electronics

MDPI

Article

A Novel Approach towards the Design and Implementation of Virtual Network Based on Controller in Future IoT Applications

Faisal Mehmood, Israr Ullah, Shabir Ahmad and Do-Hyeun Kim *

Department of Computer Engineering, Jeju National University, Jeju 63243, Korea; faisalavan@jejunu.ac.kr (F.M.); israr.ullah@jejunu.ac.kr (I.U.); shabir@jejunu.ac.kr (S.A.)

* Correspondence: kimdh@jejunu.ac.kr; Tel.: +82-10-5267-3263

Received: 27 February 2020; Accepted: 28 March 2020; Published: 2 April 2020

Abstract: The Internet of Things refers (IoT) to the billions of physical devices around the globe that are connected to the Internet, collecting and sharing data. The overall Internet of Things market is projected to be worth more than 50.6 billion U.S. dollars in 2020. IoT devices possess low processing capabilities, limited memory, limited storage, and minimal network protocol support. With the help of cloud computing technology, we can overcome the limited resources of IoT devices. A lot of research has been conducted on IoT device virtualization to facilitate remote access and control. The concept of virtualization in IoT is to provide a virtual representation of physical devices in the form of virtual objects. IoT devices are more likely to be accessed and communicate through virtual objects in the near future. In this paper, we present the design and implementation of building a virtual IoT network for a smart home. The virtual network is based on virtual objects and IoT controller. We derived the concept from Software Defined Network (SDN) and separated the control plane and data plane in the virtual IoT network. This enhanced the rapid development of diverse applications on top of the virtualization layer by establishing a dynamic end-to-end connection between IoT devices. This article briefly explains the design and development of the virtual network. Results achieved during experiments and performance analysis show that IoT controller enhances the capabilities of a virtual network by dynamically controlling the traffic congestion, handling mapping requests, and routing mechanisms.

Keywords: Internet of Things (IoT); cloud of things (CoT); Amazon Web Services (AWS); smart home; virtual network; IoT controller

1. Introduction

Presently, the Internet of Things (IoT) is one of the most important and promising technological topics. By 2020, it is estimated that there will be up to 50.6 billion connected devices [1]. The IoT consists of everyday objects such as physical devices, buildings, vehicles with embedded software, electronics, sensors, and network connectivity, that are capable to collect and exchanging data. Figure 1 shows different technologies involved in the formation of a virtual IoT network. IoT is a network of interrelated computing devices. An IoT network consists of physical devices that collect data from the surrounding environment and communicate with each other using the Internet. The IoT is capable of transmiting the data over a network without depending on human-to-machine or human-to-human interaction. The IoT plays a vital role in different fields such as smart city, smart homes, health-care, banking, and education [2,3]. The IoT network comprises of two different types of devices i.e., sensors and actuators. The sensor is a device that can detect changes in an environment. There are different sensors available in the market such as temperature, humidity, pressure, motion, heat sensor. Actuators are used to perform some specific action based on the sensor's data. The IoT not only senses and process data but activate various devices into operations based on the dynamics of the data. Automation in IoT is based on the combination of sensors and actuators. Hence we can say that sensors and actuators are the backbones of IoT.

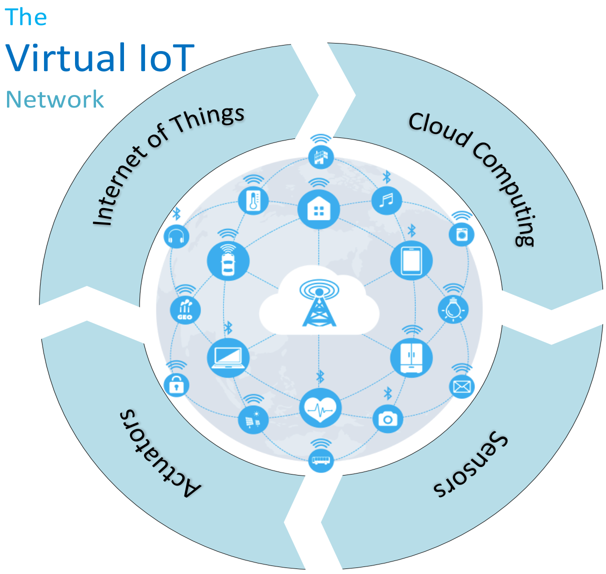

Figure 1. Convergence and evolution of network technology.

The IoT generates a huge amount of big data thus putting a tremendous load on the Internet Infrastructure [4]. Cloud computing plays a vital role in handling data storage. Cloud computing can reduce the load on the Internet infrastructure. In simple terms, cloud computing means accessing data and programs from a centralized pool of computing resources. Cloud Computing and IoT both have a complementary relationship and both serve to increase efficiency in everyday tasks [5,6]. IoT generates a massive amount of data and cloud computing provides data storage and computation. Cloud computing also provides better fraternization which is necessary for software developers. Cloud computing allows the developer to store data that can be easily accessed remotely.

Figure 2 illustrates the comparison between conventional networking and Software-Defined Networking (SDN). In conventional networking, we can see different physical devices are connected with each other and exchanging data whereas in SDN, the control and data plane are separated which makes it more flexible and agile. Machine-to-Machine (M2M) networks are very identical to Wireless Area Networks (WAN) or Local Area Networks (LAN) but are especially designed to be used for sensors, machines, and controls to communicate. M2M refers to various technologies that simply allow an end device to exchange data without human interaction. M2M is a connection between machines, whereas IoT takes M2M connectivity, integrates web applications, and connect it to a cloud. M2M is the foundation for IoT. M2M employs isolated systems of sensors whereas IoT converges disparate systems into an expansive system to enable new applications. IoT is more scalable than M2M because of its cloud-based architecture. IoT is more than physical device connectivity, as it is a network of inter-connected devices [7].

Software Defined Networking (SDN) is an architecture that aims to make the network flexible and agile. The goal is to enable cloud computing and network engineers to retort rapidly to changing requirements via centralized control. SDN technology mainly focuses on separating the network control plane from the data plane. The main feature of SDN is that it disassociates the control plane from the data plane. The control plane decides how packets should flow through the network and the data plane moves packets from one place to another [8].

M2M and IoT both relate to communication between inter-connected physical devices. M2M systems are generally segregated, stand-alone networked equipment but IoT systems take M2M to the next level, connecting various systems into a single large network. M2M systems use point-to-point communications between actuators, sensors over a wired or cellular network, while IoT systems depend on IP-based networks to transmit collected data from physical IoT devices to cloud or middle-ware platforms. M2M is used in almost every field such as manufacturing environment, smart utility management, home appliances, health-care device management, whereas IoT is used in the smart home, industrial Internet, wearables, connected cars, smart city, telemedicine, smart retail, smart supply chain, and smart farming.

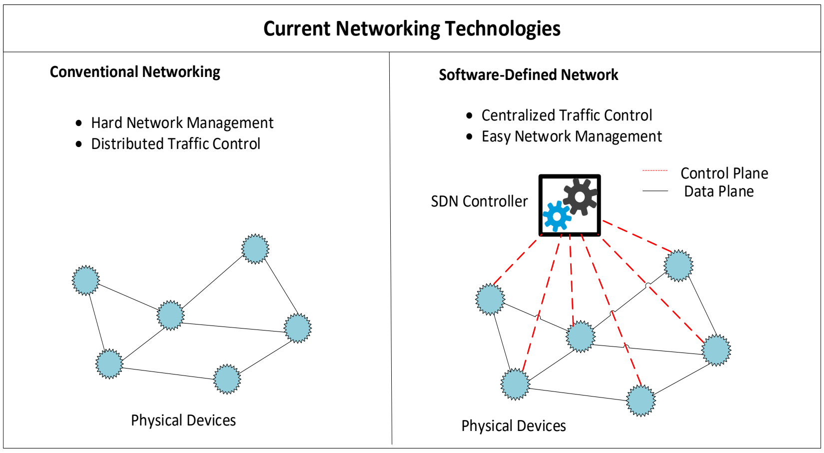

Figure 2. Current network technologies.

This paper aims to design a virtual IoT network for a smart home that is based on the concept derived from the SDN controller. We separated the control and data plane. We have designed and developed the virtual IoT network for a smart home that is based on IoT controller. The IoT controller controls the mapping and routing details for the physical IoT devices and their corresponding virtual objects. The IoT controller is updated based on user requirements. The role of IoT controller is to keep the record of registered physical IoT devices. It also keeps records of the mapping and routing information between physical IoT devices. We can easily update the mapping and routing information using the IoT controller without affecting the physical IoT devices. The user can easily map sensors and actuators and update the entries to the IoT controller using the web interface. The IoT controller then updates the routing information to each virtual object for communication. This study is about the formation of virtual network among connected IoT devices in the cloud. Previously most of the work is focused on the virtualization of the server where data collection and processing operations were handled in application logic. In this study, we separate the data and control logic which allow making changes in the network virtually without involving the physical network.

Rest of this paper is organized as follows. In Section 2, literature review is described briefly. Existing research work in the relevant field is highlighted. In Section 3, the proposed architecture of the virtual IoT network is explained. Section 4 presents the implementation of the virtual IoT network. Section 5 briefly explains the experimental environment of the proposed virtual IoT network. Section 6 presents the results achieved during experiment. Discussions are given in Section 7. Finally, the paper is concluded in Section 8 with an outlook on our future work.

2. Literature Review

The research areas of global scientific communities are often interconnected, and the community is constantly working to propose new ideas and concepts in pursuit of intellectual excellence. The results of these diverse studies are constantly changing the scientific research we are currently experiencing such as IoT. In IoT, an object or thing is connected to other objects in the world via the Internet so that these objects can communicate with each other [6]. Objects used in IoT such as RFID, sensors, vehicles, home appliances lack intelligence due to the limitations in the hardware and software [9]. In this article [10], the authors tried to fill this concept with new concepts that are called "Agent of Things" (AoT), which expands and improves the intelligence aspect of loT.

Recently, many industrial companies have invested in IoT. Different projects aim to design various IoT applications that highlight real-world problems and develop a system that provides a solution to those problems. IoT is designed to attach a small communicating device with everything that we want to monitor or control using the Internet [11]. IoT based systems have tremendous features and applications. The core of IoT networks consists of two types of devices (a) sensors that collect the context data (b) actuators that receive commands to control the environment. IoT focuses on connectivity of things (daily life objects with attached sensors or actuators) to the Internet so that users can remotely monitor certain activities or devices.

