Computers, Materials & Continua
DOI: 10.32604/cmc.2023.020851
Article

Tech Science Press

Transparent Access to Heterogeneous IoT Based on Virtual Resources

Wenquan Jin[1], Sunhwan Lim[2], Young-Ho Suh[2], Chanwon Park[2] and Dohyeun Kim[3,*]

[1]Big Data Research Center, Jeju National University, Jeju, 63243, Korea
[2]Autonomous IoT Research Section, Intelligent Convergence Research Laboratory,
Electronics and Telecommunications Research Institute, Daejeon, 34129, Korea
[3]Department of Computer Engineering, Jeju National University, Jeju, 63243, Korea
*Corresponding Author: Dohyeun Kim. Email: kimdh@jejunu.ac.kr
Received: 11 June 2021; Accepted: 05 August 2021

Abstract: The Internet of Things (IoT) inspires industries to deploy a massive number of connected devices to provide smart and ubiquitous services to influence our daily life. Edge computing leverages sufficient computation and storage at the edge of the network to enable deploying complex functions closer to the environment using Internet-connected devices. According to the purpose of the environment including privacy level, domain functionality, network scale and service quality, various environment-specific services can be provided through heterogeneous applications with sensors and actuators based on edge computing. However, for providing user-friendly service scenarios based on the transparent access to heterogeneous devices in edge computing, a consistent interface shall be provided to deliver services from edge computing to clients. In this paper, we propose transparent computing based on virtual resources to access heterogeneous IoT devices without considering the underlying network configuration at the edge of the networks. For supporting transparent access to different edge computing environments through a consistent interface, the virtual resource of edge gateway is proposed to bridge the Internet and devices which are deployed on the edge of the network. The proposed edge gateway exposes the services of the Internet of Things devices to the Internet using virtual resources that represent the resources of physical devices. The virtual resources provide a consistent interface to enable clients to access devices in edge computing without considering underlying protocols. The virtual resource is generated by the resource directory in the edge gateway through the registration of a device. Based on the device registration, the device information is stored in the gateway to link virtual resources and devices for translating messages according to the destination protocols and identifying physical devices that are represented by virtual resources. Moreover, through collaboration with the service provider, the function of device discovery and monitoring is provided to clients.

Keywords: Internet of things; edge computing; edge gateway; transparent computing; virtualization; resource directory; virtual resource

1 Introduction

Internet of Things (IoT) is an emerging industrial paradigm that is comprised of a massive number of connected devices to enable developing intelligent and autonomous systems in various industrial domains such as factories, farms, buildings, cities, education, energy and healthcare. As expected, the Internet-connected devices had surpassed the human population 10 years ago, and are increasing rapidly to be deployed around our daily life for providing sensing and actuating services ubiquitously [1]. These devices are deployed with constrained power supply, processor, storage, and network capability in a small size of equipment based on various technology providers [2,3]. Therefore, hardware limitation and heterogeneity are challenges for the increasing number of devices to provide seamless and intelligent services based on sufficient computation results [4–6]. For the limitation and heterogeneity of IoT devices, cloud computing can be a solution to be easily accessible and handle the massive IoT data which is generated by the amount of IoT devices [7–10]. Nevertheless, the network bandwidth and computation load are not sufficient to support a low-latency delivery of data between IoT devices and cloud platforms [11]. Therefore, bringing the computation and storage closer to the environment is important for supporting the constrained devices to process massive data immediately.

Edge computing has the potential to deploy computation and storage to the edge of the network for enhancing the performance of IoT devices to provide various services as well as improve data safety and privacy of the environment where the devices are deployed [12–14]. Edge computing expands the cloud to the network edge to improve the overall IoT infrastructure efficiency such as scalability, multi-tenancy, security, privacy, and flexibility [15]. As an intermediate layer between IoT and cloud, edge computing plays a crucial role to delivers the data between IoT devices and the Internet-connected elements such as clients and service providers for data visualization, computational offloading, data storing and retrieving, and device management [16–18]. According to characteristics of the edge layer, the element in this layer can be designed to assist cloud servers and IoT devices to extend the ability of computing, storage and networking for providing a sufficient computation in the network edge with low latency and high reliability [19–23]. The edge gateway is a device that includes processor, storage, and network modules to provide the function of computation and communication. Base on the characteristics of an edge gateway, that serves at the entry of edge computing for providing services to entities including web clients, cloud servers and IoT devices on the Internet as well as local networks [24–26]. For IoT systems, the edge gateway is deployed with IoT devices to provide sensing and actuating services to the Internet. The IoT services are exposed to the Internet through the edge gateway instead of heterogeneous IoT devices. Therefore, gateways can be developed to provide a consistent interface for handling requests to IoT devices.

Heterogeneity of devices in IoT is a challenge that requires an IoT client to install a corresponding communication approach for accessing the IoT device. Furthermore, for developing IoT entities to support management and communication, many solutions are proposed including protocols, frameworks, and specifications to implement applications and services [27]. However, the edge gateway can be deployed in the entry of edge computing to forward requests to the IoT devices based on translating the request messages to achieve transparent access without any configuration of communication settings on the client side [28,29]. Moreover, deploying the edge gateway can provide protocol translation, registration, discovery, management, and other major functions through interacting with elements in inside and outside of edge computing [30,31]. Therefore, accessing services of the edge gateway enables clients to achieve transparent access to IoT devices based on flexible and extendable characteristics of the edge gateway.

For supporting transparent access to heterogeneous IoT devices, in this paper, we propose an edge computing architecture based on virtual resources in the edge gateway to provide IoT services instead of IoT devices. Using the edge gateway in the entry of network edge, IoT services are delivered to clients through the proposed virtual resources which represent the resources of IoT devices such as the function of sensing and actuating. Therefore, virtual resources provide a consistent interface to clients to achieve transparent access to IoT devices in edge computing without considering underlying protocols. For generating the resources in the edge gateway on runtime, the resource directory provides the registration service to IoT devices. Through the device registration, the device information is used for mapping virtual resources and actual resources of the physical device. The virtual resources are exposed to the Internet to provide IoT services to web clients using a consistent interface. Clients access the virtual resources that bridge the client requests to the actual resources through translating messages according to the destination protocols. Also, through the IoT service provider, IoT services of edge gateways can be delivered to IoT clients based on the integration of other content using external APIs.

For developing the proposed edge computing architecture, IoT client, IoT service provider, edge gateway and IoT device are implemented. For implementing the communication between edge gateway and IoT device, Open Connectivity Foundation (OCF) is used which adopts the Constrained Application Protocol (CoAP) as the default protocol to provide services based on Representational State Transfer (REST) APIs [32]. The virtual resources provide services using Hypertext Transfer Protocol (HTTP) through translating the request and response messages between the Internet and IoT devices. Through the IoT service provider, map-based visualization is provided to IoT clients based on calling Google Map APIs, and multiple User Interfaces (UIs) are provided to presents information of virtual resources and sensing data.

The rest of the paper is structured as follows. Section 2 introduces the related works regarding IoT and edge computing frameworks including protocols, platforms and gateway solutions. Section 3 presents the proposed edge computing architecture, scenarios and methodologies. Section 4 presents the implementation details and results for the proposed gateway-centric IoT system including IoT clients, IoT service provider, edge gateway and IoT device. Section 5 presents the experimental results based on the implementations of proposed scenarios. Finally, we conclude this paper and introduce our future directions in Section 6.

