International Journal of Smart Home
Vol.8, No.5 (2014), pp.103-114
http://dx.doi.org/10.14257/ijsh.2014.8.5.10

A Resilient Device Monitoring System in Collaboration Environments

KeeHyun Park[1] and JongHwi Lee[1]

Department of Computer Engineering, Keimyung University, Daegu, Korea
{khp, dragon8829}@kmu.ac.kr

Abstract

In this paper, a resilient device monitoring system is proposed in which agents on remote devices can collaborate with each other. The system consists of collaboration groups, proxy servers and a device monitoring server. A collaboration group consists of P2P agents and a P2P host which represents the collaboration group. The system is resilient in the sense that it finds an alternative P2P host or proxy server to continue its works in the event of a failure of the original ones. Communication protocols including communication messages and flow sequences for finding alternatives are proposed in detail, and some implementation results are explained.

Keywords: *Device monitoring, Collaboration, P2P, Mobile device, Resilience*

1. Introduction

A device monitoring system can monitor the activities of remote devices in order to manage the devices remotely, without the need for operators to visit the places where the devices are located [1-5]. Typical examples of such monitoring activities are control of the device activities, value setting in the devices, error monitoring, remote install and execution of new programs in the devices, etc. Because visiting a remote device would require a great deal of time and effort, making it difficult to promptly respond to unexpected circumstances, numerous remote device monitoring systems have been developed. Examples of devices to be monitored are Personal Healthcare devices, game terminals, home appliances, lighting equipment, and various sensing devices. Usually, an agent is installed in the device to communicate with a device monitoring server for the purpose of remote device monitoring [6-10].

Most remote device monitoring systems that have been developed thus far are based on 1:1 communication between a remote device and the device monitoring server. However, as devices become smarter and as users needs become more diverse, it becomes more difficult for the existing systems to deal with the following problems:

- Lack of ability to react to environments in which collaboration is required: In order to meet the current trend of lighter, smaller and cheaper devices, devices are required to execute collaborative work to compensate for the diminished computing and communication power of the devices. Typical examples can be found in the area of distributed sensor networks, in which most sensors do not have enough communication power to send data to the remote server. All the sensed data are sent to the adjacent sink node and the sink node in turn transmits the data to the remote server [11-12], which demonstrates the collaboration work of communication.

ISSN: 1975-4094 IJSH

Another type of collaboration work can be seen in game terminals. In the past, users would enjoy their games through one game terminal, or through several game terminals connected to one game machine. Recently, users have wanted to enjoy their games in a way that is location-independent (i.e., enjoy the same game with different game terminals in different locations). For example, users want to play the same screen golf game in different locations with different screen game terminals [13]. In this case, screen golf game terminals located in geographically remote areas must be able to collaborate in order to 'place' users in the same game.

However, most of the existing device monitoring systems would suffer from degraded performance because only 1:1 communication between devices and the server is provided, and the problem of bottlenecks at the server would occur.

- Occurrence of security problems due to excessive propagation of sensitive data: The data sensed by devices may include sensitive data such as personal biomedical data. Therefore, device monitoring systems must pay special attention to the sensitive data which has to travel all the way from remote devices to the server in most existing device monitoring systems, increasing the risk of exposure of sensitive data [14]. If remote devices in a specific area (i.e., user's home) can collaborate with each other to process most of the sensitive biomedical data, only a small portion of the data would travel to the server, decreasing the exposure risk.

In this paper, a resilient device monitoring system is proposed in which agents on remote devices can collaborate with each other. The system consists of collaboration groups, proxy servers and a device monitoring server. A collaboration group consists of P2P agents and a P2P host which represents the collaboration group. The system is resilient in the sense that it finds an alternative P2P host or proxy server to continue its works in the event of a failure of the original ones. The remainder of this paper is organized as follows. Section 2 describes the previous studies related to this study. Section 3 explains the design of the system proposed in this paper. Next, Section 4 describes the results of the implementation. Finally, Section 5 draws conclusions and discusses some directions for future research.

