

Improvement of Wireless Connectivity and Efficiency in E-Healthcare Service System Using a Proxy in Body Area Device

Sang Kon Kim¹, Tae Kon Kim^{2*} and Jinhwan Koh³

¹ Dept. of Electronics and Information Engineering, Korea University
2511 Sejong-ro, Sejong City, 30019 - KOREA
[e-mail: paulka@korea.ac.kr]

² Dept. of Electronics and Information Engineering, Korea University
[e-mail: taekonkim@korea.ac.kr]

*Corresponding author: Tae Kon Kim

³ Dept. of Electronic Engineering, Gyeongsang National University
501 Jinju-daero, Jinju, South Gyeongsang Province - KOREA
[e-mail: jikoh@gnu.ac.kr]

*Corresponding author: Tae Kon Kim

*Received September 8, 2019; revised December 8, 2019; accepted December 19, 2019;
published March 31, 2020*

Abstract

E-health services have provided interoperability between personal health devices in personal area network, based the ISO/IEEE 11073 standard. In the healthcare system, the manager handles most agents concurrently through wireless communication. However, due to the distance limitation and the increased number of agents, it may be difficult to provide continuous connectivity. Recently, body area devices have been equipped with various applicable agents, which can even handle agents on behalf of the manager. A BAD may act as an intermediary device to increase system efficiency and performance. In this study, a device called “proxy”, which can be installed as software on BAD devices, is proposed. The data measured by an agent can be sent to the proxy first, and subsequently be sent to the manager again. Agents and the manager are not aware of the proxy existence and work normally without the proxy. Furthermore, a new smart proxy and modified manager are proposed. The smart proxy acts as one agent handling measurement data from several agents, which can transmit a significant amount of data at once. The proxy and smart proxy maintain compatibility with existing devices that conform to the 20601 standard. The proposed schemes are verified and the complexities of devices are analyzed. The analysis shows no significant difference among the proxy, smart proxy, and manager. Simulations exhibit that the proposed schemes can improve the system performance.

Keywords: e-healthcare system, ISO/IEEE 11073, 20601 OEP, body area device, wireless connectivity

This work was supported by the Korea University Grant.

1. Introduction

Because of the increase in the average life span of human beings in many countries, the number of seniors has drastically increased, and the government expenditure on national health has expanded correspondingly. Moreover, individuals have great interest not only in disease management but also in their daily health care, and they have started investing money and time into their health care in the pursuit of a “healthy life”. Therefore, e-health/telehealth care has developed into a cost-effective, clinically acceptable, and convenient health management system and service. Devices that measure intermittent health information such as weight scales, thermometers, and blood pressure monitors have been developed, along with devices that measure data more precisely and periodically, such as electrocardiogram (ECG) and electroencephalography (EEG). Recently, they have evolved into various types of products such as contactless, wearable, attachable, and implantable devices. In fact, people have started to focus on intelligent next-generation healthcare systems that provide a considerably improved quality of experience (QoE), quality of service (QoS), and emergency handling including real-time monitoring systems. In this service scenario, individuals do not need to be involved in the process of measurement, transmission, and feedback, and will receive appropriate healthcare service in any time and place [1].

These advanced healthcare and monitoring services are made possible by the following explosive technological developments. First, the development of a variety of measuring devices including biosensors can measure a variety of personal health information through a contact or non-contact method. In the near future, measuring devices can be planted in the human body, and body information can be measured remotely without the user's perception. A tremendous amount of the user's body information will be collected and processed in real time and sent to the healthcare service provider (HSP) including doctors in hospitals for the best feedback. Second, based on communication technology and the cloud service environment, the world is rapidly connected to advanced wired and wireless networks. Individuals are always connected to the network with sufficient bandwidth and computing power, irrespective of their location. Hence, a reliable communication environment provides seamless connectivity, sufficient transmission capacity, and minimum latency according to the type of health data and service. Third, the development of big data and artificial intelligence is changing the fundamental concepts of traditional medical and healthcare services. Thus, real-time monitoring and care service is possible without personal recognition, and artificial intelligence-based management systems become possible. Furthermore, the large amounts of information collected and accumulated will enable new levels of health care that have not been expected to date [2].

The ISO/IEEE11073 personal health device (PHD) working group (WG) initiated standards entitled “health informatics-personal health device communication” for e-health services, by providing interoperability for medical, healthcare, and wellness devices [3]. A personal health device (also called an agent) acquires personal health-related information, and a compute engine (also called a manager) collects and processes the information. An optimized exchange protocol (OEP 11073-20601) defines communication model between an agent and manager [4-5], and device specifications (11073-104zz) specify many devices including thermometer (-10408) [6], weighing scale (-10415) [7], glucose meter (-10417) [8] and basic ECG (-10406) [9]. In the standard, it is initially assumed that an agent and its corresponding manager work in pairs. For example, a user operates the blood glucose meter

directly, transmits the measurement to the corresponding manager and, submits information to a doctor or health care manager to obtain medical and health care services after a period of time, as shown **Fig. 1** (a). Recently, however, various health care devices and the matching of expensive managers to each device have proved to be inefficient. Rather, the evolution of a manager that handles a large number of health devices (**Fig. 1** (b)) and manages health care services as a portal device in the home or office environment is rapidly progressing. Hence, a smart manager manages significant amounts of health information from a number of agents and provides healthcare services in conjunction with the healthcare service provider (HSP) system (**Fig. 1** (c)) [10-12].

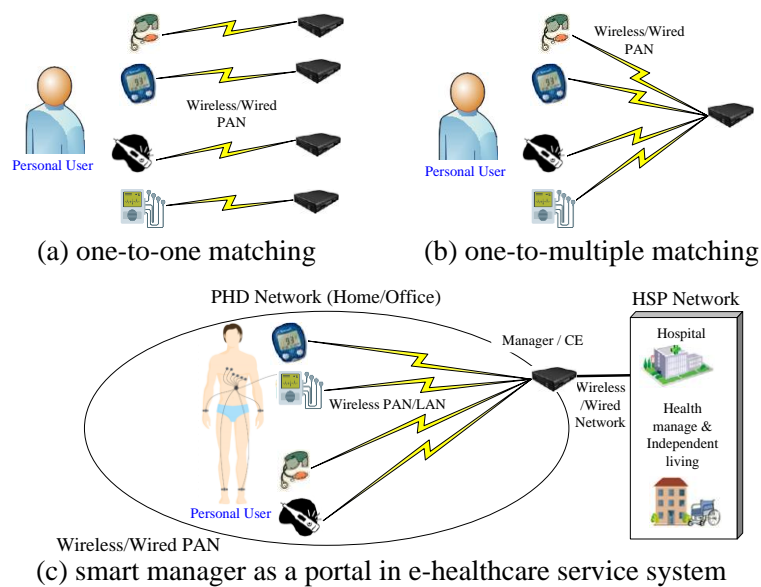


Fig. 1. Evolution of manager devices

As discussed, it is expected for the healthcare service provider to supply the satisfactory service and system with a large number of devices. However, there is not enough service products for e-health in the current consumer's market. The reason is that, for the sake of user convenience, personal health information should be obtained "at any time", "anywhere", and "in real-time", however realization of this service is presently difficult due to the following issues. First, only point-to-point (P2P) communication between an agent and a manager is possible in ISO/IEEE 11073 standards, and the manager should maintain connectivity with each device at all times. In fact, it is very difficult to maintain connectivity concurrently with agents unless a user confirms the connection in every transmission. Furthermore, both agents and the manager have to provide mobility suitable for the environment and maintain connectivity at the same time for real-time ubiquitous e-health. Hence, the manager and agents need to be connected wirelessly rather than with a wired connection. The manager is usually fixed or nomadic for electric power support and wired network connectivity in the residential area, and it is very difficult to manage the wireless connection with each agent without human intervention. Second, most agents mainly support wired or wireless personal area network (PAN) to communicate with the manager to reduce the power consumption and price. Portable agents also need a way to send measurement information efficiently due to the given battery capacity. In fact, the measured data cannot be immediately transferred, if the portable agent is

far from the manager in the service area. Hence, agents can obtain personal health information, however the measured data can be transmitted only when the connection with the manager is established. By improving the connectivity between agents and the manager in the system, it is expected that the personal healthcare service market will expand.

