Automating Configuration System and Protocol for Next-Generation Home Appliances

Eun-Seo Lee, Hark-Jin Lee, Kwangil Lee, and Jun-Hee Park

Home appliance manufacturers have recently been trying to provide smart products capable of various Internet services. For example, one health care manufacturer provides a Wi-Fi-capable scale. Once users register their information with the manufacturer's server, their weight and body fat records are automatically uploaded to the server whenever they measure their weight. The users can then watch and check their weight log easily using a smart device, such as a smartphone or tablet. One of the biggest problems, however, is that the initial configuration of the appliances and the user registration process may be quite complicated to typical users who are unfamiliar with such IT devices. This paper proposes an autoconfiguration system structure and protocol for Internet-capable home appliances, which supports the initial configuration and remote maintenance service of the device with only little user effort. Manufacturers can develop their own information appliances and provide differentiated services using the proposed system and protocol.

Keywords: Autoconfiguration, smart device, home device, internet service.

Manuscript received May 02, 2013; revised July 22, 2013; accepted Aug. 2, 2013.

This work was supported by the IT R&D program of MKE/KEIT, Korea. [I002132, Development of Interoperable Home Network Middleware for settling Home Network Heterogeneity]

Eun-Seo Lee (phone: +82 42 860 4883, eslee@etri.re.kr), Hark-Jin Lee (gausslee@etri.re.kr), Kwangil Lee (leeki@etri.re.kr), and Jun-Hee Park (juni@etri.re.kr) are with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

I. Introduction

In the future, various home appliances will be able to connect to the Internet, just like a PC or smart device, as shown in Fig. 1. Manufacturers can therefore provide smarter products that are capable of various Internet services, such as remote control, configuration, firmware/software updates, and remote maintenance.

Actually, such services are already being deployed in the real world. For example, a sonogram of a fetus taken by an obstetrician can be uploaded to a cloud server, and the mother and her family can see the image using a smart device after a user registration. When a device is connected to the Internet, a device manufacturer can provide various services, improving the competitiveness. However, one of the biggest problems with this service is that the initial configuration process may be very complicated to users who are unfamiliar with such IT devices.

Many difficult procedures are needed for the initial configuration of the device. One of the existing configuration approaches (for example, including an Internet connection and the initial setting) is shown in Fig. 2. First, the user should connect a new device to the PC using a USB cable. The user should then download and install configuration manager software on the PC from the manufacturer's homepage (even this process can be difficult for typical users). After executing the configuration manager, the user can perform the configuration process step by step according to the requirements of the configuration manager [1]. In addition to this process, the user should search and download the related smart-device application from an app store and register user information using this application. The problem is that too many procedures are required, which can be the main obstacle

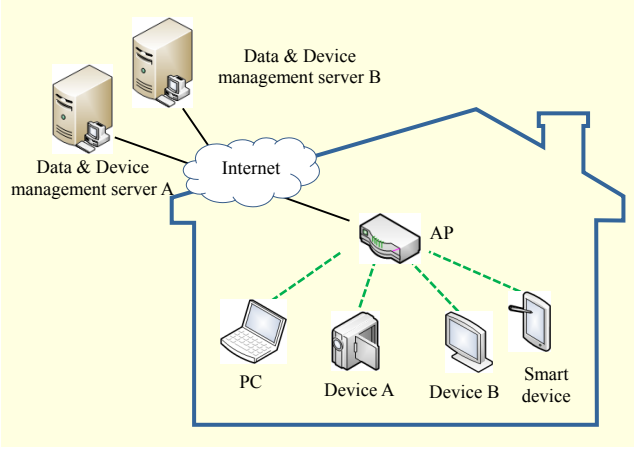

Fig. 1. Future home appliance service environment.

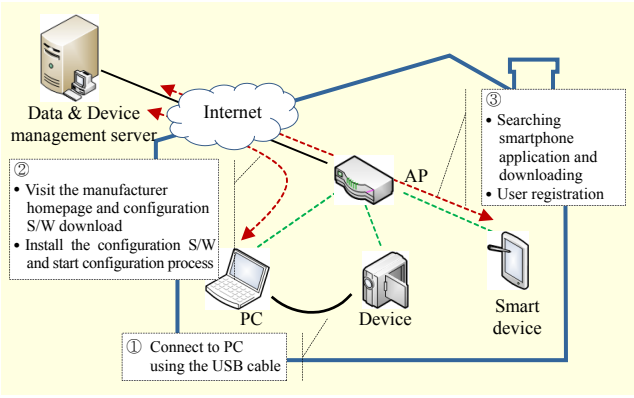

Fig. 2. Example of existing device (no display) configuration method.

in propagating Internet service to home appliances.

This paper presents a device autoconfiguration system structure and protocol to solve these problems. The outline of this paper is as follows. Section II summarizes the related works. Section III describes the device control and management protocol for device autoconfiguration. Section IV describes the automatic configuration system architecture. Section V presents the implementation results of the proposed system and protocol. Finally, section VI provides some concluding remarks regarding our proposal.

II. Related Works

There have been many efforts to support device autoconfiguration. Universal Plug and Play (UPnP) [2] is the most widely adopted interoperability technology that addresses the automatic installation and configuration of the devices and services. However, UPnP is slightly unsuitable to support the management of devices and services through a public network, as the UPnP protocol was initially designed with broadcast networks in mind.