IoT devices are used for sensing, actuating, data storage, data interpretation and exchanging data, in a situated context-aware manner. There are many challenges in IoT systems such as device heterogeneity, interoperable issues, and scalability [12,13]. IoT systems consist of many subsystems and are highly dynamic. The configuration of the IoT systems needs to be changed and the IoT resources connect and disconnect in runtime [14]. So it is necessary that IoT systems are capable of being dynamically discovered, addressed, and accessed any time. Agent-based systems provide a solution to these challenges. They provides autonomy, proactive, and reactive features in the IoT devices [15–17].

IoT devices have low processing power and storage memory so it is difficult to store and process an enormous amount of data. The proposed virtual network is deployed on the cloud and it provides enough storage and computing features to the virtual agents. The virtual agent is synchronized with the physical device, so if the physical device is facing technical issue it will keep sending signals until the issue is resolved and the device is active. This will keep other devices secure and avoid being corrupted. The virtual agent allows devices to communicate with each other. During physical device registration, we have to specify the details of the physical device so that communication is smooth and avoid compatibility issues.

Raspberry pi is a credit-card single-board computer developed by the Raspberry Pi foundation in the United Kingdom (UK). A recent development in Raspberry Pi has unlocked great potential for computing in various fields [18]. The main distinctive feature of Raspberry Pi is the GPIO (General-purpose Input/Output) module which provides interfacing with other electronics such as sensors and actuators. Arduino is easy-to-use software and hardware and it is totally open-source [19].

There are various protocols used with IoT devices such as Message Queuing Telemetry Transport (MQTT) protocol, WebSockets, and Constrained Application Protocol (CoAP). MQTT is a lightweight protocol used for exchanging messages based on the publish-subscribe standard between servers and clients [20]. WebSocket protocol is a bidirectional communication protocol. It consists of a handshake and data transfer. First, the client and server establish a communication channel using HTTP and a port 80 by default. Then the client sends a request and upon validation data is exchanged between client and server.

The development of IoT systems, such as smart homes, smart machines, smart factories, and smart cities, integration, and management in practical applications, is a complex task requiring appropriate models, methods, and techniques [21,22]. In this direction, several solutions, tools, and methods of middleware were developed for solving significant problems, such as virtualization of physical devices, management of distributed objects, and identification of guidance. However, these solutions do not provide an integrated approach to support the entire IoT system development process from analysis to implementation and often tend to solve a specific problem simultaneously. However, this partial approach led to poor interoperability of the "intranet" system, insufficient scalability or manageability of applications, leaving the original IoT embedded vision [23–25]. Therefore, we will develop a horizontal environment concept for interacting networked, physical, and networked physical systems, providing a comprehensive, application-independent methodological approach to developing IoT systems.

As the IoT is trying to integrate physical, social, and cyberspace, nodes need to become more intelligent in their applications. Thanks to agent-oriented software development, IoT can be viewed and implemented as a software agent [26,27]. Heterogeneous IoT [28] is a new area of research with enormous potential that can change our understanding of the fundamental principles of informatics and future life. Heterogeneous IoT is being used in an increasing number of areas, such as smart homes, smart cities, intelligent transportation, environmental monitoring, security systems, and advanced manufacturing. Consequently, the heterogeneous IoT, which will be distributed through powerful application areas, will fill our lives and provide many convenient services for our future. IoT's network architecture is heterogeneous, including wireless sensor networks, wireless mesh networks, mobile networks, and car networks.

Smart homes are becoming an important research area in the IoT. Smart homes monitor the interaction between the user and house components to provide appropriate services [29,30]. Smart home management system provides designers with a graphical user interface (GUI) to design a house and add home appliances [31]. Smart homes provide a user-friendly interface that is capable of monitoring and controlling home appliances [32]. Different IoT based sensors like temperature sensors, humidity sensors, motion sensors connected by wireless networks can be installed in a smart home. There are different types of sensors like physical sensors, location-based sensors. These sensors collect environmental data. This data can be used to control different devices. The smart home provides different services like control room temperature, control light bulbs, etc. These services provide a comfortable living environment [33,34].

With the development of human-computer interaction technology and the improvement of people's living standards, the IoT research has become a hot spot for research over the years. At the same time, smart home systems are attracting more and more attention to the IoT [35,36]. In the future, the IoT smart objects will be the basis for building a network penetrating system of physical intelligence in various applications, such as healthcare, transport, logistics, intelligent networks, and cities [37,38].

IoT evolved from machine-to-machine (M2M) communication. M2M refers to the intercommunication between physical computing devices using any mode of communication. M2M communication does not rely on the Internet whereas, in IoT, communication between physical devices is via the Internet. The main similarity is that it provides remote access to machine data and exchange data without human intervention. Concisely, both technologies allow machines to connect, communicate, store, exchange data with each other, and execute tasks with minimum human interference.

The authors in these article [39–42] present a complete connectivity model for designing and manufacturing products in a cloud-based production environment with IoT capabilities. The model uses social networks to connect multiple parties, promotes open innovation, uses IoT to connect physical space with cyberspace, and provides various flexible services through cloud production to create custom workspaces, interactions, information that can be shared or shared.

The Wireless Sensor and Actuator Network (WSAN) appeared as a new research area in the distributed computing environment and plays a primary role in many applications. WSAN is a group of sensors that collect data from surroundings and actuators that perform some specific tasks based on the data. Wireless Sensor Nodes (WSNs) are intelligent as compared to traditional sensors. WSN's are designed for in-network processing. In this type of network, a large number of sensor nodes span a physical geographic area. So it is difficult to manage and control such networks. In the proposed system, we designed and developed a virtual network to manage and control WSAN. In the future, WSN could be a set of autonomous nodes that communicate with each other in a decentralized manner. Such networks can change their topology dynamically when there is a change in the network due to node mobility.

Object virtualization was first introduced in Radio Frequency Identification (RFID) tags. Despite its simplicity and limitations, RFID-based identification systems bridge the gap between the physical world and the virtual world by providing a wide range of novel applications [43,44]. Fog networking represents an architecture that is based on edge devices to carry out task computation, memory storage, and communication internally and routed over the internet's backbone. Fog computing extends the working of cloud computing to the edge of network. It is highly virtualized platform that provides different services such as storage, computation, networking between end devices and cloud computing data centers [45,46].

The primary goal of integrating IoT and Web of Things (WoT) is to convert those objects, machines, or things into intelligent devices. To achieve this, devices must be provided with enough data that they are capable of acting autonomously in specific situations [47]. The agent can be referred to as a sophisticated software abstraction defining an autonomous and proactive entity [48]. IoT is a revolutionary concept but has multifacet requirements and development issues. To properly address and support IoT systems, agent-based computing represents an effective model [49].

Cloud computing is considered a promising approach that can accept challenges in IoT and create new opportunities. The combination of IoT and cloud computing make the cloud an intermediate layer between the smart objects and applications that use the data and resources they provide [50].

The demand for smart applications is rising. Smart applications are designed and developed for banking, e-health, automobile, smart homes, and smart factories [51]. The speed and accuracy are some of the major concerns in a real-time environment. With the emergence of Wireless Sensor Networks (WSN), the world is on the verge of new innovation. WSN innovation promise to integrate and optimize autonomous vehicles, smart buildings, and power grids. The use of smart applications makes our lives easier and more efficient.

IoT is generating an unprecedented amount of data which in turn puts tremendous strain on the Internet infrastructure [52]. Cloud computing services are on the rise. There are different cloud service providers such as Amazon Web Services, Google Cloud IoT, Microsoft Azure IoT Suite, Digital Ocean, IBM Watson, and Salesforce IoT cloud [53]. IoT and cloud computing both serve to increase efficiency in our everyday tasks.

There are several problems that prevent the implementation of the IoT. The most important is heterogeneity due to various functions of the device, such as network type, identifier scheme, and interactive protocol [54]. The IoT requires close coordination of devices, but devices with different functions are difficult to interact with each other. For example, wireless sensors and embedded RFID devices cannot recognize peers due to different identifiers. A session cannot be created, even if the same type of wireless sensor has a different network connection. Unfortunately, modern technologies with IoT support, such as wireless sensor networks (WSN) and machine-to-machine (M2M), do not adequately address these problems [55]. Therefore, wireless sensor networks often create isolated that cannot work with each other and can no longer work independently through IoT technology.

3. Proposed Architecture

This section briefly explains the proposed architecture of the virtual IoT network. The idea of virtual IoT network is imitated from the concept of Network Function Virtualization (NFV) in the context of SDN and cloud computing. Figure 3 presents the key idea of this study. In this figure, there are three layers i.e., application service layer, virtual IoT network layer, and physical network layer. The physical network layer consists of physical IoT devices such as sensors, actuators, or any other embedded IoT devices such as raspberry pi, and arduino. IoT uses many new competitive network technologies. IoT creates a connection between small sensing devices and enables communication among them via the Internet. Billions of devices are expected to connect and communicate with each other in the near future. Sensors are used to detect event from the surrounding environment such as BME280 sensor is used to sense temperature, pressure, and humidity from the surrounding environment. Actuators are used to control devices. Actuators receive commands to operate accordingly. Actuators are used to control devices such as light, fan, and motors. These devices will generate an enormous amount of data. Several vendors offer different technologies for different vertical markets, such as home automation, healthcare, or the industrial-IoT, and often provide alternative implementations of the same standard protocols.

In this figure, we consider an example of a smart home where different devices form a network and can communicate with each other via the Internet. Network technologies allow IoT devices to interact with other devices, applications, and services that are running in the cloud. The Internet uses standardized protocols to ensure secure communication between disparate heterogeneous devices. Standard protocols define the rules and formats that devices use to configure and manage networks, as well as to transfer data across a network. There are different communication protocols for IoT such as Message Queuing Telemetry Transport (MQTT) protocol, Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), Data-Distribution Service for Real-Time Systems (DDS), etc. In our proposed system we used the MQTT protocol to communicate between sensors and actuators via the virtual network layer. It is an extremely lightweight protocol that enables a publish/subscribe messaging model. It is useful for connections with remote locations where a small code footprint is required and network bandwidth is expensive. The sensors and actuators in the smart home are able to publish and subscribe messages based on unique topics via MQTT protocol.

