DLNA Protocol Inspection Tool for Compatibility Analysis

Se-Ho Park[1 2], Yong-Suk Park[2], Jeong-Wook Seo[3] and Jun-Rim Choi[1]
[1] Graduate School of Electronics Engineering, Kyungpook National University
Daegu, 702-701, Korea
[e-mail: sehopark@keti.re.kr, jrchoi@ee.knu.ac.kr]
[2] Contents Convergence Research Center, Korea Electronics Technology Institute
Seoul, 121-835, Korea
[e-mail: yspark@keti.re.kr]
[3] Department of Information and Communication Engineering, Samseoul Univesity
Chungcheongnam-do, 331-707, Korea
[e-mail: jwseo@nsu.ac.kr]
*Corresponding author: Se-Ho Park

Received April 7, 2014; revised June 1, 2014; accepted June 3, 2014; published June 27, 2014

Abstract

The propagation of smart devices and the rapid expansion of wireless mobile networks have increased the need for ubiquitous device connectivity and contents access. DLNA is currently being used as the de facto standard for connectivity of consumer devices in the digital home. The guidelines provided by DLNA are intended to facilitate device and contents discovery, management, sharing and distribution. However, consumer experience with DLNA has been rather poor and usability not up to speed. Different problems and issues in limited compatibility have been reported due to manufacturer customizations. Such compatibility issues are often difficult to identify by average consumers who are not savvy in information communications and multimedia technology. In this paper, the design and implementation of a DLNA protocol inspection tool is presented for compatibility analysis among DLNA certified devices. The tool monitors the home network and performs analysis of the underlying protocols used by DLNA. The tool can be used for diagnostics and troubleshooting, enabling the user to identify compatibility issues between devices or problems in the network.

Keywords: DLNA, protocol inspection, compatibility analysis, content sharing, home network

A preliminary version of this paper was presented at ICONI 2013 (“Design and Implementation of DLNA Compatibility Analysis Tool”) and was selected as an outstanding paper. This version includes changes made in the experimental setup and implementation and expands the results of the analysis. This research was supported by a research grant from the IT R&D program of MOTIE/KEIT [10041771, DLNA Automatic Test Certifiction Software Development for Contents Sharing Among Smart Devices].

http://dx.doi.org/10.3837/tiis.2014.06.017

1. Introduction

Diverse smart devices such as smartphones, tablets, and smart TVs have emerged over the past few years and have gained increasing popularity. The dissemination of such smart devices has increased the use and exchange of multimedia contents, leading to the need for easy media sharing and seamless contents access. The seamless display of multimedia contents among various devices is known as "N-screen" service. Many different technologies can be used for N-screen services. Many commercial service providers offer cloud web server based services to enable N-screen. A popular choice in the home networking environment is Digital Living Network Alliance (DLNA) technology.

In the home network, DLNA is one of the most widely used ways of multimedia sharing. Most of the networked multimedia consumer electronic devices in the market implements DLNA in one form or the other. DLNA enables you to stream media content between devices connected to the same home network, using either wired or wireless connection, without having to store the content on both devices [1]. All DLNA certified devices are tested for compatibility and interoperability before being made commercially available. However, consumer experience with DLNA has been rather poor and usability not up to speed. DLNA goes by many different commercial brand names such as "AllShare", "SmartShare", "Simple Share", etc. Therefore, consumers often do not realize that the devices that use these different brand names actually provide the same DLNA based technology and that they are supposed to be compatible with each other. In addition, manufacturers are implementing the standard pretty much according to what suits them, extending compatibility to formats that aren't included in the standard. Hence, despite being DLNA certified products, there are cases when products from different manufacturers are unable to share multimedia content, which leads to consumer frustration.

In this paper, the design and implementation of a DLNA protocol inspection tool for compatibility analysis is presented. The tool monitors the home network, detects DLNA devices connected in the network, and performs analysis of the underlying protocols used. The tool can be used to identify compatibility issues between devices or problems in the network. The user can diagnose and troubleshoot existing or potential problems in DLNA compatibility. The remaining of this paper is organized as follows. In Section 2, an overview of the DLNA guidelines is provided. Section 3 identifies the issues and problems that arise with DLNA usage in the consumer scenario. The design and implementation of the protocol inspection tool for DLNA, as well as its performance analysis, are provided in Section 4. Concluding remarks along with future work and improvements are commented in Section 5.

