A Transparent Protocol Scheme Based on UPnP AV for Ubiquitous Home

Jiyun Park, Hyunju Lee, and Sangwook Kim

Dept. of Computer Science, Kyungpook National University, Korea
{jypark,hyunju,swkim}@woorisol.knu.ac.kr

Abstract. There are a variety of AV devices and multimedia content in the ubiquitous home and the sharing of content needs to have high quality of transfer by using optimized transport protocol for the each content. This paper presents the design and implementation of our proposed scheme that is based on UPnP AV framework which can support various transport protocol transparently. We validated and evaluated our proposed scheme and experimental results show that it can stream content effectively.

1 Introduction

These days, there are various multimedia services in the ubiquitous home as much as the users of home network are increasing[1]. There are also many kinds of AV devices in the ubiquitous home and these devices perform high quality services and have various multimedia content. Today's ubiquitous home provides technology to share and use these services and content with AV devices, regardless of where the contents are stored[2-3].

The Universal Plug and Play (UPnP) forum defined UPnP AV architecture as the standard of the interoperability of the home network for sharing multimedia content and controlling AV devices[4]. The UPnP AV architecture supports zero-configuration networking and automatically discovers devices that dynamically join a home network. It leverages TCP/IP and the Web to enable seamless proximity networking so, it communicates via HTTP. However this means UPnP AV architecture can not fully provide time-based multimedia services for audio and video content because the best feature of HTTP is reliable and in-order delivery of data. We propose enhanced UPnP AV architecture which can support various transport protocols for multimedia content such as audio and video.

In this paper, we describe the design and implementation of UPnP AV System that has transparent protocol scheme. This system can select transport protocol transparently regardless of file type of content. We adopt the UPnP AV architecture for interoperability of multimedia networking among home network devices and HTTP and RTSP for transport protocol of the multimedia content. By implementing the UPnP AV architecture with proposed module, our system provides not only the real-time streaming from the media server to the media renderer but also the guarantee of QoS for transfer of audio and video.

P. Thulasiraman et al. (Eds.): ISPA 2007 Workshops, LNCS 4743, pp. 153–162, 2007.

The rest of this paper organized in the following way: it starts with overview of the transparent protocol for audio and video on the application layer. Section 3 introduces UPnP AV architecture and section 4 presents a system design and an implementation of the proposed scheme. Section 5 will show the development and the evaluation of our UPnP AV System. Finally section 6 concludes with future works.

2 Real-Time Streaming Protocol

UPnP architecture is based on TCP and UDP and it uses HTTP to communicate and transfer data between devices. HTTP is a suitable protocol for data-sharing or communication but it is not suitable protocol for multimedia content such as audio and video. This section gives the knowledge about Real-time Streaming Protocol (RTSP) and the relationship with HTTP.

RTSP is a client-server application protocol for controlling the delivery of data with real-time properties[5]. It is developed by the IETF and published in 1998 as RFC 2326 and is used in streaming media systems for audio and video which allow client to control remotely a streaming media server, issuing VCR-line commands and allowing time-based access to files on a server[6]. RTSP uses UDP as default but it can also use TCP to deliver data if necessary. It uses TCP for player control messages and UDP for audio and video data.

RTSP is similar to HTTP in syntax and operation but it differs from HTTP in using protocol on the transport layer. HTTP is entirely based on TCP for web pages and TCP guarantees reliable and in-order delivery of data from sender to receiver. HTTP has rudimentary mechanisms for random access to files therefore it is not suitable for time-based seeking.

While HTTP is a stateless protocol, RTSP is a stateful. RTSP is designed to work with time-based media over UDP. UDP does not guarantee reliability and ordering that TCP does. UDP also does not have overhead of checking whether every packet arrived, it is faster and more efficient for time-sensitive purposes. RTSP has mechanisms for time-based seeks into media clips, with compatibility with many timestamp formats. In addition, RTSP is designed to control multicast deliver of stream and therefore, it could be a framework for multicast-unicast hybrid solutions for heterogeneous network like the Internet. The differences between two protocols result from their under layer protocols which are TCP and UDP.

3 UPnP AV Architecture

UPnP is a middleware solution proposed by Microsoft. The purpose is control and operating among network devices in home using IP network and HTTP. A device can dynamically join a network, obtain IP address, convey its capabilities, and learn about the presence and capabilities of other devices. Furthermore UPnP enabled device can control remote devices and transfer data to and from remote devices. In addition, device can leave a network smoothly and automatically without leaving any unwanted state behind.