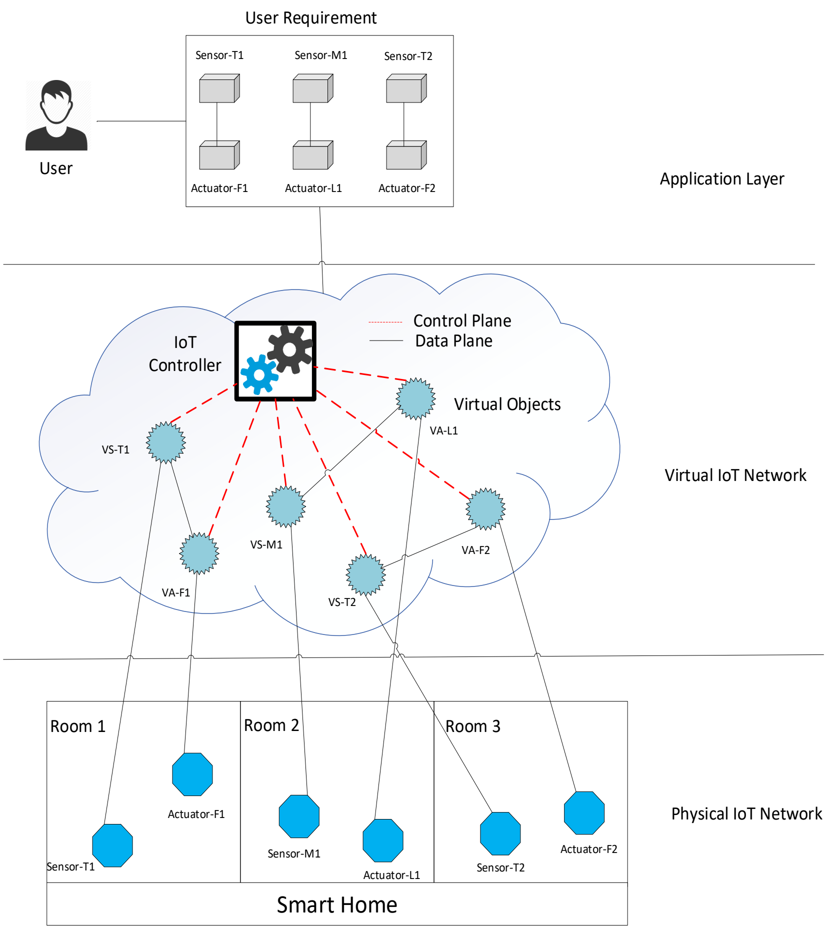

Figure 3. Proposed Architecture of Virtual IoT Network.

Many research studies suggest and prefer the use of cloud platform for IoT device connectivity because it is convenient for storing and processing a large amount of data collected from IoT devices using data analytics and big data processing schemes supported by the cloud. There are many cloud services providers in the market such as Google, Microsoft, Amazon, and Digital Ocean. These clouds provide support and APIs for the connectivity of IoT devices to their platform where the data can be easily collected and processed by respective client applications. We availed ourselves of the Amazon web services (AWS) for processing and computation of data on the cloud.

The main purpose of a virtual network is to provide the most appropriate and efficient network structure for applications hosted on a data center or service provider and use software to change the structure, without the need for physical changes in the hardware connection. The virtual IoT network layer represents the virtual representation of the physical network layer. In the virtual network layer, all the configuration, installation, and deployment of the virtual IoT network is done in the cloud.

Application service layer consists of a user-friendly web interface that allows the user to register physical IoT devices and initialize the virtual objects. The client application provides a platform for users to map virtual objects with physical devices. User can also create a routing mechanism for sensors and actuators to communicate with each other. All of this information is sent to the IoT controller. IoT controller is responsible to update the master table which includes information related to physical devices and virtual objects.

Figure 4 represents the layered architecture of the proposed virtual IoT network. In this figure, we can see that there are three layers i.e., application layer, a cloud layer, and a physical network layer. The application layer consists of the client application. It provides a user-friendly web-interface where users can register physical IoT devices and initialize virtual objects. Application layer is also used for mapping physical devices with the corresponding virtual objects. User can perform routing mechanism using the client applications. Routing is based on user requirements. We used the concept of SDN in which it separates the data and control plane. There is a concept of northbound APIs and southbound APIs. We introduced southbound and northbound APIs in the virtual network. The application layer communicates with the cloud layer via northbound APIs. All the information is transferred to virtualization server via northbound Rest APIs. After authentication and authorization, it is acknowledged and stored in the relational database.

The cloud layer comprises of repository and virtual IoT network. All the configuration, installation, and deployment of virtual network is done on the cloud. We used the Amazon EC2 service for this purpose. We introduced the IoT controller that works similar such as SDN and separates the data from the control layer. IoT controller functions include mapping of a physical device with virtual object, routing between sensors and actuators, virtual objects metadata.

The cloud layer communicates with the physical network layer via southbound Rest APIs. The IoT controller in the cloud layer fetches data from the repository and works accordingly. The IoT controller can update the mapping and routing paths according to the user requirement. In the physical layer, there are two types of devices, i.e., sensors and actuators. Physical devices communicate with virtual objects via MQTT protocol using southbound Rest APIs. It is a lightweight protocol that uses a publish/subscribe model. Sensors transfer data to a corresponding virtual object based on a unique topic via MQTT protocol. Similarly, actuators are subscribed to unique topics to receive commands and perform an action accordingly.

Figure 5 represents the sequence diagram for the proposed virtual IoT network. This figure shows the essential components of the proposed system for IoT network virtualization and its functions. There are two types of devices sensors and actuators. We categorized devices as Type A for sensors and Type B for actuators. The process is initiated with IoT device registration. In this process the profile information is sent to the virtualization server as a registration request via RESTful APIs. The virtualization server is responsible for the authentication and authorization of the physical device. After successful validation it registers the device in the repository and sends back acknowledge to the device. After acknowledgment, the virtual object of the corresponding device is initialized and is available to the users on the web-interface. The list of registered physical devices are available on the client application.

The user can perform the desired network configuration accordingly. The request is sent to the IoT controller for the list of registered virtual objects. The IoT controller fetches the list from the repository and returns back to the IoT controller. Using the list, the user can map the desired physical device with the virtual object and update the IoT controller. User can also setup the route virtually for the physical wireless sensor and actuator network. The main purpose of setting up virtual network is to provide appropriate and efficient network structure using the software without changing the hardware configuration. After setting up the desired configuration the request is sent to the virtualization server where mapping and routing configurations are deployed and IoT controller is updated. Now whenever the sensor transfer the data to virtualization server, the data is passed to IoT controller for checking routing table. The activation command is sent to the physical device according to the routing information. The corresponding device is subscribed to a specific topic and interprets the command to perform required operation. Acknowledgment is sent back to the physical device after successful operation.

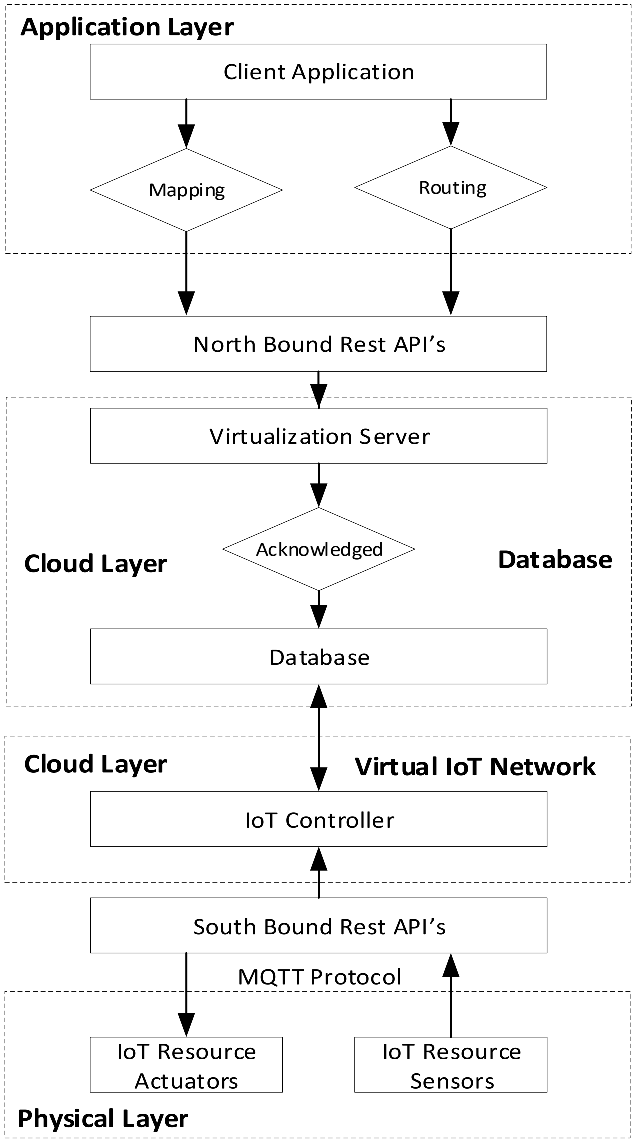

Figure 4. Layered Architecture of the Virtual IoT network.

