An Intelligent Home Environment based on Service Planning over Peer-to-Peer Overlay Network

Chuan-Feng Chiu, Steen J. Hsu, Sen-Ren Jan
Department of Information Management, Minghsin University of Science and Technology, Hsinchu, Taiwan
Email: cfchiu@must.edu.tw

***Abstract*—With the population of network usage, it is possible to connect home appliances with each other. The basic demand is to connect home appliances easily with less user intervening and home network is an complex environment which comprises several network architecture including ethernet, wireless network, or power-line etc.. Therefore, integrating heterogeneous environment in home network to reduce user intervening effort will be the first issue to realize smart home environment. On the other hand, there exists many services within home network, these services need to be integrated to make home network smart. So we propose a middleware to realize smart home environment over peer-to-peer overlay network to resolve the heterogeneous home environment problem and propose the service collaboration framework to use home services intelligently.**

***Index Terms*—home network, smart home, service collaboration, UPnP**

I. Introduction

Because of the population of network usage, traditional home network have move on digital era. Home devices in the traditional home network are operating individually and can not communicate with other devices. Through network connection home devices will be connected with each other and work cooperatively. In order to provide seamless connection within home network, UPnP is the most popular technology to realize the home network operations. The most important operation of UPnP technology is device/service discovery. Through device/service discovery, home devices can be aware which devices or services are available to use. However the device/service discovery mechanism is using broadcast messages. There are two major problems arising from the broadcast message. First, each device will send broadcast message to others, so that the network performance will be decreased. Second, because of broadcast message is sent over reserved broadcast IP address, therefore, broadcast message will not be sent across to other network segment. Otherwise, the router or gateway needs to be modified to support broadcasting across to other segments. So the heterogeneous environment will decrease the usage of home network.

Therefore, in order to resolve the above problem, we need to reduce the number of broadcast message and generic transportation architecture to support multi-segment network architecture and seamless connection between different network architecture. Fortunately, with the development of peer-to-peer overlay network architecture, multi-segment issue and seamless connection can be resolved easily without changing existing network design. In peer-to-peer network, messages can be relayed by intermediate peers and do not consider underlying network segment. Therefore, we take the advantage of the peer-to-peer overlay network architecture, we propose an implementation for UPnP operations over peer-to-peer overlay network. Through the implementation, the number of broadcast message will be reduced and message can be delivered without affecting by the mixed underlying network architecture.

On the other hand, when home user use their own mobile device, for example, PDA or mobile phone, to control home appliances within home network and move outside home network, home user will not control home device directly. Although we can deploy a home gateway to solve such kind of scenario, this will increase more complicate configuration and cost. Therefore, our work proposed in this paper can meet the demand of flexible home usage also.

Besides the seamless integration of different network, to manipulate service easily is also a basic demand for home users and intelligent usage of home service is also an important feature to realize a smart home environment. In this paper, we also propose a service collaboration framework based on service planning to provide an easy use home environment for home users.

The rest of this paper is organized as followings. Section 2 will reveal UPnP and peer-to-peer overlay network concept, and related research on home network. Section 3 describes the design of UPnP operations over peer-to-peer overlay network. Section 4 describes the system architecture of the proposed implementation of UPnP-based home environment over peer-to-peer overlay network. In section 5, a service collaboration framework is proposed to make an intelligent home environment. Finally, we give a brief conclusion of our work in section 6.

II. RELATED WORK

UPnP[1] is the most popular implementation to realize home network and it can enable data communication between home devices via a set of UPnP commands. The major operations including service discovery, service advertisement service acquiring and service control and these operations are transmitted over HTTP protocol. UPnP comprise several standards such as Simple Device Discovery Protocol, Simple Object Access Protocol and General Notification Architecture to perform service discovery, service description acquiring and service control and notification. Two kinds of roles are defined in UPnP. One is UPnP Control Point and the other is UPnP Device. UPnP Control Point is the device that can issue manipulation command to ask UPnP device to perform some actions. And UPnP Device is to provide the available services for UPnP Control Point. Figure n is the general representation between UPnP Control Point and UPnP Device.

