entropy

Article

Sleep Information Gathering Protocol Using CoAP for Sleep Care

Joo Ho Choi [1], Un Gu Kang [2] and Byung Mun Lee [2,*]

1 Department of IT Convergence, Gachon University, Seongnam-si 13120, Korea; onlysm0707@nate.com
2 Department of Computer Engineering, Gachon University, Seongnam-si 13120, Korea; ugkang@gachon.ac.kr
* Correspondence: bmlee@gachon.ac.kr; Tel.: +82-31-750-4756

Received: 9 June 2017; Accepted: 25 August 2017; Published: 28 August 2017

Abstract: There has been a growing interest in sleep management recently, and sleep care services using mobile or wearable devices are under development. However, devices with one sensor have limitations in analyzing various sleep states. If Internet of Things (IoT) technology, which collects information from multiple sensors and analyzes them in an integrated manner, can be used then various sleep states can be more accurately measured. Therefore, in this paper, we propose a Smart Model for Sleep Care to provide a service to measure and analyze the sleep state using various sensors. In this model, we designed and implemented a Sleep Information Gathering Protocol to transmit the information measured between physical sensors and sleep sensors. Experiments were conducted to compare the throughput and the consumed power of this new protocol with those of the protocols used in the existing service—we achieved the throughput of about two times and 20% reduction in power consumption, which has confirmed the effectiveness of the proposed protocol. We judge that this protocol is meaningful as it can be applied to a Smart Model for Sleep Care that incorporates IoT technology and allows expanded sleep care if used together with services for treating sleep disorders.

Keywords: IoT; healthcare; sleep care; mobile; CoAP; application protocol; 5G network

1. Introduction

The Internet of Things (IoT) is a technology that allows objects connected to a network to provide various services through communication between people and objects. In recent years, IoT technology has been applied to various fields including health care, unmanned vehicles, and smart buildings. In particular, smart care is attracting attention as an IoT-based healthcare service, and interest in personalized services using IoT platforms and wearable devices is growing [1–3].

Meanwhile, our interest in our own health is growing as our society develops into an aging society. In particular, the paradigm of health care is changing from the treatment of disease to the treatment of chronic disease, thus requiring preventive health care [4,5]. Sleep management can prevent chronic diseases such as diabetes, stroke, and coronary artery disease. Sleep is a basic resting method, which is an important means of relieving mental stress, fatigue, and restoring vitality. However, since it is difficult for us to recognize the various sleep disorders as we sleep, we can unknowingly lack sleep, leading to sleep disorders. Sleep disorders should be diagnosed and treated because they may lead to chronic illness, learning disabilities, decreased efficiency, and accidents.

However, it costs a lot of time and money to visit the hospital to both diagnose and treat sleep disorders. Therefore, sleep care systems are being studied to prevent sleep disorders and to manage sleep from home. Sleep care systems are being developed in the form of a variety of sleep care devices and mobile sleep care applications [6,7]. For example, there are wearable devices that can measure sleep disorders such as snoring and tossing and turning, etc. [8]. However, the existing sleep care systems only use a single device to measure the sleep state, which makes it difficult to accurately judge the sleep disorder(s) based on the information measured. If IoT technology is used, these problems can be solved. IoT devices equipped with various sensors are distributed around the user to measure sleep environment and sleep information. That information is transmitted to the IoT device, which comprehensively analyzes the sleep state through a network, and accurately determines the sleep state based on the collected information. To accomplish this process, a suitable protocol is needed for transmitting the information measured between IoT devices. In this process, the protocol suitable for transmitting measurement information between IoT devices is required. Typical protocols are MQTT (Message Queue Telemetry Transport), CoAP (Constrained Application Protocol), and XMPP (Existing Messaging and Presence Protocol) [9,10].

IoT devices with various sensors are connected to the internet and continuously transmit the measurement data, so the cloud federation for IoT is needed to manage this. The method of using existing single servers has limitations in collecting, managing, and analyzing large amounts of generated data. Therefore, distributed processing through hypervisor virtualization by applying the cloud to IoT data collection technology can effectively apply IoT services [11,12].

In this paper, we propose a Smart Model for Sleep Care for IoT devices with various sensors to measure the sleep state and transmit the measurement information for integrated processing. This model can measure a variety of information through distributed IoT devices and utilize the servers and mobile devices used in existing internet environment. In particular, we apply cloud federation to provide real-time IoT services that require large amounts of data. In addition, we propose a CoAP-based SIGP (Sleep Information Gathering protocol) to transmit measurement information between IoT devices. If this protocol is used for sleep care, it will be possible to provide IoT services to collect measurements and analyze an individual's sleep state. This service enables it to judge sleep information accurately based on the distributed collection and integration processing by distributed sensor devices. Based on this, we designed and implemented a protocol to verify the effectiveness of the proposed protocol.

Section 1 serves as an Introduction. In Section 2, we address the contents of the sleep care service related to this paper and the protocols used in the IoT environment to measure them. In Section 3, we propose a protocol to transmit measurement information to other IoT devices equipped with various physical sensors. In Section 4, we address the results of experiments and evaluations using various sensors to verify the effectiveness of the proposed protocol. In Section 5, we address the results of the study and propose the directions for future studies.

2. Related Studies

2.1. Sleep Care Service

Sleep disorders are characterized by failing to take normal sleep due to issues such as increased stress, irregular sleep, and an irregular sleep cycle. Some people suffer from insomnia and respiratory disorder during sleep. Insomnia is a symptom of failing to get enough sleep because the person is not able to sleep normally. Respiratory disorders during sleep, such as snoring and apnea, cause sleep quality to be degraded. Sleep care services are being developed using various sleep care devices or mobile applications to identify or prevent such sleep disorders.

Figure 1 shows a variety of sleep care services. Figure 1a shows a sleep care device that measures sleep information through various sensors. The sound sensor attached to the user detects the sleep state of the user by detecting the sounds of snoring or tossing and turning, and measures the sleep environment through a temperature and humidity sensor [13,14].

For example, a wearable device such as a bracelet as shown in Figure 1b, provides a sleep care service. FLEX, which was developed by Fitbit, is a representative example. It provides information on activity, weight, and diet management as well as informing the user of sleep efficiency by measuring sleep time, tossing and turning, and sleep patterns. This contributes to the user's comprehensive health management [15,16]. As shown in Figure 1c, the sleep care pillow measures sleep disorders such as tossing and turning, incorrect sleep posture, snoring, and apnea. It measures snoring by judging the user's breathing state through the pressure sensor and vibration sensor attached to the pillow [17].

These devices also provide measurements in tandem with the mobile applications shown in Figure 1d [18]. Some smart phones are equipped with various sensors to provide various functions, and there are applications that measure and manage the sleep state independently even if they are not linked to the sleep care devices [19].

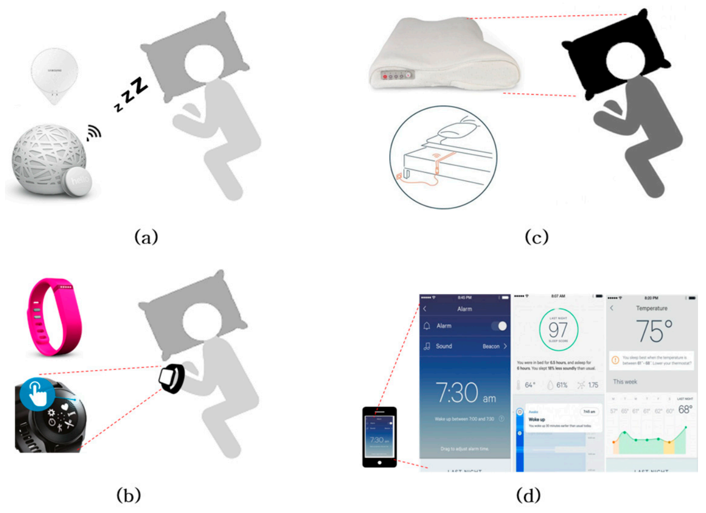


Figure 1. Sleep care services using device and mobile, (**a**) Sleep monitor device; (**b**) Sleep care pillow; (**c**) Sleep care watch; (**d**) Sleep care mobile application.

A variety of sleep care services allow users to check their sleep state or sleep environment to manage themselves, which contributes to proper sleep management. However, since sleep care devices and mobile applications operate independently, an integrated analysis of the information measured is required. If the measurements can be integrated and supplemented, it will enable more accurate sleep management. This can be achieved by using IoT technology, and can efficiently provide sleep care services. Measurements can be exchanged by autonomous data transmission among IoT devices equipped with various kinds of sensors. For this purpose, a protocol suitable for the IoT environment is needed to support communication between IoT devices and sensors.

