Device Control Protocol using Mobile Phone

Jong-hyuk Roh and Seunghun Jin

ETRI (Electronics and Telecommunications Research Institute), Korea

jhroh@etri.re.kr

***Abstract*— The capabilities of smart phone is raising the interest of the smart home and home network technologies. These has used smart phone as a remote controller. However, the technologies of each vendor are different, connections between the products of other companies are difficult. We suggest the protocol that can control multiple devices with a single application in the smart phone.**

***Keywords*— Device control, Mobile phone, Smart home services, ubiquitous computing, Home network middleware**

I. INTRODUCTION

With rapid development of smart phone make us to face in front of the boundary of the world of ubiquitous computing [1,2]. In addition to this, consumer electronics manufacturers and telecommunication companies that feel the change of society has released the connected device-related products and smart home services. However, because protocols developed by many vendors are different, there are compatibility issues with the communication protocol. And to control many devices, each application should be installed on the user mobile terminal.

By using the proposed protocol in this paper, the user can control various devices by user's mobile phone via NFC and Bluetooth. And the user interface data is downloaded from the device to the mobile phone. Then it can make it possible to control multiple devices with a single application.

In this paper, we describe the flow of the protocol and the protocol message structure. Throughout the experiment we evaluate the performance of the protocol. In addition, we show that there is a speed improvement by using a method that does not re-send the data of the user interface.

This paper is organized as follows. In section 2, we present the device control protocol, and in section 3, describe the experiment of the protocol and the simulation result. In section 4, we describe the related works. We provide a conclusion in section 5.

II. DEVICE CONTROL PROTOCOL

A. System

The system that uses the device control protocol consists of a client and a server. The app installed in the client (mobile phone) has the function of device control protocol. The server is the one of electronic devices such as electric fan, TV, and audio system. Using a mobile phone, the user controls the electronic device.

After the connection between the mobile phone and the device, the device sends data of user interface to the mobile phone. The app of mobile phone shows the user interface that can controls the device. And according to the user interface guideline, the user controls the device.

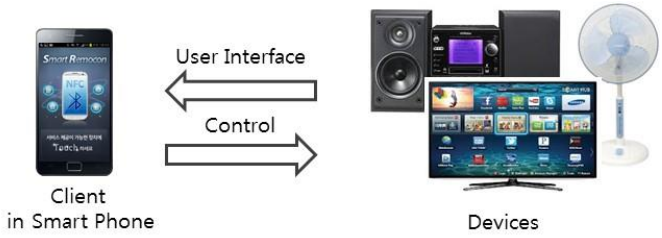


Figure 1. Device Control System

The client consists of the following modules.

- *Connect Manager*: It is responsible for connecting the device. Bluetooth is used for connection. It has 3 methods of Bluetooth pairing; basic pairing, NFC pairing, and directly connecting.
- *Protocol Processor*: It processes the device control protocol.
- *User Interface*: It decodes the data of user interface received from the device. Then it shows the user interface to the screen of mobile phone.
- *Service Device Manager*: It manages the list of devices that were connected. The Bluetooth address was saved in order to easily connect the device next time.
- *Data Base*: It manages the data of user interface received from the device.

The device has the following modules.

- *Connect Manager*: It processes in order to connect the mobile phone.
- *Protocol Process*: It processes the device control protocol.
- *Access Control*: In case that the access control is needed, it requests the authentication to the mobile phone or encrypts and decrypts the messages between them.
- *Service Manager*: It loads the data of user interface that is requested from the mobile phone and passes them to the module of protocol process. Also, it processes the service received from mobile phone.
- *Data Base*: It manages the data of user interface that will be sent to the client.

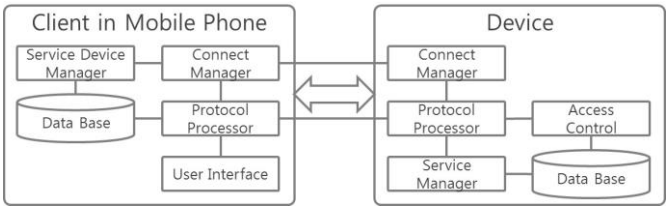


Figure 2. System Architecture

B. Device Control Protocol

Figure 3 shows the flow of the device control protocol.