Peer-to-peer network is a popular virtual network architecture in recent year. It can provide resource sharing in a distributed way. It resides on underlying network architecture and is a distributed architecture without central server involving. The first peer-to-peer network is Napster[4]. It deploy a central server as an index server, each peer can find the location of interest data items from the index server and contact the corresponding peer directly. With the development of peer-to-peer network, two peer-to-peer network architectures are appeared. One is unstructure peer-to-peer network and the other is Structure peer-to-peer network. Gnutella[3] is such kind of unstructure peer-to-peer network. Peers are connected in a distributed way and do not need to know overall topology. It uses flooding as the mechanism to send query to find the related information. In structure peer-to-peer network, it assign *key* to the data and compute a *value* for the *key*. The *(key, value)* pair is used for retrieving and locating the data item on a peer. CAN[5], Chord[6], Pastry[7] and Tapestry[8] are the famous structure peer-to-peer network system. Unlike flooding in unstructure peer-to-peer network, structure peer-to-peer network routing can be bounded in *O(logN)* hops.

III. UPNP OPERATIONS OVER P2P

In this section, we reveal the design of the UPnP message transmission over peer-to-peer overlay network. The following sub-sections describe P2P-based UPnP device/service discovery and advertisement, UPnP service description acquiring, UPnP service control and event.

A. Generic Message Middleware

When UPnP device join the home network, the UPnP device perform the join operation of peer-to-peer overlay network and establish the neighbor list of the UPnP devices. And then UPnP message could be sent over peer-to-peer overlay network. Originally, the UPnP message delivery will be sent via network interface directly. The solution of our propose method is to deliver UPnP message via peer-to-peer overlay network which do not need to consider the underlying network architecture. Figure 1 show the message flow over peer-to-peer network. When UPnP device send an UPnP message, the message will be transferred to peer-to-peer middleware which look up the address of destination device. If the middleware could not decide the destination address, the message will be sent to proper intermediate device to look up destination device via peer-to-peer routing mechanism and network transmission layer in advance. Therefore, the next device or destination device will be decided by the peer-to-peer middleware.

In order to avoid duplicate message received by nodes, each received message will be recorded in a receiving pool as a record by hash function. When message is received, the device calculates the hash value of the message and checks the hash value with receiving pool. If hash value is in the pool, the message will be dropped. Otherwise, the message will be processed. Figure 2 shows the overall processing flow chart with respect service discovery/advertisement, service description acquiring and service control.