2.2. IoT Application Protocol

IoT technology adapts its usability based on the network environment, transmission structure, and method. A transmission structure, a suitable utilization method, and an operating environment are required to provide IoT services in a restricted environment. Therefore, an application protocol should be applied according to the IoT environment and service provided. Typical protocols used for this purpose include HTTP (HyperText Transfer Protocol), MQTT, and CoAP.

HTTP is one of the typical protocols used in the internet environment as well as the IoT environment. This application-layer protocol operating on top of TCP (Transmission Control Protocol) /IP (Internet Protocol) follows the server/client model. When a client transmits a request, a server communicates with the request on a 1:1 basis. In addition, HTTP provides services in the form of RESTful APIs (Application Programming Interface), so it uses the methods such as GET, POST, PUT, and DELETE to inform the server of the location of resources and the type of request.

MQTT is a Publish/Subscribe messaging protocol optimized for an environment with limited bandwidth. Each Client operates on Broker, and based on a specific Topic, Client transmits information about Topic to Broker as Publish simultaneously receives updates to Topic as Subscribe. Therefore, it is easy to provide N: M service because Broker acts as a message bus that intermediates Topic's Publisher and Subscriber. In addition, it is designed to consume less battery, and there is no restriction on the message size due to the characteristics of Publish/Subscribe, which is advantageous in the development between heterogeneous platforms.

Finally, unlike HTTP or MQTT, CoAP operating on UDP (User Datagram Protocol) has been developed to allow devices with limited resources to communicate in a restricted network environment [9]. Although the communication using the UDP layer cannot not guarantee its reliability, CoAP supports CON (Confirmable) message and ACK (Acknowledgment) message, enabling a 1:1 reliable transmission like HTTP. It also has a feature of using the GET, POST, PUT, and DELETE methods of the RESTful API provided by HTTP.

Figure 2a shows a Request/Response Transmission Method of an HTTP message. The HTTP client requests the temperature value from the HTTP server using the GET method, and the server responds to the client with the result that the request has been successfully processed using the response code of 200. Figure 2b shows the message transmission method between CoAP Client and CoAP Server. The CoAP requests the temperature value from CoAP Server using the CON message and GET method. The CoAP Server that received the CON message responds by storing the result in the ACK message, which is a response message. In this process, it is possible to distinguish the message transaction through the message ID ([0xab12]), enabling reliable information transmission.

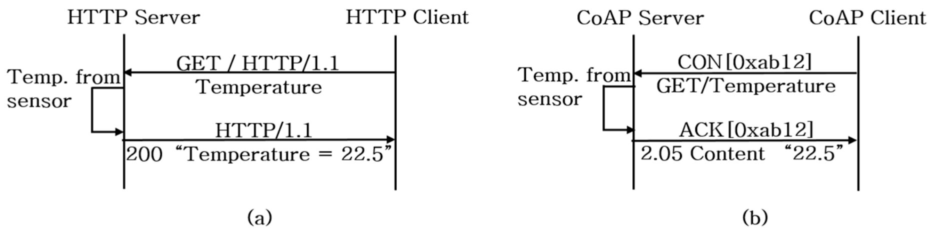


Figure 2. Message transfer method of HTTP and CoAP. (**a**) HTTP message transfer; (**b**) CoAP message transfer.

Figure 3 shows the CoAP service model that is most widely used in the IoT service. Each IoT, classified into a Server and a Client, requests and transmits the date needed to mutually provide services to users. In the Constrained Network, the IoT devices corresponding to the client request the CoAP and receive the services from the IoT devices corresponding to the server. For example, the client requests from the server the information measured by the sensor. The client requests the information measured by the sensor from the server, receives the results, and operates the actuator attached to the IoT device according to the service to be provided.

However, IoT application protocol should be able to provide various services through storage, analysis, processing and conversion of data through IoT platforms rather than simply providing services through a network interface. These services are provided in conjunction with the cloud to efficiently collect, analyze and manage large amounts of data due to the spread of IoT technology. The IoT cloud also includes various servers with virtual storage and processing capabilities, and the senor and actuator with virtual sensing and actuating capabilities. Through this, the IoT cloud collects information from all the sensors in the cloud and analyzes the collected information in an integrative way to derive the results, thereby providing various services to the user according to the result obtained [20–22]. However, the sensor and actuator with limited network environments have limitations in directly interacting with the IoT cloud. Therefore, if a protocol suitable for a limited network environment and an existing Internet environment is utilized, an interface suitable for each IoT device can be provided. The IoT service model using the CoAP, which is a typical model suitable for the limited environment, also provides the inter-HTTP interface that is commonly used in the existing internet environment through the CoAP-HTTP proxy [23,24]. In addition, since CoAP uses RESTful-based APIs like HTTP, it has the advantage of providing mapping between different protocols in a more efficient way [25,26].

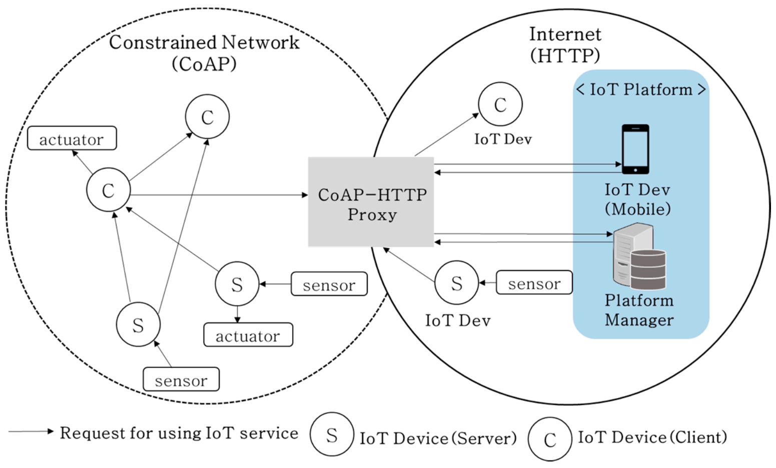


Figure 3. IoT service model using CoAP.

In this paper, we will apply a service method which can utilize both the restricted network and the existing internet environment. The sleep care service that uses this method can measure sleep information through various physical sensors and collect information necessary for judging the sleep state from the server or mobile device corresponding to the existing internet environment. Therefore, we want to use CoAP—an IoT application protocol that provides functions similar to those of HTTP together with HTT—which is most widely used in the existing internet environment. In addition, since CoAP can reliably transmit information on a 1:1 basis, it has the advantage of being able to integrate information measured by various sensors in a restricted network without loss of data. Therefore, we will propose a smart model for sleep care service using HTTP and CoAP together with an application protocol suitable for inter-IoT devices communication.

3. Sleep Information Gathering Protocol

3.1. Smart Model for Sleep Care Service

The proposed smart model for a sleep care service is a model that collects and integrates the information from an IoT device with distributed sensors to analyze a variety of sleep information. This model works by connecting each IoT device to a network to provide services to the IoT platform. It also provides two operation modes according to the IoT device connected to the network.

In Figure 4, IoT devices for sleep care are equipped with various physical sensors (such as a sound sensor, ambient light sensor, temperature sensor, and/or motion sensor) and operate at home. A physical sensor is a hardware type sensor that measures physical data that can be measured by a sound sensor, an illuminance sensor, or a temperature sensor. The data measured by each physical sensor is used to analyze the sleep state and sleep environment. The sensor that measures the data used to analyze sleep state includes a sound sensor and a motion detection sensor. The sound sensor detects the presence and magnitude of the sound generated by the user, and the motion sensor measures the motion of the user during sleep. The sensor that measures the data used to analyze sleep environment includes temperature sensors, humidity sensors, and illuminance sensors. Each sensor measures temperature, humidity, and illuminance during sleep. A virtual sensor in the form of software autonomously receives the data measured through a physical sensor and collects additional information by analyzing the information. The sleep sensor used in the smart model for sleep care service is an example of a virtual sensor. For a similar case, there is IEEE 1451, an international standard for smart sensors established by NIST (National Institute of Standards and Technology) and IEEE [27]. Although the IEEE 1451 smart sensor interface standard which has been standardized to date only specifies the point-to-point interface, the Sleep Sensor defined in this paper receives information from various physical sensors. By utilizing and analyzing information received, it generates and utilize various information such as sleep time, wakeup time, and awakening during sleep in a complex manner. As such, this model provides an operating method for analyzing sleep information by transmitting and collecting information measured by various sensors.