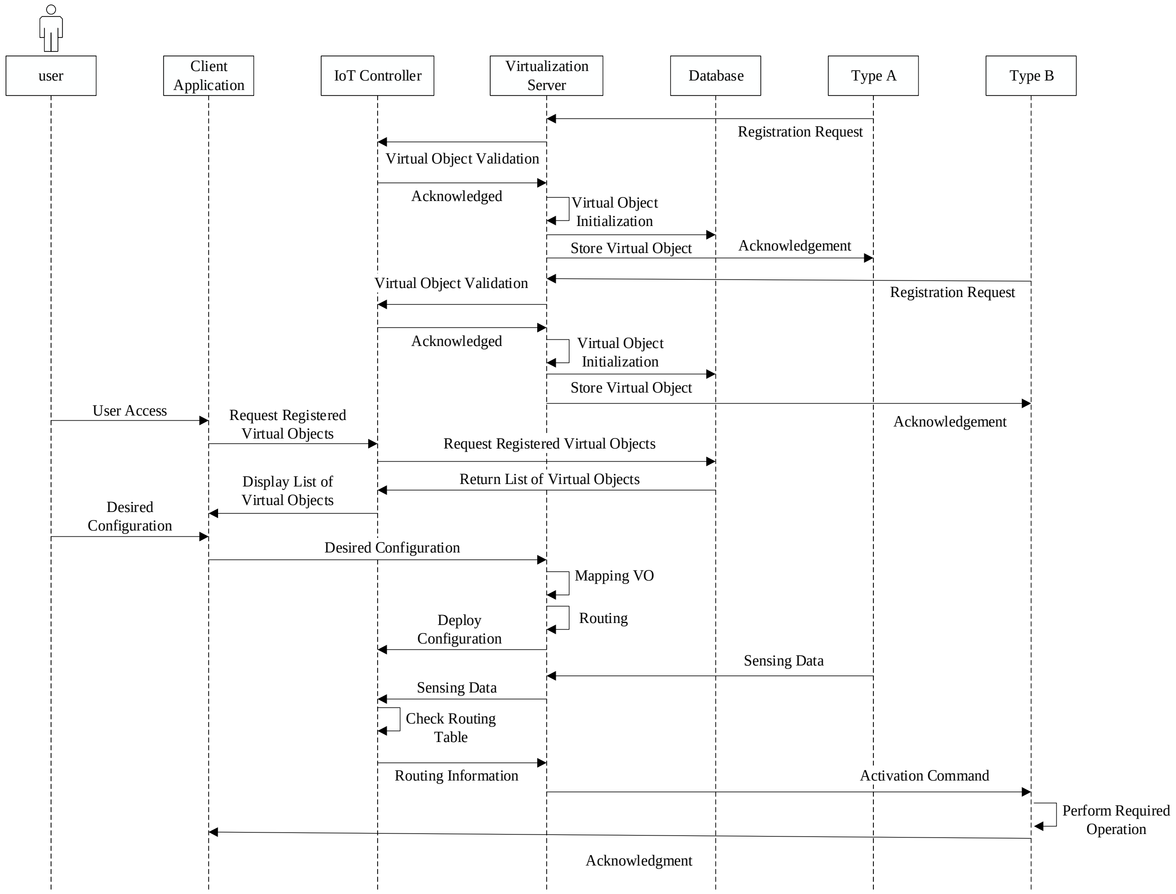

Figure 5. Sequence Diagram for Virtual IoT Network.

Figure 6 represents the functions involved in virtual objects. Virtual objects are the virtual representation of the physical devices. Smart objects are objects that improve interaction with other smart objects. These products, also known as smart connected products (SCoT), are products, assets, and other things with embedded processors, sensors, software, and connectivity that can be used to exchange data between products and its environments, manufacturers, and operators/users. Connectivity also allows some product features to exist outside of a physical device. You can then analyze the data collected by the product, provide information for decision-making, increase operational efficiency, and continually improve product performance. It can not just interact with the physical object but also with the virtual (computing environment) objects. A smart object is created by embedding electronic tags such as Radio-Frequency Identification (RFID) tags. Virtual objects are created as software objects that are essential when creating and operating a virtual environment. Virtual objects have several origins and use cases.

We can categorized virtual object functions into two i.e., virtual object registration and virtual object communication. Virtual object registration includes registration of the corresponding physical device. Before registration, the physical device information is authenticated by the virtualization server. After the initialization of the virtual object, the virtual object is synched with the physical device. The communication phase includes publish and subscribe of messages based on unique topics via MQTT protocol.

Figure 7 represents the administrative functions related to the IoT controller. At the application layer, administrative functions of IoT controller are to set policies, manage resources, and manage virtual objects. The network traffic policy determines the packet path through the network, applies the required quality of service, and protects the network from security threats. Users can also set the number of packets/second to balance the load of virtual network. Alternative route can also be set to avoid network traffic congestion. The role of IoT controller in the virtual network layer is to update mapping and routing tables. Mapping table includes information related to the connection between physical device and corresponding virtual object in the repository. Similarly routing table includes the information related to the communication between sensors and actuators via virtual network according to the user requirement and network configuration.

The IoT controller is also responsible for monitoring virtual objects. Virtual objects are synchronized to the corresponding physical devices. If the device is offline, then the virtual object is also inactive. Whenever the virtual object is inactive, the IoT controller send a signal to the corresponding physical device till the device is active. Service provisioning means to provide services to the client e.g. list of the registered physical devices, provide details of the network configuration. IoT controller also handles resource virtualization. In the control & management layer, the IoT controller role is to register resources, allocate resources, mapping devices with virtual objects and take decisions. If the virtual route is down for some reason, then it takes decision to provide some alternative route to send the command to corresponding physical device. IoT controller receives the profile of the desired network configuration from the physical layer that includes list of IoT resources, network setup, network connection, and network configuration. IoT resources information include name of device, type of device, IP of the device, protocol, and topic.

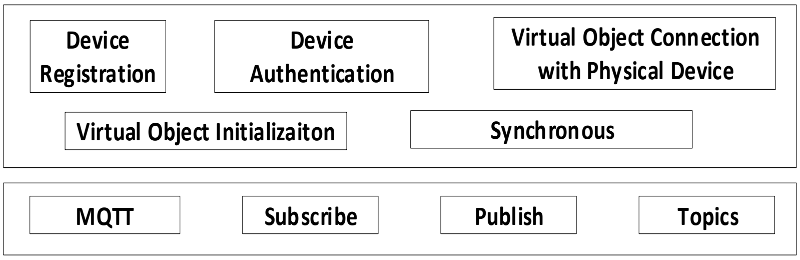

Figure 6. Functions of virtual objects in virtual IoT network.

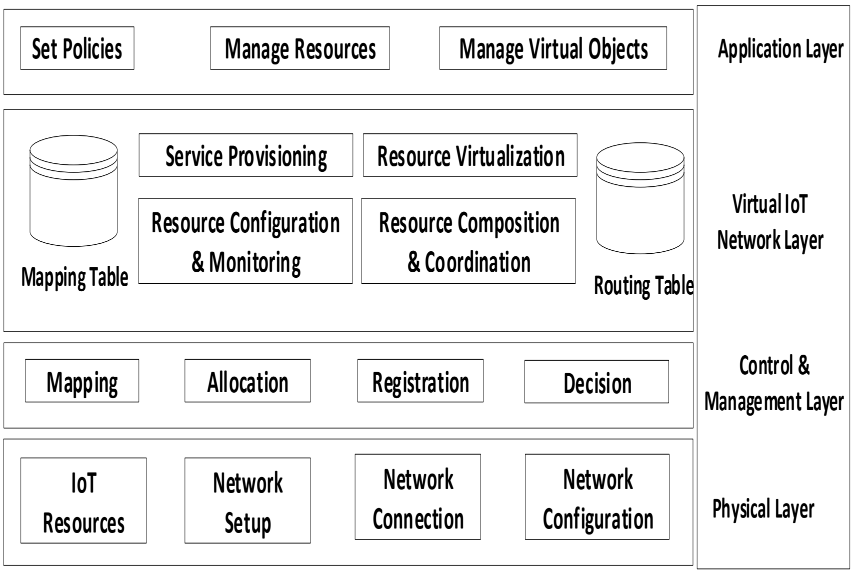

Figure 7. Administrative functions of IoT controller.

4. Implementation

Figure 8 represents the development configuration of the virtual IoT network. In this figure, there are three layers i.e., application layer, cloud layer, and physical network layer. In the application layer, there is client application that provides user-friendly web-interface where users can register physical devices and initialize virtual objects. Based on the desired network configuration users can map physical devices with the corresponding virtual objects and create a virtual route for communication among devices. In this figure, we can see that in step 1, the user sets up the virtual route for physical devices to communicate with each other. In step 2, the network configuration is sent to the IoT controller which updates the mapping and routing table in the repository. In step 3, the IoT controller updates the local table of the virtual objects. The local tables include information of virtual objects relation with the physical device. It includes physical device information and route of communication. If devices are inactive or offline, the IoT controller sends signals to physical devices until the devices are active. In step 4, all the physical devices are activated and online. The physical devices are synched with the virtual objects in the virtual network layer. In step 5, the communication flow starts and virtual objects start receiving sensor values via MQTT protocol. Based on the route setup by the user, the information flows to the next virtual objects. Virtual objects keep information in local tables that is necessary to pass the information to the next virtual object or physical device.

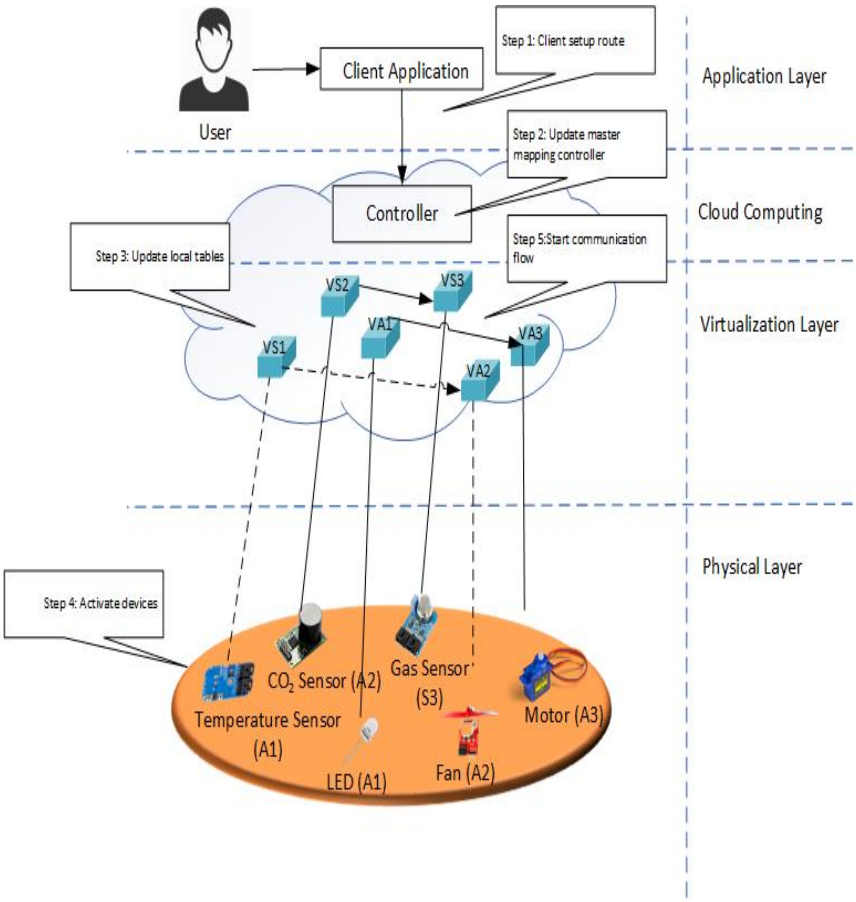

Figure 8. Development configuration and flow of the virtual IoT network.