2. Related Studies

2.1 Device Management System

The management of remote devices has been one of the hot issues in various applications [1, 15-17]. Recently, some studies of remote device management have been focused on PHD (Personal Healthcare Device) management systems, because continuous monitoring of patient health is required in ubiquitous healthcare environments [15-16]. Remote management systems of pulse oximetry and medication dispenser were studied, in [15] and [16], respectively.

The system UbiMMS is comprised of medication dispensers and a monitoring server, as shown in Figure 1 [16]. Once medication schedules and system settings are configured manually by users or automatically via the remote monitoring server, the configured information is stored in memory. When the real-time clock reaches the medication time, the alarm notifies the user through a buzzer that it is time to take medications. If the user presses the dispense button at that time, the predetermined medications are dispensed. The medication dispensers transmit the patient's medication and device status and device configurations. The

medication status is transmitted periodically, whereas the device configurations are transmitted whenever the monitoring server sends a request. Information on the occurrence of events such as a shortage of medication, medication jam, memory overload, software error, or non-adherence is transmitted immediately. The monitoring server sends the received status data and configurations to medical staff and system administrators via a medication monitor and a system monitor, respectively. In addition, it generates management operations to manage configurations, software, and medication dispenser errors if necessary [16].

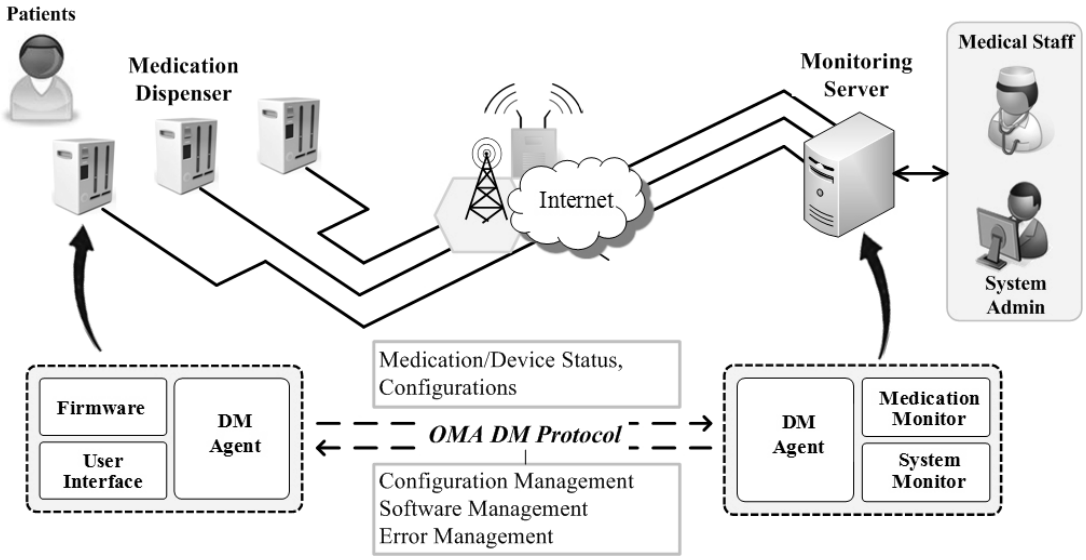

Figure 1. Ubiquitous Medication Monitoring System

2.2 Collaborating Agents

Figure 2 shows the structure of the agent-based device management system designed in the earlier studies [18, 19]. The system consists of agents, P2P hosts, and an Agent data server. Agents can exchange their local data with each other in the same group via a P2P host in the group. Upon receiving data from an agent in the group to which the P2P host belongs, the P2P host saves the data to its local database and transmits the data to every agent in the group. Only one P2P host exists in a group to represent the group, which means that transmission of data outside the group must be performed by the P2P host [18, 19**].**

P2P hosts can collaborate with other P2P hosts via the Agent server. The Agent server can multicast data to P2P hosts in response to a request from them. In addition, the server issues commands to reconfigure settings or update programs of the P2P hosts. The server also communicates with the P2P host to authenticate an agent when the agent logs in.