UPnP AV framework defines three components that are Media Server, Media Renderer and Control Point, as depicted in Figure 1. Media Server functionality provides access to multimedia content and transfers them to other devices that are located in the network. It has Contents Directory, Connection Manager and AV Transport services. Media Renderer allows playback of a variety of rich media formats on devices. It has Rendering Control, Connection Manager and AV Transport services. Control Point allows user to discover and control devices in the network and multimedia content to flow between devices.

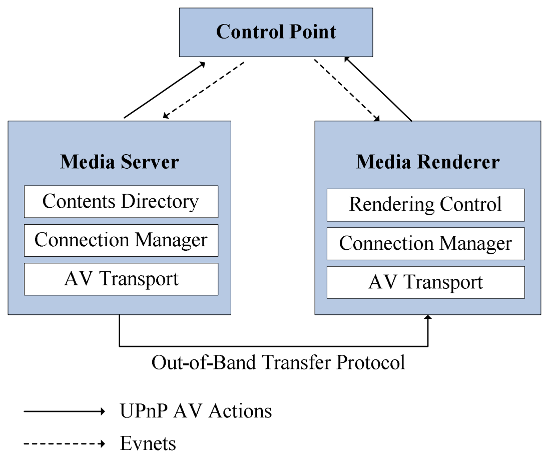


Fig. 1. UPnP AV Architecture

UPnP uses FIFO to transfer the messages among the components of UPnP AV framework. Control Point sends the requests of users to Media Server and Media Renderer and receives the event messages from them. And then Media Server streams multimedia content via HTTP to Media Renderer. Media Renderer requests the streaming to Media Server to playback multimedia content that is streaming from URL of contents that are known by Control Point. Media Server and Media Renderer send events which occur from each component to Control Point, and then users get information of multimedia content or messages from Control Point.

4 A Transparent Protocol Scheme

The main objectives of our implementation work are the validation of our proposed scheme and streaming contents between the media server and the media renderer via various transport protocols. This section describes the design, implementation and evaluation of our proposed scheme.

4.1 Our Proposed System Architecture

In the general AV system, which is based on UPnP AV framework, streams content only using HTTP-GET method. Standard UPnP AV framework defined by UPnP forum also communicates via HTTP. But it is not suitable transport protocol for time-based data such as audio and video that are used frequently in the home. For example, HTTP can not stream some kind of content such as mp4.

To overcome this problem, we proposed transparent protocol scheme and added Protocol Selection Module in the control point of our system. This module can support various transport protocols that are suitable for each content format. The architecture with Protocol Selection Module is shown in Figure 2. It consists of three components-Media Server, Media Renderer and Control Point and we integrated Media Renderer and Control Point into single GUI. The highlight of our system is the Protocol Selection Module that allows content to stream via the most suitable transport protocol for file type.

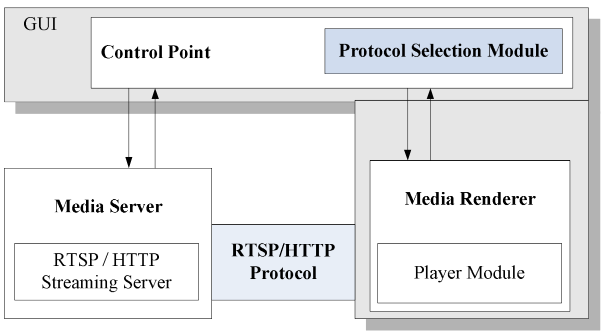


Fig. 2. Architecture of our UPnP AV System

Figure 3 helps to understand relationship between content and available transport protocols in the media servers. Media Server 1 has Content A and Content B, and Contents A can use Protocol a or Protocol b for streaming to the media renderer. Thus if Content A want to stream, then Protocol Selection Module selects transport protocol between available ones (Protocol a, Protocol b) considering the best suited for data transport.