In physical layer, there are physical hardware devices such as raspberry pi. With each raspberry pi there are sensors and actuators attached. In this figure, there are three sensors and three actuators. We used temperature, CO_2, and gas sensors. LED, fan, and motor are used as actuators. Every physical device is registered on the virtual IoT network. Each physical device has a corresponding virtual object in the cloud which is mapped.

Sensors are denoted by S_1, S_2, and S_3 whereas actuators are denoted by A_1, A_2, and A_3. The dotted lines in the figure show the communication flow between physical devices via virtual network. The main purpose of introducing IoT controller in the virtual network is to change the communication flow as desired without changing the hardware network configuration. In this way the control and data plane are separated from each other making it easy and convenient for the end-users.

Table 1 summarizes the hardware and software services being used in this experiment. Raspberry pi 3, Model B, is used for the experiment. Raspberry pi is a small IoT device that acts as a mini computer. It consumes less power. We install and configure raspbian operating system on the raspberry pi. 16 GB SD card is used as a storage for each raspberry pi. Raspberry pi 3 has 1 GB RAM built-in. Different software tools are used in the design and development of virtual IoT network. We used PuTTY to remotely access our raspberry pi devices. File Zila is used to upload application files to the cloud. Different programming languages such as Python, Node.js, Angular.js are used in development of virtual network.

Table 1. Hardware and software services used in experimental setup.

	Component	**Description**
Raspberry Pi	Model	Raspberry Pi 3, Model B
	Operating System	Raspbian Operating System
	Network	802.11n Wi-Fi
	CPU	1.2 GHz quad-core ARM Cortex A53
	GPU	Broadcom VideoCore IV 400 MHz
	SD Card	16 GB
	RAM	1 GB of LPDDR2-900 SDRAM
Software Tools	WebStorm	WebStorm 2019.2 for development in Javascript
	Sublime Editor	Sublime Text version 3 for designing web application interface
	Thonny	Thonny for development in python
	File Zila	FileZila for uploading client application.
	PuTTY	PuTTY is used to remotely access Raspberry Pi and EC2 instance.
Programming Languages	Python	Python 3.7
	Node.js	Node.js 4.8.1
	Angular JS	AngularJS 1.1
	HTML	Html is used for designing front-end interface
	CSS	CSS is used for styling the web application interface
Libraries and API's	Paho-mqtt	MQTT version 3.1
Database	MySQL	MySQL 5.5 is used to store Metadata of IoT virtual objects
Cloud Services	Amazon Web Service	Elastic Compute Cloud (EC2) Instance running on the AWS cloud
	Operating System	Ubuntu Server 16.04 LTS
	CPU	1 vCPUs 2.5 GHz Intel Xeon Family
	Memory	1 GB

Table 2 presents the various protocols used in our experiment. HTTP protocol is used for accessing web application online. MQTT protocol is used to publish and subscribe message from client to server and server to client. Secure Shell Protocol (SSH) is used to access Raspberry pi remotely. We use 5 raspberry pi and access them remotely using PuTTY software.

Table 2. Various protocols used for communication in the virtual IoT network.

Protocol(s)	Usage Description in Proposed System
HTTP	Hyper-text Transfer Protocol runs on port 80 by default and is used to access web based applications online.
MQTT	Message Queuing Telemetry Protocol is a light weight protocol that is based on publish/subscribe model. The default port is 1883.
Web Sockets	Web socket works over HTTP protocol. Web sockets are capable of providing full-duplex communication on a single TCP channel.
SSH	Secure Shell Protocol is used to access the EC2 instance on the Amazon cloud. The default port for SSH is 22.

Table 3 presents the sensors used in the smart home during the experiment. We used different types of sensors in our experiment. We used a location-based sensor, and physical sensors. In the location-based sensor we used a motion detection sensor and an ultrasonic sensor. In the physical sensors we used temperature, humidity, and pressure sensors. We used two different types of physical sensors to calculate humidity, temperature, and pressure. We used BME280 and BMP280 model sensors to calculate humidity, temperature, and pressure in our experiment in smart home.

Table 3. Detail Information of Sensors used in experimental setup.

	Sensor	Model	Smart Home Application	Function
Location based Sensor(s)	Motion Sensor	PIR Motion Sensor	Motion Detection sensor is used to detect motion of anything in the house.	Publish message to virtual object if movement of any object is detected.
Environmental Sensors	Temperature, Humidity, Pressure	BME280	BME280 sensor is used to calculate Temperature, humidity, and pressure.	Publish sensor values to web server via MQTT protocol.
	Temperature, Humidity, Pressure	BMP280	BMP280 sensor is used to calculate Temperature, Humidity, Pressure.	Publish sensor values to web server via MQTT protocol.
	Gas Sensor	MQ-2 Gas Sensor Brick	Gas detectors can be used to detect combustible, flammable and toxic gases, and oxygen depletion	

Table 3 summarizes the sensors used in the virtual IoT network. It briefly describes the model used in this experiment and role of sensors in smart home. It also explains the applications of those sensor in the smart homes. There are different functions to be performed by the sensors such as sensing the values from the surroundings and sending it to the web server via MQTT protocol. We used two types of sensors in our experiment i.e., location based sensors and environmental sensors. Location based sensor are those sensors which are used to detect motion of any object or track location of some object. Environmental sensors are those sensors which collect values from the surrounding and exchange data with other IoT devices to perform some specific action. Table 4 summarizes the actuators used in the virtual IoT network.

Table 4. Detail information of actuators used in experimental setup.

Actuator(s)	Model	Smart Home Application	Function
Motor	Tower Pro Sg90.	Servo Motors are small and efficient but critical for use in applications requiring precise position control.	To turn the arm left, right.
Fan	Keyestudio Fan.	Fan is used in smart home to cool the temperature.	To turn On/Off the Fan.
LED	Simple	LED is used in smart home for lightning.	To turn On/Off the LED.

5. Case Study

This section briefly explains the experimental environment used for the design and development of virtual IoT network. Figure 9 represents the formation of physical network in smart home. We have used five different raspberry pi as IoT devices. The sensors and actuators are attached with the raspberry pi. We have specified GPIO pins for each IoT device. We used three sensors and three actuators in our experiments. The details of sensors and actuators are given in Tables 3 and 4. The environment is setup for a smart home. There are two types of sensors used in this experiment i.e., location based sensors and environmental sensors. Location based sensors are used for detecting motion or tracking location of any object in the smart home. A motion sensor is used for this purpose. Environmental sensors are those sensors that collect data from surroundings and exchange data with other IoT devices for some specific action.

Figure 9 represents the case scenario for the proposed virtual IoT network. In this example we placed sensors and actuators to form a physical network in a smart home. Each physical deivce is registered with virtual network and has a corresponding virtual object in the cloud. The main purpose of virtual object is that we can change the network configuration and communication flow without changing the physical network. The user can map the physical devices with the registered virtual objects. After mapping the virtual objects, the user can setup the route for communication flow. In the application layer, user has map the physical devices with virtual objects. The user then created a route for communication flow e.g. Sensor-T1 to Actuator-F1. The communication flow can be changed without changing the physical network. The user can easily update the route by sending the desired network configuration to the IoT controller.

We used MQTT protocol to transfer sensor values to virtual network. Sensor publish data to the virtual network and actuators are subscribed to specific topics to perform some specific actions. In Figure 9, BME280 temperature sensor is denoted by Sensor-T1 and Fan is denoted by Actuator-F1. The user first setup the route for these devices and set a condition as if temperature is greater than 30°, then turn on the fan. The sensor will send the values to web server. The IoT controller will receive the data and check the mapping and routing information and broadcast command to corresponding virtual object, the virtual object will then publish the command to physical device. The fan will receive the command and interpret accordingly.

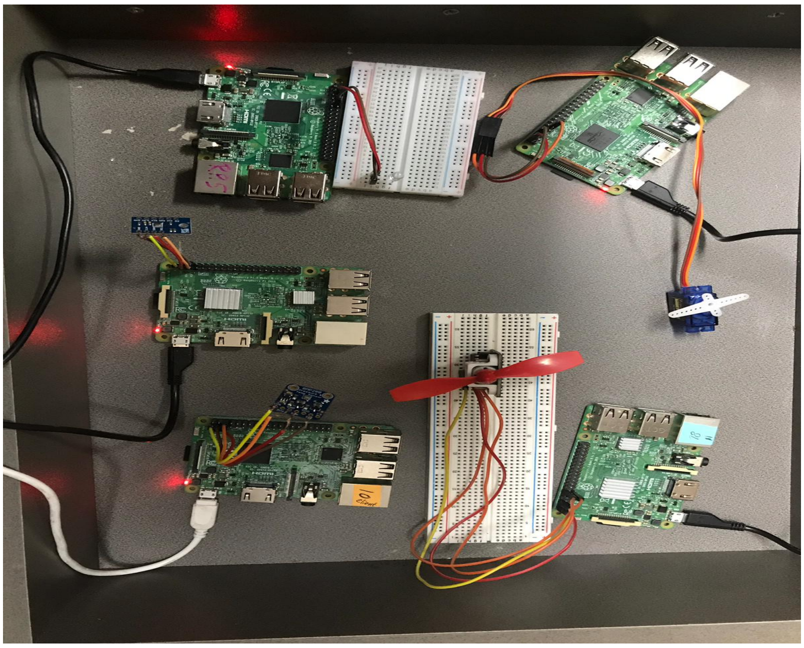

Figure 9. Physical IoT network in smart home.

6. Results

In this section, we will evaluate the performance of the proposed work. Figure 10 represents the efficiency of virtual object registration for the proposed system. Figure 10 shows the number of registered virtual objects per second. Results show that minimum number of registered virtual objects per second is 14. Maximum number of registered virtual objects per second is 29. Average number of registered virtual objects per second is 20 (stdev. 3.1).