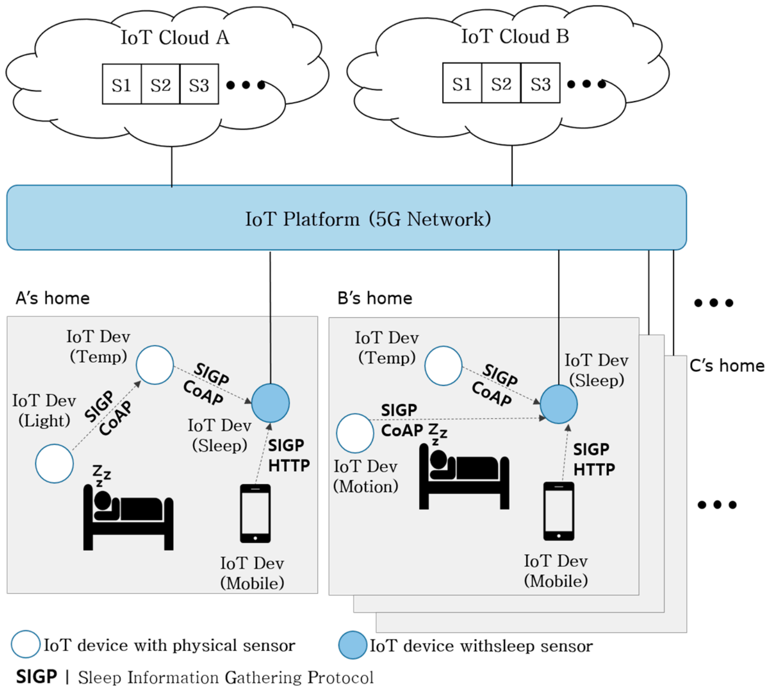


Figure 4. Smart model of information gathering for sleep care.

However, since the method of analyzing the sleep information based on sensor information cannot manage sleep systematically over the long term, an operation mode using IoT technology is required. Therefore, we propose an operation mode by applying the IoT technology to the sleep sensor. Each sleep sensor operating in our home is connected to the IoT platform, and receives the health information or sleep information of the user from the healthcare server or sleep care server connected to the IoT platform. The health information and sleep information of users stored in the virtualization storage is received from the IoT Cloud connected to the IoT platform. In addition, the information entered by the user can be utilized in conjunction with the mobile device. By integrating this information to analyze the sleep information, more accurate results can be obtained compared to the results obtained by using the sensor only. In order to provide a service to analyze sleep information in real time, the sleep sensor with two operational methods collects measurement information from the distributed physical sensor and information from IoT devices connected to the IoT platform. In this process, communication between the sleep sensor and the IoT device connected to the IoT platform utilizes an HTTP-based protocol, which is mainly used in the existing network. And in the communication between the sleep sensor and the physical sensor, CoAP, one of the IoT application protocols, is used. CoAP is suitable for communication with distributed physical sensors centered on a sleep sensor because it transmits reliable messages in 1:1 or 1:N form. In addition, it is easy to connect with the IoT platform through the sleep sensor because it has advantage of being similar to HTTP.

However, since the IoT platform provides multiple users with IoT services through a network of connected IoT devices, there may be a large amount of traffic or a delay in transmitting a large amount of data. According to a recent survey, the number of IoT devices connected to the internet by 2020 is expected to be about 20 billion to 40 billion. When using existing 4G networks, there is a limit to traffic that can accommodate all devices. However, the use of the 5G network can provide a real-time service by increasing the transmission rate 100 times compared to that of the existing 4G network and by minimizing the network latency. The 5G network is capable of high-speed data transmission using a bandwidth of more than a few hundred MHz within a high bandwidth of 300 GHz, which is beyond the limit of transmitting signals only within a low bandwidth of less than 3 GHz. The model suggested in this paper also uses IoT to provide sleep care services by transmitting data in real-time to various homes such as Home A, Home B and Home C. Therefore, the model suggested this study has considered the 5G network attracting attention as a new mobile communication network.

The sleep information analyzed by the sleep sensor is transmitted to the IoT Cloud and stored in the virtualization storage in the cloud to provide information to the IoT devices connected to the same IoT Cloud. For example, a user who has completed a sleep measurement can identify the sleep information transmitted to the IoT Cloud using a mobile application, thereby providing a customized service to induce positive sleep.

3.2. Protocol Design for the Model

In Figures 5–7, we propose the activities of the physical sensor, sleep sensor, and sleep care server operating in the Smart Model for Sleep Care service by each phase. Figure 5 shows the process of Discovery & Connection Phase in which the sleep sensor detects and connects the neighboring physical sensors.

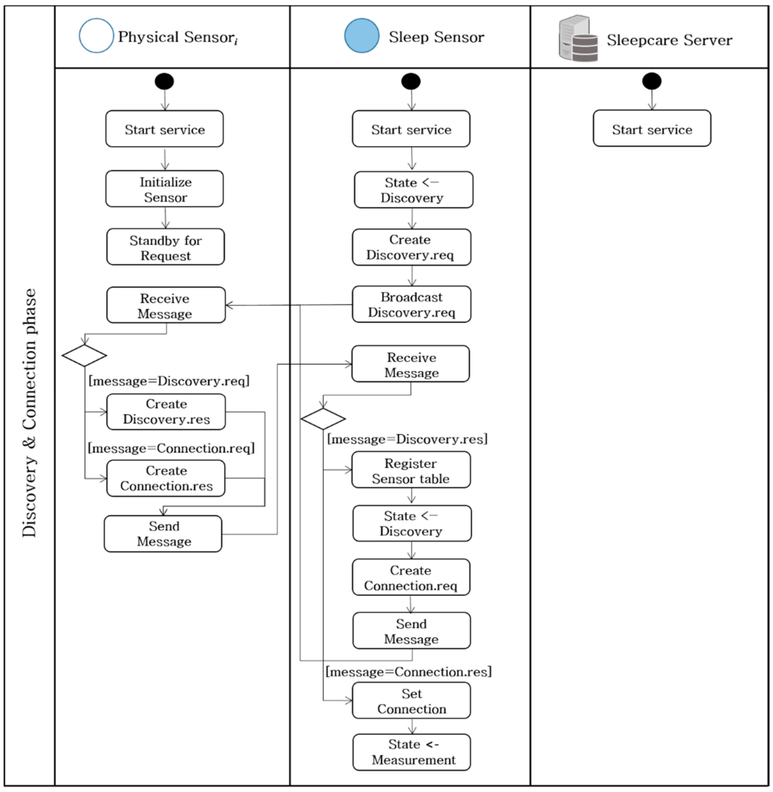


Figure 5. Activity diagram of a smart model for connecting sleep sensors and physical sensors.

When the service is started, the physical sensor resets the attached sensors and converts them into a standby state. The sleep sensor converts the state into Discovery, generates a Discovery.req message to search for the physical sensor around the user, and broadcasts the message. The physical sensor that sensed this request transmits a Discovery.res response and the sleep sensor that received the response message from the physical sensor generates the sensor table classified by the address of each physical sensor address and the type of measurement. Next, the sleep sensor converts the state to Connection and transmits a Connection.req message to the physical sensor based on the generated sensor table to request connection setup between the sensors. The physical sensor that received this message responds with a Connection.res message, and the sleep sensor establishes a setup between the sensors that received the response message. When the connection is complete, the sleep sensor converts the state into Measurement.

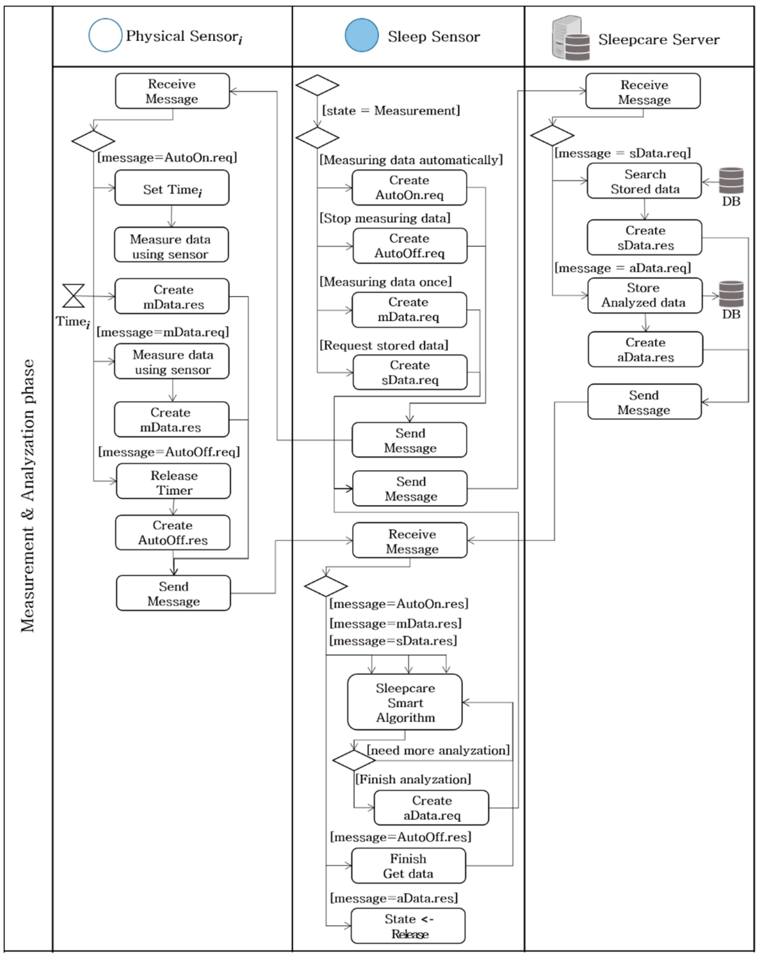


Figure 6. Activity diagram of smart model for measuring data.