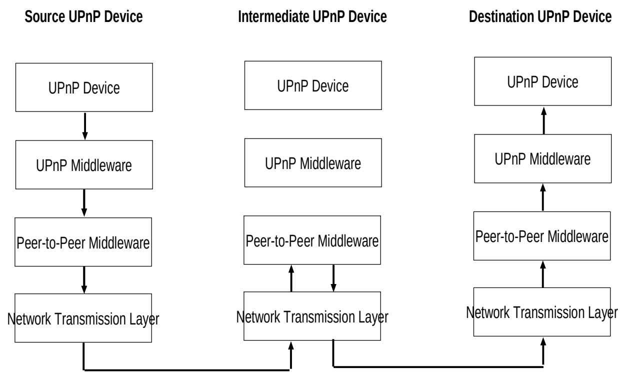

Figure 1. UPnP message over P2P network

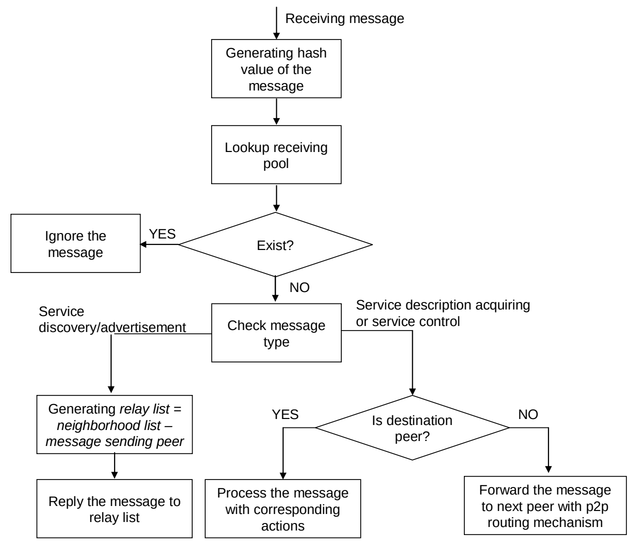

Figure 2. Overall processing flowchart of UPnP message

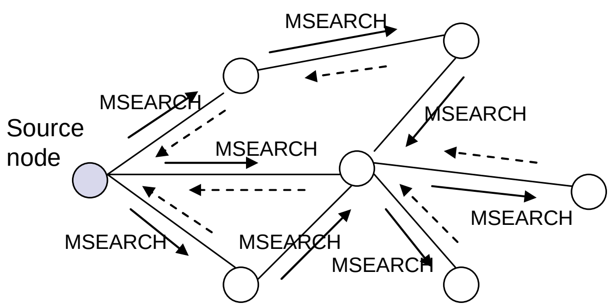

Figure 3. MSEARCH message over P2P network

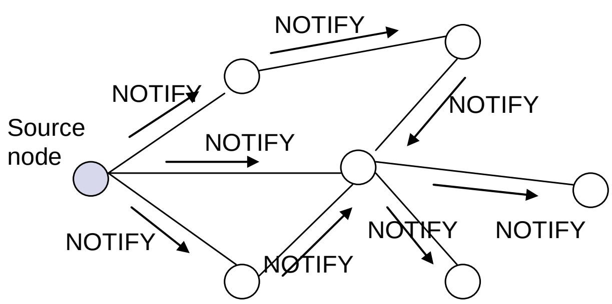

Figure 4. NOTIFY message over P2P network

B. Service discovery and advertisement

Device/service discovery and advertisement are the most important operation in UPnP. When UPnP device join or leave the home network, discovery and advertisement message is used to guarantee the availability of the devices. The discovery and advertisement mechanism comprise two kind of message including MSEARCH and NOTIFY.

If the device wants to know other devices, MSEARCH message will be sent. When MSEARCH message is sent, the message will be relayed to every node of the peer-to-peer overlay network. When a node receives the MSEARCH message, the node responses a reply message to the original node with its device identifier. Figure 3 shows the MSEARCH message over peer-to-peer network. Bold line is the MSEARCH message and dash line is the response message with respect to the MSEARCH message. From the figure, we can observe that intermediate node might receive several duplicate messages. As we described in section 3.1, first comes message will be processed and later duplicate message will be dropped. In addition, in order to avoid too many

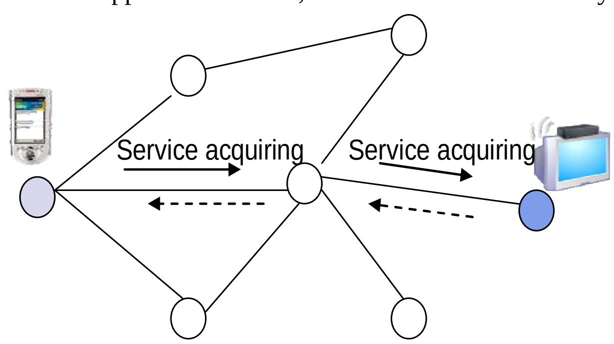

Figure 5. Service acquiring message over P2P network

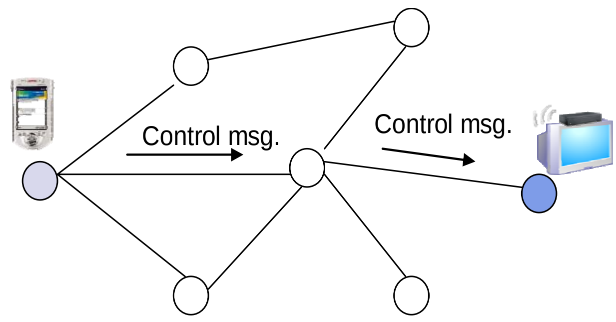

Figure 6. Control message over P2P network

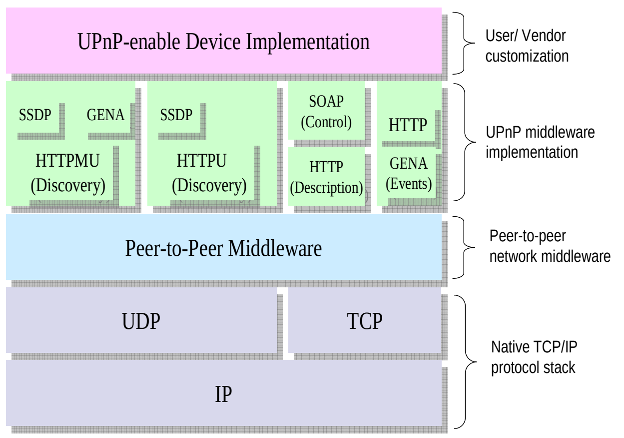

Figure 7. Protocol stack illustration

response messages on the same path, intermediate node will collect all response messages and package into one response message to the original sending node.

When UPnP device notify the awareness of the device actively, the NOTIFY message will be sent. When NOTIFY message is sent, the peer-to-peer middleware perform the join action to participate the device into peer-to-peer overlay network and intermediate node will relay NOTIFY message to its neighbors. Figure 4 shows the NOTIFY message over peer-to-peer network.