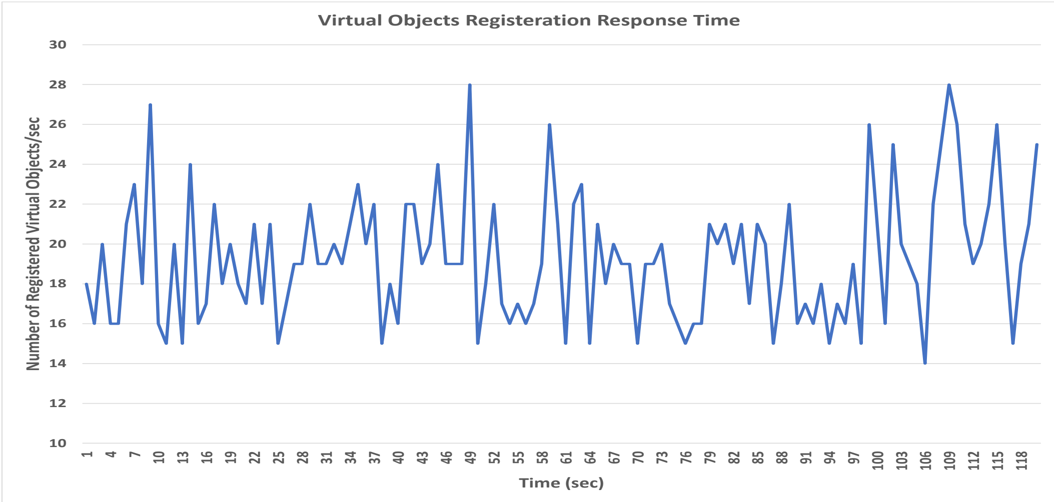

Figure 10. Virtual object registration response time.

Figure 11 represents the round trip time taken for IoT device activation and deactivation. Round trip time is the sum of time it takes for a signal to be sent and length of time it takes for an acknowledgment of the signal to be received. This time delay includes the propagation times for the paths between the communication endpoints i.e., sensor and actuator. Round trip time includes the time from sensor value to corresponding virtual object to IoT controller and from IoT controller to desired virtual object, and from the that virtual object to corresponding actuator.

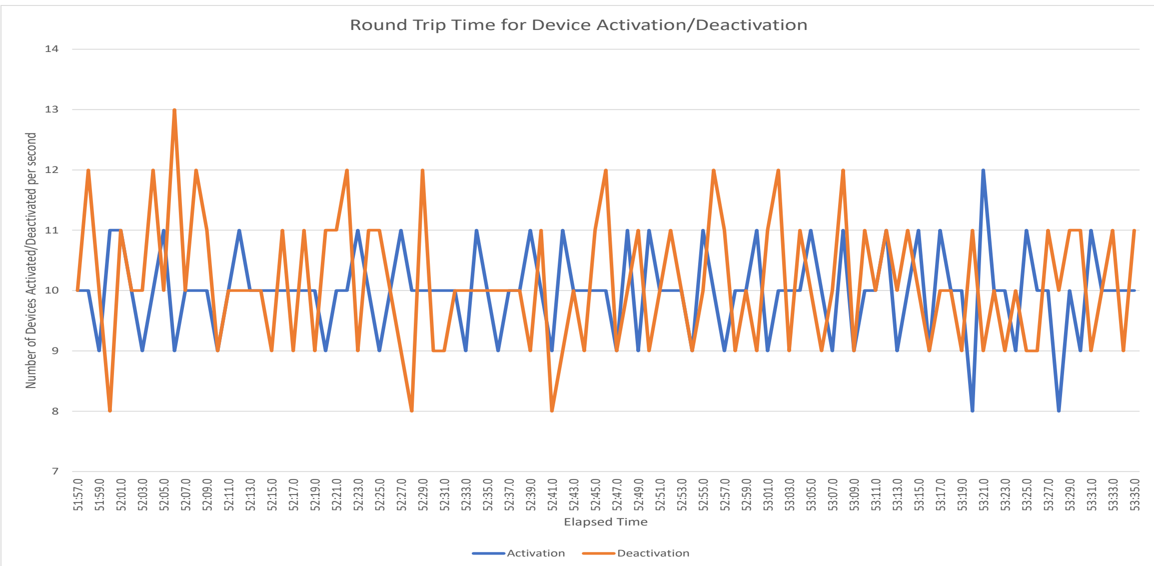

Figure 11. Round trip time for device activation/deactivation.

To calculate network latency, we have used total of 50 IoT devices. Latency means the total amount of delay in some application. Network latency means, the total amount of delay in network communication. Applications with low network latency are considered reliable and efficient. Network latency is calculated as total amount of delay in communication from source to destination. Small delays in network are normal but if there are long delays in network communication, it can create problems in the network. In Figure 12, we can see that as we increase the number of IoT devices, the network delay is lower in virtual network as compared to physical network. There is a difference of 1 sec when there are total 50 IoT devices. Results show that communication is efficient and reliable via virutal network.

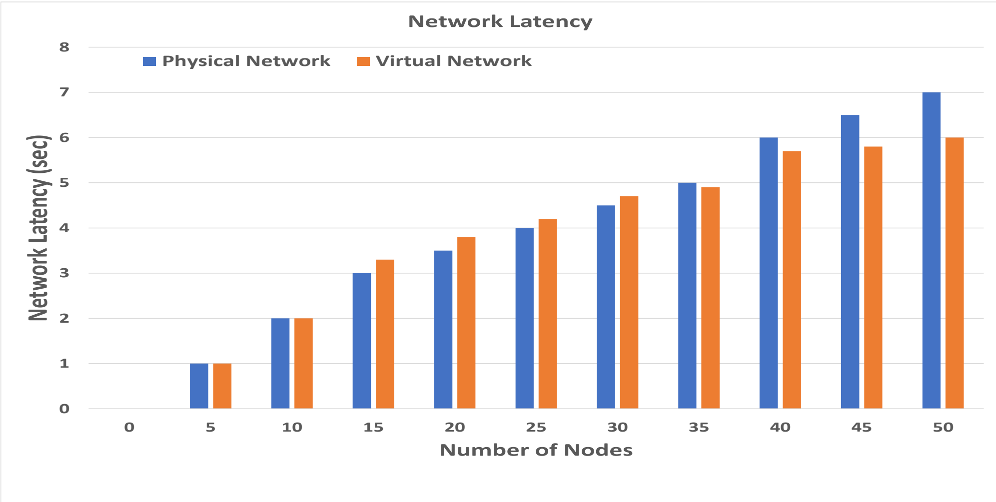

Figure 12. Comparison of network latency between virtual IoT network and physical network.

There are two ways to evaluate the impact of growing network size on throughput, either increase number of packets per second or increase number of sources. We have used the first approach i.e increase the number of packets per second to measure the performance of the proposed virtual IoT network. In these experiments, we used different sensors and actuators acting as nodes. We have calculated the throughput and end-to-end delay results by varying the date rates of 60, 80, 100, and 120 packets/second as shown in Figures 13 and 14. These results show the variation in throughput and end-to-end delay during the experiment.

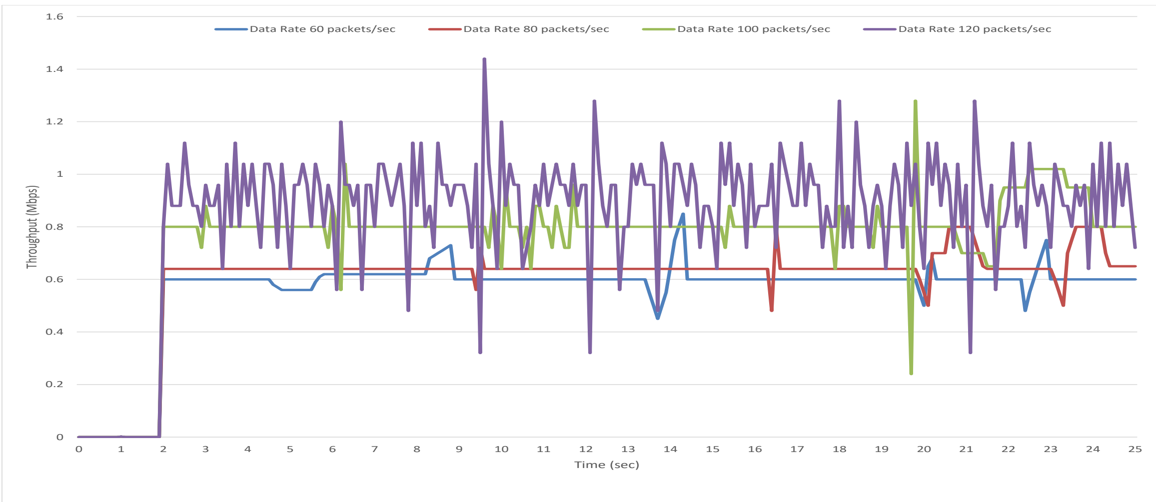

Figure 13. Throughput results.

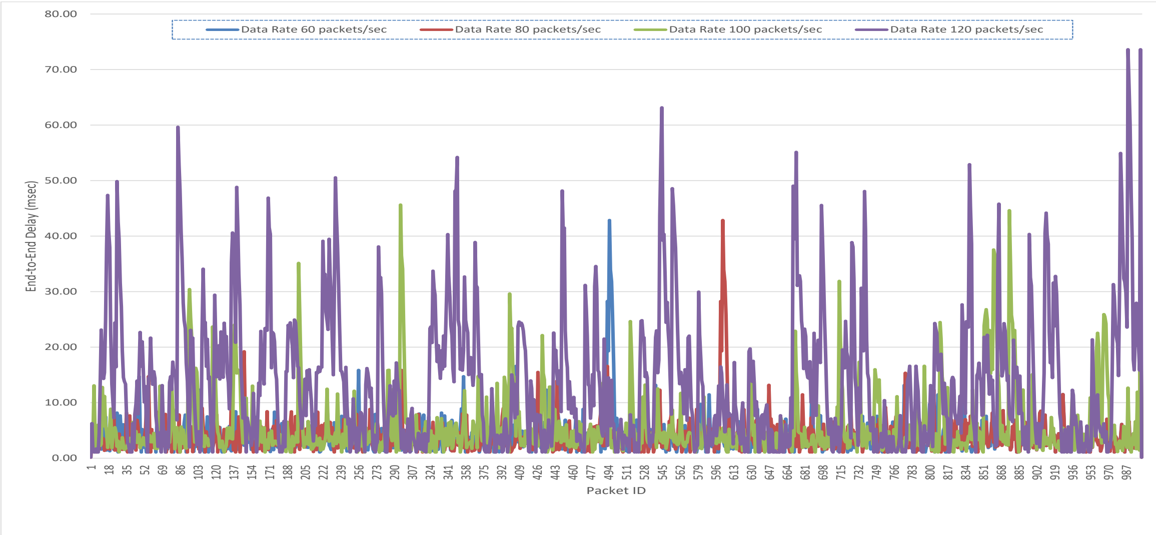

Figure 14. End to end delay.

