Design and Implementation of the HNCP-UPnP bridge using a Virtual device

Joseph Jeon†, J.M. Lee†, K.J. Myoung†, K.R. Lee†, W.H. Kwon†, and B.S. Ko‡

†Control Information Systems Lab.
School of Electrical Engineering and Computer Science
Seoul National University
Seoul, 151-742, Korea
Phone: +82-2-873-2279, Fax: +82-2-878-8933
E-mail: joseph@cisl.snu.ac.kr

‡Digital Appliance Research Lab.
LG Electronics Inc., Korea

Abstract

This paper describes a design and implementation of the HNCP-UPnP bridge using a virtual device. HNCP networked devices targeted to white goods are allowed to be controlled and monitored by UPnP Control Point. The bridge provides automated creation of a UPnP virtual device and message conversion. The developed system enables integration of a heterogeneous home network.

1. Introduction

A home network system is a collection of networked home appliances, which are considered to be distributed embedded systems connected by networks[1]. Not only whites goods, but also all sorts of devices such as PCs, mobile phones, cameras, PDA, and etc. are increasingly connecting to networks. They are using a multitude of connectivity methods. As there is no dominating technology or standard in a market, it is expected that two or more home network technologies or standards are used to implement the home network. Lack of gateways to the internet and different communication overhead allowed by devices also facilitate heterogeneity of the home network. This implies that even middleware protocol, which guarantees communication among devices with different physical media, may not exist alone. The heterogeneous home network system requires a bridge. The bridge connects two or more network technologies. Users can use not only all devices with different protocols or media, but also user several kinds of services that each standards support. In this paper, a protocol based on a power line communication (PLC) and a middleware protocol is considered to constitute the home network and the bridge that connects these two protocol is designed and implemented.

The power line has bad condition for communication: heavy noise and attenuation and hard impedance matching. In spite of these defects, technological advances and many outlets lead developers to implement a PLC module on controllable devices which require low communication overhead. HNCP which is targeted to home appliances based on low speed PLC is studied and implemented. HNCP does not support gateway in its standard. So the technology that is connected to the internet and covers different kinds of devices in the home can increase the efficiency of the home network.

Furthermore, previous works on the bridge and interoperability system have a limit of additional supports for the newly added service type. It requires administrator to program new modules and update for each services[2]. The implemented bridge reduced the labor of additional coding by implementing a virtual device creator.

This paper is organized as follows. An overview of HNCP is showed in Section 2. Section 3 presents brief introduction to UPnP. A design of the bridge is described in Section 4. Section 5 shows an experiment that shows the heterogeneous home network using the bridge. Finally, conclusions are presented in Section 6.

2. HNCP

HNCP is a communication protocol based on the low speed PLC[3]. HNCP is aimed to control home appliances. It is composed of 4 layer to reduce an overhead of the communication packet. Target networked home appliances are assumed to be operated on an 8-bit microprocessor. Multi master structure is supported so that user can use devices from multiple masters. Device address of HNCP, which is 2 bytes long, is separated into two group. The first byte announces defined device type and the second byte logical address or location address of the device. This address structure makes an easy initializing operation. Furthermore, a supported standard message set facilitates the inter-operability between home appliances. The message
set includes all the defined services and actions of each devices. HNCP is targeted to the indoor network, hence techniques that implement gateways are required to access via the internet. HNCP layer architecture is showed in Figure 1.

Figure 1: HNCP layer architecture

In [4], the home server that controls, monitors, and manages HNCP networked devices was developed. The home server receives a device description file from users or internet resources. Then it organizes a device data base and prepares user interface. In this paper, the implemented bridge get the device description file from this server.

3. UPnP

UPnP is specification for service presentation and discovery published by UPnP forum. UPnP aims to establish a communication standard based on XML. The UPnP specification[5][6] explains addressing, service advertisement, discovery, control, eventing, and presentation. To support these functions, legacy protocols are used as followed.

- **Auto IP**: In case of absence of a DHCP server, the device presents IP address actively.
- **SSDP**: Simple Service Discovery Protocol is used for service advertisement and discovery.
- **SOAP**: Simple Object Access Protocol. After discovery of services, UPnP devices use SOAP to control and to be controlled remotely based on XML and HTML.
- **GENA**: Generic Event Notification Architecture. Based on HTML, GENA notifies an UPnP subscriber event.

Figure 2 shows an UPnP architecture implemented on Linux system. Presentation URL, control URL, and event URL have to be included in XML document. The document also provides URL that has information of device’s supporting services. Control Point (CP), that can be implemented as stand alone type or embedded type, searches a service advertisement. Catching the advertisement, CP receives device description information through SSDP then allows users to control or monitor the device by SOAP.

It is hard to implement the UPnP network in real, because there are not so much devices that support the UPnP technology. [1] implemented the UPnP home network using UPnP emulators and a UPnP home server ported on embedded system. The UPnP home server enables users to control and monitor several UPnP devices in a sight and access via the internet, as well. The bridge not only provides inter-operability between UPnP and HNCP, but also helps users to access to HNCP network via the internet through the UPnP home server.

