A Design of Hybrid Appliance Local Network (HALN) Communication Architecture

Hyoung-Jun Park and Dongik Lee *

Kyungpook National University, 80 Daehak-ro, Buk-gu, Daegu 41566, Korea; hjpark40400@gmail.com
* Correspondence: dilee@knu.ac.kr

Received: 24 December 2019; Accepted: 15 January 2020; Published: 16 January 2020

Abstract: Recently, appliance networks have been widely adopted in many home applications. Usually, an appliance network requires a server. However, as the number of network users increases, there is not only the problem of costs due to extension of the server and the increase in power consumption, but also the problem that the functions of appliances are restricted when the connection to a server is unavailable. This paper presents a hybrid appliance local network (HALN) communication architecture to tackle the problems with server-based appliance networks. The HALN architecture is designed to remove and/or minimize the utilization of servers by offering the capability of communicating directly with other appliance products. The proposed architecture can also be integrated with existing server-based communication architectures. The HALN architecture is based on the simple service discovery protocol (SSDP) and HTTP protocol (RESTful HTTP server/client architecture) technologies. The effectiveness of HALN is experimentally demonstrated using a smartphone and a set of Linux-based Wi-Fi modems on which the functions that can be provided by typical appliances are implemented. Using the proposed architecture, the communication reliability is also improved by 1.6% as compared with that of an existing server-based communication architecture.

Keywords: appliances; appliance network; communication network; local network architecture

1. Introduction

Over the last decade, the importance of network-based functions has become very crucial in the wide range of consumer devices and home appliances including TVs, speakers, refrigerators, washing machines, air conditioners, door locks, plugs, and gas valves. The provision of network-based functions by home appliances enables various services for consumers’ living convenience, such as energy-saving, the enhancement of home security, and care for the elderly and children. Network-based functions will ultimately realize smart homes in which all products in the house are connected to a communication network [1,2]. For the products existing in the home to be able to provide network-based functions, diverse technologies should be selected and applied.

First, a communication media should be selected. In the past, network functions were provided using a wired communication media, such as Ethernet and power line communication (PLC). Recently, wireless communication media with a low power RF communication module, such as Wi-Fi, ZigBee, Bluetooth, and Z-Wave, have been widely exploited to provide network functions [3,4].

Second, a home gateway should be selected. Wi-Fi-based networks communicate with the servers installed and operated by product manufacturers or power companies using a wireless router already installed in the home as a home gateway. They are mainly used for consumer devices and appliances because of the merit that there is no need to purchase a home gateway separately. As ZigBee-based networks are mainly applied to home automation products, a home gateway applied with both ZigBee and Wi-Fi is required to provide remote services, such as remote monitoring and remote control [5–7].
However, from the viewpoint of consumers, this may cause the burden that a home gateway should be purchased separately when a home automation product is purchased.

Third, communication protocols should be implemented. In order for home appliances to provide network functions, a communication protocol between products and the home gateway must be defined and implemented, and a communication protocol between the home gateway and servers should be also defined and implemented. In addition, a communication protocol with smartphones, which are typical means by which servers and the consumers can check and control the network functions, should be also defined and implemented [8–14]. In addition, it is extremely difficult to maintain the interoperability between communication protocols implemented on the products that are produced by different manufacturers [15,16]. Researchers are also paying significant attention to the issues of security with communication protocols in order to reduce the risk of external intrusion that can execute the product operations that have not been originally intended [17–20].

Fourth, a server should be constructed. A server plays the role of linking the information transmitted by an appliance product in the home to a smartphone connected to the server so that the consumer, at a remote place, can identify the information. A server also plays the role of an intermediary medium for the remote control of products utilizing a smartphone conversely. Servers are mainly installed and operated by product manufacturers or power companies. In particular, power companies utilize servers as the smart grid to save energy and increase the power grid reserve rate in each household [21,22]. Manufacturers of appliances and power companies that operate servers should always consider both the initial costs for the construction of servers, and the costs for maintenance and management every year. The extra costs to install additional servers may incur as the number of network users increases. Moreover, appliance products equipped with a communication unit cannot communicate directly with each other, but can only communicate indirectly through the server. Thus, they cannot communicate when the server does not exist or the server has stopped operating.