Figure 3 (a) and (b) shows the structures of the agent and a P2P host proposed in the earlier study [15-16]. The agent consists of a session handler module, a network module, a message handler module and a manager module. The session handler module manages the communication session. Along with the network module, the module deals with communication with a P2P host. Because communication messages transmitted in the system are represented in XML [20, 21], it is the responsibility of the message handler module to parse, analyze, or generate XML messages. The manager module transmits XML messages to the P2P host in the group to which the agent belongs. As shown in Figure 4, the agent in this study consists of various classes, such as an

IPC_Controller class, a User Object class, a NativeMethod class, an IPC_Structs class, an XMLFactory class, and a P2PClient (or a P2PHost) class.

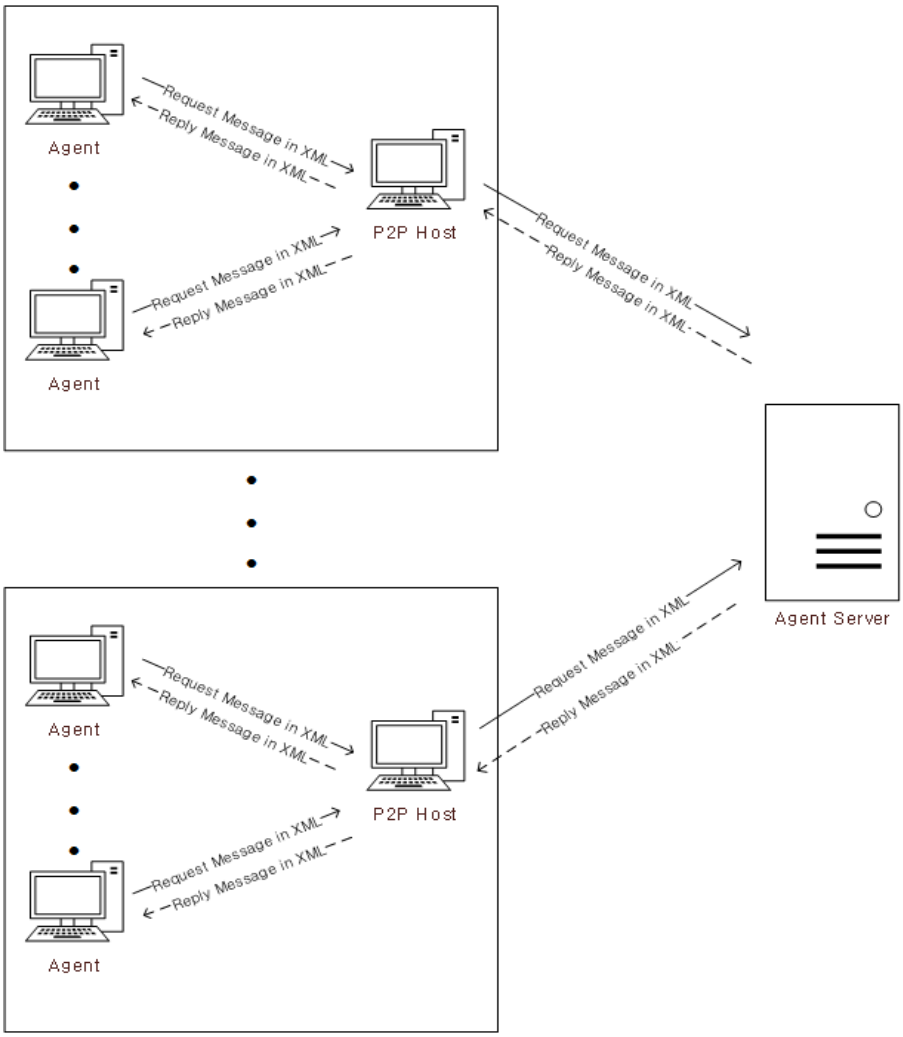

Figure 2. Structure of an Agent-based Device Management System

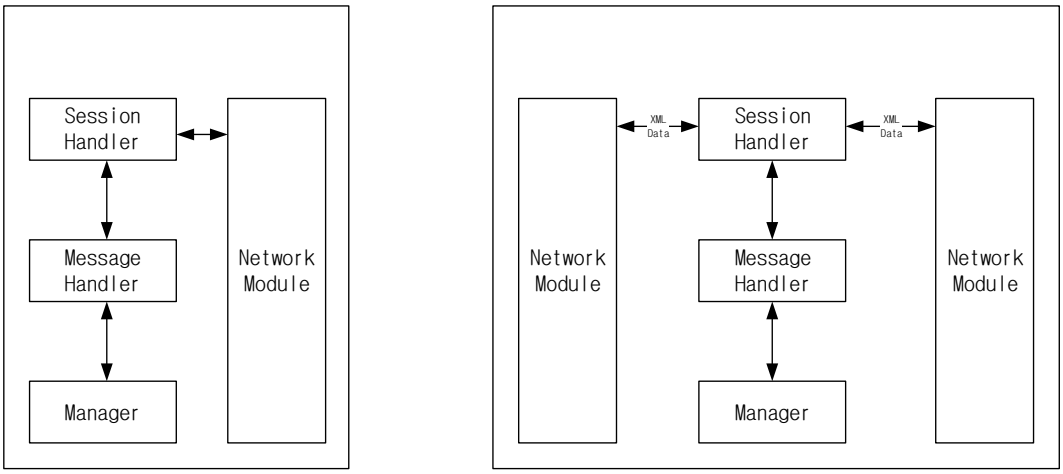

Figure 3. Structures of an agent (a) and a P2P host (b)

