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