Media Server 1	Content A	Protocol a
		Protocol b
	Content B	Protocol a
		Protocol c
		Protocol d
Media Server 2	Content C	Protocol b
	Content D	Protocol a

Fig. 3. Composition of content in the media server

Figure 4 is a flow diagram which shows how to work Protocol Selection Module according to transparent protocol scheme. At first, the control point discovers the media servers and the media renderers in the home network using UPnP standard functions (GetDivice(), GetDirectory(), GetContents().etc.). And then, the media servers and the media renderers reply to the control point with their information, which is the descriptions is written in XML, such as content directories and the specification of renderers (device type, supporting content formats. etc.). Protocol Selection Module in the control point prepares to stream the content via selecting the transport protocol which is suitable for content format and can be supported by renderering module. This module is out of the UPnP AV framework and it makes UPnP AV system serves adaptive AV service. The media renderer requests the content to the media server using URI which received from the control point. Finally the media server starts to stream the content to the media renderer via transport protocol which is selected by Protocol Selection Module.

Step 1, 2 perform when each UPnP device connects or disconnects into home network and step 3, 4, 5 perform at each service request. If the media renderer can not support the content format, the AV service is denied at step 3. If content can not use optimized transport protocol to stream, then it uses default protocol at step 5.

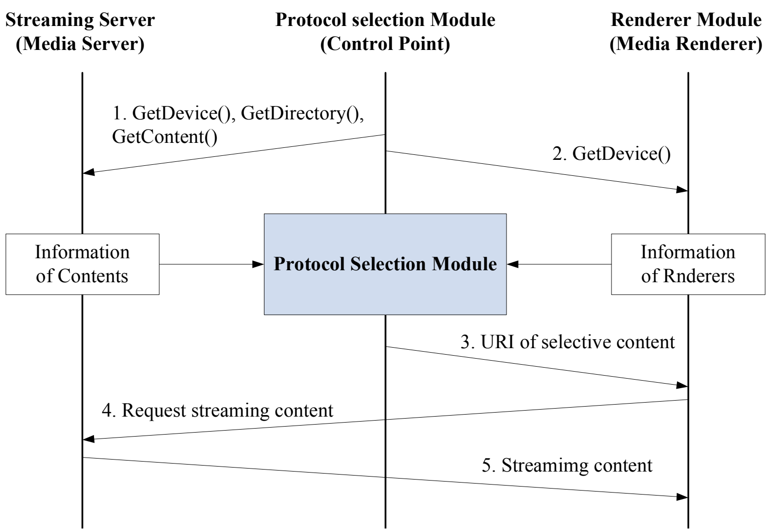


Fig. 4. Flow diagram of Protocol Selection Module

4.2 Operation of the Transparent Protocol Scheme

Our UPnP AV System consists of the media server, the media renderer and the control point like UPnP AV architecture does. We are implemented system except the media server so we used Intel AV Media Server. The media renderer used MPlayer for audio and video and ImageViewer for image as the player modules[7]. And our UPnP AV System used Freevo GUI module for the control point that is written in python. All modules of our system were relied on the UPnP API provided by the open source UPnP SDK for Linux and were formed of the C libraries.