2. DLNA Overview

2.1 Protocols Used

The DLNA guidelines consist of several industry standards based protocols, such as Universal Plug and Play (UPnP), Internet Protocol (IP), Hyper Text Markup Language (HTML), General Event Notification Architecture (GENA), Simple Object Access Protocol (SOAP), Extensible Markup Language (XML), etc. [2]

UPnP is the key protocol used in DLNA guidelines and makes the overall operation of DLNA possible. The UPnP Device Control Protocol (DCP) framework is used for device and service discovery and control [3]. The DCP framework provides automatic network self-configuration, neighbor devices and capabilities discovery, and control functionality. The UPnP Audio/Video (AV) technology provides media management and control solution for DLNA [4]. It enables devices and applications to identify, manage, and distribute media content across the home network. The UPnP AV specifications define the interaction model between UPnP AV devices and associated control point applications.

2.2 Device Classes

The DLNA guidelines define three different types of device categories. Their respective device classes as shown in **Table 1**.

Table 1. DLNA device categories and classes

Category	Class
Home Network Devices	Digital Media Server (DMS) Digital Media Player (DMP) Digital Media Renderer (DMR) Digital Media Controller (DMC) Digital Media Printer (DMPr)
Mobile Handheld Devices	Mobile Digital Media Server (M-DMS) Mobile Digital Media Player (M-DMP) Mobile Digital Media Uploader (M-DMU) Mobile Digital Media Downloader (M-DMD) Mobile Digital Media Controller (M-DMC)
Home Infrastructure Devices	Mobile Network Connectivity Function (M-NCF) Media Interoperability Unit (MIU)

Digital Media Servers (DMSs) in the Home Network Device (HND) category store content and make it available to network. Digital Media Players (DMPs) find content offered by a DMS or Mobile Digital Media Server (M-DMS). DMPs provide playback and rendering capabilities and are not visible to other devices on the network such as Digital Media Controllers (DMC) or Mobile Digital Media Controllers (M-DMC). Digital Media Renderers (DMRs) are similar to DMPs in that they render or play content received from a DMS or M-DMS. However, DMRs are unable to find content on the network and must be set up by a DMC or M-DMC. DMCs find content offered by a DMS or M-DMS and match it to the rendering capabilities of a DMR. Digital Media Printer (DMPr) devices provide printing services to the DLNA home network.

Mobile Handheld Devices (MHDs) differ from HNDs in terms of interface used and media format supported. The functionality of M-DMS, Mobile Digital Media Player (M-DMP), and M-DMC correspond to DMS, DMP, and DMC, respectively. Mobile Digital Uploaders (M-DMUs) send content to an M-DMS or DMS with upload functionality. Mobile Digital Media Downloaders (M-DMDs) find and download content exposed by an M-DMS or DMS and play the content locally on the M-DMD after download.

Home Infrastructure Devices (HIDs) provide interoperability between the HNDs and MHDs. Mobile Network Connectivity Function (M-NCF) provides a bridge between mobile handheld device network connectivity and home network connectivity. Media Interoperability Unit (MIU) provides content transformation between required media formats for home network and mobile handheld devices. Fig. 1 shows the relationship among the different DLNA device classes in terms of content distribution and control signal flow.

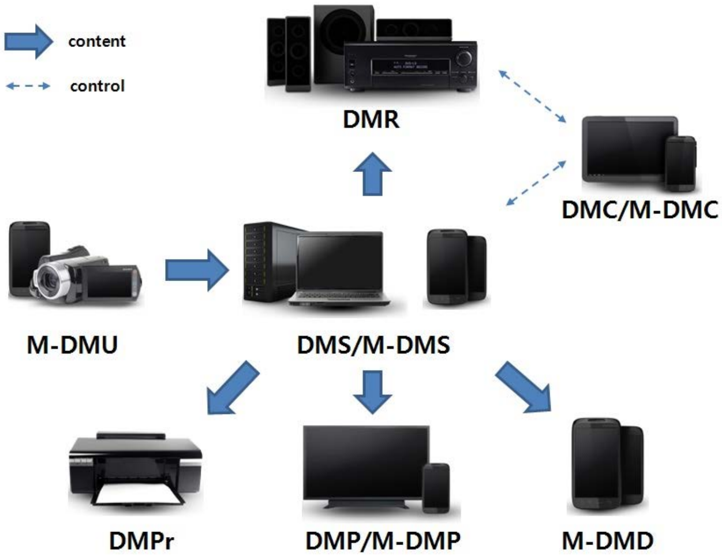


Fig. 1. Relationship among DLNA device classes

2.3 Media Format

The DLNA media format model is intended to achieve a baseline for interoperability. It defines a set of required media formats and a set of optional media formats for image, audio, and video with audio for both MHD and HND device categories as shown in Table 2.