C. Service description acquiring and control

When peer has device identifier and related information, the peer can issue service acquiring message to get the service description over peer-to-peer overlay network. And then acquired node will reply the service description to the acquiring peer. The acquiring node will cache the service description in the available service pool for future usage. On the other hand, Service description acquiring is an end-to-end action, but devices are located in different network segment. Therefore, service description acquiring is still transmitted alone with peer-to-peer overlay topology. Figure 5 shows the service acquiring message over peer-to-peer network. Service control action is similar with service description acquiring operation which is an end-to-end action also. Therefore, when a node receives the service control command, the node will work according to the received service control command. Figure 6 shows the service control message over peer-to-peer network.

IV. SMART HOME NETWORK ARCHITECTURE

In this section, we propose the design of the system architecture. First we will reveal the design of the protocol stack to realize the UPnP message over peer-to-peer network. And then the system software architecture which comprises several computing blocks or procedures will be showed.

A. Protocol Stack

In order to make less modification on existing protocol stack, we insert an add-on peer-to-peer network middleware between UPnP middleware implementation and underlying TCP/IP protocol stack. The overall protocol stack is showed in Figure 7. The overall protocol stack comprising four portions including user/vendor customization part, UPnP middleware, peer-to-peer network middleware and underlying TCP/IP stack.

User/vendor customization : Application developer can design their application with respect their scenario , idea or, usage. And they do not concern the detail of the underlying complex middleware or stack.

UPnP middleware : This is the traditional part of UPnP implementation including all functionalities of UPnP operations and messages.

Peer-to-peer network middleware : This portion is the basic transmission platform. This middleware will perform node joining, node leaving and topology maintenance. In addition, the peer-to-peer middleware will transmit the UPnP messages to destination node or route the message intermediate node also.

TCP/IP protocol stack : This is the actual network transmission layer. The destination node will be resolved by the peer-to-peer middleware and pass the destination node to the TCP/IP layer to perform actual packet transportation.

B. Software Architecture

The protocol stack in previous section reveals the protocol architecture for realizing the home network scenario. In this section, we describe the software architecture for UPnP operations over peer-to-peer network. All messages are received and sent by the underlying network interface. The received packet will passed to message processor which perform message parsing and check whether the message is duplicate or not.

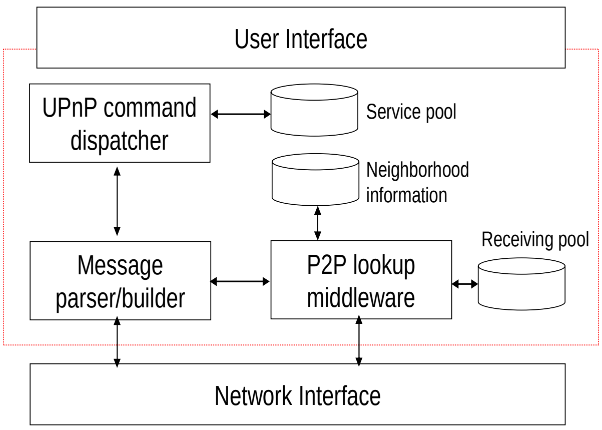

Figure 8. The middleware software architecture

Then the message will be passed to peer-to-peer lookup middleware to decide the packet will be relayed or be processing by the UPnP command dispatcher. The UPnP message dispatcher performs the major processing of the UPnP command and responsible for the corresponding actions. If the command is service advertisement, service discovery and service acquiring, the resolving result will be stored in the service pool. In addition, peer-to-peer lookup middleware will be responsible for peer-to-peer topology management, and peer join/leave processing. Figure 8 shows the proposed software architecture.

V. SERVICE COLLABORATION FRAMEWORK

In previous section, we have proposed the design of smart home environment based on peer-to-peer overlay network to seamless connecting home appliances in a feasible way. But in order to share and manipulate home service in a flexible approach, we propose a service collaboration framework in this section. Our proposed mechanism is based on service planning to have a collaboration diagram for home service so that the collaboration framework will reduce user effort.

The general principle to achieve collaboration between different services on different home appliances is outlined as following.