Figure 6 shows the Measurement & Analysis Phase in which the sleep sensor receives data from both the physical sensor and the sleep care server to measure and analyze sleep state. The sleep sensor in the state of Measurement transmits an AutoOn.req or mData.req message to the physical sensor that has been completely connected to request data measurement. The AutoOn.req message is a request to automatically receive information measured from the physical sensor at regular intervals, and the mData.req message is a request to receive one value measured by the physical sensor. The physical sensor measures the state according to the request message.

Once the AutoOn.req message is received, the physical sensor sets the time to measure the data continuously and measures the data using the attached sensor. The data measured according to the time cycle is transmitted to the sleep sensor, and the result value is included in the AutoOn.res message and transmitted to the sleep sensor. The sleep sensor that received the AutoOn.res message obtains measurement data. After that, it can obtain values by receiving messages periodically. If measurement data is no longer needed, it send the AutoOff.req message from the physical sensor to stop the data measurement. The physical sensor that received this message releases the preset time and responds with the AutoOff.res message, and the sleep sensor completes the data measurement through the periodic request. If a mData.req message is received, the physical sensor also measures the data.

The measurement data is transmitted to the sleep sensor through the mData.res message, and the sleep sensor that received the message confirms the measurement data and completes data measurement.

To utilize the existing measurement and analysis information in addition to the information measured by the physical sensor, the sleep sensor sends a data inquiry request to the sleep care server. The message used for this purpose is sData.req. The sleep care server that received this message queries the stored information from the database and responds with a sData.res message.

The sleep sensor integrates the information received through AutoOn.res, mData.res, and sData.res messages and analyzes sleep state using the Sleep Care Smart Algorithm. The Sleep Care Smart Algorithm analyzes the received information in real time and terminates the sleep analysis when the sleep measurement is complete. The information analyzed by this process is stored in the sleep care server and used for sleep analysis in the future. Therefore, the sleep sensor sends a request to the sleep care server including the analyzed information in the aData.req message. The sleep care server that received this request responds with aData.res message and completes the data storage. After that, the sleep sensor converts the state into Release.

Finally, Figure 7 is the Release Phase in which the sleep sensor and physical sensor are disconnected. A sleep sensor in a state of Release transmits a Release.req message to the physical sensor because it does not need to maintain the connection with the physical sensor. The physical sensor that received this message generates a Release.res message and responds to the sleep sensor. The sleep sensor releases the connection with the physical sensor registered in the sensor table and ends the service.

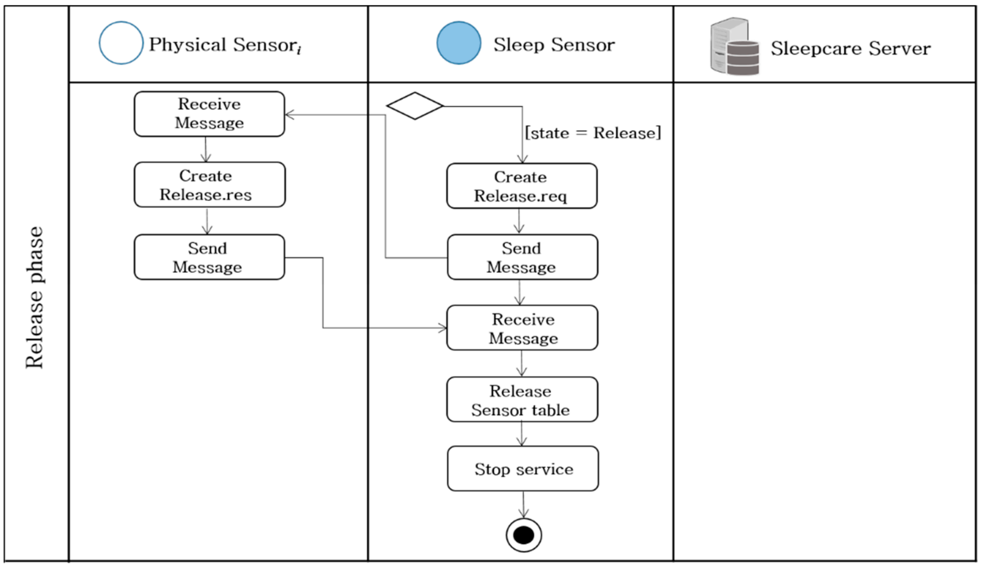


Figure 7. Activity diagram of smart model for releasing connection between sleep sensor and physical sensor.

Unlike the IoT device connected to the IoT platform in the smart model for sleep care service, the IoT device equipped with the physical sensor used in various environments needs the network protocol to provide a transmission service between Sleep sensors. If the CoAP—as described in previous studies—is used, that will allow it to utilize the IoT devices with limited resources more efficiently and to exchange the data between the virtual sensor, mobile devices, and the IoT platform through CoAP-HTTP Proxy. In addition, the secure communication in sleep care application is applied by applying the security standards provided by HTTP and CoAP. HTTPS, a security-enhanced version of HTTP, provides the ability to encrypt the session data through SSL (Secure Socket Layer) or TLS (Transport Layer Security) protocols. CoAP also provides security-enhanced CoAPS through DTLS

(Datagram Transport Layer Security). In short, it is possible to provide the smart model for a sleep care service more efficiently by using a CoAP-based Smart Information Gathering Protocol.

Figure 8 defines the structure of the transmission data of a Sleep Information Gathering Protocol for use in the smart model for sleep care service. The message header comprises the first to fourth fields. The first field SEQ indicates the serial number of message which is comprised of 2 bytes. The serial number is used most often in the AutoOn.req/AutoOn.res message which requests periodic sensor measurements and receives the results. Considering that the sensor measurement cycle, in general, is 0.8 s to 1 s, the number of messages used per sleep cycle is about 36,000 to 45,000. A message comprised of 1-byte size can utilize 65,535 messages. The second field is the ID of the IoT device that transmits packets. Since it stores the ID value for identifying each IoT device, it is comprised of 1 byte. The third field is comprised of 1 byte, indicating the length of the fourth field. The fourth field shows the address to utilize the URI in CoAP or HTTP. Each URI is used based on the REST API. Since each URI has a variable size, the field size is determined by the third field LENGTH. The decision to use the remaining items (except for the four fields described above) is determined depending on the message type. The DT field indicates the type of data measured by the sensor is configured in 1 byte. Time Field indicates the time of the sleep information measured. When the sleep sensor analyzes the measured sleep information in an integrative way, the sleep state of the user may vary according to the measurement time. Therefore, the time synchronization between the sensors is adjusted using Time Field. The Time field included in the aData.req message indicates the time at which sleep information was analyzed in the sleep sensor. The VALUE field indicates the measurement results, and the VL field configured in 1 byte is used to indicate the variable size of VALUE field. In addition, the Auto.req message for periodically receiving data sets the data transmission options through the 2-byte OPT field.