We have used six different IoT resources that act as sensors and actuators. During the experiment we have calculated the total data generated in the network by varying packets/sec. When the data rate was set to 60 packets/sec and packet size of 1024 bytes, the mean value of throughput calculated at all IoT devices was 3.4 Mbps with a standard deviation of 0.03 Mbps. This indicates that the packet delivery ratio in the network is 100% with data rate of 60 packets/sec. When we vary the data rate, the packet delivery ratio decreases as shown in Figure 15.

Figure 14 shows that there are minor variations in end-to-end delay results for data rate of 60 packets/sec. When the data rate is set to 60 packets/sec and packet size of 1024 bytes, the mean value of end-to-end delay was 3.95 ms with a standard deviation of 3.3 ms. When we increase the number of packets to 100 packets/sec, there is also variations in the end-to-end delay results as shown in Figure 14. The mean value of end-to-end delay results for data rate of 120 packets/second is noted as 6.9 ms with a standard deviation of 4.6 ms which is approximately 65% more as compared to data rate of 60 packets/sec.

Figure 15 shows that with an increase in the data sending rate there is significant decrease in the packet delivery ratio. From Figure 15, we can see that packet delivery ratio for 60 packets/sec is 100% whereas packet delivery ratio for 80 packets/sec, 100 packets/sec, 120 packets/sec is approximately 98.2%, 98%, and 88% respectively.

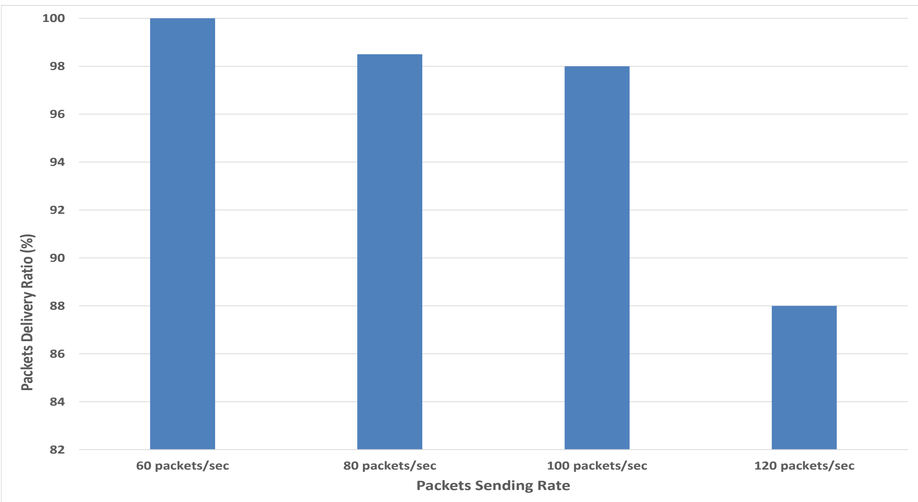

Figure 15. Packet delivery ratio.

7. Discussion

This section focuses on the challenges, implications, and limitations in the development of proposed virtual network. A virtual network based on IoT controller is developed to provide efficient, reliable, and scalable networking. There are several challenges in the development of the virtual network such as deployment and maintenance of physical devices, initialization of virtual objects, mapping between the physical device and virtual object, managing data rate, handling the network traffic by controlling the number of packets exchanged, data storage and processing. There are some unpredictable situations such as power failure of a device, Internet dis-connectivity, and network latency.

The advantages of the IoT controller based virtual network are low network latency, high fault tolerance, and high packet delivery ratio. As we increase the number of devices, the physical network communication is slow as compared to the virtual network. In the physical network, if there is some fault in the device, all other devices get effected and there is disturbance in the communication network where as if communication between devices is via virtual network, if the physical device is faulty, then IoT controller notifies the administrator so it could be fixed. Also the packet delivery ratio is higher in virtual network as compared to the physical network.

These findings have several implications. First, the proposed virtual network is based on IoT controller whom concept is driven from SDN controller. SDN technique separates the control and data plane. IoT controller also separates the data and control plane. IoT controller is different from SDN in a sense that SDN is used for switches and routers where as IoT controller is used for end-devices. The protocol between the controller and switches is OpenFlow whereas in the proposed virtual network we used light weight MQTT protocol. According to the authors best knowledge, this is a novel approach towards the design and implementation of a virtual network based on IoT controller.

Table 5 summarizes various IoT networks and their comparison with the proposed virtual IoT network. We selected the necessary attributes for any open-source tools in general to compare the performance of the IoT networks. There are many open-source and paid platforms that allow creating IoT networks such as Zetta, SiteWhere, and ThingsBoard. These platforms provide cloud computing services such as data management and computation. These platforms perform specific purpose and differ in providing services when compared to each other. The proposed system differs with other networks by separating control and data planes which makes it flexible and agile. The proposed system also provides virtualization in the IoT network to dynamically control traffic congestion, handling mapping requests, and routing mechanisms.

Even though the objective of the study was achieved, there were some limitations identified in the virtual network. Firstly, virtual objects are not intelligent enough to take decisions if the physical device is not active. The routing mechanism is solely dependent on the IoT controller. If the virtual object is down for some reason, it cannot find an alternative route for communicating with the physical device. To enhance the capabilities of the virtual network, virtual objects need to be intelligent and independent from the IoT controller. Secondly, routing tables provide an efficient and reliable way for packet forwarding. The proposed system is not dynamic to change the routing between the devices on the fly. The IoT controller is responsible for establishing desired network settings by manipulating virtual objects in the virtual network. Thirdly, network settings may change over time so it is necessary to make a network dynamic. In future we would like to work on a dynamic virtual network where users can update mapping between the physical device and virtual object, and update the routing among virtual objects.

Table 5. Comparative analysis of proposed network with existing state-of-the-art tools.

Name	Cloud Support	Open Source	Major Domain	Maintenance Status	Virtualization	Protocol(s)
ThingsBoard	Yes	Yes	Real-time data collection, processing, visualization, and device management.	Yes	No	MQTT
Kaa	Yes	No	Middleware IoT platform that is used for multipurpose and provides solutions to end-to-end devices.	Yes	No	MQTT, CoAP
SiteWhere	Yes	Yes	SiteWhere platform offers the ingestion, repository, processing, and assimilation of device inputs	Yes	No	MQTT, Stomp, AMQP
Zetta	Yes	Yes	Zetta Assembles IoT devices into data-intensive, real-time applications.	No	No	MQTT
Proposed Virtual IoT Network	Yes	Yes	Separation of data and control plane. Provides IoT services virtually.	Yes	Yes	MQTT

8. Conclusions and Future Directions

This paper presents an idea of a virtual network based on IoT controller in the cloud of things. In this study, we designed and developed a virtual IoT network that is based on the concept taken from SDN Controller. The proposed system aims to provide a platform where users can easily register physical IoT devices and initialize corresponding virtual objects. After the initialization of the virtual objects, the user can map virtual objects with physical devices to create a connection between them for further communication. The main role of SDN is to separate the control and data plane, so we introduced the IoT controller in the proposed virtual network. The main purpose of the IoT controller is to handle mapping requests and routing information between physical devices and virtual objects in the virtual network. Users can easily create a virtual network by mapping devices with virtual objects according to the desired network configuration. The proposed system allows a user to map one-to-one or one-to-many devices. We have conducted a performance analysis of the proposed system by measuring virtual object initialization response time and round trip time for device activation and deactivation. Various experiments are conducted by changing the network size and packet data transfer rate. Results show that if the network traffic increases beyond a certain limit, the performance of the network slows down. Traffic congestion can be handled dynamically by the IoT controller. The IoT controller sets the number of packets/second to avoid traffic congestion. In the future, we will extend this work by making the virtual object intelligent enough to support complex interconnection among IoT devices. This will allow the virtual object to find an alternative route if for any reasons some virtual objects are down in the virtual network. This will enhance the capabilities of the virtual objects and hence improve the performance of the virtual IoT network.