Recently, body area devices such as the smart phone, smart watch, notepad, and the like are globally used regardless of age and sex. The BAD generally has a wireless communication function, storage space, and computing ability, and its performance is rapidly improving. At the same time, its size is getting smaller, and it is expected to be improved to a level where it can be implanted into the human body in the near future, such that people can use several BADs in their daily lives. Furthermore, BADs are evolving to contain body information measuring equipment, and with the development of measuring sensors, their variety and accuracy can reach beyond imagination. In this case, a BAD will play a central role as an agent, and many agents can be managed based on the superior resources of the BAD on behalf of the manager. However, due to its limited battery power and instability of the wireless network connection, it is difficult to replace a manager. Rather, it may act as an intermediary device between agents and the manager to increase system efficiency and performance in the healthcare system. A device called "proxy" that can be installed as software on BAD devices can greatly improve network connectivity and dramatically improve system efficiency.

In this study, a novel proxy that improves connectivity and enables real-time management is proposed. A proxy installed in a BAD supports better resources and various communication methods. Existing agents and the manager are not aware of proxy's existence, and an agent and the manager are assumed to interoperate with each other directly in a 1: 1 connection. In reality, however, the proxy exists in the middle, acting as a manager for the agent, storing information received from the agent, and relaying the stored data as if it is the agent to the manager. The above operation in a proxy is denoted as Proxy in the middle (PiM) and the operation maintains compatibility with existing devices. The agents and manager work normally even in the absence of the proxy. In addition, the proxy can obtain the configuration and MDS information of any agent. This can provide improved and advanced health care services including real-time healthcare service.

With the presence of a proxy in the system, the data measured by an agent can be sent to the proxy first, and then be sent to the manager again. In terms of the overall system, there is a disadvantage in that the transmission can be doubled. To solve this problem, a new smart proxy and modified manager are proposed. In the proposed scheme, the smart proxy acts as one agent handling various heterogeneous measurement data from agents, and it can transmit a large amount of data at once to the modified manager. The improved protocol that works in smart proxy is called the smart proxy in the middle (SPiM) operation. The smart proxy and modified manager work together, while maintaining compatibility with existing devices that conform to the 20601 standard.

PiM and SPiM operations are verified using agents, a proxy, and a smart proxy implemented on the embedded systems, and using a manager and modified manager implemented on a PC. The proxy structure is a combination of agents and the manager, which is larger than that of the manager in terms of complexity. However, the analysis shows no significant difference. The smart proxy architecture consists of a proxy and an additional agent function. Its complexity increases in proportion to the number of registered agents. However, the added complexity is not significant and there is no problem in SPiM implementation. No big differences found in terms of complexity between the smart proxy and existing manager.

2. ISO/IEEE 11073 Personal Health Device Standards

In a product market, the interoperability between products of different manufacturers is not usually provided, as shown in **Fig. 1** (a). International organizations aim at addressing the interoperability of agents made by different vendors to create an environment such as the one portrayed in **Fig. 1** (b). Therefore, based on 11073 standards and certifications, devices can be interoperated.

The main domains of 11073 standards are closely related to disease management, health, and fitness applications. A technical report (11073-00103), device specifications (11073-104zz), and optimized exchange protocol (OEP) between an agent and the manager (11073-20601), shown in **Fig. 2**, are included in 11073 standards. Because 11073 standards are above the transport layer, no particular transport is not specified. However, medical profiles for USB (USB agentC) [13], Bluetooth health device profile (BT-HDP) [14], and ZigBee health care profile (ZHC) [15] were developed by special interest groups, and those are recommended by Continua Health Alliance.

2.1 Device Standards

11073-104zz standards define the specifications of existing agents over 11073-20601 OEP. Currently, around 15 agents are already standardized and revised, while others are in the process of standardization. For example, the blood pressure monitor (-10407) [16], thermometer (-10408), weighing scale (-10415), glucose meter (-10417), and independent living activity hub (-10471) [17] are the most frequently used agents, and the first standardized. Some of those are already revised.

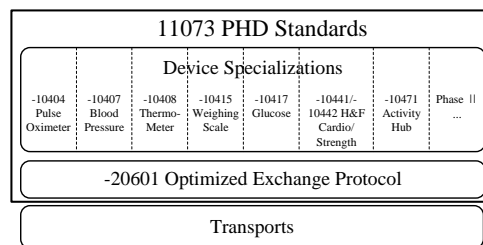


Fig. 2. Document map

2.2 11073-20601 Optimized Exchange Protocol

A common framework for constructing an abstract model of personal health data available in transport-independent transfer syntax is defined by the 11073-20601 OEP. The logical connections between an agent and the manager should be established to provide presentation capabilities and services for communication tasks [4-5]. 20601 OEP over the transport layer consists of the application layer services and the data exchange protocol. The application layer services provide the protocol for connection management, and reliable transfer of actions and data. The commands, agent configuration information, data format, and overall protocol are defined. Therefore, 20601 OEP provides the basis to support many different types of agents. In this section, 20601 OEP is briefly introduced, and then the communication procedure between agent and manager is described in detail, to capture the aim of our research work.

The system model of the 11073 agent consists of the three principal components: a domain information model (DIM), service model, and communication model.

1) Domain information model (DIM).

Because the agents specified in 11073 agent standards are defined using an object-oriented model, the information of an agent is characterized by the DIM as a set of objects having one or more attributes. The measurement data communicated to a manager, as well as elements that control behavior and report on the status of the agent are described by the attributes. The medical device system (MDS) object is at the top-most in the DIM, and there are zero or more scanner, numeric, real-time sample array (RT-SA), enumeration, or persistent metric (PM) objects.

2) Service model.

The service model provides data access primitives that are sent between the agent and manager, to exchange data specified by the DIM. Get, Set, Action, and Event Report commands are included in these primitives. In this study, Get, Action, and Event Report commands are included in the association procedure, configuration procedure, get MDS object attributes, and manager-initiated measurement data transmission.

3) Communication model.

The topology of one or more agents communicating over point-to-point (P2P) connections to a single manager is supported in 20601 OEP. For each P2P connection that is independently operated, the connection state machine specifically defines the dynamic system behavior. In addition, the states and substates of the agent and manager pair pass through, including states related to the connection, association, and operation that are defined in the connection machine.

Based on 20601 OEP, both an agent and the manager should be operated within the connection state machine, and measurement data can only be transmitted when both are in a connected/associated/operating state using the proper DIM and service model.

2.3 Measurement Data Transmission between Agent and Manager

For a single agent to measure the health information of a person and transmit the data to the manager, the following association procedure should be performed, and both the agent and the manager have to be in the associated operating state, after which measurement data can be transmitted. To better understand this study, the association and configuration procedure and measurement data transmission are described in detail in this section.

1) Association and configuration procedure

The 20601 OEP only supports 1: 1 P2P communication between the agent and the manager. After the connection is established in the transport layer, it interoperates in the associated operating state through an association procedure. Two sequence diagrams for the case where the agent is unknown and known for the association procedure are shown in [Fig. 3](#). The association procedure must be initiated only by the agent, as the agent sends an association request (AssocRequest) to the manager. The receiving manager sends a response with the accepted-unknown-config parameter (AssocResponse) if the system ID and configuration ID included in the AssocRequest are not registered in the manager, but accepted ([Fig. 3](#) (a)). The configuration procedure is initiated by the agent, and the agent sends its configuration information. After the manager checks the agent's configuration information, it sends a response with the accepted-config parameter if it can support it, and sends a response with the unsupported-config parameter if it cannot. If the agent receives a response with an unsupported-config attribute, it can send another configuration information that the agent can provide. If the agent no longer has any other configuration information that it can provide, they cannot interoperate based on the 11073 standard. On the contrary, the agent receiving the

response with the accepted-config parameter and the manager in an associated operating state can interoperate with each other. The manager identifies information such as the handle number, attribute count, ID, and value for object handles provided by the agent through a configuration procedure. If the agent is registered with the manager through the mentioned association and configuration process, then the agent is known to the manager, and the association procedure is simplified as shown in Fig. 3 (b).

After the association procedure, when both the agent and the manager are in the associated operating state, the manager can request the agent for various operations supported by the agent. First, by using the GET service, the medical device system (MDS) object corresponding to handle 0 of all agents can be obtained and managed. The MDS object contains information including type of agent, manufacturer and model number, system-id, config-id, product specification, etc. Afterward, collecting and managing the measurement data acquired by the agent becomes the main function of the manager. Because the resources of the agent are very limited, intermittent measured data can be stored temporarily in volatile memory, however if a lot of data is generated periodically, these can be stored permanently in nonvolatile memory. Temporarily stored and permanently stored measurements are defined differently by 20601. This study assumes that intermittently measured data is temporarily stored in volatile memory, however it can be extended to permanently stored cases.