2 Related Works

Many IoT frameworks have been proposed to provide intelligent, real-time and autonomous services in various industrial domains such as smart residential buildings, automatic production lines, and optimized greenhouses through interacting with heterogeneous connected devices. For communicating with connected devices, MQTT, AMQP, CoAP and RESTful HTTP are the most commonly used IoT protocols [33–35]. Therefore, the IoT platforms need to support multiple protocols. The oneM2M provides scalable and interoperable IoT standards to build a single horizontal service platform including the function of communication, management, storage and service [36]. Akasiadis et al. proposed an IoT platform that involves implementations of protocols in the server to support communications for multiple protocols [37]. However, an IoT platform based on a standalone server including many functions which is too heavy to deploy on the local environment for bridging the Internet and private space. Zamora-Izquierdo et al. proposed a flexible IoT platform using a three-tier open-source software platform at local, edge and cloud planes for deploying low-cost devices close to the environment [38]. The Lightweight M2M (LWM2M) and IoTivity are IoT frameworks to implement IoT applications that are proposed by Open Mobile Alliance (OMA) and OCF separately

[39,40]. These frameworks support communications through request and response transaction models based on CoAP that is used for constrained devices [41–43]. Therefore, these light-weight frameworks are enabled to be deployed close to the environment for supporting computational functions and data storage. However, a cloud-level service provider based on high-performance is required for proving various services to web clients such as visualization, management, and other integrated services.

Due to process the increasingly generated data and service requests at the edge of the networks, efficient computing solutions are proposed such as global edge [44], cloudlet [45], and fog computing [46] for bringing the computation to the environment where the data is generated and used. Global edge is proposed to provide an IoT platform architecture including multiple layers to reduce bandwidth costs through deploying between devices and cloud. Cloudlet is comprised of computers and abundant resources in a network to provide trust services through the service provider in a mobile device [47,48]. Satyanarayanan et al. proposed an open ecosystem for mobile-cloud convergence using the architecture of cloudlets for deploying a service provider to a mobile device that provides multiple services [49]. Fog computing is a local cloud computing platform to provide computation, storage and management service at the network edge based on distributed computing nodes [12]. Edge computing performs the computation and storage at the network edge for deploying devices close to the users to provide various services based on integrated multiple functions. The architecture of edge computing is proposed to satisfy key requirements such as scalability, multi-tenancy, security, privacy, and flexibility for novel IoT applications to provide ubiquitous and autonomous services [15]. The edge gateway is a middle layer element between IoT devices and cloud servers to enable characteristics of edge computing [20,21,24]. Through an edge gateway, the device resources are exposed to the Internet to enable accessing functions of sensing and actuating.

The edge gateway is deployed in the entry of the network edge that bridges IoT devices and the Internet to enable heterogeneous functions such as real-time sensing and actuating, computational offloading and transparent access. Mobile edge computing (MEC) is proposed to bring data service and cloud computing to the edge of a network to enable computational offloading and mobility management [50]. Industrial Internet Consortium (IIC) performs edge computing in a large-scale domain based on the layer of edge, platform and enterprise [51]. IIC considers the enterprise layer as an important element to handle the application process such as decision support systems, end-user interfaces, and operations management. Similar to GE, IIC deploys the most functions on the cloud. EdgeX framework is proposed for deploying management of device, data, and network on the edge gateway based on multiple server modules that provide microservices to allows services to scale up and down [52]. Using the EdgeX framework, an edge gateway can be satisfied to support flexible maintenance, sufficient performance and secure communication. However, deploying various microservice server modules to provide many functions including security and UIs in the edge gateway requires sufficient power supply, network bandwidth, and computing ability. Moreover, integrating data is difficult through the standalone management servers.

3 Proposed Edge Computing Based on Virtual Resources

3.1 Gateway-Centric IoT Architecture for Edge Computing Based on Virtual Resources

The proposed IoT architecture involves multiple edge computing environments that are operated based on edge gateways to expose the functions of IoT devices using virtual resources on the Internet. Based on virtual resources, IoT devices are accessed transparently from heterogeneous environments.

Fig. 1 presents the overall architecture of the proposed gateway-centric IoT system that is comprised of the Internet and network edges. In the proposed network edge, the edge gateway is an entry that provides sensing and actuating services to the Internet instead of IoT devices through virtual resources. With IoT devices, the edge gateway is deployed close to an environment where can be residential buildings, factories, and greenhouses to enable edge computing for proving heterogeneous domain-specific services. Through the edge gateway, web clients can access the edge of networks to collect sensing data, control actuators and deploy various scenarios for autonomous and intelligent tasks. Also, the request and response messages can be improved for the needs of both sides based on a further process in the edge gateway, such as translating request and response messages of virtual resources for heterogeneous IoT devices which have different communication protocols.

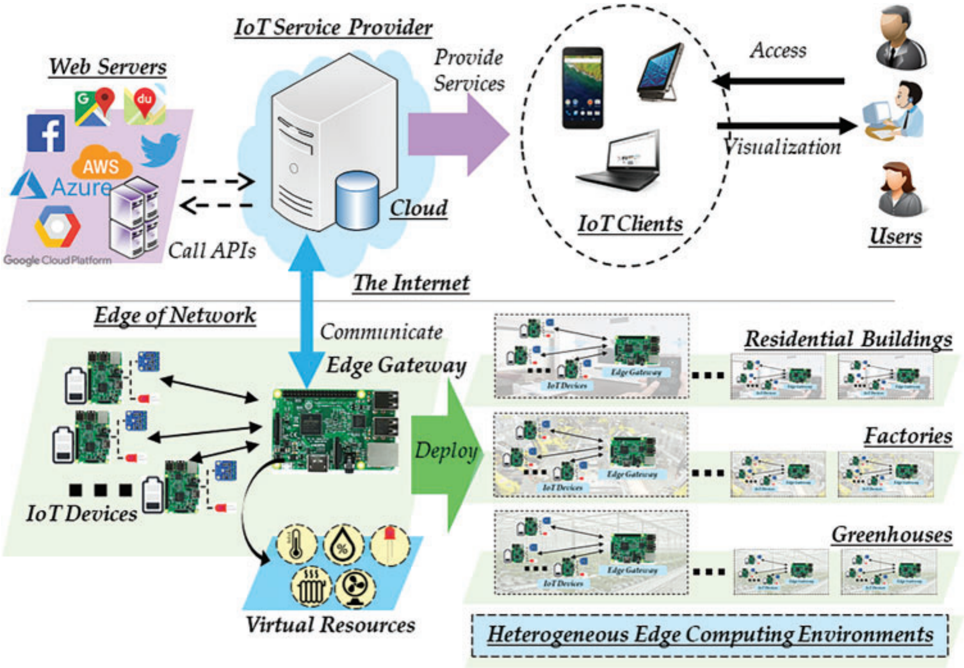


Figure 1: Proposed gateway-centric IoT system based on edge computing

For providing services to clients through the Internet, the IoT service provider is proposed to bridge users and edge computing based on an integrated visualization approach. The IoT service provider provides UIs to IoT clients based on web pages for visualizing data well-formatted form and presenting the information of edge computing on the map through calling map APIs from external web service providers. Using IoT clients, IoT devices are accessed by users. However, the visualized information represents the resources of IoT devices using virtual resources that are hosted in the edge gateways in heterogeneous edge computing environments.

