Interworking Proxy Based on OCF for Connecting Web Services and IoT Networks

Wenquan Jin[1] and Dohyeun Kim[2]
[1] Bigdata Research Center, Jeju National University, Jeju, South Korea
[2] Computer Engineering Department, Jeju National University, Jeju, South Korea
Email: wenquan.jin@jejunu.ac.kr; kimdh@jejunu.ac.kr

Abstract—The Internet of Things (IoT) is consist of heterogeneous hardware and software to enable connected devices with sensors, actuators, and applications for providing seamless services ubiquitously in our daily life. Most of the services on the Internet are provided by a high-performance processor, storage, and stable power supply. Nevertheless, IoT devices are developed for constrained environments using limited power and computational ability. In this paper, we propose an interworking proxy to bridge a constrained network to the Internet. The proposed interworking proxy is designed for Open Connectivity Foundation (OCF) network where IoT devices provide services using the OCF resources. In the OCF-based IoT network, clients access the IoT devices and web services based on the interworking proxy without considering underlying protocols. In order to access IoT devices and web services by clients, consistent registration and discovery interfaces are presented for providing services to register and discover the services based on the scheme of resource directory. For implementing the constrained network, the OCF IoTivity is included in the IoT devices to provide services.

Index Terms—Internet of things, interworking proxy, open connectivity foundation, web service, resource directory

I. Introduction

The Internet of Things (IoT) is an emerging technological paradigm to develop heterogeneous services in various industrial domains such as healthcare, healthcare, agriculture, retails, and factories. With the development of IoT technologies including communication protocols, frameworks, cloud services, and hardware platforms, a number of connected devices are deployed and will be increased to reach 20.4 billion by 2020 [1]. In the IoT networks, the heterogeneity of devices is a challenge to fulfill the requirement of the increasing number of devices with novel protocols and frameworks [2]. However, most of the web servers are developed using the HyperText Transfer Protocol (HTTP) to provide services through the Internet [3]. For constrained network environment, novel protocols are preferred which designed for low energy consumption and lossy packets in the communications [4], [5].

According to the deployment environment of IoT devices, clients can directly access the devices using physical Ethernet, Wi-Fi radio, or cellular modem [6]. In addition, a proxy can be used for forwarding a message packet from a client to a device although the client and device in a different network environment [7]. Many IoT devices are equipped with a limited power supply to support service ubiquitously. Therefore, these devices do not use pervasive network solutions [8], [9]. However, supporting the communications between IoT networks and HTTP-based servers is difficult because of the heterogeneity of IoT devices [10]. Supporting communications to the HTTP-based existing services of web servers and constrained IoT devices are important to build IoT systems for providing autonomous and intelligent novel services in industries.

For supporting protocol translation, registration, discovery, management, and other major functions, a proxy is a necessary network element to enable communications between heterogeneous networks [11], [12]. An Interworking proxy is an important network element to support interoperability in an IoT network [13]. The proxy aims to enable communication between heterogeneous networks in the IoT which [14]. In the Open Connectivity Foundation (OCF)-based IoT network, resources of OCF devices can be requested directly by the OCF clients [15]. However, the resources in the non-OCF network cannot be requested directly. The functionality of proxy is used for bridging the client and destination server.

The Resource Directory (RD) is a server that hosts information of IoT resources and provides discovery interface for retrieving registered information [16]. For a constrained IoT network, an RD server is used for enabling the IoT devices to need to be describable by other objects or services to access [17]. Discovery schemes can be considered as distributed and centralized for the IoT environment for the constrained network [18]. RDs are the discovery servers which can be deployed for a scalable distributed system that has multiple systems to handle the network [19].

In this paper, we propose an interworking proxy to enable communications between OCF-based IoT network and web servers from the Internet. The proposed interworking proxy is included in a server that provides services for registration and discovery of services including IoT services and web services. The web services are provided by web servers which can be

Manuscript received August 2, 2019; revised January 8, 2020.
Corresponding author email: author@hostname.org.
doi:10.12720/jcm.15.2.192-197

deployed in a cloud environment with high-performance equipment. In the OCF network, clients discover the registered services to access. Through the discovered service information, the IoT devices can be accessed by a client directly. For accessing the web services, the interworking proxy bridges the request from the client to the web server. Therefore, based on the proposed interworking proxy, Client devices are enabled to access IoT services and web services without considering changes of network environments.