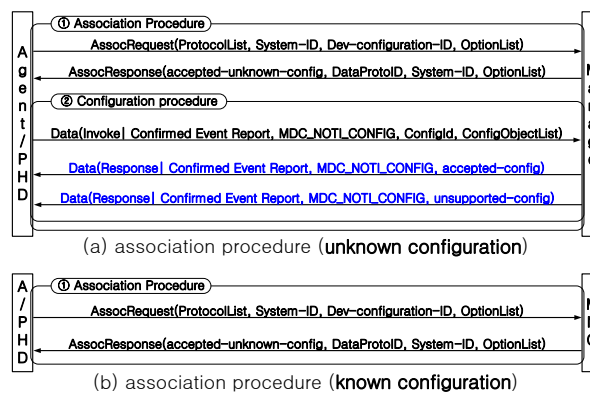


Fig. 3. Association procedure

2) Measurement data transmission from an agent to a manager

When both the agent and the manager are in the associated operating state, it is possible to transmit the measurement data, which can be started by the agent or manager. Agent-initiated transfers are typically used for agents that transfer small amounts of temporarily stored data. In contrast, agents with large data or streaming should use manager-initiated transfers, which provide a mechanism to control the data flow. In the case, it must be permanently stored and transmitted in nonvolatile memory by the PM-store class. This study focuses on the data measured intermittently by multiple heterogeneous agents. Therefore, transmission for PM-store class is not included, however it is applicable.

In the case of the confirmed service mechanism, when the agent sends measurement data to the manager using the Event Report service and receives an accepted response, the agent can delete the data. A sequence diagram of the four methods for measurement data transmission from an agent to a manager is shown in Fig. 4. As shown in Fig. 4 (a), the agent can transfer the measured data using a confirmed or unconfirmed event report in the case of agent-initiated transfer. In the case of an unconfirmed event report, there is a possibility of loss of transmitted

data, hence the use of the confirmed event report is recommended. Generally, when an agent receives a response from the manager in confirmed transfer, the stored data is deleted and no measurement is available.

There are three modes for manager-initiated transfers: single response, time period, and no time limit. In the single response mode shown in Fig. 4 (b), when the agent receives a confirmed action request, it responds with a confirmed action response containing the measurement data. The agent cannot receive an accepted response, and it cannot decide when to delete the stored data. Hence, the agent cannot delete the data without user intervention, which is not suitable for the service environment of this study. Therefore, the timed command with a timeout value of 0 is recommended by the 20601 OEP as shown in Fig. 4 (c). In this mode, after the agent sends a confirmed action response as an acknowledgement with a time value of 0, the agent sends the measured data using the confirmed event report. When an agent receives a response, the stored data can be deleted. Fig. 4 (d) shows the manager-initiated no time limit mode. In this mode, measurement data can be sent using a confirmed event report and deleted after receiving a response. Afterwards, whenever the available measurement data occurs, the process is repeated. When an agent receives a request for the stop from the manager, it stops the transfer operation. Therefore, this mode can be used when the connection can be maintained for a long time, which is not an appropriate operation for the service environment of this study.

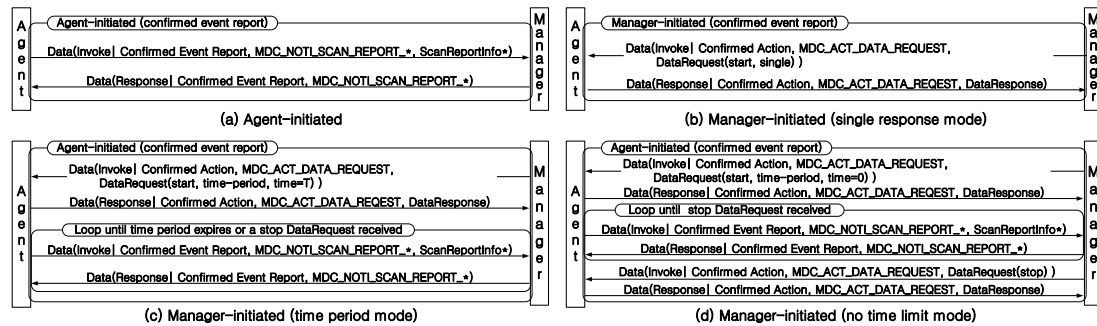


Fig. 4. Measurement data transmission

How the agent deletes the transmitted measurement data is vendor specific, however to reuse limited resources and avoid unnecessary duplication of transmission, most products delete the measured information immediately after transmission. In this study, it is clearly defined that the agent is requested to transmit the measurement data, sends the data using the confirmed event report, and shall delete the data when receiving the response. In this way, the measurement data is transmitted and then deleted to prevent duplicate transmission. This plays an important role in the scheme proposed in this study.

3. Proposed Scheme

3.1 Proxy in the Middle (PiM) Operation

In the healthcare service system, which consists of one manager and many agents, we propose the proxy with superior resources and various communication functions than the agent to improve network connectivity between the manager and an agent as shown in Fig. 5. In the absence of proxy, the existing agent and manager interoperate normally. Thus, the proposed

PiM operation performs between agents and the manager while maintaining compatibility with existing devices. In fact, the proxy can obtain the agent's configuration information and MDS information, and can use this information to provide a better service.

In fact, the proxy has a structure including both agents and the manager. It acts as a manager, collects measurement data from individual agents, and then sends the data to the manager as if it were individual agents. The physical communication network connection can utilize several technologies at the same time. The 20601 OEP is communication at the application layer, and it is not involved in the connection operation of the lower layer. Hence, only the one-to-one application layer connection is concerned and supported. Naturally, when an agent sends an association request to a manager, both the manager and the proxy can receive this request. However, since 20601 OEP only supports 1: 1 P2P communication, an agent communicates with the device that responded first, i.e., either a proxy or the manager. Agents cannot establish a connection with both the manager and the proxy at the same time. Therefore, the agent sends measurement data to either the manager or proxy and deletes the transmitted data. When measurement data is sent to the manager, it is the same as the existing system. When it is sent to the proxy, the proxy relays it to the manager as an agent.

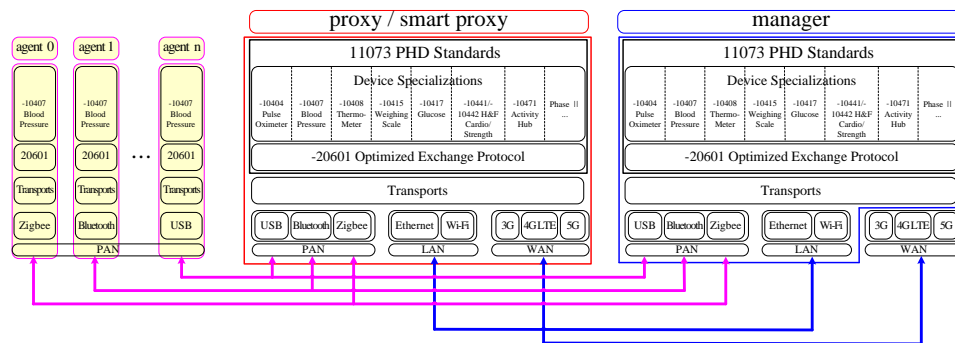


Fig. 5. Layered structures of agents, manager, proxy, and smart proxy with wired/wireless connections

Various physical communication connections can exist between the agent, proxy, and manager, and it will be clearly inspected that there is no problem when the PiM operation performs in an environment where all are in the same wired/wireless communication network. When one agent attempts to perform an association process, four cases, as shown in Fig. 6 can occur. In Fig. 6 (a), interoperation cannot be performed, because the agent cannot establish an association with neither the manager nor the proxy. In Fig. 6 (b), only the manager can establish an association with the agent, so interoperation between them can be conducted according to the existing method. In Fig. 6 (c), only the proxy can make an association with the agent, hence the agent's measurement data is sent to the proxy first, and the proxy sends it to the manager based on the PiM operation. In Fig. 6 (d), both the manager and proxy can establish association with the agent. For the association request sent by the agent, first an association is established with the corresponding device. The agent establishes an association with either the manager or the proxy, which is equivalent to (b) if the manager first sends an association response, otherwise it is equivalent to (c). The agent cannot establish two associations with both the manager and the proxy at the same time.