Table 2. DLNA media formats

Media Formats	Required	Optional
Imaging	JPEG	GIF, TIFF, PNG
Audio (HND)	LPCM (2 channel)	MP3, WMA9, AC-3, AAC, ATRAC3plus
Audio (MHD)	MP3 and MPEG4 AAC LC	MPEG4 (HE AAC, AAC LTP, BSAC), AMR, ATRAC3plus, G.726, WMA, LPCM
Video (HND)	MPEG2	MPEG1, MPEG4, WMV9
Video (MHD)	MPEG4 AVC (AAC LC Assoc. Audio)	VC1, H.263, MPEG4 part 2, MPEG2, MPEG4 AVC (BSAC or other for Assoc. Audio)

Any DMP, M-DMP, DMR, M-DMD, and DMPr device must be able to receive content from any DMS or M-DMS device by supporting the mandatory formats designated in Table 2. A DMS or M-DMS device may stream content in its native format if the receiving device supports such native format. If the format is not supported, the DMS or M-DMS device should transcode the native format to one of the applicable required formats or to a format understood by the rendering device.

3. Issues in Compatibility and Interoperability

One of the key problems with DLNA is consumer product identification. The use of DLNA certified logo is not mandatory, so it is difficult to identify devices that implement DLNA. Even if DLNA functionality is implemented, it often goes by different names, such as Samsung’s “AllShare”, LG’s “SmartShare”, Sony’s “HomeShare”, Philips’ “Simple Share”, etc.

Support for additional media formats is an issue as well. In order to increase compatibility, DLNA has restricted media formats. However, customers have been constantly demanding additional popular formats such as DTS (Digital Theater System), MKV (MatrosKa multimedia container for Video), AVI (Audio Video Interleave), etc. DLNA is currently preparing an update on media formats, but this issue will persist as new and popular media formats emerge in the market.

Another problem is interoperability. The DLNA devices on the market have been DLNA certified, so certified devices are supposed to interoperate even if they are produced by different manufacturers. The certification is performed against some key devices with required functionalities and mandatory media formats. Since DLNA is an “open” standard, there is no restriction on extending the compatibility to formats that aren’t included in the standard. The manufactures want to support popular media formats in use that are not part of the standard to increase their product competitiveness. This leads to market fragmentation. Manufacturers use different implementations, and support different media codecs. Therefore, often device the interoperability works only among products from the same manufacturer.

In addition, there is the usability issue. End-users are not necessarily technically savvy, and it can be hard to distinguish the different DLNA device classes. DLNA is supposed to be simple plug & play, zero-configuration. However, getting communication between a client and server set up can sometimes be confusing and simply proves impossible for those who don’t have a minimum of knowledge in terms of multimedia files. Consumer user experience becomes a frustration since no feedback or information is given from the devices about the problem.

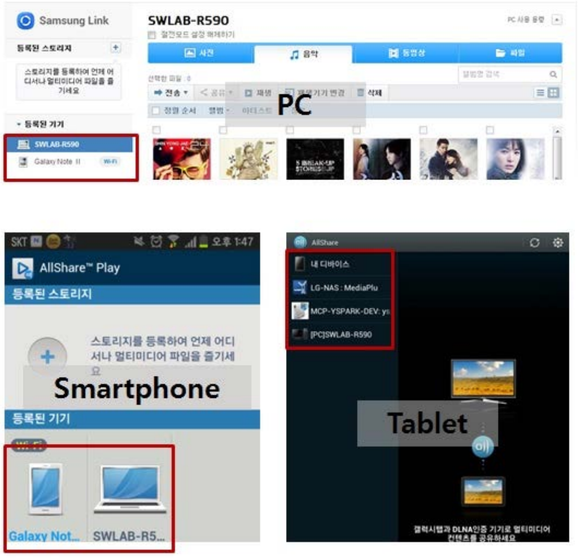


Fig. 2. Illustration of DLNA device discovery problem

Fig. 2 illustrates a simple example of device discovery problem. Three DLNA devices are connected to the same network. The PC is able to find the smartphone, but unable to see the tablet. The tablet on the other hand has seemingly discovered the PC. All the user can do is to refresh the device list until it shows up, if it ever shows up. Situations like these can lower user experience and expectations with DLNA. **Table 3** summarizes issues and problems often encountered by consumers using DLNA products. The device class shown is as first identified by the consumer, but the actual problem may reside on other interworking devices.