Fig. 2 illustrates an IoT hierarchical architecture to present functionalities of the proposed IoT system through functional blocks in four layers including a layer of device, edge, service and client. The device layer includes IoT devices to provide sensing and actuating services using IoT resources. IoT resources are used for handling requests to enable interacting with the physical world using sensors and actuators. However, the functions of sensors and actuators are exposed through the resources of IoT devices that are constrained and heterogeneous. The edge layer enables access to the device layer through the edge gateway that is comprised of virtual resources and gateway functions. The edge gateway is deployed in the entry of network edge to provide services using virtual resources to the Internet instead of IoT devices. Based on the gateway functions, registration, discovery and monitoring are supported through interacting with both sides of the gate. The service layer includes the IoT service provider to provide services to IoT clients for presenting the information of IoT resources to the users. The IoT service provider calls the APIs from external web servers to provide enhanced user experiments such as displaying IoT nodes on the map. Through interacting with edge gateways, the IoT service provider provides visualization, control and monitoring using stored and real-time data. Base on the provided information from the IoT service provider, the client layer presents UIs to the users including information on virtual resources, data visualization, and map-based node locations.

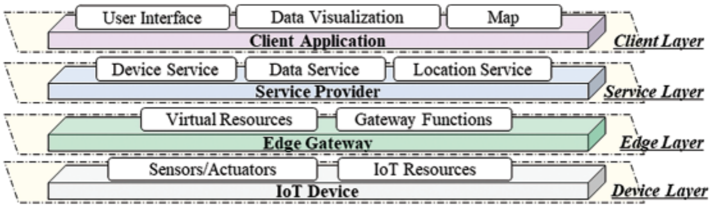


Figure 2: Proposed IoT hierarchical architecture

3.2 Edge Computing Management Scenario for Transparent Access

For supporting the transparent access to IoT devices through the proposed edge gateway, the management of IoT devices are proposed to provide registration, discovery, monitoring and access transparently. Fig. 3 depicts the sequence of registering and discovering IoT devices in the proposed IoT system. An IoT device can have one or multiple resources to provide various functions for sensing and actuating.

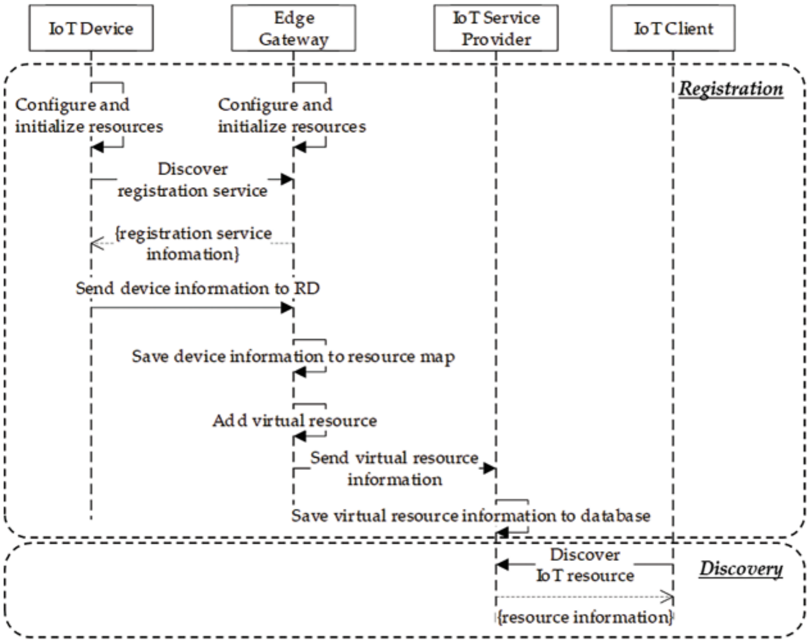

IoT Device
Edge Gateway
IoT Service Provider
IoT Client
Registration
Configure and initialize resources
Configure and initialize resources
Discover registration service
{registration service infomation}
Send device information to RD
Save device information to resource map
Add virtual resource
Send virtual resource information
Save virtual resource information to database
Discover IoT resource
Discovery
{resource information}

Figure 3: Registration and discovery of IoT device based on virtual resource

An edge gateway is also a device to provide multiple services using the resources that expose the services to the Internet and IoT devices through the same protocol or various protocols. IoT device and edge gateway configure and initialize the resources based on the deployed profile. Then, the IoT device discovers the registration service that is provided by RD in an edge gateway. Once the RD is found, the IoT device sends the device information to the RD for registering itself. The registering information includes resources information of the IoT device that is used for generating the virtual resources in the edge gateway. The IoT service provider saves the information of virtual resources to the database. The virtual resources are exposed to the Internet for being discovered by IoT clients. Through the registration process, the IoT resources are represented to the Internet for the discovery by IoT clients. IoT clients can access the IoT resources by accessing the edge gateway. The proposed edge gateway provides observation and notification mechanisms to collect the sensing data and device status from the discovered IoT devices.

Fig. 4 depicts the sequence of event observation and notification for monitoring an IoT device based on an edge gateway. The presented resource to an IoT client is a virtual resource that is provided by an edge gateway. The IoT client can discover the virtual resource and activate the monitoring function by sending a command to the edge gateway. Then the edge gateway retrieves the IoT resource from the resource map that is inserted by the registration process. Using the retrieved information, the edge gateway defines the request message, and sends to the IoT device to enable the observation of an IoT resource. Once the resource is observed successfully, the IoT service provider updates the observation status of the virtual resource, and presents to the IoT client. For notifying sensing data and status, the IoT client updates the presented information by requesting data from the IoT service provider periodically. The IoT device sends the updated data with the device ID and resource URI to the edge gateway, and the edge gateway updates the information based on the virtual resource to send to IoT service provider. The IoT service provider receives gateway ID, resource URI, and data, in which, the resource URI is used for identifying the virtual resource.

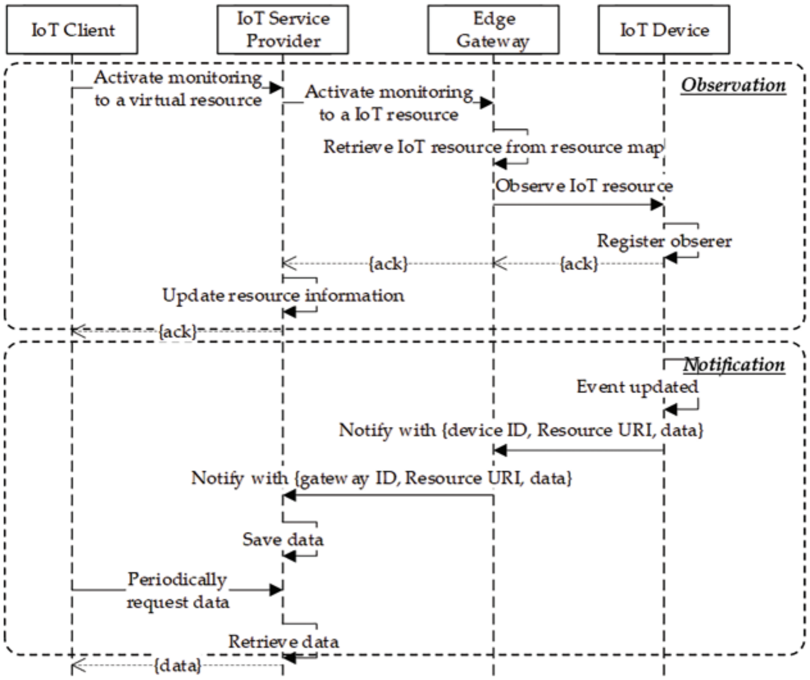

IoT Client
IoT Service Provider
Edge Gateway
IoT Device
Observation
Activate monitoring to a virtual resource
Activate monitoring to a IoT resource
Retrieve IoT resource from resource map
Observe IoT resource
Register observer
{ack}
{ack}
Update resource information
{ack}
Notification
Event updated
Notify with {device ID, Resource URI, data}
Notify with {gateway ID, Resource URI, data}
Save data
Periodically request data
Retrieve data
{data}