To overcome the problems with a server-based communication network, this paper presents a hybrid appliance local network (hereinafter HALN) that can not only directly support the communications with other appliance products, but also can be integrated with the existing server-based communication architectures. The proposed HALN architecture is designed based on Smart ThinQ®. The effectiveness of HALN is demonstrated using a set of Linux-based Wi-Fi communication modems.

Following the Introduction section, this paper is organized as below. In Section 2, the existing server-based home appliance networks are briefly described as the related work. An overview of the proposed HALN, along with a requirement analysis for communication architecture, is also presented. Section 3 describes the common network modules used for HALN in detail. Section 4 describes the communication protocol in terms of the control, event, and query for the HALN architecture. In Section 5, the proposed approach is evaluated using a set of experimental setups consisting of Wi-Fi modems and a smartphone. Finally, a concluding remark is given in Section 6.

2. Requirement Analysis for Communication Architecture

2.1. Smart ThinQ®

As shown in Figure 1, Smart ThinQ® refers to the server-based communication architecture in which a communication protocol named SA agent (smart appliance agent) communicates between smart appliances and the server installed and operated by LG Electronics. Appliances exchange information with a server using the Wi-Fi modem implemented on each device to provide customers with various services while saving energy, time, and costs. The SA agent is composed of a network connection manager module to support the network connection with a server, an RTI parser module to analyze the real-time interface protocol, and a command manager module to process RTI commands.
2.2. Desired Features of HALN Architecture

The HALN architecture proposed in this paper is designed based on Smart ThinQ® to meet the requirements as follows.

1. The HALN architecture reuses the communication media (Wi-Fi) that is commonly used in existing server-based communication architectures without any separate design or development.
2. Simple service discovery protocol (SSDP), which is part of the universal plug and play (UPnP) configuration, is used to search and connect smart appliances that can be connected on the HALN communication network.
3. The HALN architecture is based on the HTTP protocol that follows the server/client model for data transmission and reception between smart appliances. That is, HLAN uses the RESTful-based server/client structure in detail.
4. The data being transmitted between smart appliance products are defined as follows depending on the characteristics of the products and the services to be offered:
   - Data for remote control of the product;
   - Data for monitoring of the product;
   - Data for sharing the content, such as media;
   - Data used for communication between products.

2.3. Overview of the Proposed HALN Architecture

Figure 2 schematically shows network devices based on the HALN architecture proposed in this paper. The network devices may include diverse products, such as refrigerators, washing machines, cookers, cleaners, and air conditioners, which are connected through the network. There is no restriction on the types of appliance products. The above-mentioned appliance products may communicate with external servers, which may be operated by power companies, appliance manufacturers, or web servers, through a communication modem (i.e., a Wi-Fi modem) and a wireless router (i.e., an access point). In addition, the appliance products also can directly communicate with other appliance products, of which types can be the same or different.
3. Common Modules for HALN Architecture

Figure 3 depicts a typical appliance product that works with the HALN architecture shown in Figure 2. The appliance product consists of a microcontroller (MICOM) of the product, a driving unit controlled by the MICOM, and a communication modem that can communicate with the MICOM and an external server. The driving unit is not limited, but includes a motor, a heater, a lamp, and a valve.

As shown in Figure 3, each appliance product includes a module to enable local network communications with other appliance products. The module offers communication with modems through an interface. That is, each appliance product can directly communicate with other appliance products through its local network communication module even without any separate gateways.
In this case, the local network communication module may communicate using a communication protocol different from that of the modem.

Search Manager: The local network communication module may include a search manager, M-Search Manager, to detect other appliance products. M-Search Manager is a module that can multicast search messages at regular intervals to identify other application products running on the local network. In addition, the local network communication module includes a simple service discovery protocol (SSDP) that may deliver the search messages received from other appliance products to the search manager [23,24]. When the SSDP has received search messages from other appliance products, the SSDP may transmit its service description information (e.g., URL address including IP, port information, etc.) in response to the received search messages. In the case that there is no information on the relevant appliance product in the local network management table, the search manager may transmit search messages aperiodically.

Database Control Manager: The local network communication module additionally includes a database control manager. The search manager may transmit the service description of other appliance products existing in the local network to the database control manager. In this case, the search manager uses the information received from the SSDP of the other appliance products in response to the search messages transmitted by the search manager. The database control manager manages the list and information of the appliance products existing in the local network through the SSDP and the search manager. In addition, when the SSDP has received a response to the search message sent by the search manager, the database control manager may update the information in the local network management table. For example, the database control manager may store information on the other appliance products that respond to the search messages sent by the search manager in the local network management table. The database control manager can also delete information of other appliance products that do not respond to the search messages sent by the search manager from the local network management table. The information stored in the local network management table includes the IPs, port numbers, and all information (e.g., product ID, service list, etc.) of the service description for other appliance products.