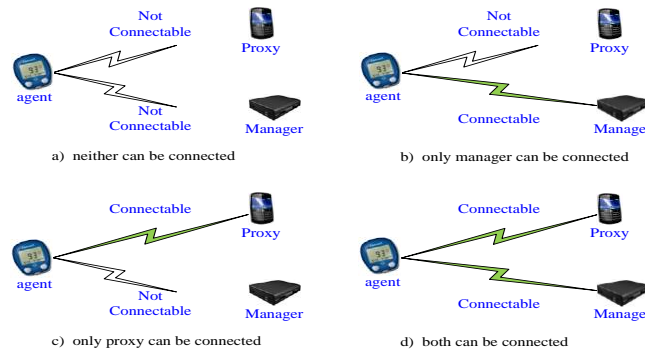


Fig. 6. The PiM operation in a communication network

As mentioned above, connection between an agent and the manager or proxy is generally made by wired/wireless PAN communication, however since the proxy and manager support various communication methods, an appropriate and efficient method can be selectively used according to service environment. For example, the proxy and manager can be connected by LAN, WAN, etc., as needed. In this case, various services are possible including a real-time monitoring service by taking advantage of the wide bandwidth and long communication distance. Moreover, if the communication method between the manager and the proxy is different from that of the agent, the connection may exist independently, because the use of the frequency is different. Hence, the proxy could be connected to the manager while maintaining the connection with the agent. Therefore, an agent’s measurement data can be immediately transmitted from an agent to a manager via a proxy.

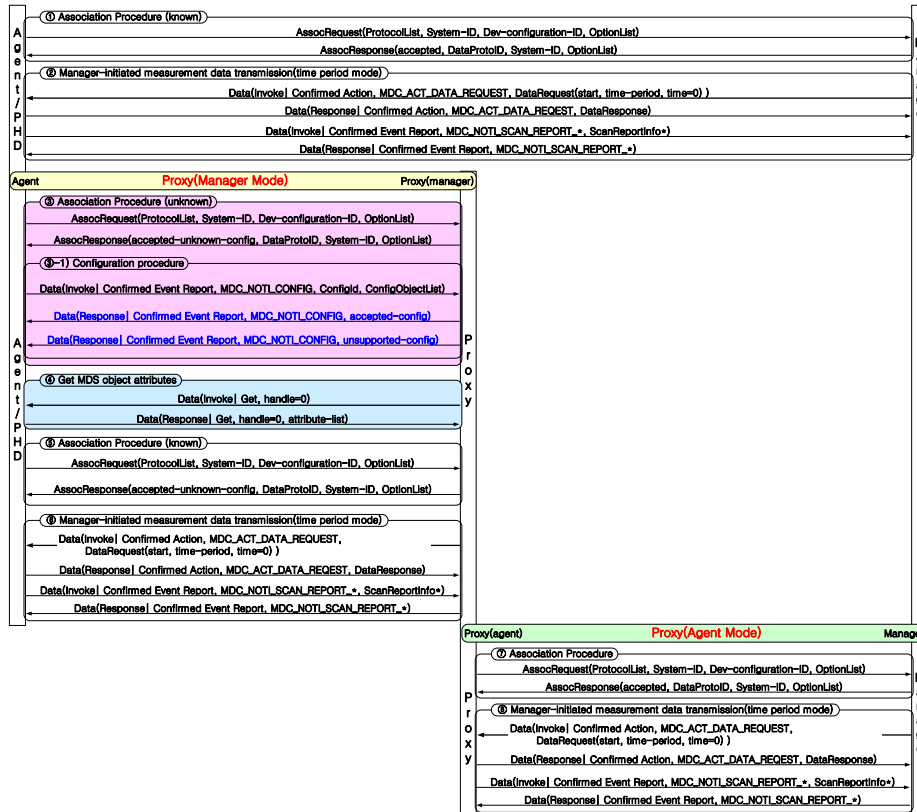


Fig. 7. Data transmission procedure for PiM operation

Fig. 7 illustrates the PiM operation in an environment where a proxy appears between an agent and the manager. Normally, when the manager sends a response to the agent's association request first, it works based on the existing sequences, as shown in **Fig. 7**. First, using the ① association procedure and ② manager-initiated measurement data transmission (time period mode), the manager and the registered agent can interoperate with each other according to the standard 11073-20601. Second, when the proxy sends a response to the request before the manager, as shown in **Fig. 7**, the proxy performs the PiM operation between the agent and the manager, including processes from ③ to ⑧. When the proxy receives the ③ association request (unknown) for the first time from an agent that is not registered to it, it performs the procedures ③ and ③-1) to obtain the agent's configuration information. At this time, the agent performs the configuration procedure with the proxy. The agent cannot recognize that it is registered in duplicate to a new proxy other than the manager that it was previously registered to. Therefore, the proxy can obtain the agent's configuration information without exchanging any information with the manager. The proxy and the agent are in associated operating state. The proxy can obtain MDS information about agent through ④ Get MDS object. The proxy is now ready to act as an intermediary between the agent and the manager.

When the proxy responds first for an association request, the PiM operation is as follows. The proxy receives an AssocRequest in ⑤ Association procedure, and it sends an "accepted" assocResponse, as if it were a manager. The agent thinks it has received an "accepted" assocResponse from the manager it is registered with. At this time, the proxy sends MDC_ACT_DATA_REQUEST in ⑥ Manager-initiated measurement data transmission (time period mode) to collect measurement data. The agent sends the measured data to the proxy in the form of a confirmed event report, MDC_NOTI_SCAN_REPORT_*. The proxy, which has received the data securely, sends a confirmed response to the agent. Then, the agent deletes the transmitted data, such that there are no available measurements.

After the proxy collects the measurement data, it sends the AssocRequest to the manager as the agent. The manager sends the "accepted" assocResponse to the proxy specified as an agent registered to it, sending MDC_ACT_DATA_REQUEST of ⑧ Manager-initiated measurement data transmission (time period mode). The proxy responds with "acknowledge", and then sends the measurement data in the form of a confirmed event report, MDC_NOTI_SCAN_REPORT_*. The proxy that receives the confirmed response deletes the related information. The proxy performs the procedures ⑤, ⑥, ⑦, and ⑧ repeatedly on the identified agents and the manager, so that they can improve connectivity without being aware of the existence of the proxy.

3.2 Smart Proxy in the Middle(SPiM) Operation

When all measurement data is transmitted through the proxy, the total transmission cost doubles as shown in **Fig. 8**. To address the drawback, a smart proxy is proposed. The smart proxy in the middle (SPiM) operation can collect measurement data from several agents and transmit them as if one agent had various object handles, as shown in **Fig. 9**. In SPiM operation, a modified manager that can interoperate with a smart proxy is required.

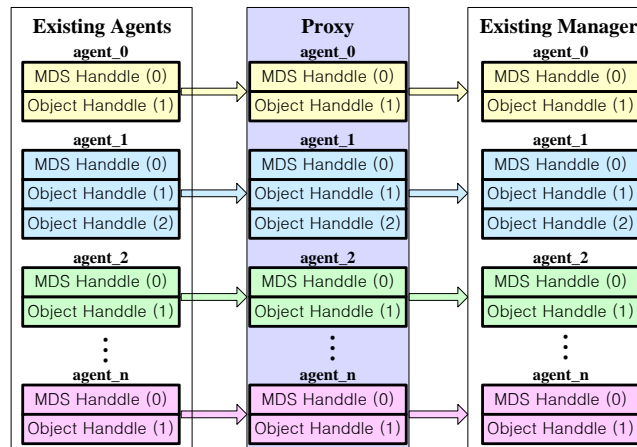


Fig. 8..PiM operation in terms of object handles

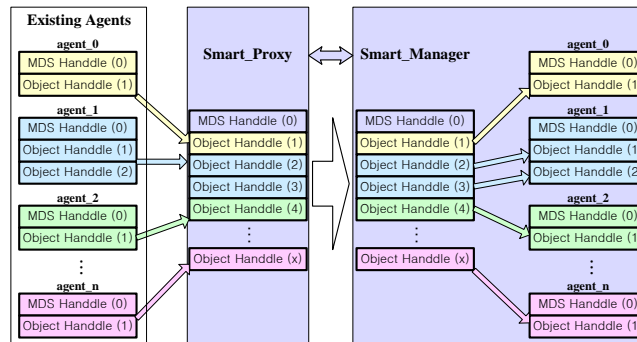


Fig. 9. SPiM operation in terms of object handles

The following problems must be solved for the SPiM operation. First, the smart proxy must inform the modified manager that it can support SPiM operations and perform mutually exclusive procedures. And the PiM operation must be performed with the existing manager. Second, the response of the modified manager must be differentiated from the response of the existing manager. For the association request, both the modified and existing managers send the same accepted-unknown-config response to the smart proxy. Hence, there is no way to distinguish whether the corresponding manager is the modified or existing manager. Third, mapping information about the handle number newly assigned by the smart proxy for each handle of each agent should be delivered to the modified manager. The smart proxy should assign a unique handle number to the different handle that can be duplicated at each handle of each agent. For example, each agent has handle 1 objects, which are actually different as shown in Fig. 8 and Fig. 9.