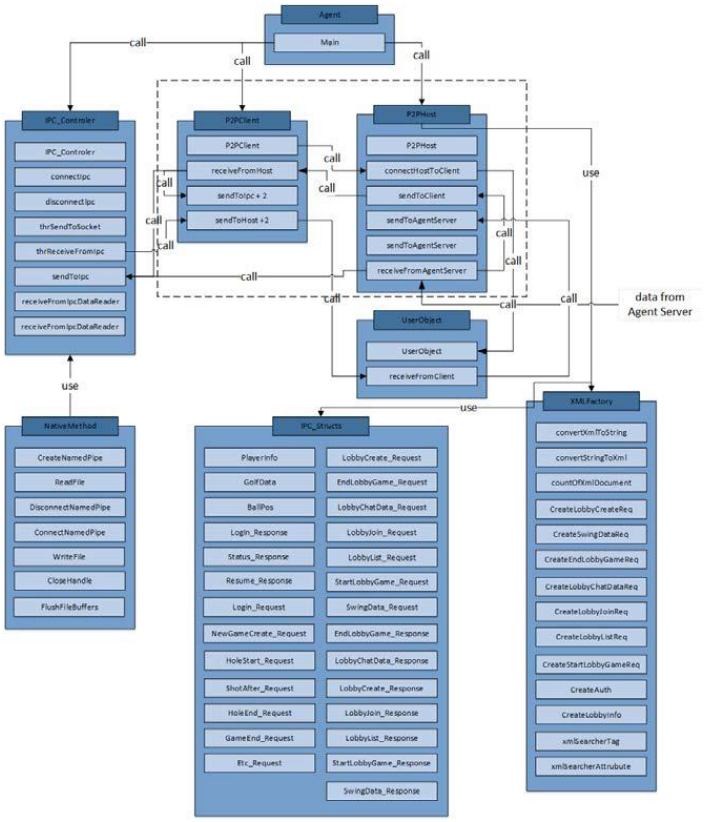

Figure 4. UML Diagram of the Agent

3. Remote Device Monitoring System

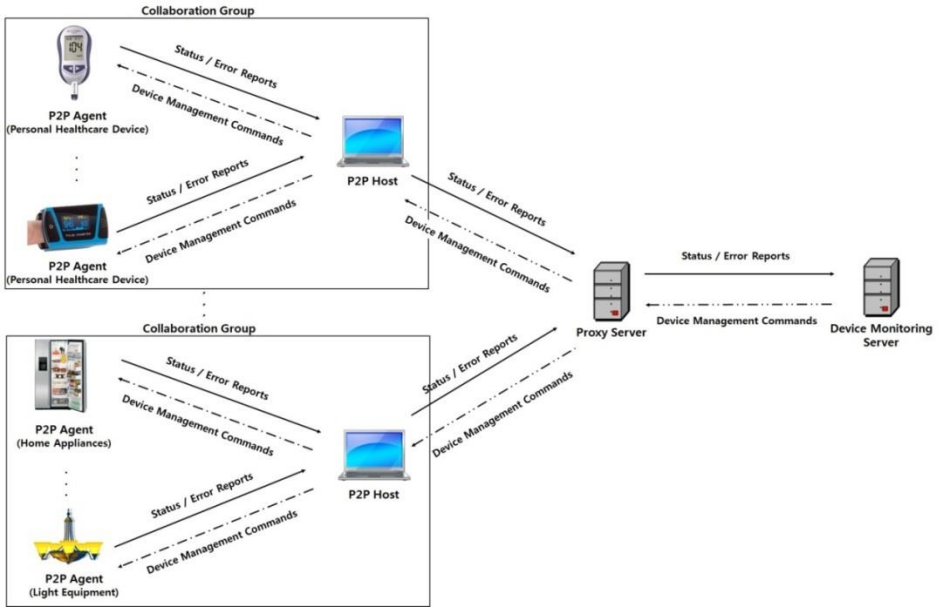

Figure 5. Device Monitoring System in Collaboration Environments

Figure 5 shows the resilient device monitoring system proposed in this study in collaboration environments. The system consists of collaboration groups, proxy servers and a device monitoring server. A collaboration group consists of P2P agents and a P2P host. When a user logs into a mobile device, the P2P agent installed in the mobile device is activated. To perform cooperative works, the agent must either create a new collaboration group by itself or join one of the existing collaboration groups. When a P2P agent creates a new collaboration group, the P2P agent becomes a P2P host of the collaboration group, representing the group. In a collaboration group, P2P agents transmit messages between them through the P2P host of the group using the P2P communication protocol [22, 23]. All the messages between P2P agents and a proxy server must pass through the P2P host of the group, and all the messages transmitted in the system are represented in XML.

The device monitoring system in this study performs device management tasks for remote device management and remote error reporting. The device monitoring server in the system manages mobile devices remotely through the following items:

- AgentStatus: represents the status of the agent in the remote mobile device to be managed
- AgentVersion: represents the version of the agent installed in the device
- OSVersion: represents the version of the operating system installed in the device
- Manufacturer: represents the manufacturer of the device
- Network Info: represents IP addresses and ports for device, separated by ‘:’