RESTful Clients/Servers: The local network communication module also includes RESTful clients and RESTful servers [25,26]. The RESTful clients can send control commands or queries to other appliance products in the local network and receive events as responses from the RESTful servers of the other appliance products. That is, when a RESTful client needs to connect to another appliance product in the local network, the RESTful client connects to the RESTful server of the appliance product. The RESTful server can receive control commands or queries from the RESTful clients of other appliance products in the local network and respond with the results of processing of the control commands or queries in the form of events.

Local Network Manager: The local network communication module includes a local network manager. When a RESTful server has received a service request (control command or query), the local network manager processes the received request. That is, the local network manager analyzes or stores the received control command or query. In addition, the local network manager calls a function of the interface to deliver the analyzed control command or query to the communication modem. It can manage control commands, events, and queries in the form of tables. A relevant management table may include a control table, an event table, and a query table. Information that may be included in each table may include service types, device IDs, device types, user IDs, and opcodes.

The control table may include the controls supported by the appliance product itself. The event table may include those event types that can be generated by the appliance product itself. The query table may include those types of queries that may be requested to the appliance product itself by other appliance products. In this case, the events stored in the event table may match the controls stored in the control table or the queries stored in the query table. That is, the information stored in the management table may include service information that the appliance product itself can support. The control commands or queries received from other appliance products may be additionally stored.
in the management table, and when responses to the additionally stored control commands or queries have been completed, the additionally stored control commands or queries may be deleted.

Monitoring Manager: The local network communication module additionally includes a monitoring manager. The communication modem saves the files generated by the function calls, analyzes the stored files, and delivers the contents of the files to the MICOM. The MICOM performs the functions corresponding to the information transmitted from the communication modem. When the functions have been completely performed, the MICOM may respond to the communication modem about the completion of processing. The communication modem also stores the response file regarding the completion of processing that came from the MICOM and sends it to the monitoring manager. The relevant monitoring manager can detect the changes in the file and read the file to deliver its contents to the local network manager.

4. Protocols for HALN Architecture

4.1. Protocol for Receiving and Processing Control Commands

In referring to Figures 3 and 4, it can be seen that the RESTful server of an appliance product may receive control commands from the RESTful clients of other appliance products.

![Figure 4. Protocol for receiving and processing control commands.](image)

When the RESTful server of an appliance product has received a control command, the RESTful server transmits the response to the other appliance product. The control command received by the RESTful server is passed to the local network manager and can be stored in the control table. The local network manager can make a function call to the interface to deliver the received control command to the communication mode. Then, the communication modem may store a file for the received control command and deliver the received control command to the product MICOM, which will perform a function in response to the received control command. In this case, the control command may include a set of commands as follows:

- A command for driving the appliance product, a command to stop driving;
- A command for operating based on energy information, a command for driving in diverse operation methods such as a course or option of the appliance product;
• A command to turn on/off or operate various components constituting the appliance product;
• A command for operation or control of additional function modules such as the camera;
• A command to save a file for updating;
• A command to start or stop diagnosis or monitoring.

When the product MICOM has completed the performance of the function in response to the received control command, the product MICOM may transmit a response regarding the completion of processing to the communication modem. The communication modem may transmit a file for the response regarding the completion of processing to the above-mentioned monitoring manager, which detects the change in the file and transmit information on the completion of processing to the local network manager. Then, the local network manager can check and delete the control command that was additionally stored in the control table. In cases where the communication modem has transmitted a control command to the MICOM but has not received any response from the MICOM, the communication modem may transmit a file for no response to the monitoring manager. The monitoring manager then detects the change in the file and delivers information on the no response to the local network manager.

The local network manager may transmit failure events to the RESTful client. Then, the RESTful client may identify the appliance products to which it should transmit the failure events through communication with the database control manager and transmit the failure events to the relevant appliance products. The other appliance products that have received the failure events may retransmit a control command to the appliance product.