Figure 4: Event observation and notification for IoT device monitoring

Fig. 5 depicts the sequence of IoT resource access by an IoT client transparently based on a virtual resource. For delivering a request message to the IoT device, the IoT client sends the message to the IoT service provider, and the message is sent to the edge gateway. Then, the virtual resource handles the message for forwarding to the destination IoT resource. For translating the message from the original protocol to the destination protocol, the edge gateway retrieves the information of the destination IoT resource from the resource map, and translates the message. Then the resource generates a new message for properties for the destination protocol. Using the new formatted message, the edge gateway requests to the IoT device, and forwards the response message to the IoT client by setting the response message to the payload of the response message which is paired with the request from the IoT service provider. In this interaction, IoT client and IoT service provider are deployed on the Internet, which communicates through the public network. The IoT device is deployed in the local network, that is not enabled to access by the public network. The edge gateway is deployed in the entry of a local network, that has a public IP to enable communication with IoT service provider and IoT device.

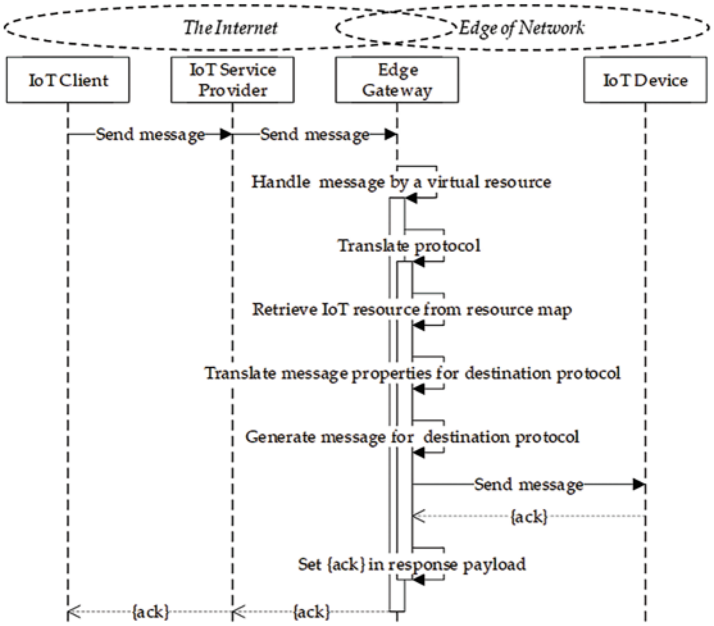

The Internet
Edge of Network
IoT Client
IoT Service Provider
Edge Gateway
IoT Device
Send message
Send message
Handle message by a virtual resource
Translate protocol
Retrieve IoT resource from resource map
Translate message properties for destination protocol
Generate message for destination protocol
Send message
{ack}
Set {ack} in response payload
{ack}
{ack}

Figure 5: IoT resource transparent access based on virtual resource

3.3 Proposed Methodologies for Edge Computing Management

In detail, IoT service provider and edge gateway include resources and functions to provide services for registration, discovery, monitoring and transparent access.

Fig. 6 illustrates the functional structure of the edge computing management including proposed characteristics. The IoT service provider is an element on the Internet, that is a server to be deployed with a public IP for providing services to IoT clients and edge gateways. The server provides the function of messaging, discovery, registration, monitor through web services. The discovery function is used for providing information of virtual resources to the IoT clients. The registration function is used for registering virtual resources to the IoT service provider. The discovery and registration functions interact with the database of IoT service provider to save and retrieve information of virtual resources.

The monitor function is used for providing data collection services to IoT clients through interacting with an edge gateway. The messaging function is used for delivering requests from IoT clients to edge gateways. In the network edge, multiple IoT devices can be deployed with multiple actual resources. Through the registration service, the edge gateway can have multiple virtual resources that are used for representing the actual resources to the Internet.

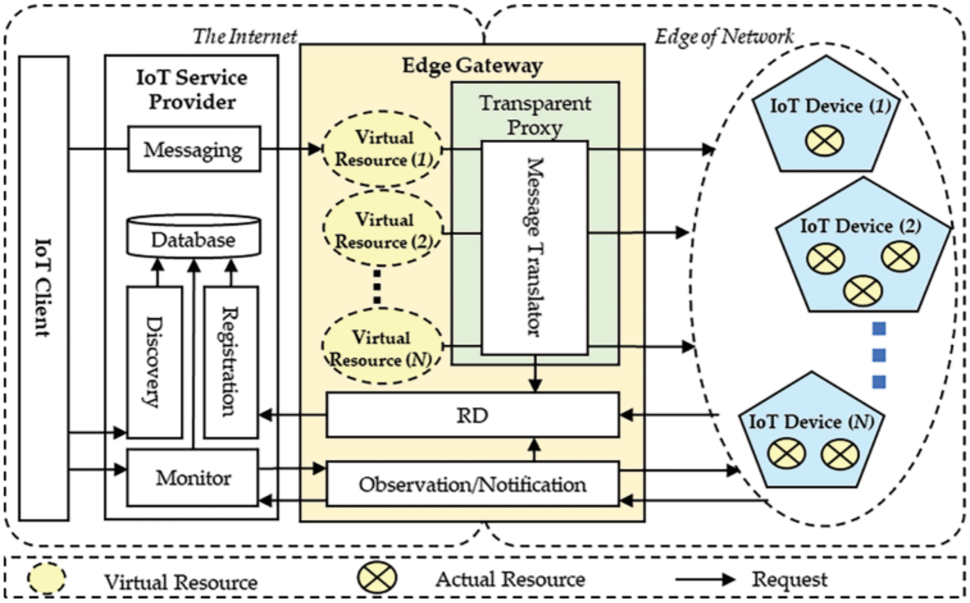


Figure 6: Functional structure of the proposed edge computing management in gateway-centric IoT network

The edge gateway includes virtual resources to provide transparent access based on the message translator. An IoT device includes resources that are registered to the edge gateway by the RD. The RD is used for providing the actual resource information to message translator and IoT service provider. The observation and notification functions set up the notification configuration to IoT devices by retrieving an actual resource by a virtual resource. Once the event is updated for the actual resource, the IoT device sends the updated data to the edge gateway and sends it to the database through the monitor function of IoT service provider.

Fig. 7 illustrates the functional structure of monitoring the IoT resources through the function of observation and notification. The IoT client requests to the monitor service of IoT service provider that sends a command to the IoT device through/observe resource of the edge gateway. Then, once the event data is updated in the resource of IoT device, the notifier sends to the data to the/notify resource of the edge gateway. Using the data service, the IoT service provider inserts the data to the database. The IoT client updates the UI by requesting the data service periodically, and the data service reads the data from the database.