The device monitoring message is generated by the agent of the mobile device to transmit in XML to the device monitoring server via the P2P host and the proxy server to which the agent belongs. An example of the message is shown in Table 1.

Table 1. Device Monitoring Message in XML (An Example)

```
<Device_Monitoring>
   <AgentStatus> 0 </AgentStatus>
   <AgentVersion> 1.0.0 </SoftwareVersion>
   <OSVersion> Windows7 </OSVersion>
   <Manufacturer> Samsung </Manufacturer>
   <NetworkInfo> 210.125.31.70:1308 </NetworkInfo>
</Device_Monitoring >
```

The device monitoring server recognizes the errors that occur in a mobile device by referring to the device management message transmitted by the device. Error items defined in this study are shown in Table 2.

Table 2. Error Items to be Reported

Error Items	Value	Comments
OK	000	No errors
Bad proxy server	001	Failure to access the proxy server
Bad agent	002	Error occurrence in the agent
Bad collaboration group	003	Failure to access the collaboration group

Examples of the error reporting message are shown in Tables 3 through 5. From a device agent, the device monitoring server receives the error reporting message shown in Table 3, when there are no errors in the device agent. Error reporting messages on a proxy server and a collaboration group are shown in Tables 4 and 5, respectively. When the device monitoring server receives an error reporting message on a bad proxy server, the device monitoring server tells the related P2P hosts which proxy server they should access. When the device monitoring server receives an error reporting message on a collaboration group, the related proxy server releases the related session between the error reporting P2P agent and the related P2P host to allow the agent to join another collaboration group.