II. IoT Architecture Based on Proposed Interworking Proxy

Fig. 1 shows proposed IoT architecture including IoT device, interworking proxy, web server, client device and user. The interworking proxy host its application, that includes functionalities of RD and interworking, and database for service information storage. Client device can be mobile devices, e.g., smartphones, tablet PCs, etc. The client run client device application for display UI and the response result. The web server is a web service provider, that supports REST API, the result data should be interpreted in the client device. A web server is used to provide services that the services are consumed by its clients. The IoT device can be constrained wireless devices with sensors, actuators to provide IoT services based on OCF resources, e.g., sensing services, actuating service, etc. The IoT device includes sensors and actuators for collecting data from its environment or the device do some works to influence the environment. In order to store device information, a data repository is needed. The storage can be a database as well as a file.

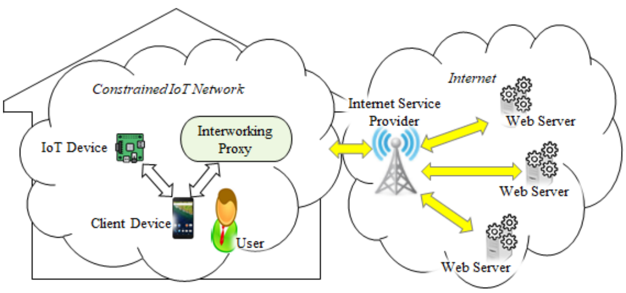

Fig. 1. Proposed IoT architecture based on interworking proxy.

Fig. 2 shows a components architecture for the proposed IoT system. In the client layer, the client device is used for interacting with proxy later and IoT device using the OCF communication, and the administrator device interacts with the proxy layer for the management. In the proxy layer, the interworking proxy with RD is included to provide services for registration, discovery and access. In the service provider later, IoT device and web server provide IoT services and web services.

The OCF network includes client devices which are used for discovering the information from the RD server, and, IoT devices are used for providing IoT services to the clients. In the internet, various web servers have been deployed to provide HTTP based services. The administrator uses a publishing client to register the resources of web servers to the RD server, due to the web servers cannot register the information by itself. For discovering and accessing those services from the OCF network and Internet, the client devices can request to the RD server using OCF communication protocol.

Administrator device is used for managing the information of IoT device and web server. For the registration, IoT device request to the RD to register IoT device information using a profile. Administrator inserts data to register web server information using a profile.

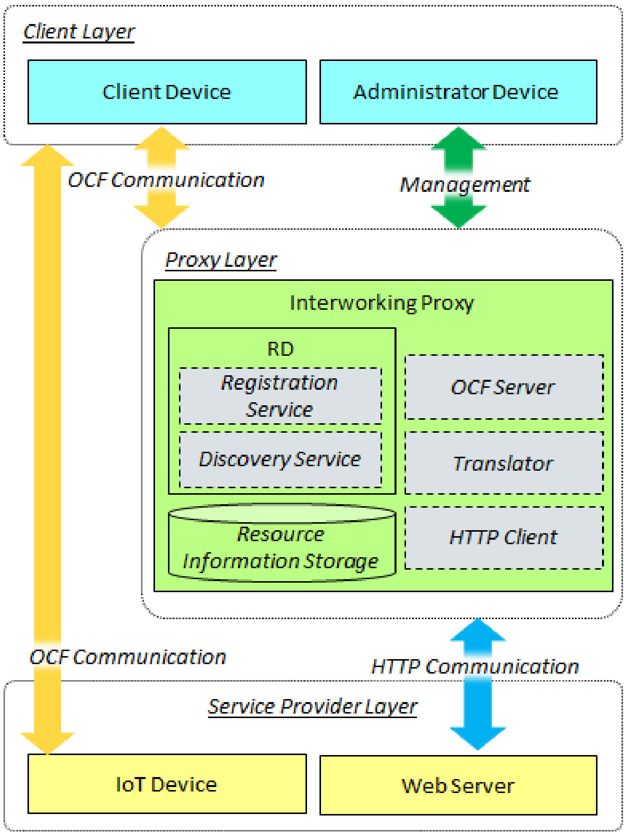

Fig. 2. Components architecture for the proposed IoT system.