Author Contributions: F.M. conceived the idea for this paper, designed the experiments and wrote the paper; I.U. and S.A. assisted in model designing and experiments. D.-H.K. conceived the overall idea of Virtual IoT Network, and proof-read the manuscript. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This research was supported by Energy Cloud R&D Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (2019M3F2A1073387), and this research was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2019-0-01456, AutoMaTa: Autonomous Management framework based on artificial intelligent Technology for adaptive and disposable IoT). Any correspondence related to this paper should be addressed to Do-Hyeun Kim.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, S.K.; Bae, M.; Kim, H. Future of IoT networks: A survey. *Appl. Sci.* **2017**, *7*, 1072. [CrossRef]
2. Park, E.; Del Pobil, A.P.; Kwon, S.J. The role of Internet of Things (IoT) in smart cities: Technology roadmap-oriented approaches. *Sustainability* **2018**, *10*, 1388. [CrossRef]
3. Wu, S.M.; Chen, T.C.; Wu, Y.J.; Lytras, M. Smart cities in Taiwan: A perspective on big data applications. *Sustainability* **2018**, *10*, 106. [CrossRef]
4. Kharrazi, A.; Qin, H.; Zhang, Y. Urban big data and sustainable development goals: Challenges and opportunities. *Sustainability* **2016**, *8*, 1293. [CrossRef]
5. Ferrández-Pastor, F.J.; Mora, H.; Jimeno-Morenilla, A.; Volckaert, B. Deployment of IoT edge and fog computing technologies to develop smart building services. *Sustainability* **2018**, *10*, 3832. [CrossRef]
6. Cirani, S.; Ferrari, G.; Mancin, M.; Picone, M. Virtual Replication of IoT Hubs in the Cloud: A Flexible Approach to Smart Object Management. *J. Sens. Actuator Netw.* **2018**, *7*, 16. [CrossRef]
7. Mehmood, F.; Ahmad, S.; Kim, D. Design and Implementation of an Interworking IoT Platform and Marketplace in Cloud of Things. *Sustainability* **2019**, *11*, 5952. [CrossRef]
8. Jamil, F.; Iqbal, M.A.; Amin, R.; Kim, D. Adaptive thermal-aware routing protocol for wireless body area network. *Electronics* **2019**, *8*, 47. [CrossRef]
9. Jin, W.; Kim, D. Consistent registration and discovery scheme for devices and Web service providers based on RAML using embedded RD in OCF IoT network. *Sustainability* **2018**, *10*, 4706. [CrossRef]
10. Mzahm, A.M.; Ahmad, M.S.; Tang, A.Y. Agents of Things (AoT): An intelligent operational concept of the Internet of Things (IoT). In Proceedings of the 2013 13th International Conference on Intellient Systems Design and Applications, Bangi, Malaysia, 8–10 December 2013; pp. 159–164.
11. Ahmad, S.; Mehmood, F.; Kim, D.H. A DIY approach for the design of mission-planning architecture using autonomous task–object mapping and the deployment model in mission-critical IoT systems. *Sustainability* **2019**, *11*, 3647. [CrossRef]
12. Leppänen, T.; Riekki, J. A lightweight agent-based architecture for the Internet of Things. In Proceedings of the IEICE workshop on Smart Sensing, Wireless Communications, and Human Probes, Wuxi, China, 4–5 March 2013; pp. 2–4.
13. Ahmad, S.; Mehmood, F.; Mehmood, A.; Kim, D. Design and Implementation of Decoupled IoT Application Store: A Novel Prototype for Virtual Objects Sharing and Discovery. *Electronics* **2019**, *8*, 285. [CrossRef]
14. Longe, O.M.; Ouahada, K.; Rimer, S.; Harutyunyan, A.N.; Ferreira, H.C. Distributed demand side management with battery storage for smart home energy scheduling. *Sustainability* **2017**, *9*, 120. [CrossRef]
15. Nakagawa, I.; Shimojo, S. IoT agent platform mechanism with transparent cloud computing framework for improving IoT security. In Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, Italy, 4–8 July 2017; pp. 684–689.
16. Leppänen, T.; Heikkinen, A.; Karhu, A.; Harjula, E.; Riekki, J.; Koskela, T. Augmented reality web applications with mobile agents in the internet of things. In Proceedings of the 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies, Oxford, UK, 10–12 September 2014; pp. 54–59.
17. Coulter, R.; Pan, L. Intelligent agents defending for an IoT world: A review. *Comput. Secur.* **2018**, *73*, 439–458. [CrossRef]
18. Zhao, C.W.; Jegatheesan, J.; Loon, S.C. Exploring iot application using raspberry pi. *Int. J. Comput. Netw. Appl.* **2015**, *2*, 27–34.
19. Schwartz, M. *Internet of Things with ESP8266*; Packt Publishing Ltd.: Birmingham, UK, 2016.
20. Oliveira, G.M.; Costa, D.C.; Cavalcanti, R.J.; Oliveira, J.P.; Silva, D.R.; Nogueira, M.B.; Rodrigues, M.C. Comparison Between MQTT and WebSocket Protocols for IoT Applications Using ESP8266. In Proceedings of the 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, Italy, 16–18 April 2018; pp. 236–241.
21. Fortino, G.; Russo, W.; Savaglio, C.; Shen, W.; Zhou, M. Agent-oriented cooperative smart objects: From IoT system design to implementation. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *48*, 1939–1956. [CrossRef]
22. Mehmood, A.; Mehmood, F.; Song, W.C. Cloud based E-Prescription management system for healthcare services using IoT devices. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju-do, Korea, 16–18 October 2019; pp. 1380–1386.

23. Ahmad, S.; Malik, S.; Ullah, I.; Park, D.H.; Kim, K.; Kim, D. Towards the design of a formal verification and evaluation tool of real-time tasks scheduling of IoT applications. *Sustainability* **2019**, *11*, 204. [CrossRef]
24. Ullah, I.; Ahmad, S.; Mehmood, F.; Kim, D. Cloud Based IoT Network Virtualization for Supporting Dynamic Connectivity among Connected Devices. *Electronics* **2019**, *8*, 742. [CrossRef]
25. Son, Y.; Jeong, J.; Lee, Y. An adaptive offloading method for an IoT-cloud converged virtual machine system using a hybrid deep neural network. *Sustainability* **2018**, *10*, 3955. [CrossRef]
26. Yu, H.; Shen, Z.; Leung, C. From internet of things to internet of agents. In Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE cyber, Physical and Social Computing, Beijing, China, 20–23 August 2013; pp. 1054–1057.
27. Mzahm, A.M.; Ahmad, M.S.; Tang, A. Enhancing the internet of things (iot) via the concept of agent of things (aot). *J. Netw. Innov. Comput.* **2014**, *2*, 101–110.
28. Qiu, T.; Chen, N.; Li, K.; Atiquzzaman, M.; Zhao, W. How can heterogeneous Internet of Things build our future: A survey. *IEEE Commun. Surv. Tutor* **2018**, *20*, 2011–2027. [CrossRef]
29. Ali, B.; Awad, A.I. Cyber and physical security vulnerability assessment for IoT-based smart homes. *Sensors* **2018**, *18*, 817. [CrossRef] [PubMed]
30. Kim, M.; Lim, K.S.; Song, J.; Jun, M.S. An efficient secure scheme based on hierarchical topology in the smart home environment. *Symmetry* **2017**, *9*, 143. [CrossRef]
31. Mehmood, F.; Ullah, I.; Ahmad, S.; Kim, D. Object detection mechanism based on deep learning algorithm using embedded IoT devices for smart home appliances control in CoT. *J. Ambient. Intell. Humaniz. Comput.* **2019**. [CrossRef]
32. Mehmood, F.; Ahmad, S.; Kim, D. Design and implementation of automation appliances control based on MVC model using distributed MQTT broker in CoT networks. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 262–269.
33. Lee, W.; Cho, S.; Chu, P.; Vu, H.; Helal, S.; Song, W.; Jeong, Y.S.; Cho, K. Automatic agent generation for IoT-based smart house simulator. *Neurocomputing* **2016**, *209*, 14–24. [CrossRef]
34. Boussard, M.; Thai Bui, D.; Douville, R.; Justen, P.; Le Sauze, N.; Peloso, P.; Vandeputte, F.; Verdot, V. Future spaces: Reinventing the home network for better security and automation in the IoT era. *Sensors* **2018**, *18*, 2986. [CrossRef]
35. Du, K.K.; Wang, Z.L.; Mi, H. Human machine interactive system on smart home of IoT. *J. China Univ. Posts Telecommun.* **2013**, *20*, 96–99. [CrossRef]
36. Khudoyberdiev, A.; Jin, W.; Kim, D. A Novel Approach towards Resource Auto-Registration and Discovery of Embedded Systems Based on DNS. *Electronics* **2019**, *8*, 442. [CrossRef]
37. Fortino, G.; Guerrieri, A.; Russo, W.; Savaglio, C. Integration of agent-based and cloud computing for the smart objects-oriented IoT. In Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design (CSCWD), Hsinchu, Taiwan, 21–23 May 2014; pp. 493–498.
38. Talari, S.; Shafie-Khah, M.; Siano, P.; Loia, V.; Tommasetti, A.; Catalão, J.P. A review of smart cities based on the internet of things concept. *Energies* **2017**, *10*, 421. [CrossRef]
39. Savaglio, C.; Fortino, G.; Zhou, M. Towards interoperable, cognitive and autonomic IoT systems: An agent-based approach. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; pp. 58–63.
40. Yang, C.; Lan, S.; Shen, W.; Huang, G.Q.; Wang, X.; Lin, T. Towards product customization and personalization in IoT-enabled cloud manufacturing. *Clust. Comput.* **2017**, *20*, 1717–1730. [CrossRef]
41. Jie, Y.; Pei, J.Y.; Jun, L.; Yun, G.; Wei, X. Smart home system based on iot technologies. In Proceedings of the 2013 International Conference on Computational and Information Sciences, Hubai, China, 21–23 June 2013; pp. 1789–1791.
42. Calvaresi, D.; Marinoni, M.; Sturm, A.; Schumacher, M.; Buttazzo, G. The challenge of real-time multi-agent systems for enabling IoT and CPS. In Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, 12–14 August 2017; pp. 356–364.
43. Nitti, M.; Pilloni, V.; Colistra, G.; Atzori, L. The virtual object as a major element of the internet of things: A survey. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1228–1240. [CrossRef]

44. Kelaidonis, D.; Somov, A.; Foteinos, V.; Poulios, G.; Stavroulaki, V.; Vlacheas, P.; Demestichas, P.; Baranov, A.; Biswas, A.R.; Giaffreda, R. Virtualization and cognitive management of real world objects in the internet of things. In Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, Besancon, France, 20–23 November 2012; pp. 187–194.
45. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 2–4 August 2012; pp. 13–16.
46. Aazam, M.; Huh, E.N. Fog computing: The cloud-iotVioe middleware paradigm. *IEEE Potentials* **2016**, *35*, 40–44. [CrossRef]
47. Rivas, A.; Chamoso, P.; Rodríguez, S. An agent-based Internet of Things platform for distributed real time machine control. In Proceedings of the 2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB), Salamanca, Spain, 12–15 September 2017; pp. 1–5.
48. Savaglio, C.; Fortino, G.; Ganzha, M.; Paprzycki, M.; Bădică, C.; Ivanović, M. Agent-based computing in the internet of things: A survey. In Proceedings of the International Symposium on Intelligent and Distributed Computing, Guimarães, Portugal, 12–14 October 2017; pp. 307–320.
49. Kwan, J.; Gangat, Y.; Payet, D.; Courdier, R. An agentified use of the Internet of Things. In Proceedings of the 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Chengdu, China, 15–18 December 2016; pp. 311–316.
50. Cavalcante, E.; Pereira, J.; Alves, M.P.; Maia, P.; Moura, R.; Batista, T.; Delicato, F.C.; Pires, P.F. On the interplay of Internet of Things and Cloud Computing: A systematic mapping study. *Comput. Commun.* **2016**, *89*, 17–33. [CrossRef]
51. Hammoudeh, M.; Arioua, M. Sensors and actuators in Smart Cities. *J. Sens. Actuator Netw.* **2018**, *7*, 8. [CrossRef]
52. Srinivasan, C.; Rajesh, B.; Saikalyan, P.; Premsagar, K.; Yadav, E.S. A review on the different types of Internet of Things (IoT). *J. Adv. Res. Dyn. Control Syst.* **2019**, *11*, 154–158.
53. Farahzadi, A.; Shams, P.; Rezazadeh, J.; Farahbakhsh, R. Middleware technologies for cloud of things: A survey. *Digit. Commun. Netw.* **2018**, *4*, 176–188. [CrossRef]
54. Jung, E.; Cho, I.; Kang, S.M. iotSilo: The agent service platform supporting dynamic behavior assembly for resolving the heterogeneity of IoT. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 608972. [CrossRef]
55. Kibria, M.G.; Kim, H.S.; Chong, I. IoT learning model based on virtual object cognition. In Proceedings of the 2016 International Conference on Information Networking (ICOIN), Kinabalu, Malaysia, 13–15 January 2016; pp. 369–371.