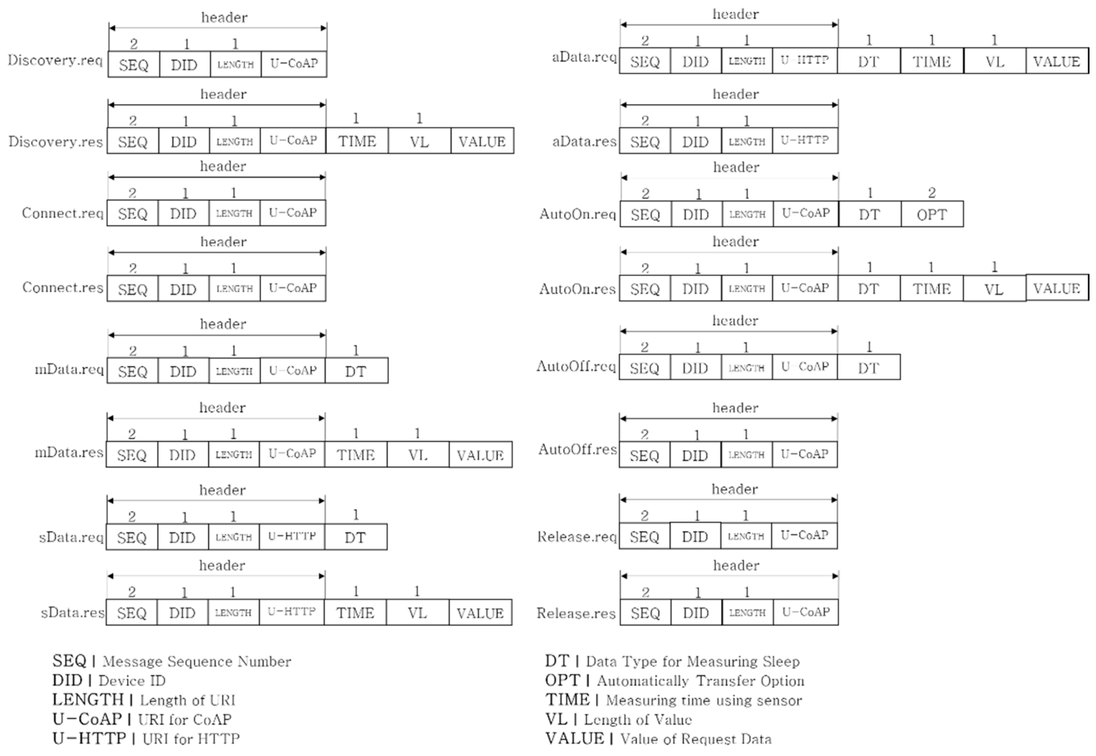

Discovery.req: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP
Discovery.res: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP | TIME 1 | VL 1 | VALUE
Connect.req: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP
Connect.res: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP
mData.req: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP | DT 1
mData.res: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP | TIME 1 | VL 1 | VALUE
sData.req: header | SEQ 2 | DID 1 | LENGTH 1 | U-HTTP | DT 1
sData.res: header | SEQ 2 | DID 1 | LENGTH 1 | U-HTTP | TIME 1 | VL 1 | VALUE
aData.req: header | SEQ 2 | DID 1 | LENGTH 1 | U-HTTP | DT 1 | TIME 1 | VL 1 | VALUE
aData.res: header | SEQ 2 | DID 1 | LENGTH 1 | U-HTTP
AutoOn.req: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP | DT 1 | OPT 2
AutoOn.res: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP | DT 1 | TIME 1 | VL 1 | VALUE
AutoOff.req: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP | DT 1
AutoOff.res: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP
Release.req: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP
Release.res: header | SEQ 2 | DID 1 | LENGTH 1 | U-CoAP
SEQ | Message Sequence Number
DID | Device ID
LENGTH | Length of URI
U-CoAP | URI for CoAP
U-HTTP | URI for HTTP
DT | Data Type for Measuring Sleep
OPT | Automatically Transfer Option
TIME | Measuring time using sensor
VL | Length of Value
VALUE | Value of Request Data

Figure 8. Transfer data format.

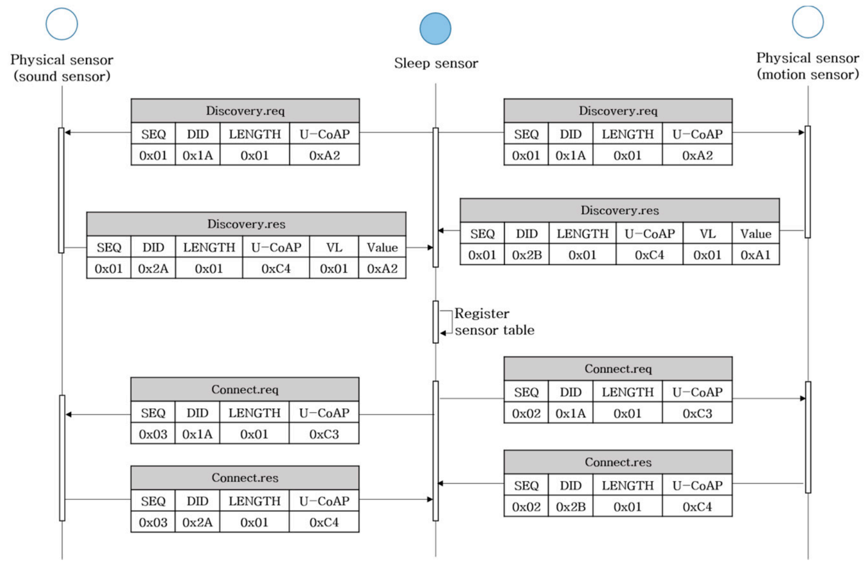
3.3. Interface Design

Table 1 defines the method, URI, and internal protocol used by applying the REST API for the respective message data formats of the Sleep Information Gathering Protocol. The sleep sensor uses the GET method when requesting data and processes the result with the response message. In addition, the data transmission message of the sleep sensor uses the POST method, and the result of the message transmission is confirmed through the response message. The message of the Sleep Information Gathering Protocol uses different internal protocols according to the communication partner. CoAP is used to transmit messages between physical sensors in a restricted network environment, and HTTP is used to transmit messages between existing IoT devices.

Table 1. Sleep Information Gathering Protocol (SIGP) message data format for using REST API.

Method	SIGP Data Format	URI	Internal Protocol
GET	Discovery.req/Discovery.res	/sensor/discovery	CoAP
	Connect.req/Connect.res	/sensor/connection	CoAP
	mData.req/mData.res	/sensor/measurement	CoAP
	sData.req/sData.res	/server/storage	HTTP
	AutoOn.req/Auto.res	/sensor/automaticon	CoAP
POST	AutoOff.req/AutoOff.res	/sensor/automaticoff	CoAP
	aData.req/aData.res	/sensor/analysis	HTTP

Figure 9 shows the sequence diagram of the process by which the sleep sensor searches for and connects with the neighboring physical sensors by applying the defined message data format of the Sleep Information Gathering Protocol. The Sleep Information Gathering Protocol operates based on CoAP for message transmission between physical sensors. The sleep sensor broadcasts the Discovery.req message to identify neighboring physical sensors. Discovery.req stores the message sequence number, the device ID of the sleep sensor, and the URI corresponding to the U-CoAP field.


Physical sensor (sound sensor)
Sleep sensor
Physical sensor (motion sensor)
Register sensor table

Discovery.req

SEQ	DID	LENGTH	U-CoAP
0x01	0x1A	0x01	0xA2

Discovery.req

SEQ	DID	LENGTH	U-CoAP
0x01	0x1A	0x01	0xA2

Discovery.res

SEQ	DID	LENGTH	U-CoAP	VL	Value
0x01	0x2A	0x01	0xC4	0x01	0xA2

Discovery.res

SEQ	DID	LENGTH	U-CoAP	VL	Value
0x01	0x2B	0x01	0xC4	0x01	0xA1

Connect.req

SEQ	DID	LENGTH	U-CoAP
0x02	0x1A	0x01	0xC3

Connect.req

SEQ	DID	LENGTH	U-CoAP
0x03	0x1A	0x01	0xC3

Connect.res

SEQ	DID	LENGTH	U-CoAP
0x02	0x2B	0x01	0xC4

Connect.res

SEQ	DID	LENGTH	U-CoAP
0x03	0x2A	0x01	0xC4

Figure 9. Sequence diagram of discovery & connection phase using SIGP.

The physical sensor that received this message responds by generating a Discovery.res message. Discovery.res contains the same sequence number as the request, its device id, and the URI corresponding to the Discovery.res message. It also contains the address and type of the physical sensor in the value field, and the sleep sensor that received the Discovery.res message generates the sensor table with the contents stored in the value. Based on the generated sensor table, the sleep sensor transmits a Connect.req message to the address of the physical sensor, and the physical sensor responds with Connect.res.

Figure 10 shows Measurement & Analysis Phase in which the sleep sensor requests data from the physical sensor and sleep care server, then receives results. The sleep sensor transmits an AutoOn.req message to the physical sensor for periodic data measurement. The AutoOn.req message has a DT field for identifying the data type for measurement and an OPT field for setting options for periodic data measurement. The physical sensor that received this message responds with an AutoOn.res message and periodically responds by including measurement data in the value. The sleep sensor uses the AutoOff.req message to indicate it does not need more data, and transmits the data type for which it wants to stop the measurements. The sleep sensor receives an AutoOff.res message in response to the request. It uses mData.req message when it needs only one measurement value, not periodic measurement. This message also identifies the data type for measurement through the DT field, and the physical sensor that received a request responds by storing the result in the value field of the mData.res message.