III. Design of Proxy-Based IoT System

Fig. 3 shows the overall interaction scenario. Firstly, the resource information is registered by the IoT device and administrator in the registration process. Then, in the discovery process, the client device retrieves the resource list and detail information. In the service access process. The client device accesses the IoT device service directly and accesses the web service through the interworking proxy.

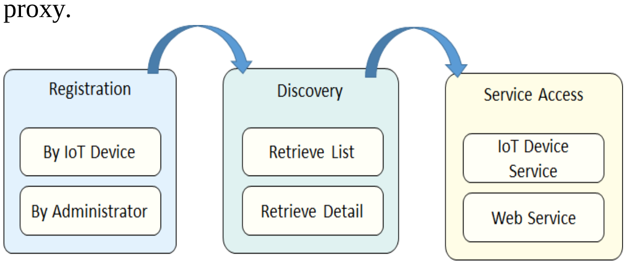

Fig. 3. Overall interaction scenario.

Fig. 4 shows a sequence diagram for a web service registration process. Web services are provided web servers which are deployed in the Internet to support storage and computing functions through services based on HTTP communication. The publishing client is used for registering service information of web services using the consistent registration interface of interworking proxy through the OCF communication network. Firstly, the publishing client reads the profile file to get the service information. Then, the publishing client gets the name and data from the profile to put in the payload of the OCF request message. The publishing client requests the message using the POST method with URI coap://{RD's IP:5683}/rd and the payload. Once the interworking proxy receives the message, the server parses the profile data and inserts it to the DB.

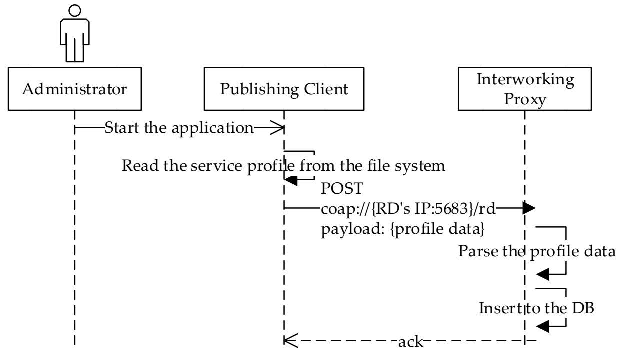

Fig. 4. Web service registration process.

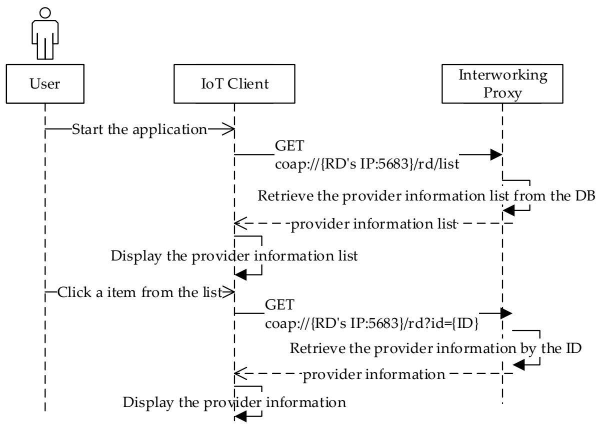

Fig. 5. Discovery process for registered IoT device and web server.

Fig. 5 shows the sequence diagram for discovering the registered IoT device and web server through their resources. In the response message, the payload includes the information of OCF resources as well as the HTTP resources which belong to IoT devices and web servers. The discovery service is consumed by the client device that requests to /rd/list resource of the interworking proxy with the query parameter using the GET method. The client device displays the resource list on the UI through the returned information. Then a user clicks an item from the list to request /rd resource of the interworking proxy with id query parameter using GET method for getting the detail information of a resource.