Table 3. Error Reporting Message in XML (No Errors)

```
<ErrorReportingRequest>
   <ErrorStatus> 000 </ErrorStatus>
</ErrorReportingRequest>
```

Table 4. Error Reporting Message in XML (Bad Proxy Server)

```
<ErrorReportingRequest>
   <ErrorStatus> 001 </ErrorStatus>
   <ProxyInfo> 210.125.31.70:1305 </ProxyInfo>
</ErrorReportingRequest>
```

Table 5. Error Reporting Message in XML (Bad Collaboration Group)

```
<ErrorReportingRequest>
      <ErrorStatus> 003 </ErrorStatus>
      <CollaborationGroupID> 0012 </ CollaborationGroupID >
</ErrorReportingRequest>
```

Figure 6 shows the flow sequence of a Bad proxy server error. When an agent tries to login and finds that the related proxy server cannot be reached, the agent sends the device monitoring server an error reporting request message (error reporting message on Bad proxy server), as shown in Table 4, in order to request an alternative proxy server to login. Upon receiving the error reporting message from the agent, the server recognizes the failure of the proxy server and sends the agent an error reporting response message to tell the agent the alternative proxy server to use for login. Then, the agent tries to login to the alternative proxy server and performs the normal login procedure.

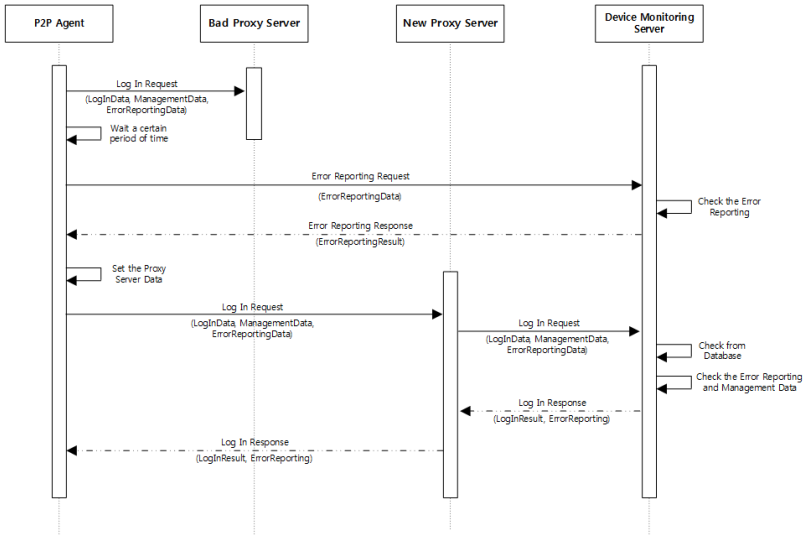

Figure 6. Flow Sequence on the Bad Proxy Server Error

Figure 7 shows the flow sequence of a Bad collaboration group error. When an agent tries to join a collaboration group and finds that the related P2P host cannot be reached, the agent sends the related proxy server an error reporting request message (error reporting message on Bad collaboration group), as shown in Table 5. Then, the proxy server relays the message to the device monitoring server. Upon receiving the message from the proxy server, the server recognizes the failure of the collaboration group and allows the related proxy server to choose a new P2P host for the collaboration group. Then, the new P2P host sends response messages for connection to all the P2P agents in the group. Upon receiving the message from the new P2P host, the agents update information of their collaboration group.