As many home appliances are being connected to the Internet, the UPnP forum has proposed a UPnP Device Management (DM) architecture [3], which is a mechanism used for managing devices on a network. It involves provisioning and configuration services, updating the software/firmware, diagnosing faults, and so on. UPnP DM operations can be used from outside through any remote access protocol using a local proxy gateway from the remote management protocol to UPnP. Such a remote management implementation requires the implementation of a proxy between the UPnP DM and a remote management protocol (for example, customer-premises equipment [CPE] WAN Management Protocol [CWMP] [4], Open Mobile Alliance Device Management [OMA-DM] [5], or Simple Network Management Protocol [SNMP] [6]). However, there are issues in that the proxy gateway is needed for the device management service, and the proxy gateway has the burden of considering various remote management protocols.

The ISO/IEC JTC SC25 WG1 Home Electronic System working group defines the standards for control communication within homes. The scope of this working group includes the control of equipment for heating, lighting, audio/video, telecommunication, security, and any equipment within the home. For this purpose, several device networks, including UPnP and LonWorks [7], are specified. The Home Electronic System also includes residential gateways between the internal Home Electronic System network and external wide-area networks, such as the Internet. However, ISO/IEC JTC1 SC25 WG1 [8]-[10] is targeted only on the device control using home equipment and does not provide device management functions. In addition, since each device network specification is targeted on a specific network and/or on some device types (for example, UPnP is applicable only to IP networks), multiple device networks usually co-exist in a home. It is therefore necessary to implement multiple device networks in a device or provide interoperability among heterogeneous device networks.

The OMA-DM specification is designed for the management of small mobile devices, such as mobile phones, PDAs, and palm-top computers. The device management is intended to support the configuration, software updates, and fault management services. A device may optionally implement all or a subset of these features. Since OMA-DM targets the management function of mobile devices, it does not provide device control functions or the management of wired devices. For the device management, a device must be connected to the server directly. Therefore, all devices are required to be connected to public networks, such as Global System for Mobile Communications (GSM)/CDMA.

The CWMP defines an application layer protocol for the

remote management of end-user devices. As a bidirectional SOAP/HTTP-based protocol, it provides communication between CPE and autoconfiguration servers.

As home appliances are being connected to the Internet and remote management services are becoming required, various remote management methods [11]-[14] have been proposed as a combination of remote management protocols (for example, UPnP, SNMP, CWMP, and OMA-DM). However, there are two different issues between the existing research and the proposed automating configuration method. First, the proposed system considers all initial configuration procedures, including an Internet connection, device registration, and service registration. In addition, this paper proposes a new device control and management protocol that can be operated in a local network and public network simultaneously.

One of the most basic initial configuration procedures of Internet-capable devices is the Internet connection (for example, router or access point [AP] connection) of target devices. In the wireless network environment, however, some devices (for example, non-display devices) are difficult to connect to a secured AP since those devices have no interface for entering a password. For this reason, some appliance manufacturers are trying to solve this initial connection problem [1], [15]. One of the latest methods is to use the Bluetooth interface on smart devices [15]. After the Bluetooth pairing process between a smart device and a target device, users can download a configuration application to the smart device. When an AP connection information is entered using the application, this information can be transferred to the target device via the Bluetooth channel. However, in this case, the target device should have a Bluetooth interface and a Bluetooth pairing process is required additionally.

There is another approach to provide the Internet service of home appliances without using the Wi-Fi interface [16], [17]. Wireless networks, such as ZigBee, Bluetooth, and Wi-Fi networks, are integrated through a common home gateway. The home gateway provides a simple and flexible user interface, and remote access to the system. However, this approach additionally requires a coordinator, which is responsible for starting the ZigBee network and a gateway device supporting the various wireless interfaces for the initial configuration. Interoperability among the appliances is difficult to guarantee since the configuration methods between the ZigBee (or Bluetooth) device and gateway can be diverse according to the manufacturers.

III. Device Control and Management

This paper proposes a new device control and management (DCM) protocol that supports the device autoconfiguration, including the function of Internet connection and device/service registration. In addition, the proposed protocol can cover not only a local network but also a public network. The standardization work of the DCM protocol is also currently underway [18]-[20].

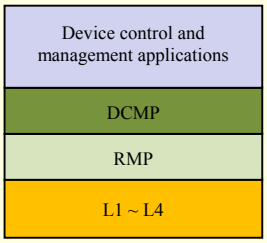

Fig. 3. Protocol stack of DCM.

DCM can support various control and management services, regardless of the network protocols or interfaces. DCM is composed of two protocols: a device control and management protocol (DCMP) and reliable message delivery protocol (RMP). DCM provides various functions for the device control and management. DCM supports the device and network status information retrieval, device and network initialization, firmware and software updates, file transmissions, and so on. In an administrative domain (that is, a network area where a single administrator configures and manages a network with the same policy), there may be a device management server that collects, controls, and manages devices using DCMP. To exchange DCMP messages among the devices, RMP is needed. RMP is a message exchange protocol among the devices, regardless of the network protocols or interfaces. The detailed protocol stack of DCM is shown in Fig. 3.

Basically, DCMP messages can be exchanged using the RMP. RMP has node information, which maintains the mapping information between the DCM device identifier and physical network identifier, such as the IP address and port number in the IP network. If there is a device management server (DMS) in an administrative domain, the RMP might be able to obtain the node information about all devices that are connected in the administrative domain from the DMS. After the RMP retrieves the target node information, the DCMP messages (for example, "device information request" or "device control request") can be transferred to the target device using the RMP.