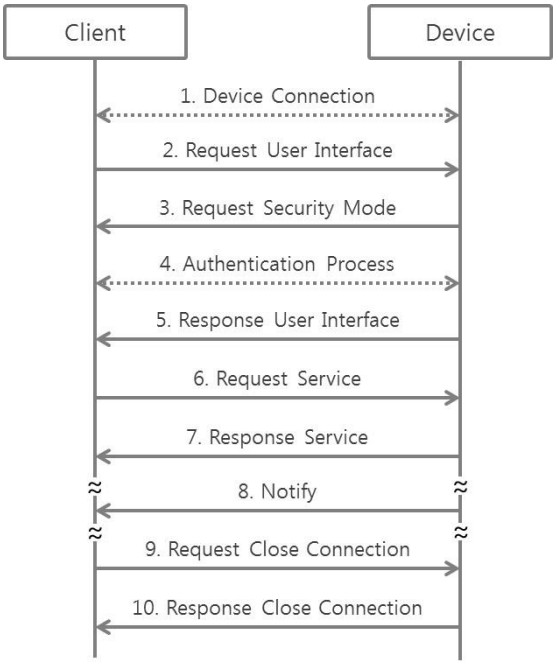


Figure 3. Protocol flow

1. *Device Connection*: The client and the device are connected using Bluetooth. The pairing process is needed for Bluetooth connection.
   - Basic paring: It is a typical method of Bluetooth pairing. Device sets the discoverable mode and the client detects the discoverable device within range.
   - NFC pairing: NFC tag that has Bluetooth address of device is attached on the device. When the mobile phone is touched to the NFC tag, the paring between them is built.
   - Directly connecting: The client has the list of devices that were connected. The user selects the device in the list then the client sends the request message to the selected device.
2. *Request User Interface*: After the connection is established, the client sends a request message for the data of user interface to the device.
3. *Request Authentication*: The device can request an authentication process.
   - Authentication mode: If the device requests an authentication mode, the authentication is done in the phase 4. During authentication, the session key is created. In the following messages, HMAC is included.
   - Encryption mode: If the device request an encryption mode, the authentication is done and the session key is created in the phase 4. The following messages is encrypted by the session key.
   - Normal mode: No security is required, the client and device process the phase 5 without the phase 4.
4. *Authentication Process*: The authentication process is done.
5. *Response User Interface*: The device sends the data of user interface. The client shows a user interface in the screen of the mobile phone.
6. *Request Service*: The user selects the service, which he wish to use, in the user interface. The client sends the message for selected service.
7. *Response Service*: After the requested service is processed, the device sends a result message about success/failure of the service.
   When the user interface is needed to change, the device should send the message of *response user interface* instead of the message of *response service*.
8. *Notify*: The device sends the message about that the client should sends the message of *request user interface* to the device. Most of the flows are the response of the device to the request of client except this message.
9. *Request Close Connection*: Either the client or the device requests closing the connection.
10. *Response Close Connection*: It is the response corresponding to the request for close connection.

C. User Interface

The important feature of the device control protocol is to be able to control more than one device using only on application in the mobile phone. The method that the user interface is transmitted from the device to the application of the mobile phone makes this possible.

HTML is used as the format of the data of the user interface. User interface module in client has the HTML parser and renderer.

The use of a standard web browser is that if the user enters the URL in the address bar, the web browser connects the web server and receives a response message containing the HTML from the web server and then shows that on the screen of the browser. When the user clicks on the hyper link, the web browser sends a HTTP request message to the web server related this hyperlink. Web server sends the HTTP response message corresponding to the request.

Because it does not use the Internet, this system works in a different way with a common web browser. The user requests a service by touching the screen, the touched location has the anchor tag of HTML. Generally, URL about the hyperlink is included in the href attributed of the anchor tag, but in this system, the command string is included

Ex) OKButton.png

When the user touches the OK button, the message of *Request Service* that has the command string, "command3", is sent to the devices.

D. Message

Message consists of a header and a body. In the header, the information such as the type of message, session ID, and number of data is included.

TABLE 1. HEADER FIELD

Field	Size(Octets)	Information
type	1	Message type
msgID	3	Session ID. Identifier for the session
numOfData	1	Number of data in body
lenOfData	3	Length of body

Table 2 shows the type of message.