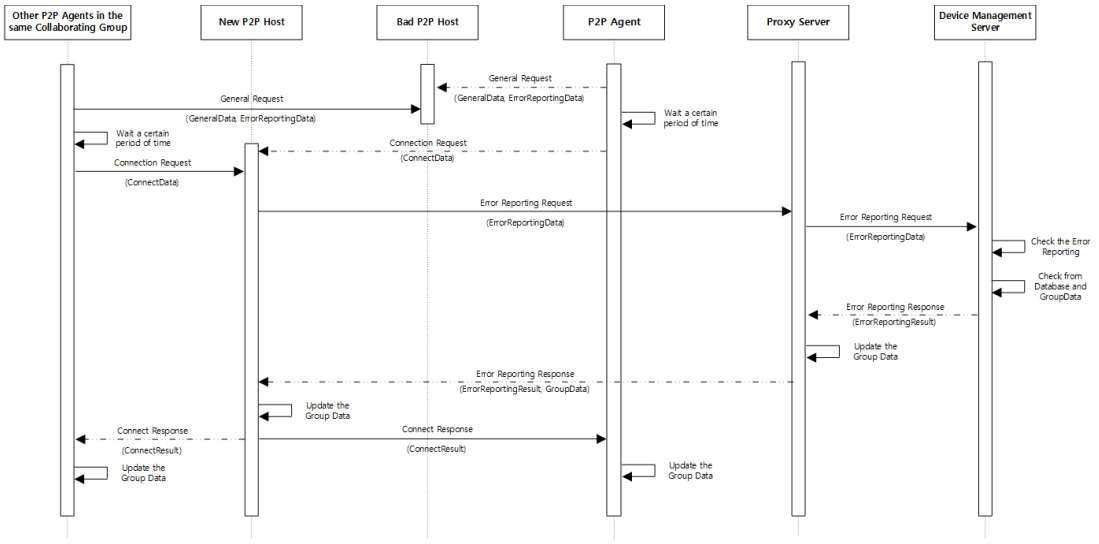

Figure 7. Flow Sequence on the Bad Collaboration Group Error

4. Results

The system explained above is implemented. Figure 8 shows a screen shot that illustrates how the system is working when the Bad proxy server error occurs. In this figure, a P2P sends an ErrorReportingRequest message in XML to the device monitoring server after the P2P agent finds it cannot access the related proxy server whose IP address is 210.125.31.72. The content of the message is shown in the upper part of the screen ('Send XML'). Upon receiving the ErrorReportingRequest message, the device monitoring server sends the agent an ErrorReportingResponse message in XML, the content of which is shown in the lower part of the screen ('Receive XML'). The server tells the agent that the alternative proxy server is 210.125.31.70.

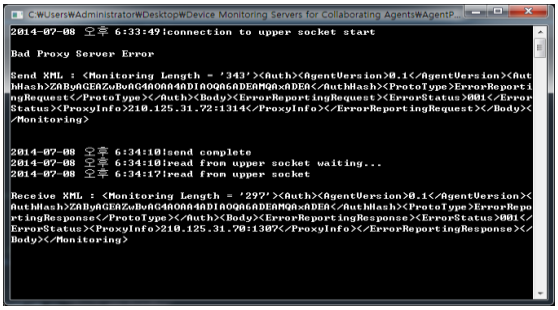

Figure 8. A Screen Shot on the Bad Proxy Server Error

5. Conclusion and Future Research

In this paper, a resilient device monitoring system is proposed, in which agents on remote devices can collaborate with each other. For the original P2P host representing a collaboration group, the system finds an alternative P2P host to maintain the related collaboration group work. Also, the system can find an alternative proxy server when the original server cannot be accessed. Communication protocols including communication messages and flow sequences for finding alternatives are proposed in

detail, and some implementation results are explained. For future research, efficient methods of selecting from among alternative candidates will be studied.

Acknowledgements

This research was supported by the Bisa Research Grant of Keimyung University in 2004.