Fig. 6 shows the sequence diagram for accessing a web service through the proposed interworking proxy. The client device sends a request message to the interworking proxy using OCF communication. In this request, the method type is GET, and URI prefix is coap. Once the interworking proxy receives the request message, it generates an HTTP client for forwarding the message to the web server. The result from the web server is included in the HTTP response message. Once the HTTP message is delivered to the interworking proxy, then the proxy converts the HTTP message to the OCF message and forwards to the client device.

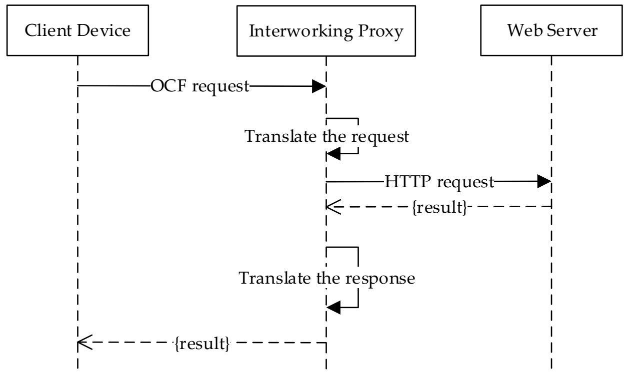

Fig. 6. Web service access process through interworking proxy.

IV. Implementation Results

For developing proposed IoT system, interworking proxy, IoT device and client device are implemented. For applying the interworking proxy, a weather service provider is used. The service provider provides various APIs for accessing the web server to get data.

TABLE I: DEVELOPMENT ENVIRONMENT

	Interworking Proxy	IoT Device	Client Device
Hardware	Intel Edison Board	Raspberry Pi 3 Model B	Samsung Galaxy S4
Platform	Android Things 0.2 (Build: compile SDK 25, min SDK 24)		Android 5.0 Lollipop (Build: compile SDK 25, min SDK 21)
Library and Framework	IoTivity 1.2.1(x86), raml-parser-2 1.0.13, jackson-core 2.9.0		IoTivity 1.2.1(armeabi), jackson-core 2.9.0

Table I presents the development environment for the proposed IoT system. Interworking proxy is implemented on the Intel Edison board with the Android Things 0.2 that is built by compiling Android software development kit (SDK) 25. The minimum version of Android SDK is required higher than 24. The implementation of IoT device is also based on Android Things. The hardware of IoT device is Raspberry Pi 3 Model B. For the client device, Samsung Galaxy S4 is the hardware and the runtime Operating System (OS) is Android 5.0 Lollipop that is built by compile SDK 25 and min SDK 21. For implementing OCF communications including OCF resources and client functions, IoTivity framework is used that is compiled on Ubuntu 16.4 64 bit for Android OS on x86 CPU. RESTful API Modeling Language (RAML) is used for registering the resources of IoT device web servers as the profile. For the RAML parser, a Java-based RAML parser library is used that supports RAML version 0.8 and 1.0.

For registering an IoT device, an RAML definition is used to illustrate the service information. According to the registered RAML definition, the IoT device provides two services through resoruces of /led and /temperature. The /led resource has handlers of GET and PUT for handling the requests from clients. The /temperature resource has GET handler for handling the request. For each handler of resources in this IoT device, the query parameters and response body are defined. The response body is defined using JSON schema, and example also is included in this RAML definition.

For registering an HTTP service provider, an RAML definition is used to illustrate the service information which are deployed in the Internet. The HTTP service provider provides one service through the /weather resource. The /weather resource has GET handler for handling the request. For the handler of resource in this HTTP service provider, the query parameters and response body are defined. The response body is defined using JSON schema, and example also is included in this RAML definition. The HTTP service provider is a weather service provider that provides several weather-related services using open APIs. The service that is included in the presented RAML definition is a server for providing current weather information.