1. DCM Operation

Each device sends the node advertisement message to the device management server using the RMP. The physical address information on the management server can be installed on the RMP module when the device is manufactured or can

ETRI Journal, Volume 35, Number 6, December 2013
http://dx.doi.org/10.4218/etrij.13.0113.0216

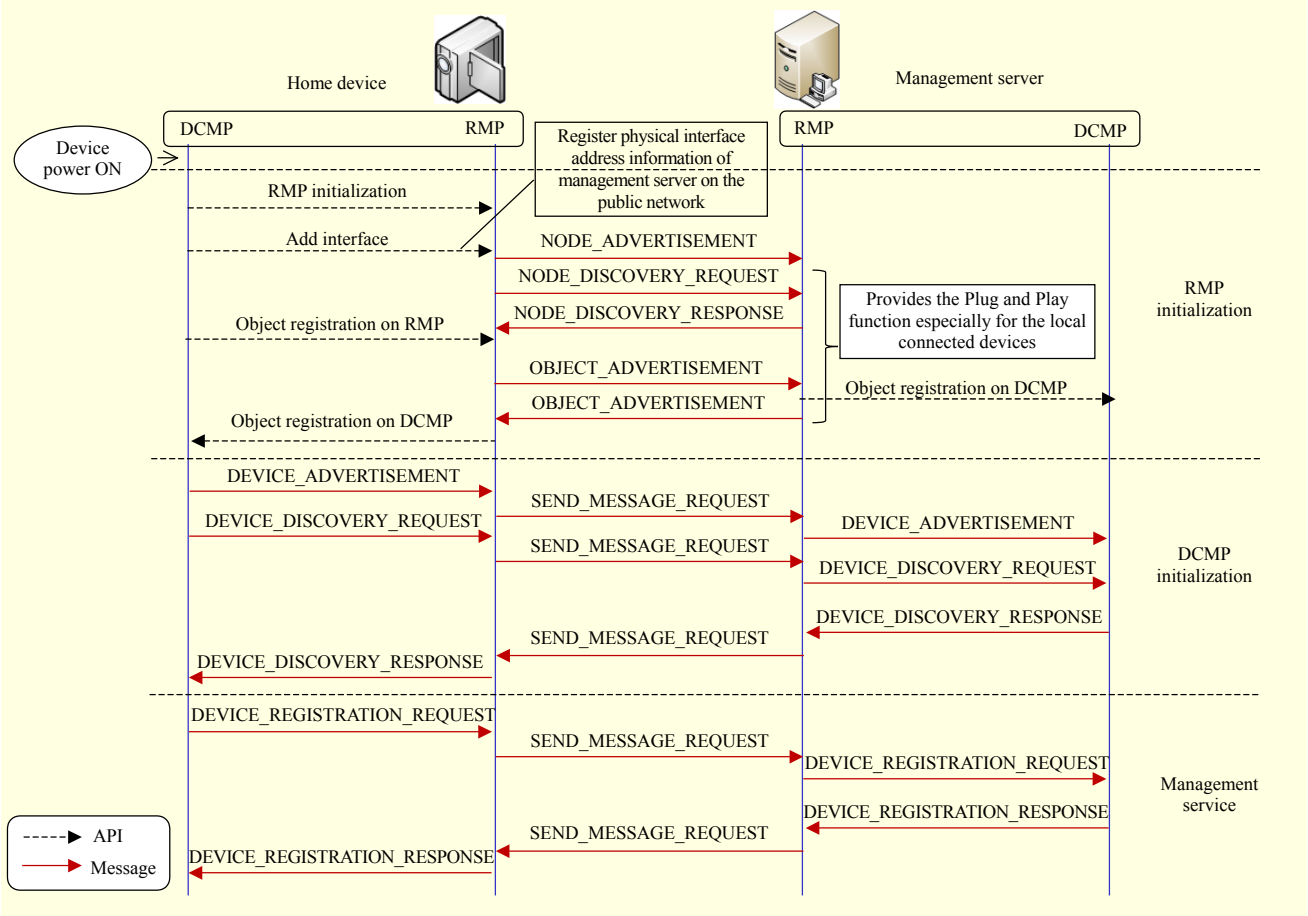

Fig. 4. Example of RMP and DCMP operation.

be added by an API of the RMP module (that is, add_interface). The DCMP also sends the device advertisement message, device registration message, and user registration message to the device management server using the RMP. After these processes, the DCMP is able to upload the user's own data to the device management server using the RMP. An example of a DCM operation is shown in Fig. 4.

The RMP manages the mapping information between the device identifier and physical network identifier. For an advertisement of the physical network identifier information, a NODE_ADVERTISEMENT message is broadcast by the RMP. To advertise the device identifier information, an OBJECT_ADVERTISEMENT message is then broadcast by the RMP. The RMP broadcasts a NODE_DISCOVERY_REQUEST message and receives a NODE_DISCOVERY_RESPONSE message from the other RMP with the OBJECT_ADVERTISEMENT message because the RMP needs to know the node information of the other devices. After these processes, the DCMP can send the message or event to the other devices using the RMP.

The physical address information of the management server can be added in the RMP module of a home device using the API of RMP. The RMP module of the home device sends a NODE_ADVERTISEMENT message using a broadcast to the other devices connected in the local network, but the RMP module of the home device uses a unicast when sending the NODE_ADVERTISEMENT message to the management server. The RMP module of the management server manages the physical address information of the home device using the socket information, that is, the IP address and port information of the AP or router.

2. Device Control and Management Protocol

The number of smart devices has recently increased significantly and is expected to exceed more than 50% of the number of PCs in the near future. Major consumer electronics companies produce smart appliances, which control and manage other appliances. In addition, the smart applications become an important issue, which are developed to control and manage smart devices through the network. However, the control and device functions are different for different device

Table 1. Operations of DCMP.