Fig. 8 illustrates the functional structure of transparent access. The edge gateway provides sensing and actuator services to web clients based on virtual resources. Through the registration process, the edge gateway generates virtual resources using the virtual resource handler. The handler supports method of GET, POST, PUT, DELETE to handle requests through the virtual resource. Each method calls the message translator that uses the mapping table and Device client to generate the request to the IoT device. In this case, the IoT device is developed using the IoTivity and HTTP protocols for providing IoT services. For other protocols, message translator can use other libraries to generate the request. The mapping table links URI of virtual resources to the IoT resources. The message translator retrieves the destination resource information using the URI of a virtual resource.

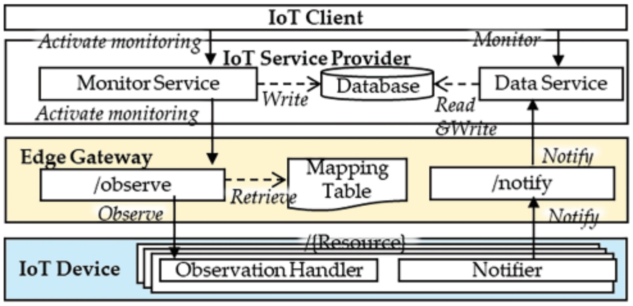


Figure 7: Functional structure of monitoring using the proposed edge gateway

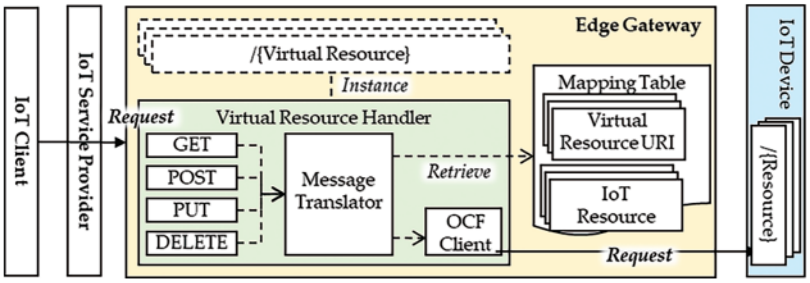


Figure 8: Functional structure of transparent access using the proposed edge gateway

4 Implementation Details

For presenting the transparent access based on virtual resources in gateway-centric edge computing, the IoT service provider, edge gateway, IoT device are developed and deployed.

The applications of edge gateway and IoT device are developed in Android, that are separately deployed on a device of Raspberry Pi 3 Model B based on Android Things 1.0 using SDK API Level 28. The edge gateway application includes Jetty 9.1.0.v20131115 to implement the HTTP server for providing services to the IoT service provider and Volley 1.1.0 to implement the HTTP client to access the IoT service provider. For experimenting the multiple protocols using the IoT device, the IoT resources are developed in IoTivity and HTTP to provide sensing and actuating services. Therefore, the edge gateway includes IoTivity 1.3.1 and Volley to implement the IoTivity and HTTP client to communicate with the IoT devices. The IoT device is developed for IoTivity and HTTP network environments based on IoTivity and Jetty libraries to implement the servers and calls the bmx280 library to use sensors in the Android Things platform.

The IoT service provider is deployed on a Windows 10 system using JDK 8 and Tomcat to run the server for providing HTTP services to edge gateways IoT clients. The IoT service provider application includes Spring Framework 4.3.3 to implement the server application, and call MyBatis library to interact with the MySQL database. The Commons File Upload 1.3.1 library is used for providing file upload service to create the plan of an indoor map. Apache HTTP Client 4.5.3 library is used for accessing edge gateways. The IoT client is implemented for Google Chrome 77 in Windows 10 system.

Nevertheless, the application can run on any web browsers including mobile phones. The jQuery 3.1.1 and Bootstrap v3.3.7 libraries are used for implementing the IoT client.

Fig. 9 shows the implementation scenario and components of the proposed IoT. The IoT device includes temperature and pressure sensor for collecting data from the environment, and fan actuator for changing the parameters of the environment. For these sensors and actuator, the IoT device provide resources to access these functions. The hosted resources are/temperature, /pressure, and/fan. Through the registration, the edge gateway generates virtual resources/vr/jnu-mcl-423-001/temperature, /vr/jnu-mcl-423-001/pressure, and/vr/jnu-mcl-423-001/fan. These virtual resources are exposed to the Internet where the IoT client can access through the IoT service provider. The IoT service provider is a server that provides the information of gateway and resource, sensing and actuator status data, and indoor map and location of gateway and resource based on map-based visualization. The IoT service provider provides UIs to the IoT client for visualizing data, forms, map to the users. With the UIs, the IoT service provider provides the outdoor map through calling map APIs from Google Map API server.

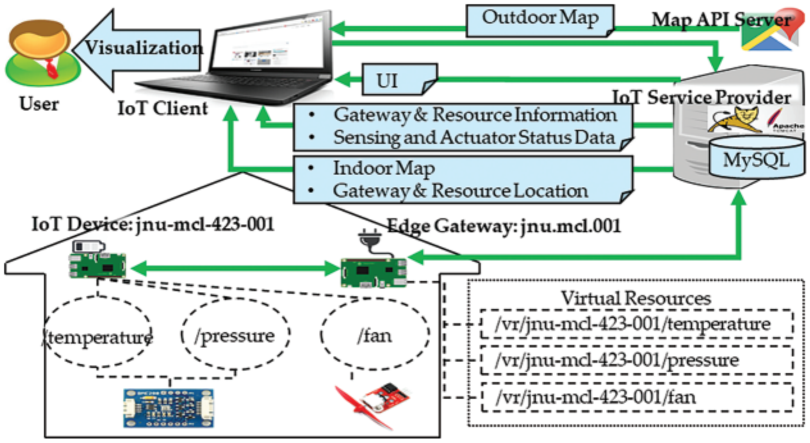
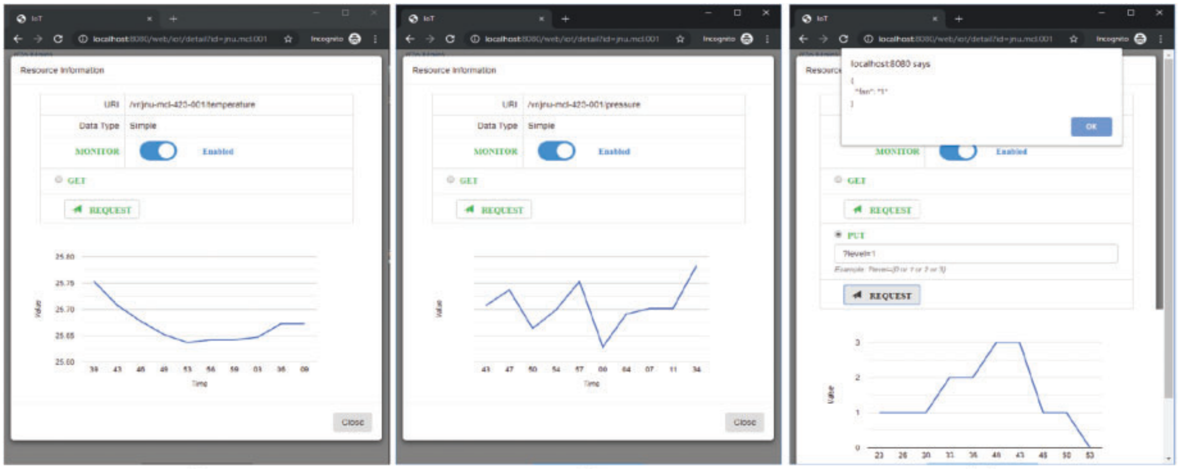