To clearly understand the solutions to the above problems, the PhdAssociation information and its properties included in the association request are described in detail. It includes the following information.

- 1) protocol-version
- 2) encoding-rules
- 3) nomenclature-version

- 4) **functional-units**
- 5) system-type
- 6) **system-id**
- 7) **dev-config-id**
- 8) data-req-mode-capab
- 9) option-list

The first problem can be solved using functional-units information, as follows.

```
FunctionalUnits ::= BITS-32 {
    fun-units-unidirectional(0),
    fun-units-havetestcap(1),
    fun-units-createtestassoc(2),
    fun-units-smartproxy(3),    }    -- indicates that smart proxy function is supported
```

By adding a **fun-units-smartproxy(3)** bit among the reserved bits for future use, a smart proxy can provide the related information. This bit is meaningful to a modified manager, whereas it is meaningless to an existing one. Therefore, the SPiM operation can be performed only between the smart proxy and the modified manager. The second problem can be solved by modifying the AssociateResult value included in the modified manager's association response as follows:

```
AssociateResult ::= INT-U16 {
    accepted(0)
    rejected-permanent(1)
    rejected-transient(2)
    accepted-unknown-config(3)
    rejected-no-common-protocol(4)
    rejected-no-common-parameter(5)
    rejected-unknown(6)
    rejected-unauthorized(7)
    rejected-unsupported-assoc-version(8)
    accept-SPiM-config(9)    }
```

By adding a **accept-SPiM-config(9)** value among the reserved association result values to indicate that the SPiM operation is supported, a modified manager can provide the information of configuration. This value is meaningful to a smart proxy, whereas it is meaningless to an agent and proxy. Therefore, when the smart proxy receives the “accept-SPiM-config” response, it performs the SPiM operation. The third problem can be solved by using MDS information, handle 0, which is the top-level object of all agents. The MDS represents the identification and status of the agent through its attributes. The information of the MDS object includes Handle, System-Type, System-Model, System-Id, Dev-Configuration-Id, Attribute-Value-Map, etc. Of these, the System-Model contains the manufacturer name and manufacturer specific model information, which is determined by the agent vendor and must be in printable ASCII form. Hence, it has string information of the following structure.

```
SystemModel ::= SEQUENCE {
    manufacturer    OCTET STRING,
    model-number    OCTET STRING    }
```

The smart proxy can insert the following information into the manufacturer's area as the specified format of a string, so that it can send handle mapping information to the modified manager. The format must be specified in advance.

- 1) start of mapping information
- 2) newly assigned handle number by smart proxy
- 3) System-Id of existing agent
- 4) Dev-Configuration-Id of existing agent
- 5) handle number of existing agent
- 6) end of mapping information

According to the 20601 OEP, when both the agent and the manager are in associated operating state, the manager may receive MDS information corresponding to handle 0 of all agents by using a Get service. Therefore, the smart proxy and the modified manager recognize the SPiM operation through the association procedure, and the modified manager can obtain the MDS information and the handle mapping information with the above format. For reference, the number of handles is already known in the configuration process, and mapping information of 2) ~ 5) will be repeatedly presented as the number of handles. Fig. 10 shows the procedures of the SPiM operation. The smart proxy already manages both configuration and MDS information for the registered agents by the PiM operation described in section 3.1. It interoperates with the modified manager as a new agent having all object handles that are related to the measurement data of registered agents. Moreover, the smart proxy should newly assign a unique handle number to the different handle of registered agents, and it should also generate its MDS object having this handle mapping information.

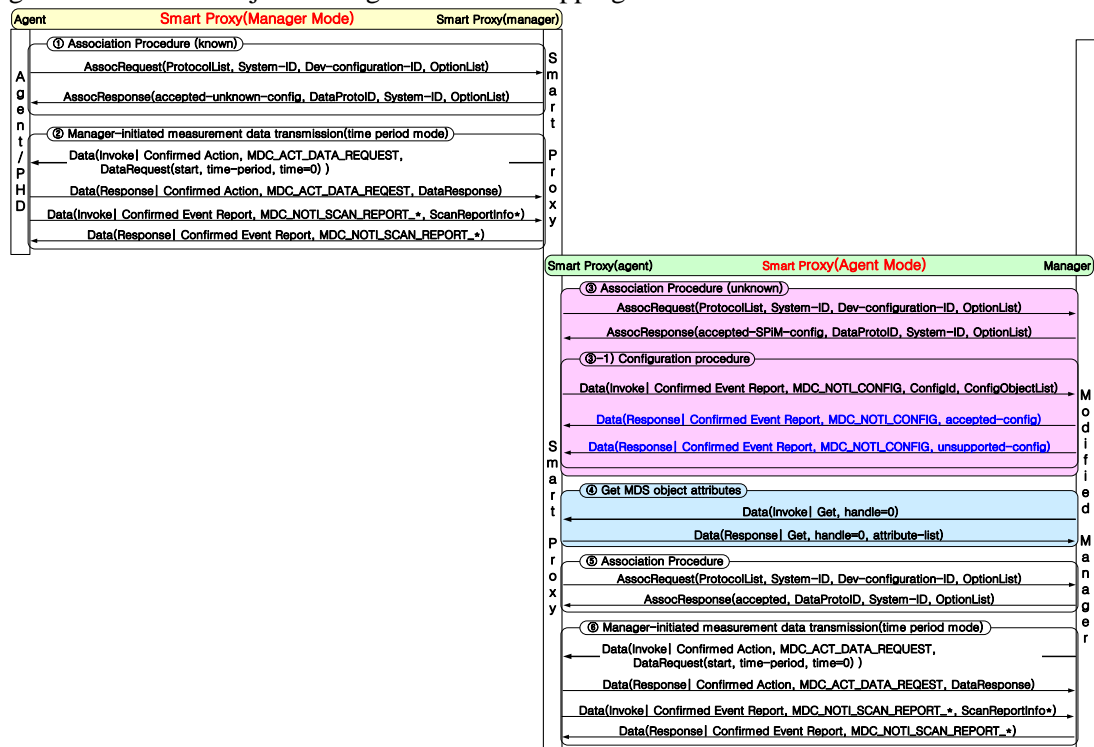


Fig. 10. Data transmission procedure for SPiM operation

Like a proxy, a smart proxy collects and manages measurement data from registered agents using ① association procedures and ② Manager-initiated measurement data transmission (time period mode). The smart proxy initiates an association procedure by sending an association request with a fun-units-smartproxy (3) bit set to indicate that the SPiM operation is possible. The smart proxy is at first also not recognized by the manager, and an association procedure (unknown) is performed and must be registered in the manager through the configuration process. For a modified manager, it recognizes that the requesting agent is a smart proxy. However, if the manager is an existing, it treats the smart proxy as an existing agent. The modified manager sends an "accept-SPiM-config" response, whereas the existing manager sends an "accepted-unknown-config". For the existing manager, the smart proxy sends the association abort (AssocAbort), and then starts the PiM operation like a proxy. And, for the modified manager, the smart proxy performs the SPiM operation. By performing the ③-1) configuration procedure, the modified manager obtains the configuration information. The modified manager registers the smart proxy as a new agent, and they are in associated operating state. The modified manager can obtain MDS information by the ④ Get MDS object procedure. According to the solution proposed in this study, the modified manager grasps the mapping information of the handles inserted in the System-Model property in the MDS information. In the SPiM operation, the modified manager obtains information about object handles in a bundle of smart proxy from configuration information and handle mapping information from MDS information. In the example of [Fig. 10](#), the modified manager manages the value of handle 1 of the smart proxy as the value of handle 1 of the designated agent_0 according to the mapping information. Similarly, handle 2 and handle 3 of the smart proxy are mapped and managed as handle 1 and handle 2 of agent 1 respectively. Therefore, if there is a change in configuration and mapping information, such as inserting a new agent to a smart proxy or deleting a registered agent from a smart proxy, the configuration procedure including ③ and ③-1) and Get MDS object procedure between the smart proxy and modified manager must be performed.