Operation	Description
Device discovery	When a DEVICE_DISCOVERY_REQUEST message is broadcast, all devices fit into the requested information response, a DEVICE_DISCOVERY_RESPONSE message.
Device advertisement	A DEVICE_ADVERTISEMENT message is used to inform the device's plug-in or plug-out.
Device info	A DEVICE_INFORMATION_REQUEST message is used when a device needs to know the system and network information of the other devices.
Device control	When a device receives a DEVICE_CONTROL_REQUEST message, the device executes the requested control and returns the result.
Event	When some events occur in a device, the event can be reported to all devices by an EVENT message.
Event subscription	Event information should be reported to only interested devices. For this reason, event handling operation includes event subscription/un-subscription operations.
Get file	A GET_FILE_REQUEST message is an essential function for the device management since the software update and firmware update requires the updated file to be transferred to the target device.
Put file	A PUT_FILE_REQUEST message is an essential function for the device management since the software update and firmware update requires the updated file to be transferred to the target device.
Apply	An APPLY_REQUEST message is used when a firmware update, reboot, or configuration is needed in the system.
Device registration	A DEVICE_REGISTRATION_REQUEST message is used when a device registers its own information to the manufacturer server.
Service registration	A SERVICE_REGISTRATION_REQUEST message is used when a user wants to register personal information to the manufacturer server.
Version info	A VERSION_INFORMATION_REQUEST message is used when a device wants to know the latest device version information.

types, capabilities, and device manufacturers. In addition, the network environment for devices is too diverse. Therefore, it is necessary to control and manage devices uniformly, regardless of the device type and underlying network environment. The DCMP is applicable to many different smart applications, such as smart home appliances, e-health, smart cars, and smart works. The objective of the DCMP is to define a common device control and management protocol for various smart devices. The operations of the DCMP are described in Table 1.

3. Reliable Message Delivery Protocol

The RMP is a protocol for the message exchange among

Object ID	Node ID	MP Type	MC Type	MAddress	MPort	SP Type	SAddress	SPort
0005	000H	UDP	Uni	214.31.5.2	Fixed	UDP	214.31.5.2	Fixed
0002	000B	UDP	Broad	Fixed	Fixed	UDP	192.168.0.8	Fixed
0003	000C	UDP	Uni	19.25.8.5	Fixed	UDP	19.25.8.5	Fixed
0004	000D	TCP	Uni	29.`25.8.5	Fixed	TCP	29.`25.8.5	Fixed

Fig. 5. Example address translation table of RMP.

devices. Since each device is connected to a different network, the data transport and network protocols are also different. In addition, some devices are connected directly with different interfaces. Therefore, the RMP provides a uniform and reliable message exchange protocol among devices, regardless of the network protocols or interfaces. Each node has an address translation table, which maintains the mapping information between the device identifier (that is, object ID) and physical network identifier (that is, node ID and IP address in an IP network). An example of an address translation table is shown in Fig. 5, and a description of each field is as follows.

MPType is a protocol type for the multi-target message reception (that is, UDP or TCP), MCType is a casting type for multi-target message reception (that is, unicast, broadcast, or multicast), MAddress is a network address for multi-target message reception, MPort is a network port for multi-target message reception, SPType is a protocol type for single-target message reception (that is, UDP or TCP), SAddress is a network address for single-target message reception, and SPort is a network port for single-target message reception.

When a node receives an initialization signal by an API, that node sends a NODE_ADVERTISEMENT message using the broadcast address. Node information, such as the IP address, port number, and so on, is included in the NODE_ADVERTISEMENT message. When an object is registered in the node, that node should send the OBJECT_ADVERTISEMENT message, which includes the information of the object ID and application type. After sending the OBJECT_ADVERTISEMENT message, that node can send a NODE_DISCOVERY_REQUEST message using the broadcast address. If some nodes receive a NODE_DISCOVERY_REQUEST message, these nodes should send a NODE_DISCOVERY_RESPONSE message to the node that sent the NODE_DISCOVERY_REQUEST message. After sending the NODE_DISCOVERY_RESPONSE message, that node should send an OBJECT_ADVERTISEMENT message if that node has some objects. After these processes, each node can send an application message or event to the other nodes using an API. The operations of the RMP are described in Table 2.

ETRI Journal, Volume 35, Number 6, December 2013
http://dx.doi.org/10.4218/etrij.13.0113.0216

Table 2. Operations of RMP.

Operation	Description
Node advertisement	When a network is enabled or reconfigured, then a node advertises its presence.
Node discovery	Node discovery is used when a device wants to discover nodes in the network.
Send message	Send Message operation is used to send a DCMP message to the other node.
Send event	Send Event Message is used to send a DCMP Message to the other node without any response.
Object advertisement	Each node has an address translation table, which maintains the mapping information between device identifier and physical network identifier such as IP address in IP network. So when a device is added in the node, that node should advertise that fact.

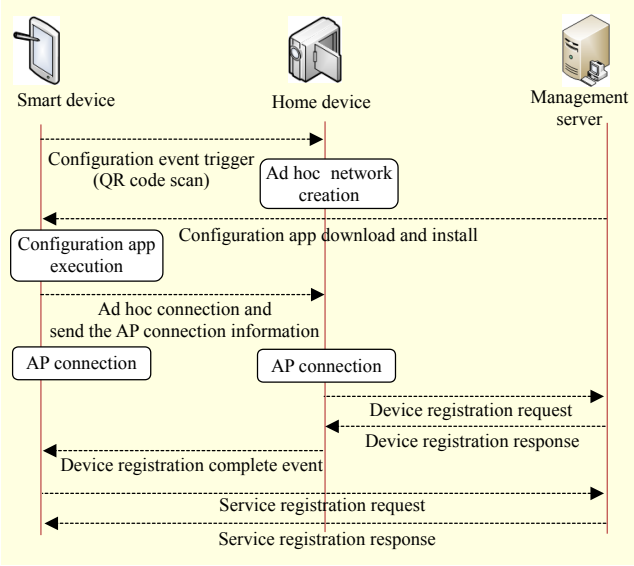

Fig. 6. Automating configuration service procedure.