TABLE 2. TYPE OF MESSAGE

Type	Value
Request User Interface	0x01
Response User Interface	0x02
Request Service	0x03
Response Service	0x04
Request Close Connection	0x05
Response Close Connection	0x06
Notify	0x07
Request Security Mode	0x08
Authentication Process	0x09

Body of the message depends on the message type. In case of *request user interface*, *request close connection*, *response close connection* and *notify*, there is no body. In the body of the *request service* message, command field is included. In the *response user interface* message, the data of the user interface is included. In the body of the message *response service*, status field, which contains the results of the service execution is included.

E. Efficiency

The user interface data of the devices has not been changed frequently. Therefore, the client should be saved and reused user interface data of the device connected once. This method improves sufficiently the speed of user's connectivity.

F. Security

If the device requires security, it is possible to authenticating the user or to encrypt a message. In protocol, the device can select either authentication mode or encryption mode.

We use the SRP (Secure Remote Protocol), RFC 2945[2], as the authentication algorithm. The SRP is an augmented password-authenticated key agreement protocol. It does not require a trusted third party and it is resistant to dictionary attacks. By exchanging data of four times between the client and the server, the SRP processes the authentication and the key exchange.

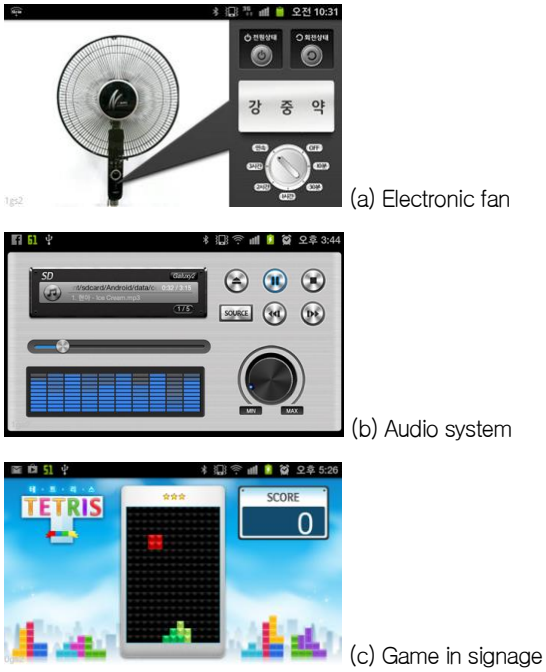
HMAC-SHA256 is used for the message authentication. The key for HMAC-SHA256 is the session key made by the SRP. Message authentication code is appended the end of the body.

3DES is used for the encryption. A message that has no body is not encrypted.

III. EXPERIMENT

A. Environment and simulation

For the experiment, we use android smart phones that support NFC and Bluetooth. The client application is installed in a smart phone. For simulating devices such as an electronic fan, an audio system and a signage system, we use android smart phones and NFC tags. The following images show simulated devices.



(a) Electronic fan

(b) Audio system

(c) Game in signage

Figure 4. Simulated Devices

The following images are the screen images of the client that connected each device. The (a) is the basic client screen. The (b) is the screen of the client that connected the electronic fan. The (c) is the audio system. The (d) is the game.

(a) Basic screen (b) Electronic fan

(c) Audio system (d) Game in signage

Figure 5. User Interfaces

B. Performance Test

Performance test was to measure the speed of connection between the client and the device. The measured time is started when the pairing begins between them, and is ended when the smart phone shows the user interface.

For the test, we checked the time of the process 50 times. In android phone, because the basic processes have worked in the background, the measured error occurred frequently. Especially, the measured value of the error is large in the Bluetooth pairing session.

The following table shows the time of the measurement.

TABLE 3. THE TIME OF THE MEASUREMENT

Process	Time (ms)
NFC connection	43.9
Bluetooth pairing	1872.9
Transfer the interface data (non-save mode)	490~1587
Transfer the interface data (save mode)	138~868

The save mode, described in the section *Efficiency*, is that the client saved and used the user interface data of the device connected once. The efficiency was improved more than 50%.

IV. RELATED WORKS

The device control protocol is associated with the home network technology and the smart home technology in terms of controlling various devices by using a mobile terminal. In this chapter, we shortly explain the technology of the home network middleware technology and the smart home market, and also describe the characteristics as compared with the proposed protocol.