Table 3. Problems and issues in DLNA encountered by consumers

Device Class	Problems
DMS	Album images not displayed for certain music files Seek command not working during audio play Subtitles are not displayed
DMR	Multimedia play stops abruptly Network disconnection after a few minutes of streaming play
DMC	Folders are shown as empty Unable to play certain video formats Unable to play certain audio formats Listing of contents are duplicated Unable to connect with certain DMSs or DMRs Multimedia files are recognized but unable to play Long delays (7 to 30 seconds) until start of multimedia play

Various works have been previously carried out to identify DLNA operation issues and to improve device interoperability. In order to improve DLNA interoperability, [5] presents a scheme to dynamically match capabilities of content sinks and sources based on media formats. The scheme proposes the use of transcoders located in external networks such as Internet when a match is not locally found. In [6], the authors propose a way to detect anomalous events by analyzing the users' operation history of DLNA devices using ARP spoofing. A collaborative content play method is proposed in [7] that automatically associates the main-content with the sub-content based on attribute information matching in order to enhance user entertainment experience. While the existing works focus mainly on analyzing content attributes, the proposed DLNA Inspection Tool goes a step further and uses the information to construct a topology of contents and devices to facilitate compatibility and interoperability analysis to the user.

4. DLNA Inspection Tool

4.1 Tool Design

In order to provide the user with useful information about the devices connected in the home network, the type of media formats supported, as well as analysis on errors and misoperation, the following design of the inspection tool is proposed, illustrated in **Fig. 3**.

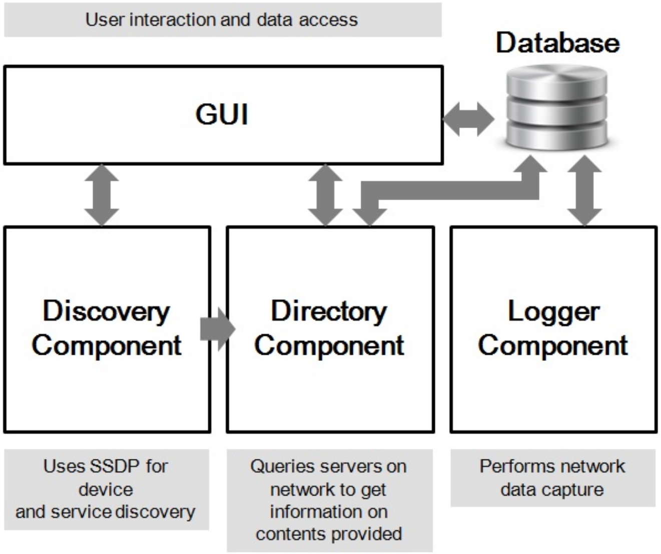

User interaction and data access
Database
GUI
Discovery Component
Directory Component
Logger Component
Uses SSDP for device and service discovery
Queries servers on network to get information on contents provided
Performs network data capture

Fig. 3. High-level block diagram of the inspection tool

The Discovery Component uses SSDP (Simple Service Discovery Protocol) to discover devices on the network. Both active discovery using M-SEARCH and passive discovery by observing NOTIFY are used. The tool can issue M-SEARCH to get responses directly from devices on the network and listen to self-notifications of devices newly joining the network or updating their status. The information on devices discovered is passed to the Directory Component.

The Directory Component queries the servers on the network to return information on the contents provided, including the protocols and media formats supported. This is done by using the server's ContentDirectory::Browse() or Search() actions. The Directory Component also queries the renders to return transfer protocols and media formats supported. The render's ConnectionManager::GetProtocolInfo() action is used to perform this. Then, the Directory Component matches the protocols and formats from servers and renderers and stores the information in the database. For each media type and for each transport protocol type, the lists of servers and clients that support them are maintained in the database. This facilitates device and content matching.