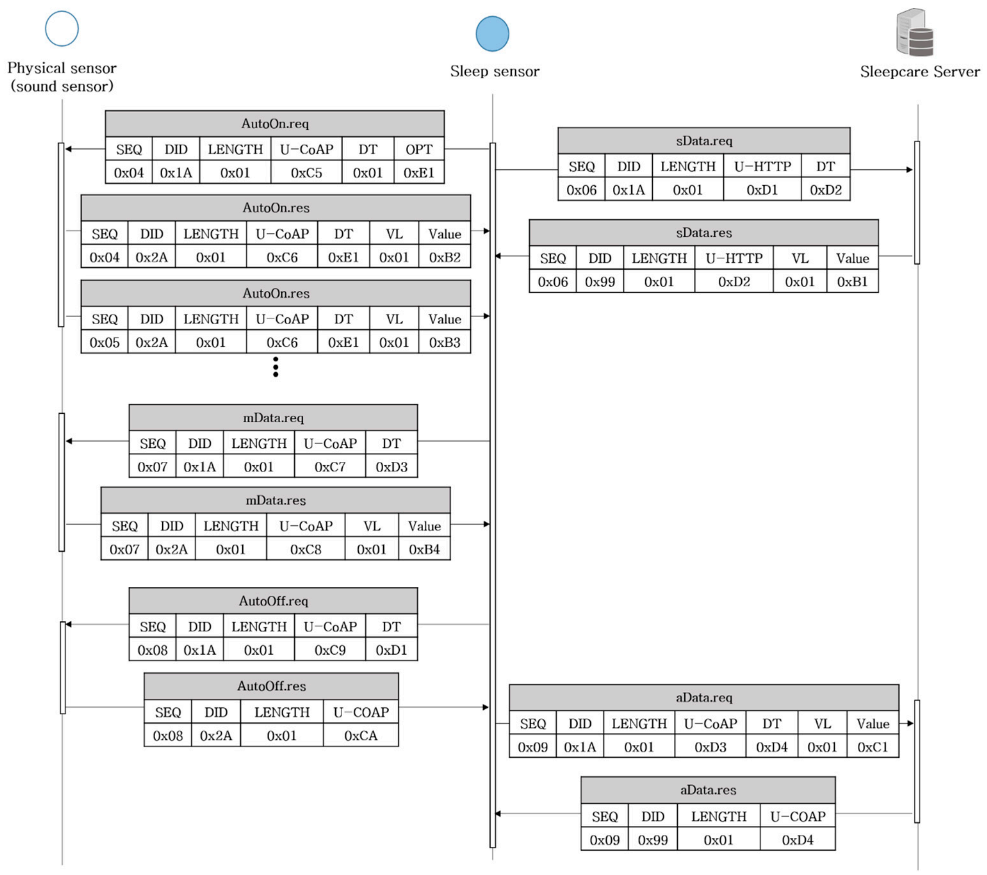

Physical sensor (sound sensor)
Sleep sensor
Sleepcare Server
AutoOn.req
SEQ DID LENGTH U-CoAP DT OPT
0x04 0x1A 0x01 0xC5 0x01 0xE1
AutoOn.res
SEQ DID LENGTH U-CoAP DT VL Value
0x04 0x2A 0x01 0xC6 0xE1 0x01 0xB2
AutoOn.res
SEQ DID LENGTH U-CoAP DT VL Value
0x05 0x2A 0x01 0xC6 0xE1 0x01 0xB3
mData.req
SEQ DID LENGTH U-CoAP DT
0x07 0x1A 0x01 0xC7 0xD3
mData.res
SEQ DID LENGTH U-CoAP VL Value
0x07 0x2A 0x01 0xC8 0x01 0xB4
AutoOff.req
SEQ DID LENGTH U-CoAP DT
0x08 0x1A 0x01 0xC9 0xD1
AutoOff.res
SEQ DID LENGTH U-COAP
0x08 0x2A 0x01 0xCA
sData.req
SEQ DID LENGTH U-HTTP DT
0x06 0x1A 0x01 0xD1 0xD2
sData.res
SEQ DID LENGTH U-HTTP VL Value
0x06 0x99 0x01 0xD2 0x01 0xB1
aData.req
SEQ DID LENGTH U-CoAP DT VL Value
0x09 0x1A 0x01 0xD3 0xD4 0x01 0xC1
aData.res
SEQ DID LENGTH U-COAP
0x09 0x99 0x01 0xD4

Figure 10. Sequence diagram of measurement & analysis phase using SIGP.

When the sleep sensor requests information stored from the sleep care server, the requested data are stored in the DT field using the sData.req message. The result values for the request are stored in the value field of the sData.res response message. When the sleep sensor, on the other hand, requests data storage from the sleep care server, it uses aData.req message and transmits the message by including the data value to be stored in the value field. The sleep care server that received the request responds with aData.res message, which confirms the data is stored. For data transmission between the sleep sensor and the sleep care server, HTTP is used instead of CoAP. Since the server used in the existing IoT environment is not a restricted environment, it transmits data based on HTTP. Therefore, it transmits the message by storing the URI value of HTTP in the U-HTTP field between the sleep sensor and sleep care server.

Figure 11 shows the Release Phase in which the sleep sensor requests a release from the physical sensor. The sleep sensor transmits a Release.req message to each physical sensor to be released.

The Release.req message has a message sequence number, the device ID of the sleep sensor, and a URI value corresponding to Release.req. The physical sensor that received this message responds to the sleep sensor by including the same message sequence number, its device ID, and the URI value corresponding to release.res. The sleep sensor that received the response message releases the sensor table registered based on the device ID and completes the disconnection between the physical sensors.

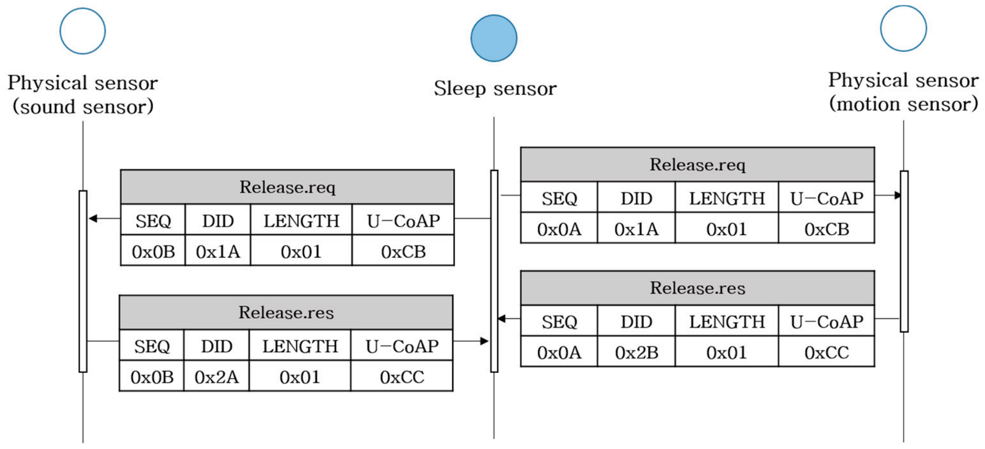


Figure 11. Sequence diagram of release phase using SIGP.

4. Experiment Protocols

4.1. Testbed and Its Scenarios

As shown in Figure 12, in order to verify the effectiveness of the Smart Model for Sleep Care service, we made a testbed by applying the Sleep Information Gathering Protocol to various IoT devices. The IoT device used in the testbed includes hardware (A) and (E) to which physical sensors are connected for measuring information for sleep care and hardware (D) which operates as a sleep sensor that receives the information from the physical sensor through CoAP. For each piece of hardware, we used a Raspberry Pi 3 supporting Wi-Fi, and utilized a mobile application (H) to check the communication results between devices. A Samsung Galaxy S5 was used as the smartphone to operate mobile applications. The Raspberry Pi 3 used in the experiment is very powerful hardware to apply to the sleep monitoring service. However, more information should be measured, collected, and transmitted in order for the extension of the Smart Model for Sleep Care service to provide health care services. Therefore, the authors have experimented using Raspberry Pi 3 considering the scalability of the service, and experimented by applying the Sleep Information Gathering Protocol.

For Raspberry Pi hardware (A), (D), and (E), we implemented the program using JavaScript on the Node.js framework based on Raspbian OS (A). We used Raspbian OS (version 4.4.49) and Node.js framework (version 4.4). It has a human body sensor (HC-SR501) (B) and a sound sensor (SZH-EK033) (C) for measuring a sleep state. (B) detects motion during sleep and (C) detects sound during sleep, and they transmit the result to (A). (A) analyzes tossing and turning, snoring, and apnea based on the measurement values that it received. (E) also has a connected temperature-humidity sensor (DHT22) (F) and an illuminance sensor (GL55) (G), and each sensor is used to measure information about the sleep environment. (A) and (E) are equipped with these physical sensors, so a CoAP-based network module is implemented as a program for data transmission between devices. Unlike (A) and (E), the sleep sensor (D) is not equipped with sensors, but it receives measurement values from both devices and analyzes the sleep state based on this information. We also implemented CoAP-HTTP Proxy to enable data request through a mobile application (H).