Implementing Protocol Selection Module is key point of this paper. It extracts URI of the multimedia content from parsed DIDL (Digital Item Declaration Language) description. In most cases, only URI is enough to decide its transport protocol. But for reliability, our system further checks the file type of content. After the phase of checking it, Protocol Selection Module gives the URI of content to the media renderer, and then the media renderer requests media server to stream content via selective protocol. The following piece of XML code is the description of multimedia content.

```
<xsd:element name="CurrentURI">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="upnp:string">
        <xsd:anyAttribute namespace="##other"
          processContents="lax"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

5 Development

This section presents development of our enhanced UPnP AV System that has our proposed scheme, which can select transport protocol transparently. And we evaluate our system in comparison with the standard UPnP AV architecture.

5.1 System Development Based on UPnP AV Architecture

We validated the possibility of RTSP in the UPnP AV framework on our previous version[8]. Protocol Section Module in the new version can select transport protocol between HTTP and RTSP among a variety of transport protocols, because these two are the typical transport protocols in the application layer for the AV services in the home network.

RTSP is not supported by standard UPnP SDK, on the other side HTTP is. To support this on MPlayer, our system adopted live555 Library. There are some reasons for choosing of this library. First, it can be complied for various OS (UNIX, Window, QNX) can be used to build streaming applications[9]. The live555 Library can be used to stream, receive and process MPEG, H.263 or JPEG video, and several audio codec. Another reason is that it can be easily extended to support additional codec, and can also be used to build basic RTSP client and servers. And this library has been used to add streaming support to existing media player applications, such as MPlayer. This is the main reason that we adapted.

Figure 5 illustrates the demonstration. We ran Intel AV Media server on PC running Window and our UPnP AV System on PC running Linux. As shown, the user can choose content on the friendly GUI of the control point that integrated media renderer, as mentioned. The content are classified into categories that are stored in the media server. As the content plays, it is immediately delivered to the player module for rendering.

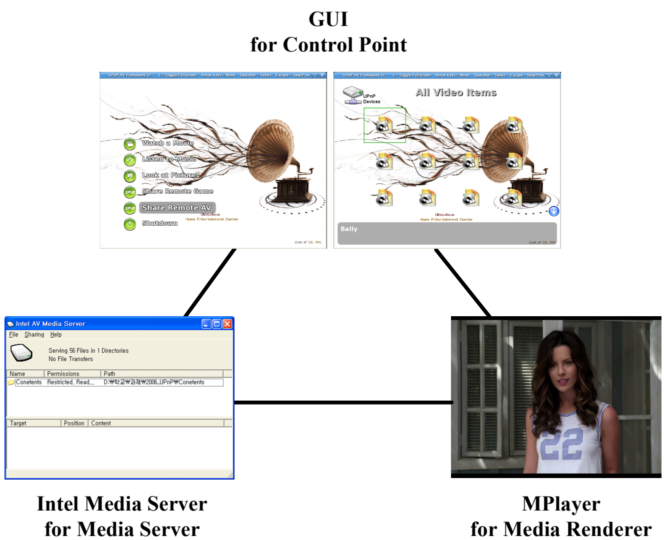


Fig. 5. Scenes of demonstration

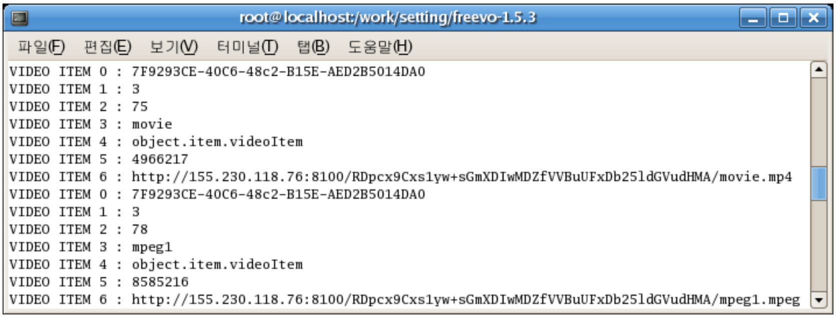
5.2 System Evaluation

In this section, we present the evaluation of transparent protocol scheme that can choose transport protocol by file type of contents and support various file types.

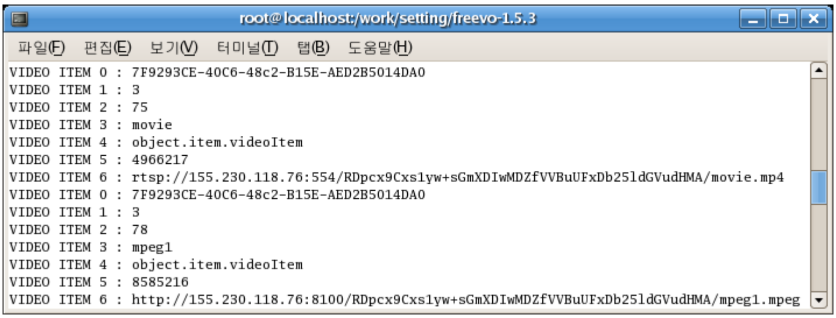
5.2.1 Validation of Scheme

Our implementation of the proposed scheme was tested against a number of functional trials. For test, we set up two PCs hosting the media server and the media renderer. The media renderer was integrated into the control point that has our Protocol Selection Module.

Figure 6 shows the URI of content that is stored in the media server which is detected by the control point. Figure 6(a) shows the URI list which is detected by standard UPnP AV architecture, Figure 6(b) shows it which is detected by our implementation system. As figures demonstrate, even if content is the same, the URI is not. Because our Protocol Selection Module select the most suitable transport protocol for content and makes URI of content. In result, the media server sends it to the control point and then the media renderer receives the content using that URI.