Service preference : In order to describe different service capability or constrain, to have a easy understanding preference for service is very important. Fortunately, UPnP protocol described in previous section

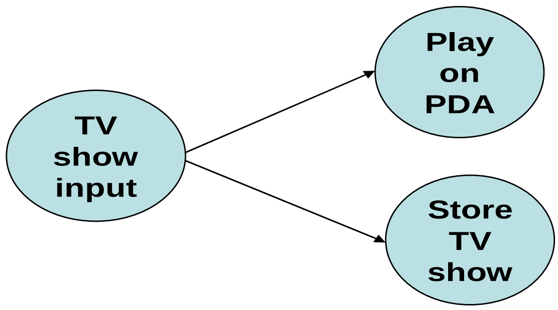

Figure 9. Service Planning Example

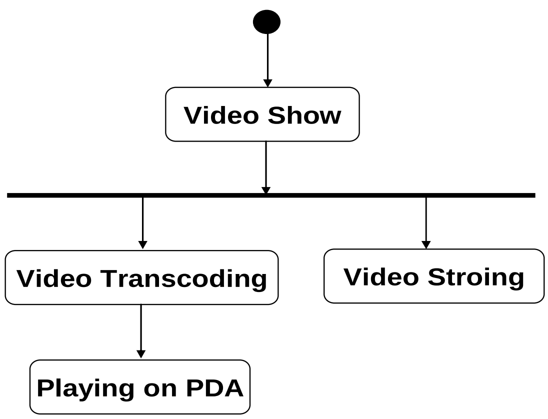

Figure 10. Service Diagram I

have defined service description based on XML language. The service preference can be described in common and easily.

Service workflow determination : Service is a basic unit for specified task in home environment. Home user will not understand each atomic service very well. And home user manipulates home service that comprises several services involving usually. Therefore to define a workflow that integrates each service is the tool for home users.

Service composition : After defining service workflow, different services can be composed from different home appliances as an atomic-like service so that user effort will be reduced and manipulate service in

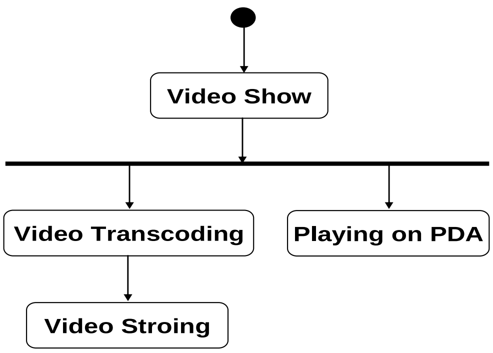

Figure 11. Service Diagram II

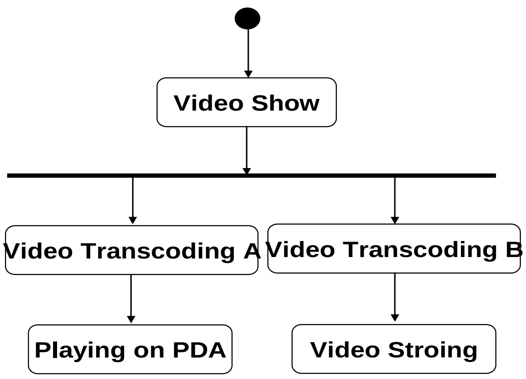

Figure 12. Service Diagram IV

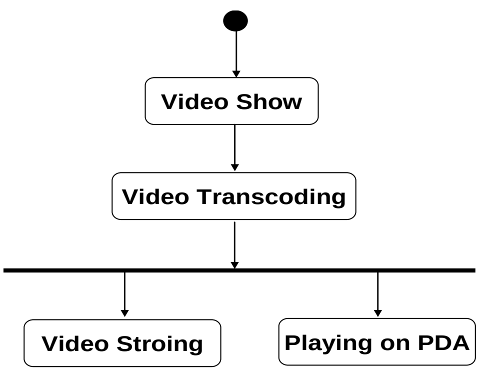

Figure 13. Service Diagram III

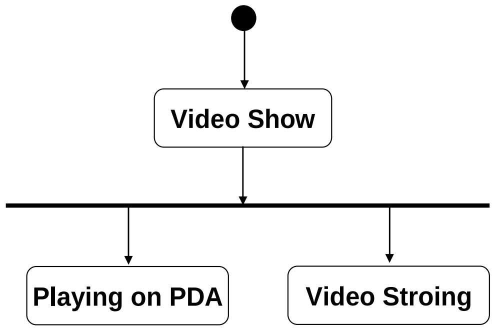

Figure 14. Service Diagram V

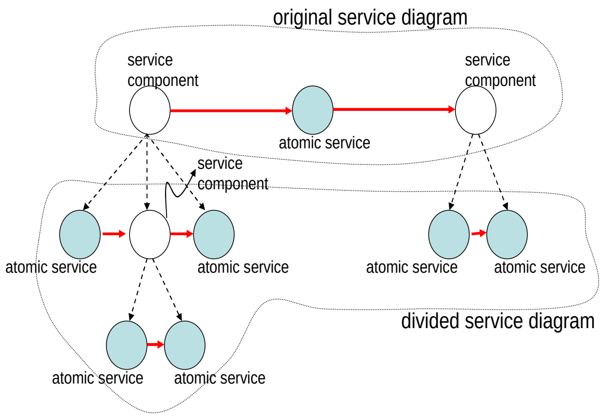

Figure 15. Service Planning Example

a more flexible way. The stage is to collect all of the available services and merge into service collaboration diagram.