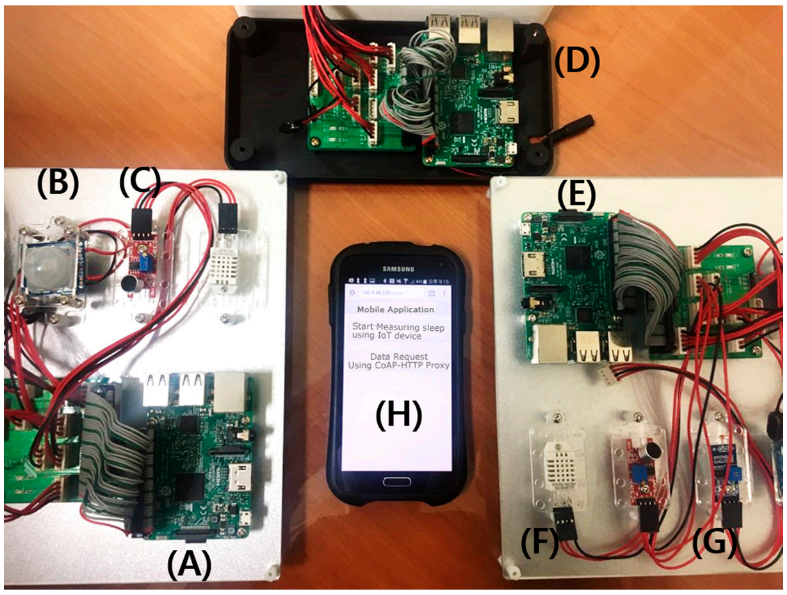

(D)
(B)
(C)
(E)
Mobile Application
Start Measuring sleep using IoT device
Data Request Using CoAP-HTTP Proxy
(H)
(A)
(F)
(G)

Figure 12. Testbed for verifying smat model for sleep care.

In this experimental environment, (D) searches neighboring (A) and (E) and establishes a connection between devices. At this moment, (D) can request the data from (A) and (E) for analyzing and processing sleep information, and measures the information about sleep state and sleep environment through the sensors installed. (H) can request data measurement from (A) and (E) through (D).

We conducted 3 experiments. In the first experiment, we compared the response time for HTTP Request/Response between (H) and (D) and the response time for CoAP Request /Response between (D) and (A). In the second experiment, we compared the response time to the HTTP request of (H) to (A) as in the existing sleep care service, and the response time to the HTTP request of (H) to (A) through CoAP Proxy of (D). In the third experiment, we measured and compared the throughput and consumed power for each request by comparing when the HTTP request was transmitted from the sleep sensor (D) to (A) equipped with the physical sensor and when SIGP was used.

4.2. Result and Evaluation

In this experiment, we implemented a system by applying the Sleep Information Gathering Protocol to each hardware operating in the Smart Model for Sleep Care. Figure 13 shows the response time of (D) to the request for data measurement from (H), and the time that took for (D) to receive the results after requesting (A) and (E) for the measurement of the corresponding data. (D) converts it into CoAP, and requests the data from (A) and (E), respectively. In order to analyze sleep information in (D), we conducted experiments using two approaches: Request for sleep state measurement (Request/Response Time (D) to (A)) and Request for sleep environment measurement (Request/Response Time (D) to (E)). Respective response times were directly measured through the code and confirmed.

The response time to the HTTP request from (H) to (D) ranges from about 20 to 40 ms. (D) that receives the HTTP request, converts the request through CoAP-HTTP Mapping, and then transmits it to (A) and (E). Response Time (D) to (A) indicates the response time that it took to receive the sleep state measurement values of (B) and (C), and Request/Response Time (D) to (E) receives the sleep environment measurement values of (F) and (G). Based on the results, it was confirmed that the two CoAP communications had a response time of about 50–55 ms and took about twice as long as HTTP. However, the HTTP had a response time of about 0.03 s, and CoAP had a response time of about 0.05 s, indicating that the difference in response time does not reduce effectiveness.

In the second experiment Figure 14, we compared (H) to (A), which is a method in which the mobile directly transmits a HTTP request for sleep measurement, and (H)–(D)–(A), which is a request method applied to the proposed Smart Model, and then compared the response time of the results of measurements. In this experiment, the response times were also measured through the code. First, when looking at the Request/Response Time (H) to (A), in which an HTTP request was transmitted, it has a response time of about 20 to 40 ms, which is similar to the HTTP communication in the first experiment. On the other hand, when looking at Request/Response Time (H)–(D)–(A), it has the response time from minimum 110 to maximum 140 ms. It can be judged that since (D) underwent CoAP-HTTP Mapping, request and response are executed twice respectively, which consumes time. However, since this request also has a response time difference of about 0.1 s, there is no problem in providing an IoT service in which CoAP and HTTP are combined. It also allows for more efficient resource management for (A) operating in a restricted environment by transmitting a request to (A) via (D) rather than by transmitting a direct request to (H) to (A).

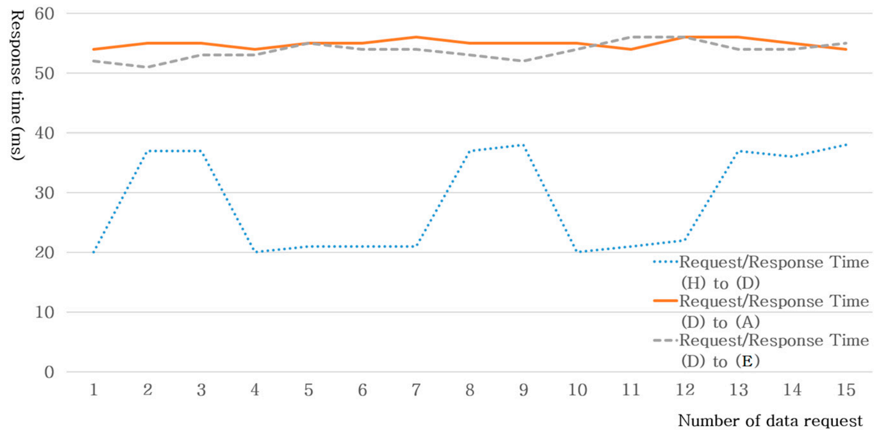


Figure 13. Comparing request/response time when using HTTP and CoAP.

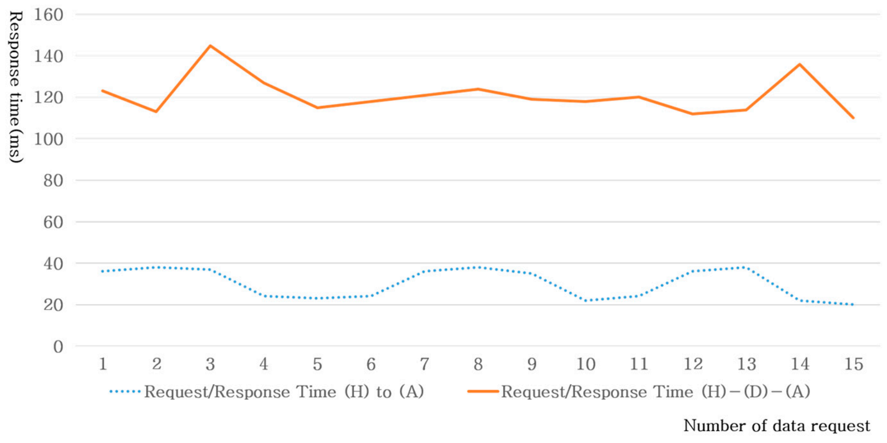


Figure 14. Comparing request/response time in (H) to (A) and (H)–(D)–(A).

In the last experiment, we compared the throughput and consumed power when (D) requests the data from (A) by applying HTTP and SIGP. Figure 15 shows that the average throughput for 1 min is about 110 responses when requesting data using HTTP. On the other hand, when SIGP was used, the throughput for 1 min was about 242 responses, which is over twice as efficient as the HTTP. In addition, there was a difference in the amount of power consumed by processing data requests for 20 min. We attached a battery of 1000 mA to (A) and conducted tests for 20 min using the same protocols. When HTTP was used, 800 mA was consumed, whereas when SIGP was used, 650 mA was consumed. The results confirmed that SIGP consumes fewer resources than HTTP while processing the same amount of data.

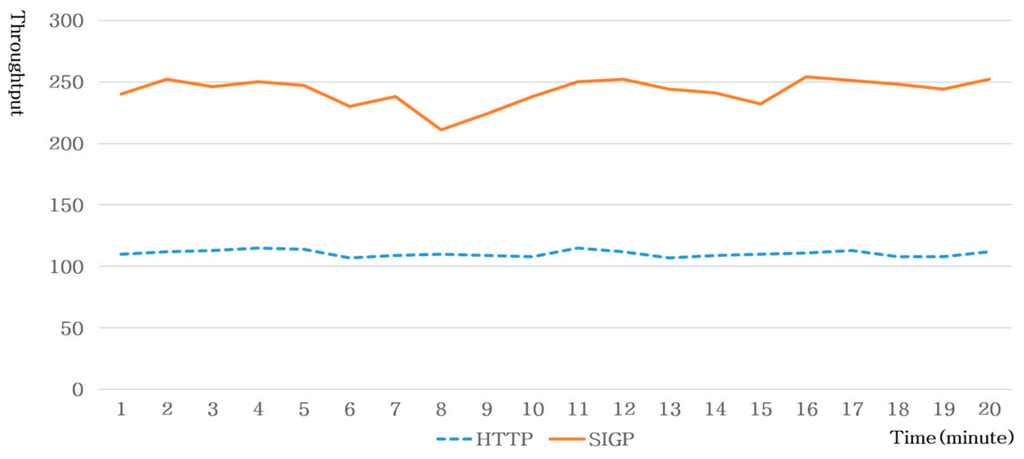


Figure 15. Comparing throughput when using HTTP and SIGP.