The Logger Component performs network data capture and stores network traffic data in the database. The Logger Component filters only DLNA relevant traffic. The filtering is based on devices discovered using SSDP. The stored traffic data is used for error detection and analysis. The Logger Component is also required in order to observe and analyze content transfer data, since the server and client use out-of-band transfer protocol to directly transmit content.

A high level graphical user interface (GUI) is also provided to enable easy user interaction and access to data.

4.2 Test Environment Setup

The basic experimental environment is illustrated in **Fig. 4**. A wireless access point router manages connection of wireless DLNA devices. It also dynamically assigns local IP address to all DLNA devices on the network through its DHCP (Dynamic Host Configuration Protocol) server. A dummy hub is connected to the wireless access point router via a single Ethernet cable. All wired DLNA devices are connected to the dummy hub, including the DLNA inspection tool.

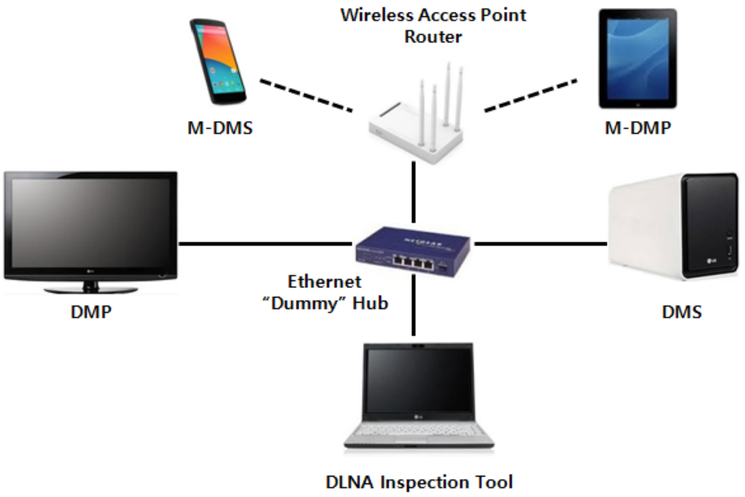
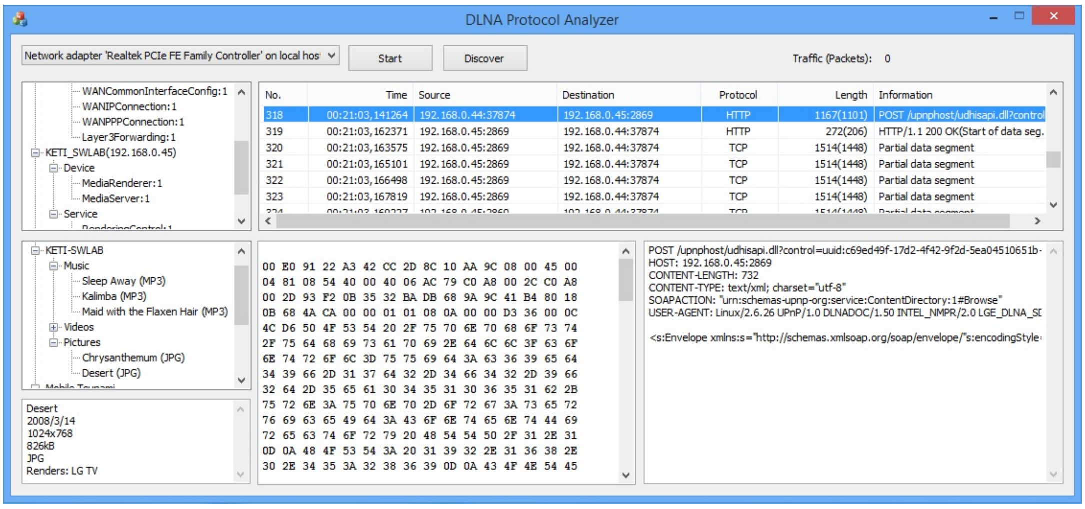


Fig. 4. Test environment setup

A dummy hub is used in order to facilitate network packet capture. Most network devices today operate in switching mode, which means that unicast traffic will be forwarded only to its associated or connected port. This increases network performance, but capturing of all traffic data within the network is not possible. The use of a dummy hub creates a shared media network allowing all traffic data to be received on all its ports. Therefore, the DLNA inspection tool with its network adapter operating in promiscuous mode will be able to see and capture all packets in the network.