4. Verification, Analysis, and Simulation

4.1 Implementation and Verification

To verify the proposed PiM and SPiM operations, three agents, a proxy, and a smart proxy are implemented on the embedded systems, and a manager and modified manager are implemented on a PC. According to the guiding principles of the 11073-20601 standard, agents typically have very limited computing capabilities and a fixed configuration such the protocol intentionally places more load on the managers. Therefore, in the consideration of limited system resources and price of an agent, blood pressure monitor, thermometer and weight scale are implemented on the embedded system environment with the 8-bit processor as shown in [Table 1](#).

Table 1. System resources for agents and proxy/smart proxy

Agents	Proxy/smart proxy
8-bit μ Controller	64-bit μ Controller
20MHz System Clock	1.2 GHz
1024 bytes SRAM	1GB SDRAM
256 bytes Data EEPROM	Up to 32GB microSD storage
14Kbytes Program Memory	Linux O/S
4 8-bit Timers & 1 16-bit Timers	Ethernet 10/100 wired LAN
1 UART	802.11n Wi-Fi wireless LAN
Bluetooth version 2.0	Bluetooth 4.0 wireless PAN

The blood pressure monitor (agent 1) with a pulse rate numeric object, thermometer (agent 2) with a body temperature numeric object, and weighing scale (agent 3) with a body weight numeric object were implemented on three embedded systems. With consideration to the BAD's resources, a proxy and a smart proxy are implemented in the embedded system environment with a 64-bit processor, as shown in [Table 1](#). The proposed PiM and SPiM operations were tested and verified based on the association procedure, configuration procedure, get MDS object, and measurement data transmission as shown in [Fig. 3](#) and [Fig. 4](#).

First of all, it is verified that a manager operates with each of three agents (blood pressure monitor, thermometer, and weighing scale) based on 11073 standards. For a proxy, the PiM operation is implemented and verified. When all devices are in the same Bluetooth wireless PAN environment, it is verified that an agent properly makes an association with either a proxy or a manager exclusively, in the four cases shown in [Fig. 6](#). Hence, we verified that either a proxy or a manager receives the measurement data sent by an agent, and the measurement data is not sent in duplicate. Furthermore, when the devices except the manager are in one hop PAN area, a proxy collects and manages measurement data from agents using Bluetooth. The data is sent to the manager using the PiM operation in various wired/wireless communications such as Bluetooth, Ethernet, or Wi-Fi as shown in [Fig. 5](#). Finally, the manager properly stores and manages the received data as the value of the corresponding agent's object handle.

In the operation between the smart proxy and the existing manager, the smart proxy ascertains that the corresponding manager does not support the SPiM operation in the association procedure, and it sends an abort request to the manager to terminate the current association procedure. The smart proxy sends an association request with a fun-units-smartproxy (3) bit clear to initiate the PiM operation. For a modified manager, they identify that the SPiM operation can be supported during the association procedure. Meanwhile, a configuration procedure each object handle (handle 1) of the three heterogeneous agents is sequentially assigned to the smart proxy's object handle 1, 2, and 3. The modified manager obtains the MDS object that contains the handle mapping information, and it maps that handles 1, 2, and 3 correspond to handle 1 of blood pressure monitor (agent 1), handle 1 of thermometer (agent 2), and handle 1 of weighing scale (agent 3), respectively. The smart proxy efficiently transmits three different types of measurement data in one confirmed event report. The overall SPiM operation is implemented and verified.

4.2 Complexity Analysis

The complexity analysis in this study is limited to the 20601 OEP layer and the device specifications layer in [Fig. 11](#). The vendor specific parts are excluded. Complexity is analyzed

in terms of the required memory space. As shown in Fig. 11 and Fig. 12, implementations of 20601 for an agent and manager are different and a manager is more complex than an agent. However, it is assumed that the manager has sufficient resources. When each of the blood pressure monitor, thermometer, and weighing scale are implemented on the embedded system with a standard configuration and an object handle except the MDS object, the required memory size is not very large, with a maximum of 25 KB.

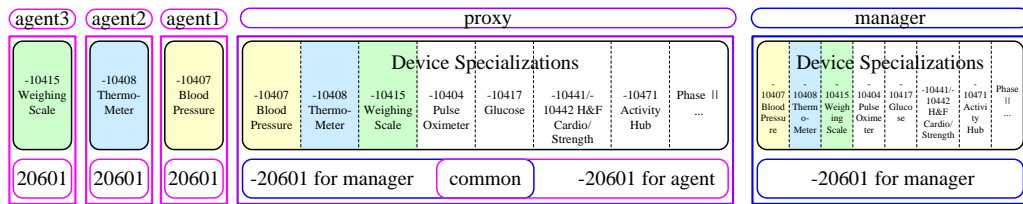


Fig. 11. Document maps of proxy, agents, manager structure over transport layer

The proxy has a structure including both agents and the manager, and the complexity of a proxy is larger than that of the manager. Hence, the complexity of the proxy seems to be the sum of the registered agents and the complexity of the manager. However, the complexity of the agents in the proxy is much lower than the sum of the complexity of each agent. In fact, the 20601 layer occupies a large part of the complexity of an agent, and is used redundantly by other agents. The proxy is not required to independently implement the agents, and the 20601 layer is commonly used. Only the device specification of the upper layer needs to be implemented separately. According to the implementation results of this study, the complexity of the proxy is confirmed to be much smaller than 75 KB. The complexity of the manager part in the proxy is similar to that of the manager. Moreover, because the 20601 and the device specification have a common part in the agent and the manager, when the proxy is implemented efficiently so as not to be overlapped, the complexity of the proxy can be reduced more effectively. As shown in Fig. 11, agents and managers can share device specifications on the proxy. Therefore, the complexity of the proxy is not a problem for the BAD environment. For example, it is small enough to be applied in the form of an app such as the smart phone.

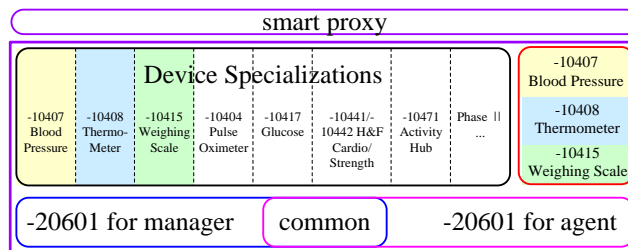


Fig. 12. Document map of smart proxy structure over transport layer

When working with a modified manager, the smart proxy acts as an agent with a new device specification, as shown in Fig. 12. The smart proxy is similar to adding one independent representative agent to the proxy. Hence, the smart proxy seems to be another agent with a new extended configuration. The added complexity of the smart proxy involves parts for extended configuration and MDS object including handle mapping information. The added complexity

of the smart proxy can be calculated and it is closely related to the following parameters.

- nA: number of agents
- nC(i): number of configurations in i_{th} agent
- nH(i,j): number of handles in j_{th} configuration, i_{th} agent
- newMDS: memory space for a mapping information in a new MDS

The complexity for creating extended configuration (C_{EC}) can be formulated as follows.

$$C_{EC}(i, j) = a g nA + b g \sum_{i=1}^{nA} nC(i) + c g \sum_{i=1}^{nA} \sum_{j=1}^{nC} nH(i, j) \quad (1)$$

Where a, b, and c are the proportional constants of memory required for nA, nC, and nH, respectively. The complexity of creating MDS object with mapping information about handles (C_{MDS}) can be described as follows.

$$C_{MDS}(i) = p \sum_{i=1}^{nA} \sum_{j=1}^{nC} \sum_{k=1}^{nH} newMDS \quad (2)$$

Here, p is a proportional constant of memory required for MDS including mapping information about handles. The added complexity of the proxy implemented in this experiment is calculated using the above formulas, and the result is smaller than 75 KB as expected. The complexity of the modified manager is slightly more complicated than that of the general manager, however it remains lower in complexity compared to smart proxies.