5. Conclusions

In this paper, we proposed a Smart Model for Sleep Care for efficient sleep management by applying IoT for efficient sleep management, and then proposed a Sleep Information Gathering Protocol that will be used in this model. The Smart Model for Sleep Care allows various devices to share information through a network through the IoT to existing sleep care services, which in turn enables them to analyze the sleep information more accurately. We also defined the Sleep Information Gathering Protocol using CoAP, which is a protocol suitable for IoT to receive information related to a user's health and sleep state from the mobile devices and the sleep care server.

To verify the effectiveness of the service model using such a protocol, we made a testbed to compare the response time to the sleep information request. As a result, it was confirmed that the defined protocol did not show a significant difference in response time compared with the existing sleep care services, indicating its applicability to Smart Model for Sleep Care.

To analyze the user's sleep and diagnose sleep disorders, more extended sleep care services should be provided. In particular, there is a need for a method to protect the sleep information of an individual as well as a method for using the security protocol provided by CoAP or HTTP. In addition, the proposed protocol should be supplemented to provide extended services, and the effectiveness of the proposed protocol in integrating the therapy service to treat sleep disorders rather than simply measuring a sleep state should be verified in future studies.

Acknowledgments: This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2016S1A5B6913737).

Author Contributions: Joo Ho Choi and Byung Mun Lee conceived and designed the experiments; Joo Ho Choi performed the experiments; Joo Ho Choi and Byung Mun Lee analyzed the data; Un Gu Kang contributed reagents/materials/analysis tools; Joo Ho Choi wrote the paper. All authors have read and approved the final manuscript.

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

References

1. Datta, S.K.; Bonnet, C.; Gyrard, A.; Ferreira da Costa, R.P.; Boudaoud, K. Applying Internet of Things for personalized healthcare in smart homes. In Proceedings of the Wireless and Optical Communication Conference, Taipei, Taiwan, 23–24 October 2015.
2. Pantelopoulos, A.; Bourbakis, N.G. A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis. *IEEE Trans. Syst. Man Cybern.* **2009**, *40*, 1–12. [CrossRef]

3. Choi, J.H.; Kang, U.K.; Lee, B.M. IoT Service Model for Measuring Sleep Disorder using CoAP. *Int. J. Mob. Device Eng.* **2017**, *1*, 9–14. [CrossRef]
4. Dhawan, A.P.; Heetderks, W.J.; Pavel, M.; Acharya, S.; Akay, M.; Mairal, A.; Wheeler, B.; Dacso, C.C.; Sunder, T.; Lovell, N.; et al. Current and Future Challenges in Point-of-Care Technologies: A Paradigm-Shift in Affordable Global Healthcare With Personalized and Preventive Medicine. *IEEE J. Transl. Eng. Health Med.* **2015**, *3*, 2800110. [CrossRef] [PubMed]
5. Perez, A.J.; Leff, D.R.; Ip, H.M.D.; Yang, G.Z. From Wearable Sensors to Smart Implants—Toward Pervasive and Personalized Healthcare. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 2750–2762. [CrossRef] [PubMed]
6. Chang, D.W.; Liu, Y.D.; Young, C.P.; Chen, J.J.; Chen, Y.H.; Chen, C.Y.; Hsu, Y.C.; Shaw, F.Z.; Liang, S.F. Design and Implementation of a Modular-ized Polysomnography System. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 1933–1944. [CrossRef]
7. Chen, Z.; Lin, M.; Chen, F.; Lane, N.D.; Cardone, G.; Wang, R.; Li, T.; Chen, Y.; Choudhury, T.; Campbell, A.T. Unobtrusive Sleep Monitoring using Smartphones. In Proceedings of the 2013 7th International Conference on Pervasive Computing Technologies for Healthcare, Venice, Italy, 5–8 May 2013.
8. Shin, H.; Cho, J. Unconstrained snoring detection using a smartphone during ordinary sleep. *BioMed. Eng. OnLine* **2014**, *13*, 116. [CrossRef] [PubMed]
9. Fuqaha, A.A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
10. Shelby, Z.; Hartke, K.; Bormann, C. *The Constrained Application Protocol (CoAP)*; Internet Engineering Task Force: California, CA, USA, 2014.
11. Celesti, A.; Fazio, M.; Giacobbe, M.; Puliafito, A.; Villari, M. Characterizing Cloud Federation in IoT. In Proceedings of the 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Crans-Montana, Switzerland, 23–25 March 2016.
12. Agarwal, R.; Fernandez, D.G.; Elsaleh, T.; Gyrard, A.; Lanza, A.; Sanchez, L.; Georagntas, N.; Issarny, V. Unified IoT ontology to enable interoperability and federation of testbeds. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–13 December 2016.
13. Sense. Available online: http://www.hello.is/ (accessed on 30 April 2017).
14. Lin, F.; Zhuang, Y.; Song, C.; Wang, A.; Li, Y.; Gu, C.; Li, C.; Xu, W. SleepSense: A Noncontact and Cost-Effective Sleep Monitoring System. *IEEE Trans. Biomed. Circuits Syst.* **2016**, *11*, 189–202. [CrossRef] [PubMed]
15. Fitbit. Available online: http://www.fitbit.com/ (accessed on 30 April 2017).
16. Jowbone. Available online: http://www.jawbone.com/ (accessed on 30 April 2017).
17. Zhang, J.; Zhang, Q.; Wang, Y.; Qiu, C. A real-time auto-adjustable smart pillow system for sleep apnea detection and treatment. In Proceedings of the 2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Philadelphia, PA, USA, 8–11 April 2013.
18. Behar, J.; Roebuck, A.; Shahid, M.; Daly, J.; Hallack, A.; Palmius, N.; Stradling, J.; Clifford, G.D. SleepAp: An Automated Obstructive Sleep Apnoea Screening Application for Smartphones. *IEEE J. Biomed. Health Inf.* **2014**, *19*, 325–331. [CrossRef] [PubMed]
19. Cao, L.; Zheng, G.; Shen, Y. Research on design of military ammunition container monitoring system based on IoT. In Proceedings of the Prognostics and System Health Management Conference, Chengdu, China, 19–21 October 2016.
20. Celesti, A.; Mulfari, D.; Fazio, M.; Villari, M.; Puliafito, A. Exploring Container Virtualization in IoT Clouds. In Proceedings of the 2016 IEEE International Conference on Smart Computing (SMARTCOMP), St. Louis, MO, USA, 18–20 May 2016.
21. Morabito, R. Virtualization on Internet of Things Edge Devices with Container Technologies: A Performance Evaluation. *IEEE Access* **2017**, *5*, 8835–8850. [CrossRef]
22. Krejcar, O.; Jirka, J.; Janckulik, D. Use of Mobile Phones as Intelligent Sensors for Sound Input Analysis and Sleep State Detection. *Sensors* **2011**, *11*, 6037–6055. [CrossRef] [PubMed]
23. Sulaeman, A.B.; Ekadiyanto, F.A.; Sari, R.F. Performance evaluation of HTTP-CoAP proxy for wireless sensor and actuator networks. In Proceedings of the 2016 IEEE Asia Pacific Conference on Wireless and Mobile (AP-WiMob), Bandung, Indonesia, 13–15 September 2016.

24. Ludovici, A.; Calveras, A. A Proxy Design to Leverage the Interconnection of CoAP Wireless Sensor Networks with Web Applications. *Sensors* **2015**, *15*, 1217–1244. [CrossRef] [PubMed]
25. Sommer, N.L.; Touseau, L.; Mahéo, Y.; Auzias, M.; Raimbault, F. A Disrup-tion-Tolerant RESTful Support for the Web of Things. In Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), Vienna, Austria, 22–24 August 2016.
26. Teklemariam, G.K.; Hoebeke, J.; Abeele, F.V.D.; Moerman, I.; Demeester, P. Simple RESTful sensor application development model using CoAP. In Proceedings of the 2014 IEEE 39th Conference on Local Computer Networks Workshops (LCN Workshops), Edmonton, AB, Canada, 8–11 September 2014.
27. Lee, K. IEEE 1451: A standard in support of smart transducer networking. In Proceedings of the Instrumentation and Measurement Technology Conference, Baltimore, MD, USA, 1–4 May 2000.