4.3 Implementation and Results

The DLNA protocol analysis tool was implemented on a PC running Windows operating system. **Fig. 5** shows a snapshot of the initial implementation of the DLNA inspection tool.



No.	Time	Source	Destination	Protocol	Length	Information
318	00:21:03,141264	192.168.0.44:37874	192.168.0.45:2869	HTTP	1167(1101)	POST /upnphost/udhisapi.dll?control
319	00:21:03,162371	192.168.0.45:2869	192.168.0.44:37874	HTTP	272(206)	HTTP/1.1 200 OK(Start of data seg.
320	00:21:03,163575	192.168.0.45:2869	192.168.0.44:37874	TCP	1514(1448)	Partial data segment
321	00:21:03,165101	192.168.0.45:2869	192.168.0.44:37874	TCP	1514(1448)	Partial data segment
322	00:21:03,166498	192.168.0.45:2869	192.168.0.44:37874	TCP	1514(1448)	Partial data segment
323	00:21:03,167819	192.168.0.45:2869	192.168.0.44:37874	TCP	1514(1448)	Partial data segment

```
00 E0 91 22 A3 42 CC 2D 8C 10 AA 9C 08 00 45 00
04 81 08 54 40 00 40 06 AC 79 C0 A8 00 2C C0 A8
00 2D 93 F2 0B 35 32 BA DB 68 9A 9C 41 B4 80 18
0B 68 4A CA 00 00 01 01 08 0A 00 00 D3 36 00 0C
4C D6 50 4F 53 54 20 2F 75 70 6E 70 68 6F 73 74
2F 75 64 68 69 73 61 70 69 2E 64 6C 6C 3F 63 6F
6E 74 72 6F 6C 3D 75 75 69 64 3A 63 36 39 65 64
34 39 66 2D 31 37 64 32 2D 34 66 34 32 2D 39 66
32 64 2D 35 65 61 30 34 35 31 30 36 35 31 62 2B
75 72 6E 3A 75 70 6E 70 2D 6F 72 67 3A 73 65 72
76 69 63 65 49 64 3A 43 6F 6E 74 65 6E 74 44 69
72 65 63 74 6F 72 79 20 48 54 54 50 2F 31 2E 31
0D 0A 48 4F 53 54 3A 20 31 39 32 2E 31 36 38 2E
30 2E 34 35 3A 32 38 36 39 0D 0A 43 4F 4E 54 45
```

```
POST /upnphost/udhisapi.dll?control=uuid:c69ed49f-17d2-4f42-9f2d-5ea04510651b-
HOST: 192.168.0.45:2869
CONTENT-LENGTH: 732
CONTENT-TYPE: text/xml; charset="utf-8"
SOAPACTION: "urn:schemas-upnp-org:service:ContentDirectory:1#Browse"
USER-AGENT: Linux/2.6.26 UPnP/1.0 DLNADOC/1.50 INTEL_NMPR/2.0 LGE_DLNA_SI

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"s:encodingStyle=
```

Desert
2008/3/14
1024x768
826kB
JPG
Renders: LG TV

Fig. 5. Implementation of the DLNA inspection tool

To perform analysis of DLNA data by capturing network traffic, WinPcap was used [8]. WinPcap provides a Windows OS based network interface API (Application Program Interface). Network packets are captured from the network adapter set to operate in promiscuous mode. The captured data is saved to a database. SQLite was used as the database for its simplicity of use. In addition, unlike other client-server databases, SQLite can be embedded into the program and become an integral part of it. Information on the data captured can be retrieved from the database for further analysis.

The right-top window of the GUI displays IP traffic captured. In order to reduce computation and processing burden, the IP stream is filtered and only packets that are relevant to DLNA are captured. If a particular data is selected, the captured data is retrieved from the database, and its hexadecimal data and payload information are displayed in the windows below.