IV. Automating Configuration System Architecture

This paper proposes an autoconfiguration system structure and protocol for Internet-capable home appliances, which supports the initial configuration of the device, such as the Internet connection and device registration, with only little user effort.

1. Automated Configuration Procedure

The proposed automated configuration service procedure is shown in Fig. 6. The user scans a Quick Response (QR) Code on the new device with a smartphone, and the configuration application can then be downloaded and installed. When the power of the device is turned on, the device creates an ad hoc network using the promised SSID (for example, DCMPAdhoc).

After executing the configuration application on the smartphone, the application connects to the ad hoc network. The user can then input the AP connection information (that is, SSID and password) and user information (that is, username, account, e-mail address, and so on) for the Internet service. The AP connection information and user information are delivered to the target device using the DCM protocol, and the device and smartphone are then connected to the Internet through an AP.

The target device performs the device registration process using the DCM protocol. The device sends a device registration request message to the management server with the device information (that is, device name, device ID, serial number, and so on, and this information can be set when the device is manufactured). After receiving the device registration response message from the management server, the device sends the complete device registration event message to the smartphone. Finally, the smartphone application sends a service registration request message to the management server and receives the service registration response message.

2. System Architecture

The system function block of the auto device configuration system is shown in Fig. 7, and additional details of each function block are shown in Table 3. The DCMP messages (that is, messages for the device configuration, control, and management) can be delivered to the target device by the RMP module.

According to the DCMP message type, the message can be handled by each manager (that is, device manager, control manager, event manager, apply manager, and so on). When the DCMP module creates the DCMP message, each manager retrieves various information from the application according to each function, and this information is delivered to the message creator. After the message creator creates the appropriate message header and payload (that is, DCMP message), that message is transmitted to the target device using the RMP module.

When the RMP module receives the DCMP message, the RMP sends that message to the message parser of the DCMP module. The message parser analyzes the DCMP message header and delivers the DCMP payload message to the appropriate manager (that is, the device manager, control manager, event manager, apply manager, and so on) according to the DCMP message type in the header. Each manager then performs its own function, such as device control, service

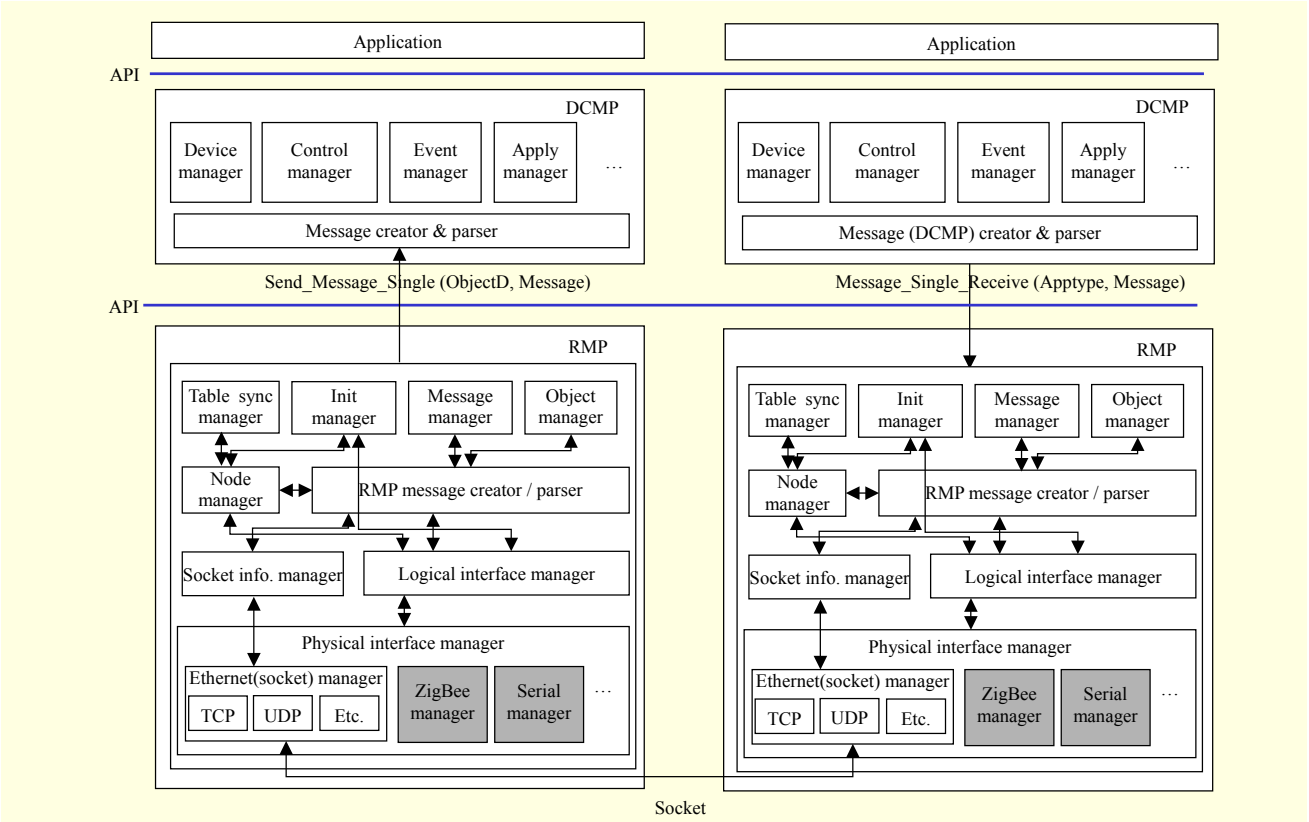

Fig. 7. Automated configuration system architecture.

Table 3. Description of automated configuration system function block.