A. Smart home market and technologies

Smart home market began with the spread of the Internet in the late 1990s. The home appliance manufacturers and IT companies have developed the technologies to connect various devices in the home. These technologies make it possible for us to control and use them. However, the market was deflated because of the lack of connectivity between devices of other companies, the problem of role-sharing between vendors and the cost of the deployment.

Recently, smart pad, smart TV, and smart phone is beginning to be widely used. Due to this, the application service, such as smart grid and u-health using a smart device, has been introduced and the smart home market is expected to grow.

Leading manufacturing companies of smart home are Apple, Google, Microsoft, Sony and AT&T. These companies are actively investing in smart home technology under the plan of its own.

In smart home, the user controls the device using a mobile phone or a dedicated remote controller. The home gateway server is used to control devices, rather than a direct connection between the device and the mobile terminal. The home gateway, which is always connected with devices, such as air conditioner, a refrigerator, TV, and so on, makes the user to know the status of home devices and helps to control them [4,6].

And there are problems of the initial configuration of the devices and the user registration process. Because these processes may be difficult to the user who is unfamiliar with the smart home service. Recently, studies for solving these problems is underway [8].

B. Home network middleware

Home network middleware is a protocol for data exchange between devices. The widely used middlewares include UPnP, DLNA, Jini, and HAVi.

UPnP(Universal Plug and Play) is a set of protocols for pervasive peer-to-peer network connectivity of devices and is designed to bring easy to use and flexible [7]. DLNA(Digital Living Network Alliance) is as a de-facto standard for multimedia sharing in the home network [5]. Jini is a java-based technology for dynamically arranging devices on networks and mutually providing device functions [9].

In order to solve the problem of user interface, CEA has announced the CEA-2014 which includes the web technologies and UPnP RUI. Structure proposed by CEA-2014 is a system that uses the server which provides the user interface. The client requests the profile-based user interface to the server.

V. CONCLUSIONS

The proposed protocol, which is the device control protocol, solves the problem of the compatibility of the connection

between a device and a smart phone. This protocol can make it possible to control various devices by using an application in the smart phone. Because it downloads the user interface data from the device through Bluetooth.

In this paper, we presented the message structure and the flow of the protocol. And we evaluated the performance of the protocol throughout the experiment. In addition, it was shown that there is a speed improvement by using a method of reusing the data of the user interface.

REFERENCES

[1] Rafael Ballagas, Jan Borchers, Michael Rohs, Jennifer G. Sheridan, "The Smart Phone: A Ubiquitous Input Device," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 70-77, Jan.-March 2006.

[2] Wu. T. "The SRP Authentication and Key Exchange System", *RFC 2945*, 2000.

[3] "Ubiquitous computing," http://wikipedia.org.

[4] Richard Harper, *Inside the Smart Home*, Springer, 2003.

[5] DLNA, "Digital Living Network Alliance Networked Device Interoperability Guidelines: Expanded," http://www.dlna.org.

[6] GSMA, "Vision of Smart Home the Role of Mobile in the Home of the Future," www.gamaembeddedmobile.com, 2011.

[7] UPnP, "UPnP Technology – The Simple, Seamless Home Network," http://upnp.org.

[8] Eun-Seo Lee et al., "Automating Configuration System and Protocol for Next-Generation Home Appliances," *ETRI Journal*, vol. 35, no. 6, 2013.

[9] Kadowaki, K. et al., "Design and Implementation of Adaptive Jini Systemto SupportUndefined Services," *Proceedings of the 6th Communication Networks and Services Research Conference*, 2008.

Jong-hyuk Roh received the B.E., M.E, Ph.D. degrees in computer engineering from the Inha University, Korea, in 1996, 1998, and 2006. He is currently a senior researcher at ETRI since 2000. His research interests are computer security and privacy.

Seunghun Jin received the B.S. and M.S degrees in computer science from the Soongsil Uiversity, Korea, in 1993 and 1995, and the Ph.D. degree in computer science from the Chungnam national university, Korea in 2004. He is a managing director in the cyber security core technology research department at ETRI. His research interests are computer/network security, PKI, and Identity management.