4.2. Protocol for Receiving and Processing Events

In referring to Figures 3 and 5, it can be seen that the RESTful server of an appliance product may receive an event from the RESTful clients of other appliance products. When the RESTful server of the appliance product has received an event, the RESTful server sends a response regarding the receipt of the event to the other appliance product. The event received by the RESTful server may be delivered to the local network manager and stored in the event table. Hereinafter, only those parts that are different when compared to the case where a control command has been received as illustrated in Figure 4 will be described.

![Figure 5. Protocol for receiving and processing events.](image-url)
The event received from the RESTful server may be a failure event in which an appliance product sends a control command or query to another appliance product but the latter the appliance product fails to process it and transmits it to the former appliance product. In cases where the event received by the RESTful server is a failed event, the local network manager attempts to retransmit the control command or query to the RESTful client and the event additionally stored in the event table may be deleted. Then, the RESTful client may identify the appliance product to which it may retransmit the control command or query through communication with the database control manager and retransmit the control command or query to the relevant appliance product.

4.3. Protocol for Receiving and Processing Queries

In referring to Figures 3, 6 and 7, it can be seen that the RESTful server of an appliance product can receive queries from the RESTful clients of other appliance products.

![Figure 6. Protocol for receiving and processing queries (part-A).](image1)

![Figure 7. Protocol for receiving and processing queries (part-B).](image2)
When the RESTful server of the appliance product has received a query, the RESTful server transmits the response to the receipt of the query to other appliance products. The query received by the RESTful server can be delivered to the local network manager and stored in the query table. The local network manager may make a function call to the above-mentioned interface to deliver the received query to the communication modem. Then, the communication modem can save a file for the received query and deliver the received query to the MICOM, which may perform the function corresponding to the received query. In this case, the performance of the function in relation to the received query may be verification for response or preparation for response to the query. In addition, the query may be an inquiry into the current state of the appliance product or a request for the information held by the appliance product or the past operation history of the appliance product.

In cases where the product MICOM has completed the performance of the function in response to the received query, the response regarding the completion of processing can be transmitted. The communication modem can transmit a file for the response regarding the completion of processing to the monitoring manager, and the monitoring manager detects changes in the file and transmits information on the completion of processing to the local network manager. Then, the local network manager can check and delete the query stored in the query table. The local network manager then delivers the event, which is the response to the received query, to the RESTful client. Then, the RESTful client may identify the appliance products to which it should transmit the event through communication with the database control manager and transmit the event, which is a response to the query, to the relevant appliance product.

In cases where the communication modem has transmitted a control command to the product MICOM but has not received any response from the MICOM, the communication modem may transmit a file for no response to the monitoring manager. The monitoring manager then detects the change in the file and delivers information on the no response to the local network manager.

5. Experimental Result

5.1. Experimental Setup

The proposed HALN architecture is implemented on a Wi-Fi modem with embedded Linux as shown in Figure 8. As an example, for the local network-based services, Figure 9 illustrates an experimental setup for the remote monitoring of power consumption in each appliance product through the HALN architecture. The appliance products used in this experiment include a washing machine, a dryer, and a display unit on the refrigerator. The power consumption by the washing machine and the dryer are remotely monitored and the total power consumption level is shown on the refrigerator display.

<table>
<thead>
<tr>
<th>Wi-Fi Modem</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ARM Processor</td>
<td></td>
</tr>
<tr>
<td>- Core : ARM926EJ</td>
<td></td>
</tr>
<tr>
<td>- Frequency : 454MHz(max)</td>
<td></td>
</tr>
<tr>
<td>• SDRAM : 64MB</td>
<td></td>
</tr>
<tr>
<td>• NAND Flash : 128MB</td>
<td></td>
</tr>
<tr>
<td>• Wi-Fi Chipset</td>
<td></td>
</tr>
<tr>
<td>- support IEEE802.11a/b/g/n</td>
<td></td>
</tr>
<tr>
<td>• OS : Embedded Linux</td>
<td></td>
</tr>
<tr>
<td>• PCB</td>
<td></td>
</tr>
<tr>
<td>- Size : 44mm x 60mm</td>
<td></td>
</tr>
<tr>
<td>- 6-Layer</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8. Wi-Fi modem and its specification used for experiments.
information on the completion of processing to the local network manager. Then, the local network manager can check and delete the query stored in the query table. The local network manager then delivers the event, which is the response to the received query, to the RESTful client. Then, the RESTful client may identify the appliance products to which it should transmit the event through communication with the database control manager and transmit the event, which is a response to the query, to the relevant appliance product.