User
Visualization
IoT Client
Outdoor Map
Map API Server
UI
IoT Service Provider
• Gateway & Resource Information
• Sensing and Actuator Status Data
MySQL
• Indoor Map
• Gateway & Resource Location
IoT Device: jnu-mcl-423-001
Edge Gateway: jnu.mcl.001
/temperature
/pressure
/fan
Virtual Resources
/vr/jnu-mcl-423-001/temperature
/vr/jnu-mcl-423-001/pressure
/vr/jnu-mcl-423-001/fan

Figure 9: Implementation scenario and components of the proposed IoT architecture based on virtual resources of edge gateway

The IoT client is operated through a web browser that provides information of edge gateways on a list. Based on the registered information of edge gateways, the IoT service provider provides the search service using keywords that are used to retrieve edge gateways from the database. Once an edge gateway is clicked from the list, the IoT client presents detail information of the edge gateway that includes a list of virtual resources. In the virtual resource list, each item provides a button for presenting an UI to access the virtual resource. Fig. 10 shows the implementation result of monitoring resources that are included in the edge gateway as virtual resources to represent actual resources of IoT devices. Therefore, using these virtual resources to access the actual resources of IoT devices. The implemented IoT device includes resources of temperature, pressure, and fan that are represented in the edge gateway as virtual resources and identified by URIs. Through accessing the detail page of a virtual resource, the monitoring function is provided for the virtual resource of temperature, pressure, and fan. For the monitoring function of sensors, the page displays the sensing data of the sensor by requesting the corresponding resource of the sensor to get the current sensing data. In this process, the resource of actuators responds the status of the actuator to be presented in the page. The monitoring data is presented by time sequence to provide visualization of sensing data to users.



Resource Information

URI /vr/jnu-mcl-423-001/temperature

Data Type Simple

MONITOR Enabled

URI /vr/jnu-mcl-423-001/pressure

Data Type Simple

MONITOR Enabled

localhost:8080 says

```
{
  "fan": "1"
}
```

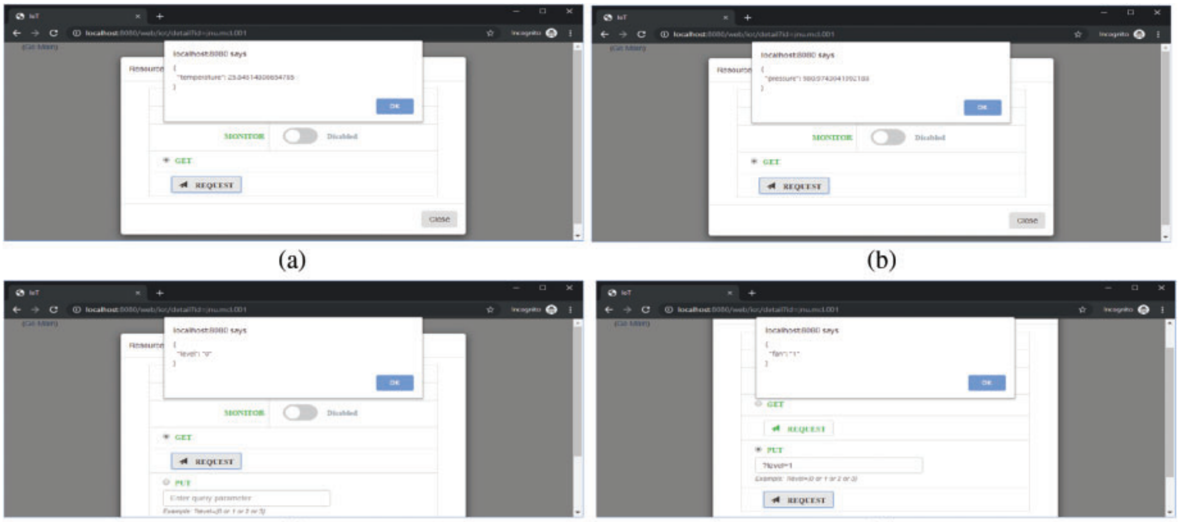
?level=1

Example: ?level=(0 or 1 or 2 or 3)

(a) (b) (c)

Figure 10: Implementation result of monitoring resources (a) Temperature resource (b) Pressure resource (c) Fan resource

Fig. 11 shows the pages of accessing resources of IoT devices through virtual resources. The sensors of temperature and pressure are used for implementing the resources of IoT device to collect sending data from the environment, and the fan actuator is used for implementing the actuator resource. The sensor resources provide sensing data through the edge gateway to clients. The actuator resource provides the status of the actuator and receives the command from the client through the edge gateway. Figs. 11a–11c shows accessing sensing data using GET method to request resources. Fig. 11d shows accessing an actuator using PUT method to request the resources for updating the status of the actuator., and mark the locations of resources on the plan. Users can click the mark to access the resource.



localhost:8080 says

```
{
  "temperature": 25.84514300654755
}
```

MONITOR Disabled

localhost:8080 says

```
{
  "pressure": 980.9742041092108
}
```

MONITOR Disabled

localhost:8080 says

```
{
  "level": "0"
}
```

MONITOR Disabled

Enter query parameter

Example: ?level=(0 or 1 or 2 or 3)

localhost:8080 says

```
{
  "fan": "1"
}
```

?level=1

Example: ?level=(0 or 1 or 2 or 3)

(a) (b)

(c) (d)

Figure 11: Implementation result of resource access (a) Get temperature data (b) Get pressure data (c) Get fan status (d) Update fan status

5 Performance Evaluation

For providing the performance of the proposed edge computing architecture, we evaluate the interaction delays of the IoT scenarios including registration, accessing, observation and notification using the IoTivity and HTTP network communication solutions. Each scenario is performed 100 times to collect the Round-Trip Time (RTT) and summarized to get the maximum, minimum and average.

The IoTivity device sends a less message packet than the HTTP device. Nevertheless, the delays from the IoT device (ID) to IoT service provider (ISP) through the edge gateway (EG) take almost the same time. Currently, the IoTivity library is not mature to provide outstanding performance. Especially, the message sending process in the OCF requires the resource discovery step that also takes time although very small. In the future version of the IoTivity library can be updated to provide less delay in message delivery. Then, the message packet size and delivery delay both present a better performance in the proposed gateway-centric edge computing architecture.

For evaluating the performance of registering the IoT device to generate virtual resources in the edge gateway, 100 times interaction delays of the registration process are collected including local and total registration as shown in Figs. 12a and 12e respectively using IoTivity and HTTP. The average total registration delay of IoTivity and HTTP takes 1232 and 1237 ms that are collected by the interaction between IoT device and IoT service provider. Approximately, there are differences between IoTivity and HTTP. However, the message packet size is 883 bytes from the IoTivity device and 1269 bytes from the HTTP device for registering the same registration profile.