Function block	Description
Device manager	Device management module through the device advertisement, device discovery request/response, device information request/response message
Control manager	Creating the device control request message or actuating the device with the received device control request message
Event manager	Creating the event message or actuating the device with the received event message
File manager	Processing module for the file transmission function
Apply manager	Processing module for the firmware update function, file execution, and rollback with the applied message
Message (DCMP) creator/parser	After retrieving various device information for the device management, creates a DCMP message or parses the received DCMP message
Table sync manager	Performs the synchronization of routing information, which is managed by a logical interface manager
Init manager	Module for the initialization of the RMP module
Massage transmission manager	Performs the message transmission according to the routing table information
Object manager	Management module of the device identifier
Node manager	Management module of the routing information
RMP message creator/parser	After retrieving various node information for the network management, creates the RMP message or parses the received RMP message
Socket information manager	Extracts the IP address and port information (i.e., AP information) of home appliances that are trying to connect the management server in the public network.
Logical interface manager	Makes and updates a routing table
Physical interface manager	Manages various physical interfaces, such as TCP/IP, RS485

http://dx.doi.org/10.4218/etrij.13.0113.0216

execution, firmware updates, and so on.

V. Implementation and Result

This paper described the implementation of the DCM protocol and prototype system (that is, home device, smart device application, and management server) to verify the proposed automated configuration system. The testbed environment, implemented device structure, service scenario, and implementation results can be described as follows.

1. Testbed Environment

The environment of the implemented automated configuration system is shown in Fig. 8. The target home device is a coffee machine, which provides the ability to adjust the concentration of coffee, the amount of water, and the type of coffee (that is, Americano, latte, or espresso). A Linux (ARM)-based bridge device is implemented for the DCM protocol operation, and this device provides Wi-Fi and a physical RS232 interface. For communication between the bridge device and coffee machine, a printed circuit board (PCB) (that is, micro-computer) capable of RS232 serial communication is implemented and installed in the coffee machine. The smart device application (that is, performs the initial configuration, remote control, and Internet service through the DCM protocol) is implemented on a tablet PC with the Android OS, and the management server is implanted based on Linux. The management server has a unique public network address and provides device registration, service registration, control information retrieval (that is, information on the user's coffee consumption), and information service according to the user's coffee consumption pattern.

2. Implemented Devices

The system structure of the target device (that is, a coffee machine) is shown in Fig. 9. The bridge device is implemented for the DCM protocol operation, and this device has Wi-Fi and an RS232 interface. The bridge device receives various DCMP messages from the smart device application or management server and controls the target home device through the RS232 interface. A PCB is installed in the coffee machine for the RS232 serial communication. This PCB receives the RS232 signal from the bridge device and controls the coffee machine. In addition, it sends an internal signal (for example, various device events, such as water existence, coffee bean existence, filter status, and so on) to the bridge device through the RS232 interface.

A screenshot of the implemented smart device application is

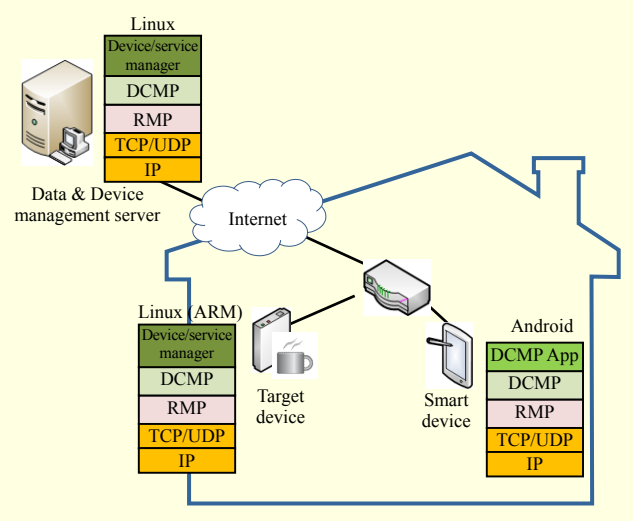

Fig. 8. Implemented automating configuration system environment.

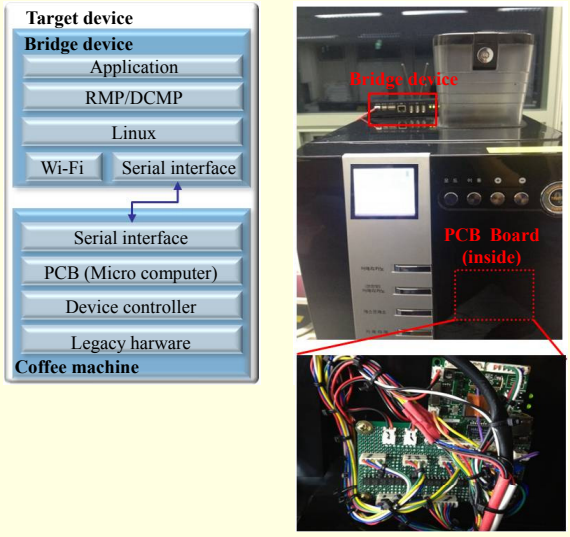

Fig. 9. Implemented home device (coffee machine).

shown in Fig. 10. The smart device application is implemented based on the Android OS platform and supports the DCM protocol. The major function of this application is the initial configuration, remote device control, and Internet information service. Users can select the desired function on the left menu.

Users can also input AP connection information (that is, SSID name and password) for the initial configuration and user information (that is, user's name, user account, password, e-mail address, and so on) for the service registration. When the user fills in those fields and touches the start configuration button, the Internet connection, device registration, and service registration are performed automatically.