(a) URI of standard UPnP AV System



(b) URI of our UPnP AV System

Fig. 6. URI of content

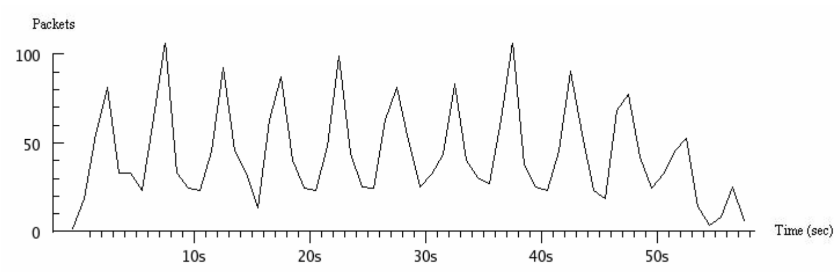
5.2.2 Analysis

Data communication between AV devices in the home network should be done and controlled via UPnP protocol on the IP layer. On this condition, HTTP works well via TCP and RTSP works well via UDP generally. We measured packet rates and byte rates on our UPnP AV System and standard UPnP AV architecture with mp4 video data using Ethereal[10]. The size of video data is 2.33M and we play it for 60sec.

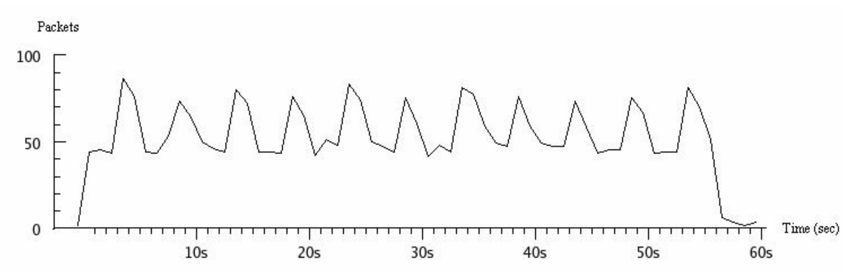
Figure 7(a) shows the packet flow on standard UPnP AV System and Figure 7(b) does ours. The horizontal line is time and the vertical line is number of packets. The following table summaries this experiment, our UPnP AV System that used RTSP achieves more packet rate and bit rate comparison with standard UPnP AV

Table 1. Comparison between systems

	Ours	Standard
Avg. packets/sec	56.017	43.353
Avg. bytes/sec	35574	34249
Avg. MBit/sec	0.285	0.274



(a) Packet flow on standard UPnP AV System



(b) Packet flow on our UPnP AV System

Fig. 7. Comparison of packet flow

architecture. This experiment validated RTSP is likely that the alternative application layer protocol for home network will support real-time streaming.

6 Conclusions

Transparent protocol scheme has been proposed in this paper. Based on UPnP AV architecture, this scheme enables a transport protocol selection service transparently and it provides a suitable protocol for the each content. This scheme that is described in this paper has already implemented. Our UPnP AV System is implemented on Linux, so it can be easily embedded in most AV devices. In other word, this system can be used widely in ubiquitous home network.

The implementation of the enhanced UPnP AV System for mobile devices is ongoing. Control Point for PDA is already implemented. So, a short-term perspective is to apply our framework to the various mobile devices. This makes real universal play in ubiquitous environment.

Acknowledgements

This research was supported by the MIC of Korea, under the ITRC support program supervised by the IITA(IITA-2006-C1090-0603-0026).

References

1. Jun, G.: Home Media Center and Media Clients for Multi-room Audio and Video Applications. In: Proceeding of Consumer Communications and Networking Conference, Las Vegas, pp. 257–260 (2005)
2. Giovanelli, F., Bigini, G., Solighetto, M.: A UPnP-based bandwidth reservation scheme for In-Home Digital Networks. In: Proceeding of International Conference on Telecommunications 2, 1059–1064 (2003)
3. Choi, S., Kang, D., Lee, J.: An UPnP based Media Distribution System supporting QoS in a Converged Home Network. Proceeding of Network Operations and Management Symposium, Vancouver, 1-4 (2006)
4. UPnP Forum.: http://www.upnp.org
5. RTSP.: http://www.rtsp.org
6. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.) RFC 2326, http://www.ietf.org/rfc/rfc2326.txt7
7. Mplayer.: http://www.mplayerhq.hu
8. Park, J., Lee, H., Kim, S., Kim, S.: UPnP AV framework for Real-time AV Streaming. In: The 6th International Conference on Applications and Principles of Information Science, Kuala Lumpur, pp. 174–177 (2007)
9. Live 555 Library.: http://live555.com/mplayer
10. Ethereal.: http://www.ethereal.com/