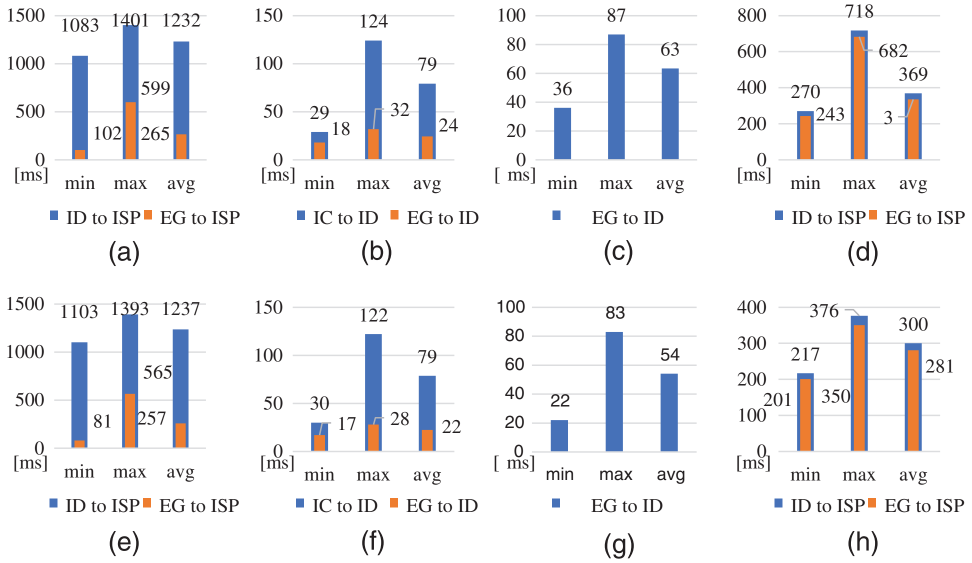


Figure 12: Performances (a) IoT device registration performance using IoTivity (b) IoT device accessing performance using IoTivity (c) IoT device observation performance using IoTivity (d) Data notification performance using IoTivity (e) IoT device registration performance using HTTP (f) IoT device accessing performance using HTTP (g) IoT device observation performance using HTTP (h) Data notification performance using HTTP

Figs. 12b and 12f show the performance of transparent access to the IoT device based on the virtual resource in the edge gateway. In this test, the most of total delay times take less than 100 ms from the IoT client (IC). From the edge gateway, the forwarded requests to the IoT device using the virtual resource take 18 ms for the minimum total delay, and 32 ms for the maximum total delay based on IoTivity. Using HTTP communication, the network delays are similar to the IoTivity. For the average delay using IoTivity, from the edge gateway to the IoT device takes 24 ms. The size of network packet is 96 bytes for responding messages from the IoTivity device and 233 bytes for responding message from the HTTP device. The performance result illustrates that the IoTivity takes a similar time with a less packet size for the less network cost. Also, in the gateway-centric local network, the communication delays take little time, which enables further operation in the edge gateway using real-time data.

Figs. 12c and 12h show IoT device observation performance using IoTivity and HTTP communications. The request is sent by the edge gateway to the IoT device. In this process, the request delays take very small time. Figs. 12d and 12f shows the data notification performance using IoTivity and HTTP communications. The request is sent by the IoT device and the packet is delivered to the IoT service provider through the edge gateway. Between edge gateway and IoT device, IoTivity and HTTP communications are performed. Using the IoTivity, the IoT device sends 82 bytes for the sensing data. For the same data, the IoT device send 526 bytes using HTTP. Obviously, the HTTP-based packet size is bigger than the IoTivity-based packet size. However, the communication delays are less than the IoTivity.

6 Conclusions and Future Directions

This paper proposed gateway-centric edge computing to provide IoT device registration, monitoring and transparent access based on virtual resources. The edge gateway exposes virtual resources to the Internet to present the resources of physical devices. The virtual resource bridges the Internet and IoT devices, therefore, web clients access a virtual resource to consume a service from an IoT device. Instead of IoT devices, the proposed edge gateway is deployed in an entry of a network edge to provide sensing and actuating services to the web clients on the Internet. Based on virtual resources, the edge gateway provides a consistent interface to enable clients to access devices in edge computing without considering underlying protocols. Through the device registration, the resource directory generates the virtual resource and links to the stored device information for identifying the physical device which is represented on the Internet. Furthermore, the edge gateway provides device discovery and monitoring to the web clients through mapping the virtual resources to actual resources on IoT devices. Using the proposed edge computing, the services of resource discovery, monitoring, data visualization and map-based visualization are developed. According to the performance evaluation, the edge gateway is satisfied to provide the proposed edge services on the current hardware. Moreover, accessing IoT device data through the edge gateway takes a short time for operating the IoT devices in real-time.

As future directions, we will apply the proposed edge gateway to smart buildings to operates IoT device in real-time based on Artificial Intelligence (AI). The AI approach can be deployed in the edge gateway, and triggered by events of IoT device in edge computing. The mechanism of mapping the virtual resource with actual IoT resources enables various intelligent scenarios to be deployed by connecting the virtual resources in scenario sequences.

Acknowledgement: This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2020-0-00048, Development of 5G-IoT Trustworthy AI-Data Commons Framework). Any correspondence related to this paper should be addressed to Dohyeun Kim.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Evans, “The internet of things: How the next evolution of the internet is changing everything,” *CISCO White Paper*, vol. 1, no. 2011, pp. 1–11, 2011.

[2] R. Want, B. N. Schilit and S. Jenson, “Enabling the internet of things,” *Computer*, vol. 48, no. 1, pp. 28–35, 2015.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[4] P. Sethi and S. R. Sarangi, “Internet of things: Architectures, protocols, and applications,” *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 9324035, 2017.

[5] G. A. Akpakwu, B. J. Silva, G. P. Hancke and A. M. Abu-Mahfouz, “A survey on 5G networks for the internet of things: Communication technologies and challenges,” *IEEE Access*, vol. 6, pp. 3619–3647, 2017.

[6] W. Jin 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, pp. 539–545, 2018.

[7] C. Doukas and I. Maglogiannis, “Bringing IoT and cloud computing towards pervasive healthcare,” in *2012 Sixth Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing*, Palermo, Italy, IEEE, pp. 922–926, 2012.

[8] W. Jin and D. H. Kim, “Design and implementation of e-health system based on semantic sensor network using IETF yang,” *Sensors*, vol. 18, no. 2, pp. 629, 2018.

[9] A. Abdelaal, T. Hegazy and M. Hefeeda. “Event-based control as a cloud service,” in *2017 American Control Conf. (ACC)*, Seattle, WA, USA, pp. 1017–1023, 2017.

[10] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu *et al.,* “An IoT-oriented data storage framework in cloud computing platform,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1443–1451, 2014.

[11] H. Chang, A. Hari, S. Mukherjee and T. Lakshman, “Bringing the cloud to the edge,” in *2014 IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, IEEE, pp. 346–351, 2014.

[12] Y. Ai, M. Peng and K. Zhang, “Edge computing technologies for internet of things: A primer,” *Digital Communications and Networks*, vol. 4, no. 2, pp. 77–86, 2018.

[13] W. Shi and S. Dustdar, “The promise of edge computing,” *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[14] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[15] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar and J. Ott, “Consolidate IoT edge computing with lightweight virtualization,” *IEEE Network*, vol. 32, no. 1, pp. 102–111, 2018.

[16] V. Cozzolino, A. Y. Ding and J. Ott, “Fades: Fine-grained edge offloading with unikernels,” in *Proc. of the Workshop on Hot Topics in Container Networking and Networked Systems*, Los Angeles, CA, USA, pp. 36–41, 2017.