Service execution : When service composition based on the service workflow, the composed service will be executed. And according to different service state in the service workflow, some of atomic service can be executed in parallel. Therefore, the service execution performance can be improved compared with traditional service manipulation which uses each service in step by step way. The service execution chain is called service execution flow.

A. Service Planning

In order to make several services working cooperatively, we need to define a workflow is achieved by which services. At first we provide a service planning interface to home user to define service workflow. In the tool, home users just pick service component from the toolbar to link these picked service as a overall service flow. For example, we can consider the following scenario. A home user wants to watch a TV show on its PDA and wants to record the show in a storage place in home. Figure n shows the workflow of the above scenario and each circle represents a service task. The workflow will be called service planning in this paper. Service planning is high-level process to describe the service component which might be an atomic service or comprising sub-service components.

B. Service Composition Mechanism

When the service planning is defined, the planning is complied as a service composition diagram. The compile process will recognize the planning into several detail service components and schedule different execution flow to produce different service diagrams that might have synchronous or asynchronous execution path. The service composition mechanism is a recursive process. Each service component from the service planning will be divided into more detail service component that might be atomic service or another service planning. Figure 15 illustrates the result of service diagram extraction process in a hierarchy architecture. The synchronous or asynchronous execution path is produced according to the following definition.

Sequencing Flow : When the service A is followed service B, service B has a directed flow to service A. That is $A \to B$.

Concurrent Flow : When service A and service B are independent, service diagram will split two concurrent flows to perform service A and B. That is $A \cap B = \emptyset$.

Figure 10~14 shows several service diagrams of the example that is described in previous sub-section. Service diagram I in figure 10 means a video stream, called TV show in previous section, is the input for playing on the PDA and storing in home storage. Before playing on PDA, the video stream needs to be transferred to another format to play on PDA. On the other hand, video stream can be stored in home storage without transcoding. Service diagram II in figure 11 means a video stream is the input for playing on the PDA and storing in home storage. Before storing in the home storage, the video stream needs to be transferred to another format. On the other hand, video stream can be played on PDA directly. Service diagram III in figure 12 means video stream needs to be transferred to another format first and the format is suitable for both PDA and home storage. Service diagram IV in figure 13 means the video stream needs to be transferred to another two formats to play on PDA and store in home storage. Because of the PDA and home storage require different video format. Service diagram V in figure 14 means the video stream does not need to be transferred to another format and can be played on PDA and stored in home storage directly.

After service planning is divided as different service diagram representation, the home appliances will send the service query or discovery message to invoke service from other home appliances. After all atomic services are ready to execute, the overall service is prepared to be executed in next step.

C. Service Execution

After service composition mechanism by the service component extraction process, the service execution phase will compose the actual execution object for each service diagram. First, the service execution phase will send a service acquiring message to peer-to-peer based home environment to have all home appliances' status that will provide the service. After gathering all possible supported home appliances, the service execution phase will compute all possible service execution flow and compute the cost for each service execution flow. For example, if two service S1 and S2 can satisfy user's requirement, and home appliance A and B support service S1 and home appliance C and D provide service S2, the possible service execution flow will have four choices, that is (A, C), (A, D), (B, C) and (B, D). The service execution phase will select the minimum processing time of service execution flows. The service processing time is defined as the sum of the home appliances processing time and network processing time. The home appliances processing time will also be influenced by different home appliances' capabilities including the remained CPU time to process the service, the specified service processing time etc.. The network processing time includes all network delay, network unit processing time etc.. we will select the minimum cost from these service execution flows as the actual service execution flow. Equation (1) is the formula to compute processing cost with respect to each service execution flow.

$$Cost = \alpha \sum_i P_i + (1-\alpha) \sum_j L_j \quad (1)$$

α is an estimated factor to represent the importance of the home appliance processing time and $0 \le \alpha \le 1$. $1-\alpha$ shows the importance factor of the network transmission. P_i is the home appliance processing time of device i and L_j is the network transmission time for edge j between two consequence execution devices. The network transmission time comprise the network delay, propagation delay and queueing delay.