4.3 Simulation

As discussed above, devices including agents, the proxy, and the manager need not be in the same wireless communication environment, e.g., in a Bluetooth. Hence, the connections among the devices in the system may be different, and there could be multiple wireless methods. However, to fairly analyze the performance of the system, including the repeated transmission through the proxy, it is assumed that all devices are in one wireless communication environment. The number of data transmissions can be increased up to double.

In this study, the performance of the existing system, along with the system that involves the proxy, is analyzed and compared. The analysis includes delay, throughput, and wireless channel occupation time (i.e., wireless transmission power consumption). In **Fig. 13** (a), there are agents and the manager only and an agent sends its data to the manager through a wireless channel. In **Fig. 13** (b), a proxy is involved and an agent sends its data to the proxy first, after which is transmitted to the manager. In **Fig. 13** (c), a smart proxy is added instead of the proxy, and this system is operated by the SPiM operation.

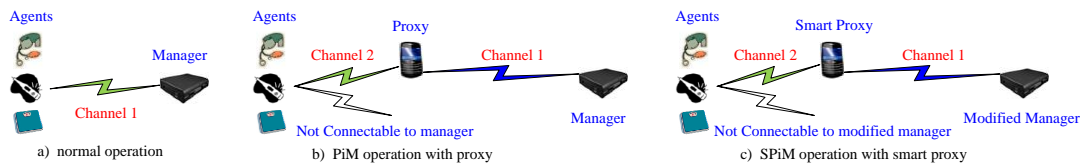


Fig. 13. Simulation environments

The same number of agents are assumed in all three scenarios. It is also assumed that all agents measure body information intermittently and generate the same amount of information. When an agent sends measurement data, the simulation includes all the related messages defined in 20601 OEP. Further, all agents are considered to be associated, and the transmission

channel is modeled as a wireless transport one at the application layer. In fact, 20601 OEP belongs to the application layer, and all the layers below (including the physical layer) are mapped into the wireless channel model for simplicity in this simulation.

For a realistic simulation, Bluetooth, the most widely used PAN communication, is adopted. In Bluetooth version 1.2, the data rate is 1 Mbps and the maximum throughput at the application is about 0.7 Mbps [18]. A trial of measurement data transmission between devices takes approximately 1,460 μ S for a measured value of one object handle. When a smart proxy sends a maximum of 25 measurement data in one transmission, it takes about 6030 μ S. The transmission capacity (throughput) includes only measurement data successfully transmitted in the application layer. In this simulation, fixed 4-byte measurement data is assumed.

The channel condition is assumed as a slowly fading in the wireless PAN. The two-state Markov Gilbert-Elliot model [19-20] is applied. This GE model has been widely used in the related research of wireless channels [21-23]. The model with two “good” (G) and “bad” (B) states is shown in Fig. 14. The transition probabilities from G to B and from B to G are denoted P_{GB} and P_{BG} , respectively. P_{GG} and P_{BB} denote the transition probabilities from G to G (stay in G) and from B to B (stay in B), respectively. Then, the steady-state probabilities of the states G and B, π_G and π_B , are given by $\pi_G = P_{BG} / (P_{BG} + P_{GB})$ and $\pi_B = P_{GB} / (P_{BG} + P_{GB})$, respectively.

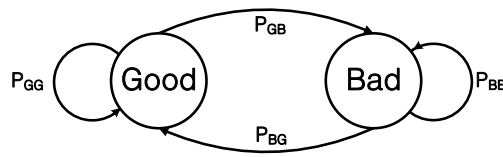


Fig. 14. Two-state Markov Gilbert-Elliot channel model

For a given i_{th} agent among the total U agents, it is assumed that measurement data is episodically generated according to a discrete Poisson random process, with a mean λ_i in a second. The incoming measurement in a proxy is also a discrete Poisson random process, with a mean λ_T , due to the superposition property of the Poisson, and it is expressed as follows:

$$\lambda_T = \sum_{i=1}^U \lambda_i \quad (3)$$

The connection between an agent and the proxy/smart proxy is modeled as a channel with low error rate ($\pi_G = 0.95$ and $P_{BG} = 0.8$). The reason is that at the time of the agent measurements, the proxy on the BAD and the agent are located closely to each other. The channel model between the agent and manager is assumed to vary widely from good to bad conditions. For a fair comparison, the channel between the proxy/smart proxy and the manager is equivalent to the channel between the agent and the manager.

In terms of delay, scenario 1 exhibits the best, whereas scenario 2 exhibits the worst result. Assuming bundles of 25 measurement data in scenario 3, the delay is improved in comparison to that of scenario 2. As the wireless channel condition exacerbates, the difference in the delay performance is decreased gradually (Fig. 15 (a) and (b)). In a good channel environment, the throughputs of Scenario 1 and Scenario 3 are almost the same, whereas in a bad channel

condition, the performance of Scenario 3 is further enhanced (Fig. 15 (c) and (d)). Channel occupation time is closely related to power consumption, and Scenario 2 uses twice as many channels as Scenario 1. However, Scenario 3 utilizes almost the same amount of wireless channels as Scenario 1, and it occupies the channel more efficiently than Scenario 1 in a bad channel condition (Fig. 15 (e) and (f)). As expected, simulation results show that the use of proxy can degrade overall the system performance. However, in case of a smart proxy, the system performance degradation is insignificant. On the contrary, when the channel is poor, the performance is significantly improved.