[17] Y. Zhou, F. R. Yu, J. Chen and Y. Kuo, “Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11339–11351, 2017.

[18] C. Wang, F. R. Yu, C. Liang, Q. Chen and L. Tang, “Joint computation offloading and interference management in wireless cellular networks with mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.

[19] O. Salman, I. Elhajj, A. Kayssi and A. Chehab, “Edge computing enabling the internet of things,” in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Milan, Italy, IEEE, pp. 603–608, 2015.

[20] R. Morabito, R. Petrolo, V. Loscrí and N. Mitton, "Enabling a lightweight edge gateway-as-a-service for the internet of things," in *2016 7th Int. Conf. on the Network of the Future (NOF)*, Buzios, Brazil, IEEE, pp. 1–5, 2016.

[21] R. Morabito, R. Petrolo, V. Loscri and N. Mitton, "LEGIoT: A lightweight edge gateway for the internet of things," *Future Generation Computer Systems*, vol. 81, pp. 1–15, 2018.

[22] X. Xu, B. Shen, X. Yin, M. R. Khosravi, H. Wu *et al.,* "Edge server quantification and placement for offloading social media services in industrial cognitive IOV," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2910–2918, 2020.

[23] S. Wan, Z. Gu and Q. Ni, "Cognitive computing and wireless communications on the edge for healthcare service robots," *Computer Communications*, vol. 149, pp. 99–106, 2020.

[24] C. -H. Chen, M. -Y. Lin and C. -C. Liu, "Edge computing gateway of the industrial internet of things using multiple collaborative microcontrollers," *IEEE Network*, vol. 32, no. 1, pp. 24–32, 2018.

[25] S. K. Datta, C. Bonnet and N. Nikaein, "An IoT gateway centric architecture to provide novel M2M services," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Seoul, Korea (South), IEEE, pp. 514–519, 2014.

[26] Q. Zhu, R. Wang, Q. Chen, Y. Liu and W. Qin, "IoT gateway: Bridgingwireless sensor networks into internet of things," in *2010 IEEE/IFIP Int. Conf. on Embedded and Ubiquitous Computing*, Hong Kong, China, IEEE, pp. 347–352, 2010.

[27] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego and J. AlonsoZarate, "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.

[28] W. Jin and D. Kim, "Development of virtual resource based IoT proxy for bridging heterogeneous web services in IoT networks," *Sensors*, vol. 18, no. 6, pp. 1721, 2018.

[29] W. Jin and D. Kim, "Improved resource directory based on DNS name self-registration for device transparent access in heterogeneous IoT networks," *IEEE Access*, vol. 7, pp. 112 859–112 869, 2019.

[30] C. -L. Zhong, Z. Zhu and R. -G. Huang, "Study on the IoT architecture and gateway technology," in *2015 14th Int. Symp. on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, Guiyang, China, IEEE, pp. 196–199, 2015.

[31] H. Chen, X. Jia and H. Li, "A brief introduction to IoT gateway," in *IET Int. Conf. on Communication Technology and Application (ICCTA 2011)*, Beijing, China, IET, pp. 610–613, 2011.

[32] S. Park, "OCF: A new open IoT consortium," in *2017 31st Int. Conf. on Advanced Information Networking and Applications Workshops (WAINA)*, Taipei, China, IEEE, pp. 356–359, 2017.

[33] S. Sharma and S. Kumar, "A review on IoT: Protocols, architecture, technologies, application and research challenges," in *2020 10th Int. Conf. on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, IEEE, pp. 559–564, 2020.

[34] D. Glaroudis, A. Iossifides and P. Chatzimisios, "Survey, comparison and research challenges of IoT application protocols for smart farming," *Computer Networks*, vol. 168, pp. 107037, 2020.

[35] T. Yokotani and Y. Sasaki, "Comparison with http and mqtt on required network resources for IoT," in *2016 Int. Conf. on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, Bandung Indonesia, IEEE, pp. 1–6, 2016.

[36] H. Park, H. Kim, H. Joo and J. Song, "Recent advancements in the internet-of-things related standards: A oneM2M perspective," *ICT Express*, vol. 2, no. 3, pp. 126–129, 2016.

[37] C. Akasiadis, V. Pitsilis and C. D. Spyropoulos, "A multi-protocol IoT platform based on open-source frameworks," *Sensors*, vol. 19, no. 19, pp. 4217, 2019.

[38] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez and A. F. Skarmeta, "Smart farming IoT platform based on edge and cloud computing," *Biosystems Engineering*, vol. 177, pp. 4–17, 2019.

[39] D. Tracey and C. Sreenan, "OMA LWM2M in a holistic architecture for the internet of things," in *2017 IEEE 14th Int. Conf. on Networking, Sensing and Control (ICNSC)*, Calabria, Italy, pp. 198–203, 2017.

[40] K. Elsayed, M. Ibrahim and S. Hamza, "Service discovery in heterogeneous IoT environments based on OCF/IoTivity," in *2019 15th Int. Wireless Communications & Mobile Computing Conf. (IWCMC)*, Guilin, China, pp. 1160–1165, 2019.

[41] C. A. L. Putera and F. J. Lin, "Incorporating OMA lightweight M2M protocol in IoT/M2M standard architecture," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Milan, Italy, IEEE, pp. 559–564, 2015.

[42] S. Rao, D. Chendanda, C. Deshpande and V. Lakkundi, "Implementing 1WM2M in constrained IoT devices," in *2015 IEEE Conf. on Wireless Sensors (ICWiSe)*, Melaka, Malaysia, IEEE, pp. 52–57, 2015.

[43] J. C. Lee, J. H. Jeon and S. H. Kim, "Design and implementation of healthcare resource model on IoTivity platform," in *2016 Int. Conf. on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea, IEEE, pp. 887–891, 2016.

[44] I. Sittón-Candanedo, R. S. Alonso, J. M. Corchado, S. RodríguezGonzález and R. Casado-Vara, "A review of edge computing reference architectures and a new global edge proposal," *Future Generation Computer Systems*, vol. 99, pp. 278–294, 2019.

[45] M. Satyanarayanan, P. Bahl, R. Caceres and N. Davies, "The case for VMbased cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[46] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. of the First Edition of the MCC Workshop on Mobile Cloud Computing*, Helsinki Finland, pp. 13–16, 2012.

[47] A. Ceselli, M. Premoli and S. Secci, "Cloudlet network design optimization," in *2015 IFIP Networking Conf. (IFIP Networking)*, Toulouse, France, IEEE, pp. 1–9, 2015.

[48] Z. Sanaei, S. Abolfazli, A. Gani and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369–392, 2013.

[49] M. Satyanarayanan, R. Schuster, M. Ebling, G. Fettweis, H. Flinck *et al.,* "An open ecosystem for mobile-cloud convergence," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 63–70, 2015.

[50] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[51] M. Tseng, T. Canaran and L. Canaran, "Introduction to edge computing in IIoT," *Industrial Internet Consortium (IIC) White Paper*, Tech. Rep, 2018.

[52] W. Jin, R. Xu, T. You, Y. Hong and D. Kim, "Secure edge computing management based on independent microservices providers for gateway-centric IoT networks," *IEEE Access*, vol. 8, pp. 187975–187990, 2020.