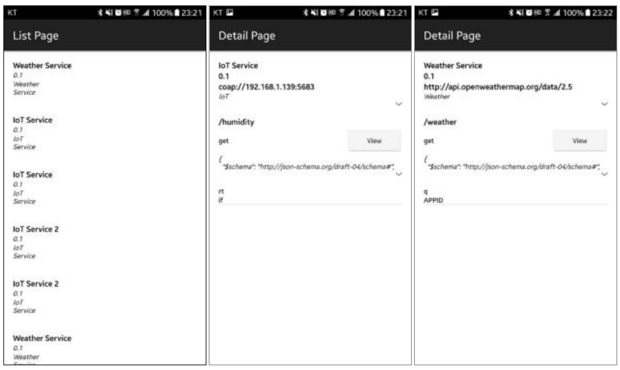

Fig. 7. Implementation results of displaying registered services.

Fig. 7 shows the implementation results of displaying registered services through the list page and detail page. The list page displays the registered services where each item of the list displays the profile name, version, and description. For requesting the list page, the client device needs to include query parameters in the request URI. The parameter is the OCF resource interface, and the parameter keyword is used to retrieve the list with the query. The request method is GET, and the handler of the method is in the RD server that is used for getting the service list. The detail page displays the registered service detail information. The detail page includes the resource list because a provider can include multiple resources. For requesting the detail page, the client device needs to include query parameters in the request URI. The parameter is on the OCF resource interface, and id is used for retrieving provider information by its ID. The request method is GET, and this method handler is in the RD server that is used for obtaining service information.

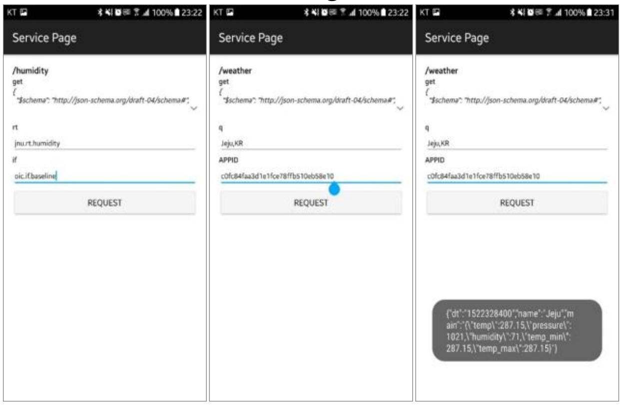

Fig. 8. Implementation results of access registered services.

Fig. 8 shows the results of accessing the IoT services and web services from the OCF network and Internet. The access page is displayed in the client device once the resource information discovered. For accessing a service, a user needs to input the parameters. For the OCF request to an IoT device directly, resource type and resource interface are required to input. For the HTTP request, the required query parameters are required to input. Client device is developed to recognize that the request is sent to an IoT device or a web server based on the information that is registered. The response message includes the payload that carries the result from resources. Once the result is delivered to the client device, the client displays the result on the screen. In the implementation result, the web server returns a JSON format message which includes weather information.

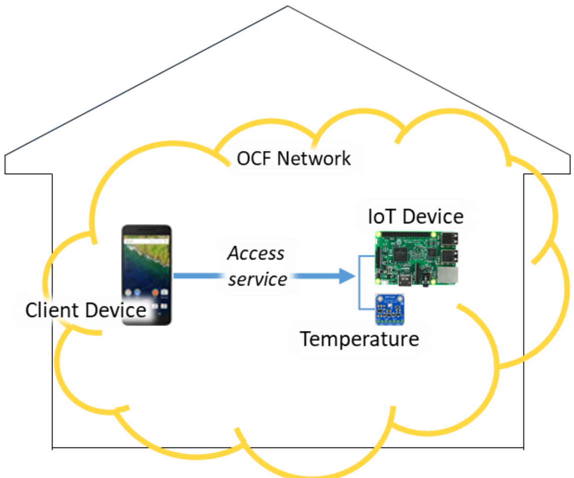

Fig. 9. IoT device sensing service access scenario.