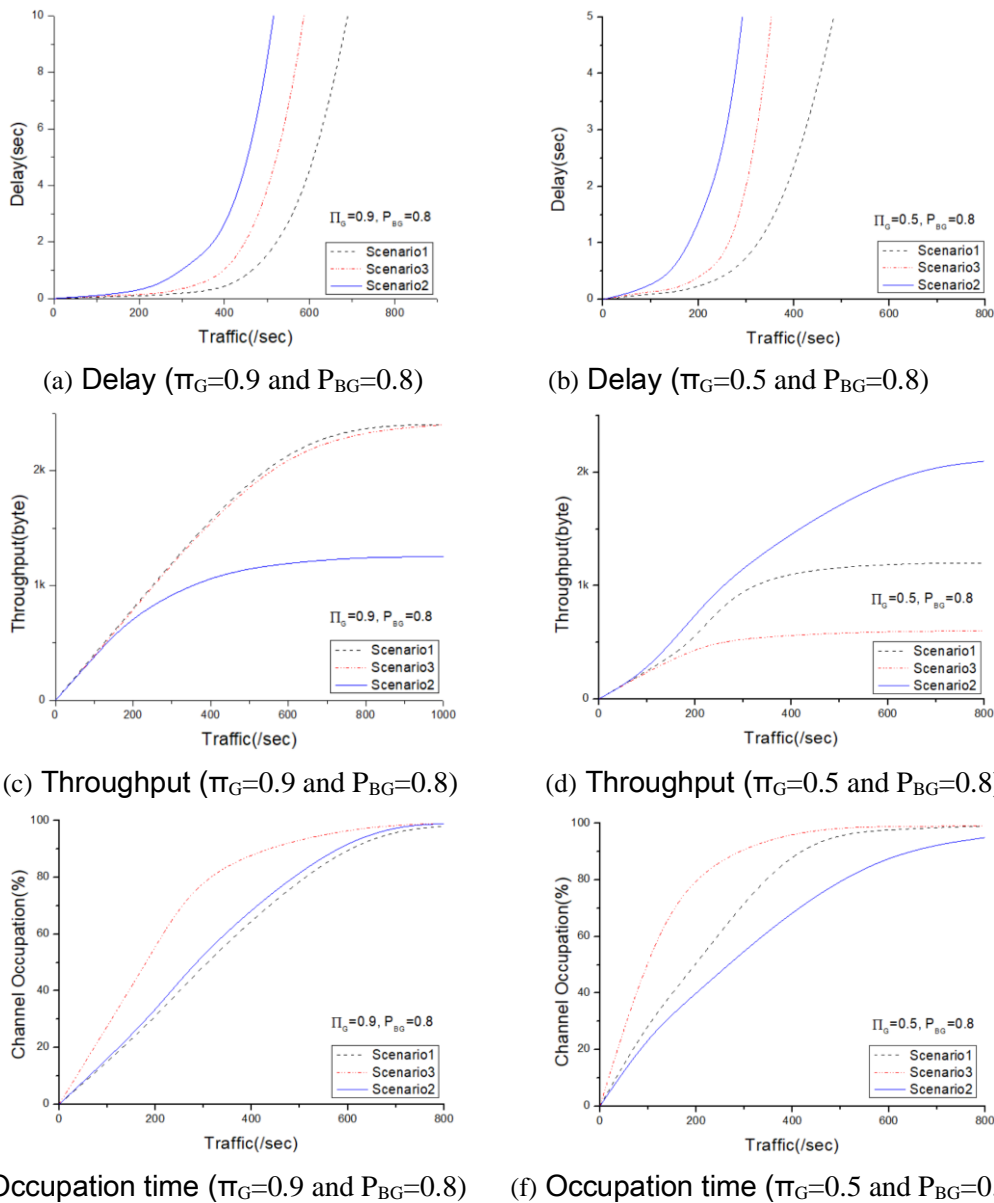


Fig. 15. Simulation results of delay, throughput, and channel occupation

5. Conclusion

The advanced healthcare service system consists of many agents and a manager, where the manager controls most agents concurrently through wireless communication, for the user's convenience. It is crucial for service to provide continuous connectivity between devices. In this study, a novel proxy is proposed, which can greatly improve network connectivity and system efficiency. The PiM operation relays the measurement from an agent to the manager. A smart proxy is proposed, and its SPiM operation transmits large amount of data at once to the modified manager. The proxy and smart proxy maintain compatibility with existing devices that conform to the 20601 standard. PiM and SPiM operations are verified on an embedded system, and their complexity is analyzed. The required memory of the proxy/smart proxy is found to be sufficiently small to be applied in the form of an app to the BAD. Although the use of a proxy can degrade overall system performance, simulation results show that the smart proxy can significantly improve the performance of the system, particularly when the channel condition is bad.

References

- [1] S. Raj and K. C. Ray, "A Personalized Point-of-Care Platform for Real-Time ECG Monitoring," *IEEE Trans. on Consumer Electronics*, vol. 64, no. 4, pp. 452-460, November, 2018. [Article \(CrossRef Link\)](#)
- [2] X. Wang and Z. Jin, "An Overview of Mobile Cloud Computing for Pervasive Healthcare," in *IEEE Access*, vol. 7, pp. 66774-66791, May, 2019. [Article \(CrossRef Link\)](#)
- [3] Health Informatics-Personal Health Device Communication, ISO/IEEE 11073. Available: <http://standards.ieee.org/>.
- [4] Health Informatics-Personal Health Device Communication Part 20601: Application Profile-Optimized Exchange Protocol. ISO/IEEE Std. 11073-20601-2008. Available at: <http://standards.ieee.org/findstds/standard/11073-20601-2008.html>
- [5] Health Informatics-Personal Health Device Communication Part 20601: Application Profile-Optimized Exchange Protocol Amendment 1. ISO/ IEEE Std. 11073-20601a-2010. Available at: <http://standards.ieee.org/findstds/standard/11073-20601a-2010.html>
- [6] Health Informatics-Personal Health Device Communication Part 10408: Device Specialization-Thermometer. ISO/IEEE Std. 11073-10408-2010. Available at: <http://standards.ieee.org/findstds/standard/11073-10408-2010.html>
- [7] Health Informatics-Personal Health Device Communication Part 10415: Device Specialization-Weighing Scale. ISO/IEEE Std. 11073-10415-2010. Available at: <http://standards.ieee.org/findstds/standard/11073-10415-2010.html>
- [8] Health Informatics-Personal Health Device Communication Part 10417: Device Specialization-Glucose Meter. ISO/IEEE Std. 11073-10417-2010.
- [9] Health Informatics-Personal Health Device Communication Part 10406: Device Specialization-Basic Electrocardiograph. ISO/IEEE Std. 11073- 10406-2011. Available at: <http://standards.ieee.org/findstds/standard/11073-10406-2011.html>
- [10] S.-K. Kim, T.-K. Kim and H.-K. Lee, "A Novel Transmission Scheme for Compressed Health Data Using ISO/IEEE11073-20601," *KSII Transactions on Internet and Information Systems*, vol. 11, No. 12, pp. 5855-5877, Dec., 2017. [Article \(CrossRef Link\)](#)
- [11] Tzong-Shyan Lin, Pei-Yu Liu, and Chun-Cheng Lin, "Home Healthcare Matching Service System Using the Internet of Things," *Mob. Netw. & Appl.*, vol. 24, no. 3, pp. 736-747, June 2019. [Article \(CrossRef Link\)](#)

- [12] P. Nguyen Huu, V. Tran-Quang, and T. Miyoshi, "Low-Complexity and Energy-Efficient Algorithms on Image Compression for Wireless Sensor Networks," *IEICE Transactions on Communications*, vol. E93-B, no.12, pp. 3438-3447, December 2010. [Article \(CrossRef Link\)](#)
- [13] Universal Serial Bus Device Class Definition for Personal Healthcare Devices Release 1.0. Available: http://www.usb.org/developers/devclass_docs
- [14] Bluetooth Health Device Profile Version 1.0 Revision 00. Available: <https://www.bluetooth.org/Technical/Specifications/adopted.htm>
- [15] ZigBee Health Care Profile Specification Version 1.0 Revision 15. Available: <http://www.zigbee.org/Standards/ZigBeeHealthCare/download.aspx>
- [16] Health Informatics-Personal Health Device Communication Part 10407: Device Specialization-Blood Pressure Monitor. ISO/IEEE Std. 11073-10407-2010. Available at: <http://standards.ieee.org/findstds/standard/11073-10407-2010.html>
- [17] Health Informatics-Personal Health Device Communication Part 10408: Device Specialization-Independent Living Activity Hub. ISO/IEEE Std. 11073-10471-2010.
- [18] Specification of the Bluetooth System, Core Version 1.2, Bluetooth Special Interest Group, 2003. Available at: <http://www.bluetooth>
- [19] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell Systems Technical Journal*, vol. 39, pp. 1253-1265, Sept., 1960. [Article \(CrossRef Link\)](#)
- [20] E. O. Elliot, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell Systems Technical Journal*, vol. 42, pp. 1977-1997, Sept. 1963. [Article \(CrossRef Link\)](#)
- [21] A. Wiling, "Redundancy Concepts to Increase Transmission Reliability in Wireless Industrial LANs," *IEEE Trans. on Industrial Informatics*, vol. 1, no. 3, pp. 173-182, 2005. [Article \(CrossRef Link\)](#)
- [22] W. Chen, J. Wang, D. Shi and L. Shi, "Event-Based State Estimation of Hidden Markov Models Through a Gilbert-Elliott Channel," *IEEE Trans. on Communications*, vol. 65, no. 7, pp. 3626-3633. Jul. 2017. [Article \(CrossRef Link\)](#)
- [23] M. B. Khuzani, H. E. Saffar and P. Mitran, "On Adaptive Power Control for Energy Harvesting Communication Over Markov Fading Channels," *IEEE Trans. on Automatic Control*, vol. 65, no. 2, pp. 863-875. Feb. 2017. [Article \(CrossRef Link\)](#)



Sang-Kon Kim He received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2008. He is currently an assistant professor of Electronics and Information Engineering in Korea University. His research interests include wired and wireless networks and communications, network and computer security, and e-health.



Tae-Kon Kim He received the Ph.D. degree in electrical engineering from the Pennsylvania State University, University Park, PA, in 2001. From 2001 to 2002, he was with Technology & Research Labs, Intel Corporation, Chandler, AZ. From 2003 to 2004, he was with Digital Media R&D center, Samsung Electronics, Suwon, Korea. Since 2005, he has been with Korea University, Korea, where he is an associate professor of Electronics and Information Engineering. His research interests include multimedia signal processing, wireless networks and communications, and e-health.



Jin-Hwan Koh He was born in Taegu, Korea. He received the B.S. degree in Electronics from Inha University, Incheon, Korea, in 1991 and the M.S. and Ph.D. degrees in Electrical engineering from Syracuse University, Syracuse, NY, in 1999. He formerly worked for LG Electron Semiconductor Company in Korea. From 2000 to 2003, he was an Assistant professor at the Kyungpook National University, Taegu, Korea. He is now a Professor in the Department of Electronic Engineering, Engineering Research Institute, GyeongSang National University, Jinju, Korea. He is a member of the IEICE, KICS, and IEEK and a reviewer for the IEEE transactions on Antennas and Propagation. His research interests include Signal processing for antennas, Antenna pattern measurements, DOA estimations and EM data interpolation/extrapolation.