In cases where the communication modem has transmitted a control command to the product MICOM but has not received any response from the MICOM, the communication modem may transmit a file for no response to the monitoring manager. The monitoring manager then detects the change in the file and delivers information on the no response to the local network manager.

5. Experimental Result

5.1. Experimental Setup

The proposed HALN architecture is implemented on a Wi-Fi modem with embedded Linux as shown in Figure 8. As an example, for the local network-based services, Figure 9 illustrates an experimental setup for the remote monitoring of power consumption in each appliance product through the HALN architecture. The appliance products used in this experiment include a washing machine, a dryer, and a display unit on the refrigerator. The power consumption by the washing machine and the dryer are remotely monitored and the total power consumption level is shown on the refrigerator display.

In addition to the remote power monitoring system, a smart oven is used to compare the communication reliability of the HALN architecture with a server-based communication network. The experimental setup consists of ten Wi-Fi modems in which a server-based communication network is implemented, and 10 Wi-Fi modems in which the HALN architecture is implemented. The experiment is carried out as follows:

- A five-minute cooking in the smart oven is carried out. Cases where the monitoring information (i.e., remaining cooking time displayed on a smartphone application as shown in Figure 10), which is required to update every five seconds, is not updated twice or more are judged as a communication NG.
- The above process is repeated 100 times and the communication success rate is calculated.

5.2. Experimental Results

As compared in Figures 11 and 12, the communication success rate with the server-based network shows 97.6% on average, while the proposed HALN architecture achieves 99.2% on average. Based on this result, it is seen that HALN architecture improves the communication success rate by 1.6% as compared to the server-based network architecture. The improvement results from the reduction in data loss thanks to the simplified communication architecture with HALN.
5.2. Experimental Results

As compared in Figures 11 and 12, the communication success rate with the server-based network shows 97.6% on average, while the proposed HALN architecture achieves 99.2% on average. Based on this result, it is seen that HALN architecture improves the communication success rate by 1.6% as compared to the server-based network architecture. The improvement results from the reduction in data loss thanks to the simplified communication architecture with HALN.

![Figure 11. Comparison of communication success rates.](image)

<table>
<thead>
<tr>
<th>Sample</th>
<th>N</th>
<th>Average</th>
<th>Standard Deviation</th>
<th>SDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server-based Network</td>
<td>10</td>
<td>97.600</td>
<td>0.516</td>
<td>0.16</td>
</tr>
<tr>
<td>HALN</td>
<td>10</td>
<td>99.200</td>
<td>0.632</td>
<td>0.20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Difference Estimate</th>
<th></th>
<th>Confidence Intervals for Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difference</td>
<td>-1.600</td>
<td>(-2.145, -1.051)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hypothesis Test</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Null Hypothesis</td>
<td>H0: μ1 = μ2 = 0</td>
<td></td>
</tr>
<tr>
<td>Alternative Hypothesis</td>
<td>H1: μ1 ≠ μ2</td>
<td></td>
</tr>
<tr>
<td>T-Value</td>
<td>-20</td>
<td></td>
</tr>
<tr>
<td>DF</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>P-Value</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

![Figure 12. Statistics analysis (2-sample t verification using Minitab) of communication success rates.](image)

6. Conclusions

The HALN architecture proposed in this paper has demonstrated the advantage of being able to directly communicate with other appliances to transmit or receive information. In addition, HALN was able to offer the same services as those provided by the existing server-based communication network architectures. In terms of the communication performance, the HALN architecture has improved the communication success rate by 1.6% as compared to the existing server-based network architecture. Since the proposed architecture can reduce the server utilization, it is expected to save the cost for the server extension even when the number of network users increases.

For future works, additional studies are required for determining the access priority in cases where both the server-based communication architecture and the HALN architecture access simultaneously. It is also necessary to develop the policy for supporting multi-users, and to improve the network reliability in the event of communication failure.
Author Contributions: Conceptualization, H.-J.P. and D.L.; methodology, H.-J.P.; software, H.-J.P.; validation, H.-J.P. and D.L.; formal analysis, H.-J.P.; investigation, H.-J.P. and D.L.; resources, H.-J.P. and D.L.; data curation, H.-J.P.; writing—original draft preparation, H.-J.P.; writing—review and editing, D.L.; visualization, H.-J.P.; supervision, D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References