Fig. 9 shows the IoT device sensing service accessing scenario an using a client device in the OCF network. The client device accesses the IoT device based on the information of the resource for the sensing service in the service access page. The IoT device is equipped with a BMP 280 sensor and collects temperature sensing data. It provides temperature sensing services through the /temperature resource. IoT devices is equipped with LEDs that can be turned on and off. It provides the services to turn on/off the LED through the /led resource. Communications between the client device and the IoT device proceeds over an OCF network.

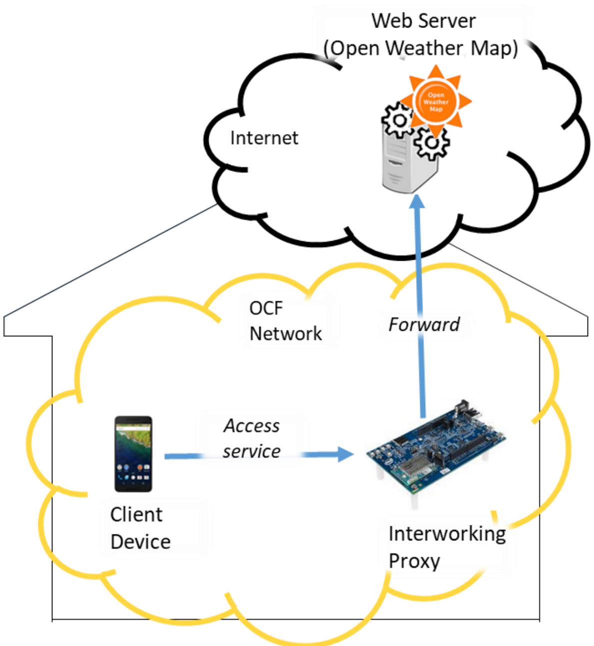

Fig. 10. Web server service access scenario.

Fig. 10 shows an architecture of the web server service accessing scenario through the interactions between client device, interworking proxy and web server. The web server service scenario of accessing the services of the WSP comprised of OCF network and the Internet. The client device accesses the current weather service from the service access page in the mobile device. The service request is made through the interworking proxy. The interworking device sends a request of the client device to the web server and returns a response message according to the request to the client device. Client device and interworking proxy are communicated in the OCF network, and interworking proxy and web server (weather service provider) are communicated through the Internet.

Fig. 11. Performance results of access web services.

Fig. 9 shows the performance results of access web services using the proposed interworking proxy. The proposed interworking proxy is used for accessing a weather service provider. From the service provider, Current Weather (CW), Weather Forecast (WF), Weather Daily Forecast (WDF) and Current Ultraviolet Index (CUI) are used for evaluating the performance of interworking proxy. The results are collected by interactions of client device, interworking proxy, and web server. WF part is provided by the IoT device directly through the OCF network.

V. Conclusions and Future Directions

For enabling communications between OCF-based IoT network and web services to access web services, an interworking proxy is proposed with consistent interfaces of service registration and discovery. The proposed interworking proxy is included in a server that provides services for registration and discovery of services including IoT services and web services. The web services are provided by web servers which can be deployed in a cloud environment with high-performance equipment. In the OCF network, clients discover the registered services to access. Through the discovered service information, the IoT devices can be accessed by a client directly. For accessing the web services, the interworking proxy bridges the request from the client to the web server.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Wenquan Jin and Dohyeun Kim designed the overall system. Wenquan Jin implemented the overall system and performed experiments. Wenquan Jin and Dohyeun Kim wrote this paper.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (NRF-2019R1I1A1A01062456), and this work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-01456, AutoMaTa: Autonomous Management framework based on artificial intelligent Technology for adaptive and disposable IoT). Any correspondence related to this paper should be addressed to Dohyeun Kim.

References

[1] L. Shancang, L. D. Xu, and S. Zhao. "5G internet of things: A survey," *Journal of Industrial Information Integration*, pp. 1-9, 2018.