Users can control the concentration of coffee, the amount of water, and the type of coffee using this application after the user registration has been completed. These user preferences can be

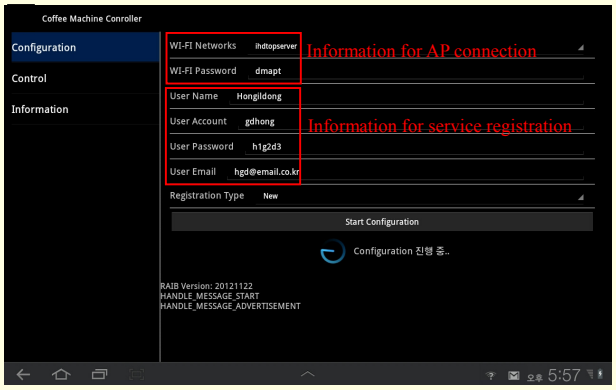

Fig. 10. Implemented smart device application for initial device configuration.

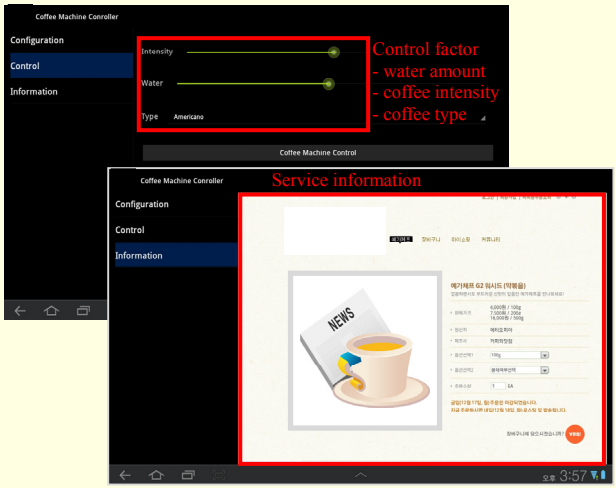

Fig. 11. Control and service screen of implemented smart device application.

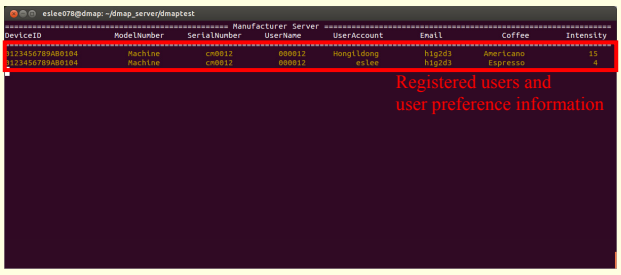

Fig. 12. Screenshot of implemented management server.

uploaded to the management server through the DCM protocol, and the management server can provide the coffee information (for example, information on the coffee beans and coffee bean purchases) to suit the user's tastes using the smart device application. The control and service screen of the smart device application is shown in Fig. 11.

Table 4. Required processing time for each procedure.

	DCM initialization	AP connection	Device registration	Service registration
Management server	1.2 sec	-	1.5 sec	2.1 sec
Home device	2.4 sec	3.2 sec	1.5 sec	-
Application	2.2 sec	4.7 sec	-	2.1 sec

Table 5. Comparison of experiment results.

	Proposed system	Commercial system 1 [1]	Commercial system 2 [15]
Network configuration time	33.0 sec	More than 3 min	More than 1 min
Total configuration time	37.0 sec	More than 5 min	More than 90.0 sec
Secured AP connection	Support	Support	Support
Additionally required device	Smart device	Smart device, PC	Smart device (with BT)
User intervention	5 times	11 times	7 times
Control latency	0.7 sec	Less than 2.0 sec	Less than 2.0 sec

The DCM protocol operation module is installed in the management server, which is connected to a public network, and the management server can perform the device and service registration. In addition, the management server retrieves the user's coffee consumption information and provides adapted coffee information to the user. The registered user information, device information, and user preference information are shown in Fig. 12.

3. Implementation Results

The required processing time is measured for each procedure of the proposed automated configuration system. It takes 5.6 seconds and 6.9 seconds for the DCM module initialization and AP connection at the home device (that is, coffee machine) and smart device application, respectively. It takes 3.6 seconds for the device registration and service registration. The results of the measured processing time are described in Table 4.

In this paper, device initial configuration procedures include the Internet connection, device registration, and service (that is, user) registration. A comparison of experiment results for several initial configuration systems, including those of recent commercial products, is shown in Table 5. "Network configuration time" refers to how long it takes before a target device is connected to the Internet after the power of the target

ETRI Journal, Volume 35, Number 6, December 2013
http://dx.doi.org/10.4218/etrij.13.0113.0216

device is turned on. In the case of commercial system 1 [1], it takes several minutes because the system uses a PC for the Internet connection of the target device. "Total configuration time" refers to how long it takes to perform all initial configuration procedures (that is, Internet connection and device and service registration). The proposed system only takes about 37 seconds to complete all initial configuration procedures. The procedure for the secured AP connection is considered in all the systems. The proposed system and commercial system 2 [15] need only one additional device to complete all initial configuration procedures. "User intervention" refers to how many times the user intervenes during the initialization process. The proposed system requires that the user intervene five times: scanning QR Code, installing an application, executing the application, entering AP connection information, and entering user information. In the case of commercial system 1, the user must intervene 11 times: turning on the PC power, connecting a USB cable, visiting a manufacturer website, registering user, downloading, installing and executing a configuration manager, entering AP connection information, searching, downloading, and executing a related smart device application. In the case of commercial system 2, the user must intervene seven times: Bluetooth pairing initialization, searching for the Bluetooth device, connecting the Bluetooth device, installing and executing an application, registering user, and entering AP connection information. "Control latency" refers to the delay time during the information upload or device control after finishing the initial configuration process. The proposed system performs well in all the features.