4. Bridge design

The bridge receives information of HNCP networked devices and makes UPnP virtual devices. During initialization step, the bridge set up its HNCP properties by communicating with the HNCP home server. After the set up, the bridge requests a connected device list and receives the description file one by one. Parsing received files, it compares the command code from the HNCP document with that of a message map. The message map contains a defined message match table between HNCP and UPnP. Reading the UPnP action name and properties, the bridge writes down UPnP documents required to create and publish the virtual device. Newly added device in the HNCP network is announced to the bridge by the HNCP home server and the matched virtual device is created following the sequence.

4.1. Bridge modules

To implement the HNCP-UPnP bridge, software modules are divided into three parts. A HNCP interface, a UPnP
The HNCP interface implements 4-layer communication protocol of the HNCP standard. HNCP defines 3 types of interface between a PLC modem and an embedded system. The bridge system uses a D-type modem so the application layer and the network layer should be implemented. Communication interface to the PLC modem have to follow HIP defined in the HNCP specification.

The UPnP interpreter includes an UPnP stack that advertises services, notifies events, and processes control messages. To do these, legacy TCP/IP, SSDP, GENA, and SOAP protocols are used with the XML parser. UPnP accepts all physical media that support TCP/IP.

The bridging module is comprised of 4 functional blocks - a message manager, a document manager, a virtual device creator, and a virtual device manager. Figure 3 shows the functional architecture of the bridge. The message manager keeps defined message set and compares it to the received specified message set. If received message is not defined in the message map, the manager converts HNCP messages using HNCP service codes and command codes. The manager gathers all the services and returns them to the virtual device creator.

The virtual device creator receives the HNCP device description document then parses the document and converts it to the UPnP type. When the message manager returns converted result, the creator writes down UPnP documents and creates the UPnP virtual device from the documents. The created device is published and then managed by the virtual device manager.

The virtual device manager basically converts UPnP messages to HNCP messages and vice versa. Received messages are compared with that of the message map. Finding pair of the message, the manager sends control commands from the UPnP domain to the HNCP domain and response or event message from the HNCP domain to the UPnP domain. The manager always look out whether the device is disconnected or not. In case of disconnection, the manager deletes information of the device and ignores requested converting messages.

Finally, the document manager keeps created UPnP documents: device description, service description, and presentation page.

### 4.2. Message matching

The virtual device creator and manager refer the message map to compare the UPnP message with the HNCP message. The message map is made out based on the HNCP message. The HNCP message is composed of a service code, a command code, and arguments. The service code is selected from "Write", "Read", "Notification", and "Response". Each command code represents services that device supports. Combining these two sets and arguments with various type, users can create all the commands that are required in real device. In UPnP service description, however, each action name represents a action. As shown in Figure 4, the command "Power" in the HNCP message set is directly matched to the "SwitchPower" in the UPnP device template. For the service "power" with command "0x01" which means "off" to the UPnP message, this information is matched to the "SwitchPowerOff" in the UPnP service description. In case of command "0x00" which means "on", the combination becomes pair of "SwitchPowerOn". If the matched ServiceID does not exist in the UPnP device template like "random" in the HNCP message set, the message map only keeps the HNCP command. The virtual device creator regards it as a undefined service in UPnP, so writes down action name with combination of the service code and the command code such as "OnRandom", "OffRandom", and "ReadRandom". In this case, delivered UPnP command is parsed and divided into two parts: command part of "on", "off", "read" or "write" and service name part. If new service type which is not defined in the UPnP domain is added to the device, the virtual device creator creates the UPnP action names. The virtual device manager refers the message map and do the converting sequence in this manner.

### 5. Experiment

Figure 5 shows experiment using the implemented bridge. The bridge connects the HNCP network and the UPnP network. To organize the HNCP network, HNCP prototype devices - an air-conditioner and a microwave oven - and the HNCP home server implemented on a PC are used. The UPnP network is composed of UPnP emulators and the UPnP home server that takes a role of control point. The bridge module is implemented on a PC to support TCP/IP.
and the HNCP power line modem that uses serial port simultaneously. Figure 6 shows structure of the HNCP-UPnP heterogeneous home network with the bridge. The experiment proves that user can access to the HNCP network through the UPnP control point and control the HNCP device. As it is possible to access the UPnP home server using socket program, external access is allowed to the HNCP network, as well.

6. Conclusions

The HNCP-UPnP bridge that creates a virtual UPnP device of HNCP device using the message map was designed and implemented. The message map keeps matched command set between the HNCP message set and the UPnP device template. Because two technologies support the standard message set, it is easy to manage the message map and update. The experiment shows that the implemented bridge guarantee inter-operability between the HNCP network and the UPnP network. Reducing the time required to receive the HNCP device description file from the HNCP server via the power line is demanded as a future work.

References


Figure 4: Message matching between the sample UPnP device template and the HNCP service description

Figure 6: The HNCP-UPnP heterogeneous home network using the bridge