[2] Akpakwu, G. Anuga, *et al*., "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619-3647, 2017.
[3] N. Nitin, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in *Proc. IEEE International Systems Engineering Symposium*, 2017.
[4] A. Al-Fuqaha, *et al.*, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.
[5] J. Wenquan and D. Kim. "Consistent registration and discovery scheme for devices and web service providers based on RAML using embedded RD in OCF IoT network," *Sustainability*, vol. 10, no. 12, p. 4706, 2018.
[6] W.Roy, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *Computer*, vol. 1, pp. 28-35, 2015.
[7] J. Wenquan and D. Kim, "Development of virtual resource based IoT proxy for bridging heterogeneous web services in IoT networks," *Sensors*, vol. 18, no. 6, p. 1721, 2018.
[8] R. M. Abdur, *et al*., "Middleware for internet of things: A survey," *IEEE Internet of things Journal*, vol. 3, no. 1, pp. 70-95, 2015.
[9] K. Vasileios, *et al*., "A survey on application layer protocols for the internet of things," *Transaction on IoT and Cloud Computing*, vol. 3, no. 1, pp. 11-17, 2015.
[10] Z. Qian, *et al*., "Iot gateway: Bridgingwireless sensor networks into internet of things," in *Proc. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2010.
[11] Z. Chang-Le, Z. Zhu, and R. Huang, "Study on the IOT architecture and gateway technology," in *Proc. 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science*, 2015.
[12] C. Hao, X. Jia, and H. Li, "A brief introduction to IoT gateway," in *Proc. IET International Conference on Communication Technology and Application*, 2011.
[13] G. André, F. Cercas, and N. Souto, "Design and implementation of an IoT gateway to create smart environments," *Procedia Computer Science*, vol. 109, pp. 568-575, 2017.
[14] M. Roberto, *et al*., "LEGIoT: A lightweight edge gateway for the internet of things," *Future Generation Computer Systems*, vol. 81, pp. 1-15, 2018.
[15] P. Soohong, "OCF: A new open IoT consortium," in *Proc. 31st International Conference on Advanced Information Networking and Applications Workshops*, 2017.
[16] J. I. N. Wen-Quan, Y. Hong, and D. Kim. "Design and implementation of node discovery and registration based on RD using IETF CoAP protocol in IoT environment," *International Journal of Control and Automation*, vol. 9, no. 5, pp. 151-160, 2016.
[17] J. Wenquan and D. Kim, "A sleep scheme based on MQ broker using subscribe/publish in IoT network," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 2, pp. 639-645, 2018.
[18] B. Djamaa, A. Yachir, and M. Richardson, "Hybrid CoAP-based resource discovery for the Internet of Things," *J. Ambient. Intell. Humaniz. Comput*., vol. 8, pp. 357–372, 2017.
[19] A. Boukhadra, K. Benatchba, and A. Balla, "Efficient distributed discovery and composition of OWL-S process model in P2P systems," *J. Ambient. Intell. Humaniz. Comput*., vol. 7, pp. 187–203, 2016.

Wenquan Jin received the B.S. degree in computer science from the Yanbian University of Science and Technology, China, in 2013, and the M.S. and Ph.D. degrees in computer engineering from the Jeju National University, Korea, in 2015 and 2019, respectively. He is currently a post-doc in Bigdata Research Center at the Jeju National University, Korea, since March 2019. His research work mainly focused on IoT protocols and frameworks, cyber-physical systems and intelligent services.

Dohyeun Kim received the B.S. degree in electronics engineering from the Kyungpook National University, Korea, in 1988, and the M.S. and Ph.D. degrees in information telecommunication the Kyungpook National University, Korea, in 1990 and 2000, respectively. He joined the Agency of Defense Development (ADD), from Match 1990 to April 1995. Since 2004, he has been with the Jeju National University, Korea, where he is currently a Professor of Department of Computer Engineering. From 2008 to 2009, he has been at the Queensland University of Technology, Australia, as a visiting researcher. His research interests include sensor networks, M2M/IoT, energy optimization and prediction, intelligent service, and mobile computing.