C. Implementation

In previous sub sections, we have illustrated the major steps for collaborating services in home network. In order to build the service collaboration framework to process the results of each stages of service collaboration mechanism, we define several description languages to represent the results. All of designed description language is based on XML technology which is the famous and popular tool to design new description language in a

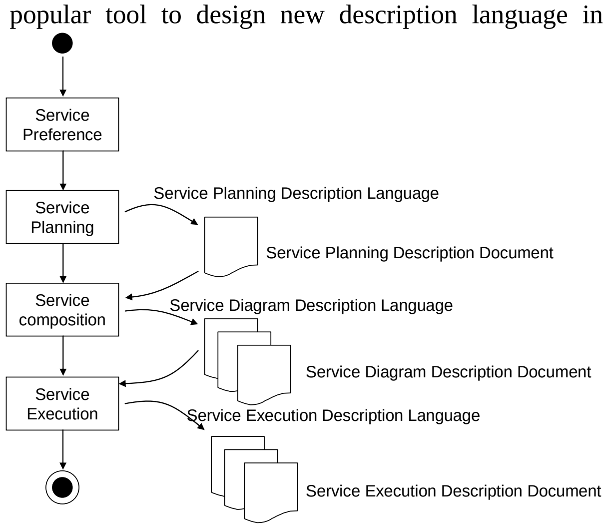

Figure 16. Description Language Set of Service Collaboration Framework

flexible way. On the other hand, because the atomic service description and device description of UPnP-based home environment are also XML-based. Therefore, using XML technology as the core of the description language is the trivial method to implement the service collaboration framework. The complete set of description language in the service collaboration framework is showed in figure 16. In service planning phase, Service Planning Description Language(SPDL) is used to produce Service Planning Description Document(SPDD) to illustrate the service planning. Figure n shows an example of Service Planning Description Document(SPDD) using Service Planning Description Language(SPDL). Service Diagram Description Language(SDDL) is used to produce Service Diagram Description Document(SDDD) to illustrate the service planning that can be performed by the different composition. Figure 17 shows an example of Service Diagram Description Document(SDDD) using Service Diagram Description Language(SDDL). Service Execution Description language(SEDL) is used to produce the Service Execution Description Document(SEDD) to illustrate the actual execution objects. Figure 18 shows an example of Service Execution Description Document(SEDD) using Service Execution Description language(SEDL).

```
<Service_Planning name=“…..”>
        <Service_Component>
                <Component_ID>…</Conponent_ID>
                <Component_Name>…</Component_Name>
                <Parent_Component>… </Parent_Component>
        </Service_Component>
        …Other Service Component……
</Service_Planning>
```

Figure 17. An Example of Service Planning Description Document

```
<Service_Diagram name=“…..”>
        <Service_Planning_Name>...</Serivce_Planning_Name>
        <Service_List>
                <Sequence_Number>...</Sequence_Number>
                <Atomic_Service Name=“...”>
                        <Ancestor_Service>...</Ancestor_Service>
                        <Service_Action Name=“...”>
                                <Service_Parameter>
                                        <Paramter_Name>...</Parameter__Name>
                                        <Paramter_Value>...</Parameter__Value>
                                <Service_Parameter>
                                ....other service parameter.....
                        </Service_Action>
                        ...other service action...
                </Atomic_Service>
                ...other atomic service......
        </Service_List>
        ...Other Service List......
</Service_Diagram>
```

Figure 19. An Example of Service Execution Description Document

```
<Service_Execution Name=“…..”>
        <Execution_Flow Name=“...” Sequence=“...”>
                <Atomic_Service Name=“...”>
                        <Ancestor_Service>...</Ancestor_Service>
                        <Device_URL>...<Device_URL>
                        <Service_Action Name=“...” URL=“...”>
                                <Service_Parameter>
                                        <Paramter_Name>...</Parameter__Name>
                                        <Paramter_Value>...</Parameter__Value>
                                <Service_Parameter>
                                ....other service parameter.....
                        </Service_Action>
                        ...other service action...
                </Atomic_Service>
                ...other atomic service....
        </Execution Flow>
        ...other service execution flow......
</Service_Execution>
```

Figure 18. An Example of Service Diagram Description Document

VI. CONCLUSION

In this paper, we propose a middleware implementation for home network to extend the UPnP operations which is originally restricted by broadcast message. Based on the proposed middleware, the broadcast message can be transmitted cross different network segment and do not need to modify the underlying network devices. And based on the message transmission over peer-to-peer network, the number of broadcast message can be reduced and make the network usage efficient. On the other hand, in order to provide a smart home service environment, we also propose a service collaboration framework which is based on XML technology. Because our proposed middleware is based on peer-to-peer overlay network, there exists security issue including DoS attack between these peers. Therefore, a better security mechanism will be the most important task in the future. On the other hand, service planning in the service collaboration framework is generated by home users. In order to provide an automatic environment, service semantic composition is also a future research direction.