References

[1] OMA, http://www.openmobilealliance.org, accessed in July (2014)
[2] OMA DM Protocol Specification, http://www.openmobilealliance.org, accessed in July (2014)
[3] J. Pak and K. Park, "Advanced Pulse Oximetry System for Remote Monitoring and Management", Journal of Biomedicine and Biotechnology, vol. 2012, Article ID 930582, 8 pages, doi:10.1155/2012/930582 (**2012**).
[4] J. Pak and K. Park, "UbiMMS: an ubiquitous medication monitoring system based on remote device management methods", Health Information Management Journal, vol. 41, no. 1, (**2012**), pp. 26-30.
[5] K. Park and J. Park, "A Collaborating Agents for a Device Management System", International Journal of Software Engineering and Its Applications, submitted (**2014**).
[6] Wikipedia, http://en.wikipedia.org/wiki/Agent, accessed in July (2014)
[7] M. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice", Knowledge Engineering Review, vol. 10, no. 2, (**1995**), pp. 115-152.
[8] S. Yu and Jun Ai, "Software Test Data Generation Based on Multi-Agent", International Journal of Software Engineering and Its Applications, vol. 4, no. 3, (**2010**).
[9] R. O. Legendi and L. Gulys, "Agent-based dynamic network models: Validation on empirical data", Advances in social simulation, Springer, (**2014**), pp. 49-60.
[10] H. Du, S. Li and S. Ding, "Bounded consensus algorithms for multi-agent systems in directed networks", Asian Journal of Control, vol. 15, no. 1, (**2013**), pp. 282-291.
[11] W. Dargie and C. Poellabauer, "Fundamentals of wireless sensor networks: theory and practice", John Wiley and Sons, (**2010**).
[12] D. Silva, M. Ghanem and Y. Guo, "WikiSensing: An Online Collaborative Approach for Sensor Data Management", Sensors, vol. 12, no. 12, 13295. doi:10.3390/s121013295 (**2012**).
[13] K. Park, "An XML Based Communication System for a Ubiquitous Game Simulator", International Journal of Smart Home, vol. 7, no. 6, (**2013**), pp. 367-376.
[14] W. Stallings, "Cryptography and Network Security: Practice and Principles", 4th. ed., Prentice Hall (**2006**).
[15] J. Pak and K. Park, "Advanced Pulse Oximetry System for Remote Monitoring and Management," Journal of Biomedicine and Biotechnology, vol. 2012, Article ID 930582, 8 pages, doi:10.1155/2012/930582 (**2012**).
[16] J. Pak and K. Park, "UbiMMS: an ubiquitous medication monitoring system based on remote device management methods," Health Information Management Journal, vol. 41, no. 1, (**2012**), pp. 26-30.
[17] M. O. Balitanas and T. Kim, "Architecture for Automatic Management of ParcTab Ubiquitous Computing," International Journal of Advanced Science and Technology, vol. 15, February (**2010**), pp. 1-12.
[18] K. Park and J. Park, "Design of a Monitoring System for Many Collaboration Agents," Proceedings of Software2013 the second workshop, Jeju (**2013**).
[19] K. Park and J. Park, "Collaborating Agents for a Device Management Systems," International Journal of Software Engineering and Its Applications, vol. 8, no. 5, pp. 23-28, http://dx.doi.org/10.14257/ijseia.2014.8.5.03 (2014)
[20] S. Y. Hong and D. H. Han, "XML Principles and Applications", Hanbit Media Inc. (**2011**).
[21] Y. Song, K. Choo and S. Lee, "Design of Index Schema based on Bit-Streams for XML Documents", International Journal of Software Engineering and Its Applications, vol. 6, no. 4, (**2012**), pp. 131-13.
[22] E. P. Duarte Jr. and F. B. Godoi, "Reliable Content Distribution in P2P Networks Based on Peer Groups", International Journal of Internet and Distributed Systems, vol. 2, no. 2, Article ID:45447, 10 Pages, (**2014**).
[23] Y. Wang, J. bo1 and W. Xu, "Model Research of p2p VoD System based on Fluency," International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 7, no. 2, pp. 153-164, http://dx.doi.org/10.14257/ijsip.2014.7.2.15, (**2014**).

Authors

KeeHyun Park, (khp@kmu.ac.kr) He received his B.Sc. and M.Sc. degrees in Computer Science from Kyungbook National University, Korea, and from KAIST, Korea, in 1979 and 1981, respectively, and his Ph.D. degree in Computer Science from Vanderbilt University, USA, 1990. He has been a professor of Computer Science and Engineering Department at Keimyung University, Korea since March 1981. His research interests include Mobile/Network Communication System, Embedded System and Parallel Processing System.

JongHwi Lee, (dragon8829@kmu.ac.kr) He received his B.Sc. Computer Engineering from Keimyung University, Korea, in 2014. His research interests include Mobile Device Management/Data Synchronization and Personal Health Device Management System.