VI. Conclusion

Because of the difficulty in the initial configuration process, device manufacturers who try to propagate Internet-capable devices are facing difficulties. This paper proposed an autoconfiguration system structure and protocol for Internet-capable home appliances, which supports the initial configuration and remote maintenance service of the device with only little user effort. In addition, this paper verified the proposed automating configuration system by implementing the proposed protocol and prototype system.

Security services may be necessary according to the application environments in which the proposed system is applied. However, network security was not considered in this paper. The proposed system may suffer from many network-specific threats. To countermeasure those threats, some security mechanism should be deployed in the future.

The proposed system is able to solve problems faced by manufacturers and provide a new service environment in which manufacturers can provide differentiated services with their own information appliances. In addition, this research provides useful guidelines for the implementation of home appliances for applications of an automated configuration system.

References

[1] Withings, *Wi-Fi Body Scale — Quick Start Guide V 7.0*, Mar. 2012. http://www.withings.com

[2] UPnP Forum, *UPnP Device Architecture 1.0*, Apr. 2008.

[3] UPnP Forum, *UPnP DM BasicManagement 1.0*, July 2010.

[4] DSL Forum, *CPE WAN Management Protocol*, technical report, TR-069, May 2004.

[5] OMA, *OMA Device Management V1.2 Approved Enabler*, Feb. 2007.

[6] W. Stallings, *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*, 3rd ed., Addison Wesley, 1998.

[7] ECHELON, *Introduction to the LonWorks System 1.0* (*078-0183-01B*), 2009.

[8] CD text of ISO/IEC 30100-1, *Home Network Resource Management — Part 1: Requirements*, Jan. 2012.

[9] CD text of ISO/IEC 30100-2, *Home Network Resource Management — Part 2: Architecture*, Jan. 2012.

[10] CD text of ISO/IEC 30100-3, *Home Network Resource Management — Part 3: Management Application*, Sept. 2012.

[11] C.G. Park et al., "NAT Issues in the Remote Management of Home Network Devices," *IEEE Netw.*, vol. 22, no. 5, 2008, pp. 48-55.

[12] A.E. Nikolaidis et al., "Automating Remote Configuration Mechanisms for Home Devices," *IEEE Trans. Consum. Electron.*, vol. 52, no. 2, 2006, pp. 407-413.

[13] T. Cruz et al., "Using UPnP-CWMP Integration for Operator-Assisted Management of Domestic LANs," *IEEE Int. Consumer Commun. Netw. Conf.*, Las Vegas, NV, USA, Jan. 2012, pp. 73-78.

[14] H. Rachidi and A. Karmouch, "A Framework for Self-Configuring Devices Using TR-069," *IEEE Int. Conf. Multimedia Comput. Syst.*, Ouarzazate, Morocco, Apr. 2011, pp. 1-6.

[15] Withings, *Smart Body Analyzer — Quick Installation Guide V 1.1*, Mar. 2013. http://www.withings.com

[16] K. Gill et al., "A Zigbee-Based Home Automation System," *IEEE Trans. Consum. Electron.*, vol. 55, 2009, pp. 422-430.

[17] C. Park, *Apparatus for Managing Home-Devices Remotely in Home-Network and Method Thereof*, US Patent App. 10/901,169, Sughrue Mion PLLC, Washington, DC, 2004.

[18] CD text of ISO/IEC 17811-1, *Device Control and Management — Part 1: Architecture*, Jan. 2013.

[19] WD text of ISO/IEC 17811-2, *Device Control and Management*

— *Part 2: Specification of Device Control and Management Protocol*, Sept. 2012.
[20] WD text of ISO/IEC 17811-3, *Device Control and Management — Part 3: Specification of Reliable Message Delivery Protocol*, Sept. 2012.

Eun-Seo Lee received his B.S., M.S., and Ph.D. degrees in electrical and electronics engineering from Chung-Ang University, Rep. of Korea, in 2003, 2005, and 2008, respectively. Since 2009, he has been a researcher at ETRI, where he has worked on home network middleware, especially device control and management. Also, he has participated in standardization activities on ISO/IEC JTC1 SC6. His current research interests are smart home appliances, device autoconfiguration, device control, and management systems.

Hark-Jin Lee received his B.S. and M.S. degrees in computer science from Chung-Ang University, Rep. of Korea, in 2005 and 2007, respectively. He is a researcher in the Green Computing Research Department at ETRI, where he develops home network middleware. His research interests include home network middleware, the Linux system, and embedded computing.

Kwangil Lee received his BS, MS, and Ph.D. from the Dept. of Computer Science at Chungnam National University in 1993, 1996, and 2001, respectively. From 2000 to 2002, he worked as a guest researcher at the National Institute of Standards and Technology (NIST), USA. Then, he worked as a research associate at the University of Maryland (2002-2004) and the University of Texas (2005), USA. Since 2006, he has been a senior researcher for the IT Convergence Middleware Team at ETRI, Rep. of Korea. His research interests include home networks, smart ship, security/privacy, QoS, routing, power-line communication, and wireless/mobile networks.

Jun-Hee Park received his B.S., M.S., and Ph.D. degrees in computer science from Chungnam National University, Rep. of Korea, in 1995, 1997, and 2005, respectively. He was a researcher at the System Engineering Research Institute from 1997 to 1998, where he worked on network computing and clustering systems. From 1998 to 2009, he was a senior researcher at ETRI, where he worked on home network middleware, especially the interoperability framework. Since 2010, he has been the team leader of the Emotion-IT Convergence Middleware Research Team. He has conducted research in the area of ship and ICT convergence and has developed ship area network technology. His current research interests are smart home and smart ship.

ETRI Journal, Volume 35, Number 6, December 2013
http://dx.doi.org/10.4218/etrij.13.0113.0216