REFERENCES

[1] UPnP Forum, avaliable: http://www.upnp.org.

[2] E Lua,J Crowcroft,M Pias,et al., "A survey and comparison of peer-to-peer overlay network schemes", IEEE Communications Survey and Tutorial, Vol.7, No.2, 2005, pp. 72-93.

[3] Gnutella development forum, the gnutella v0.6, 2001, Avaliable: http://groups.yahoo.com/group/the_pdf/files/.

[4] Napster.[Online]. available: http://www.napster.com/

[5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network", in Proceedings of the ACM SIGCOMM, 2001, pp. 161-172.

[6] Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," IEEE/ACM Transitions on Networking, vol.11, no.1, 2003, pp. 17-32.

[7] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems", IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, November, 2001, pp. 329-350.
[8] Y. Zhoa, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowica, "Tapestry: A resilient global-scale overlay for service deployment," IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, January 2004, pp. 41-53.
[9] Dong-Sung Kim, Jae-Min Lee, Wook Hyun Kwon, In Kwan Yuh, "DESIGN AND IMPLEMENTATION OF HOME NETWORK SYSTEMS USING UPnP MIDDLEWARE FOR NETWORKED APPLIANCES", IEEE Transactions on Consumer Electronics, Vol. 48, No. 4, NOVEMBER 2002, pp. 963-972.
[10] Young-Guk Ha, Joo-Chan Sohn and Young-Jo Cho, "ubiHome: An Infrastructure for Ubiquitous Home Network Services", in Proceedings of IEEE International Symposium on Consumer Electronics(ISCE 2007), Irving, TX, USA, 20-23 June 2007, pp.1-6.
[11] Xie Li and Wenjun Zhang, "The Design and Implementation of Home Network System Using OSGi Compliant Middleware", IEEE Transactions on Consumer Electronics, Vol. 50, No. 2, MAY 2004, pp.528-534.
[12] Kyeong-Deok Moon, Young-Hee Lee, and Young-Sung Son, Chae-Kyu Kim, "Universal Home Network Middleware Guaranteeing Seamless Interoperability among the Heterogeneous Home Network Middleware", IEEE Transactions on Consumer Electronics, Vol. 49, No. 3, AUGUST 2003, pp.546-553.

Chuan-Feng Chiu received his Ph. D. degree in Computer Engineering from Tamkang University, Taiwan in 2002. He also received his MS and BS degrees in the same major in 1999 and 1995, respectively. From 2002 to 2006 he was a senior engineer in Panasonic Taiwan Laboratory in Taiwan. Since 2006 he has been working as a Assistant Professor in Department of Information Management, Minghsin University of Science and Technology, Xinfeng Hsinchu, Taiwan. His research interests include multimedia system, peer-to-peer computing, home network and network security. Since 1999, he published many refereed papers in international conferences and journals. He is an honor member of Phi-Tau-Phi Scholastic Honor Society of the Republic of China.

Steen J. Hsu was born in Taipei County, Taiwan, Republic of China, on September 10, 1963. He received the M.S. and Ph.D degrees in computer science and information engineering in 1992 and 1999, respectively, from the National Chiao Tung University, Hsinchu,Taiwan.

Form August 1999 to July 2005, he was an Assistant Professor of the Department of Information Engineering at I-Shou University, Kaohsiung County, Taiwan. Since August 2005, he has been an Assistant Professor of the Department of Information Management at Minghsin University of Science and Technology, Hsinchu County, Taiwan. His research interests include wireless LAN, multimedia communication, information security, and P2P networks.

Dr. Hsu is a member of the IEEE Communication Society and the Phi Tau Phi honor society of the Republic of China.

Sen-Ren Jan received the BS degree in computer science from Tamkang University, Taipei, Taiwan, in 1983 and the MS, PhD degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 1991, 2000, respectively. He is currently an assistance professor in the Department of Information Management at Minghsin University of Science and Technology, Hsinchu, Taiwan. His current research interests include mathematical morphology, image processing, data hiding and pattern recognition.