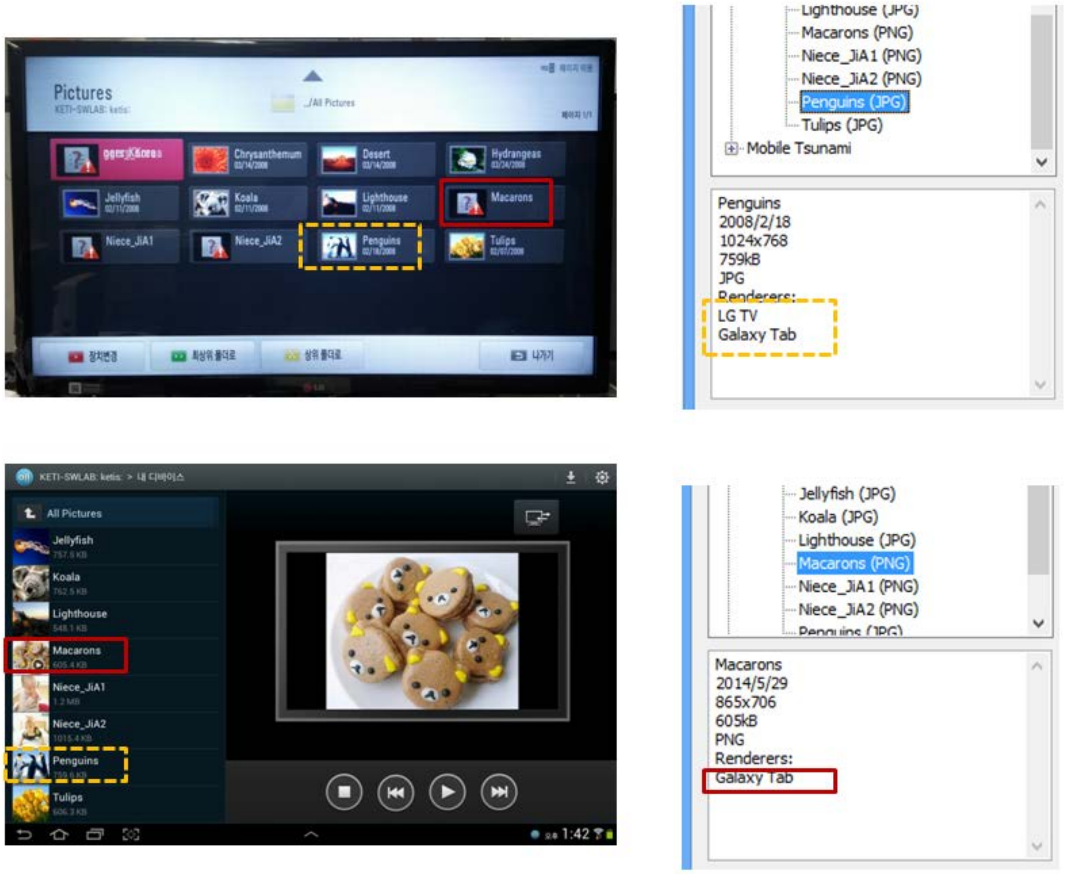
The three windows on the left side show DLNA device information in the network. The top window displays device and service information for each DLNA device discovered. This information is obtained by analyzing SSDP relevant traffic data. The middle window shows all media files discovered. Content Directory information is processed to extract information on media formats supported and media files currently available in the network. The last window displays additional information about a device, service, or content selected. For media files, it also provides information about compatible devices.

This initial implementation is able to detect devices and list the available multimedia contents, as well as identify compatibility in media formats. Currently, the GUI lists the devices and provided contents, and shows the details about media formats. The GUI also shows all DLNA relevant data traffic, so analysis on network or protocol relevant problems can be performed by tracing the data captured. Since manual data tracing of captured data can be a tedious task, we intend to enhance the tool and include protocol flow graphs in order to automate and facilitate analysis of traffic flow.

In addition, the use of Address Resolution Protocol (ARP) spoofing may be necessary for more detailed analysis as described in [9]. ARP spoofing is a technique where fake ("spoofed") ARP messages are sent onto a Local Area Network. The aim is to associate the analysis tool's MAC address with the IP addresses of other DLNA devices, causing any traffic meant for those IP address to be sent to the analysis tool instead. ARP spoofing is an effective method to collect all detailed information in a home network, since all traffic will be sent to the analysis tool before reaching their actual destinations. Currently, the analysis tool is limited to data monitoring and information gathering. ARP spoofing capability will enable active testing of devices by enabling change and manipulation of commands and events.

4.4 Performance Analysis

Fig. 6 shows the functional performance of the tool implemented. A DMS (KETI-SWLAB) is sharing content in the home network. A DMP (LG TV) and M-DMP (Galaxy Tab) are accessing content from the DMS. The DMP is unable to process certain formats, and displays some contents as question marks. The user isn’t given any further information about these files. By using the DLNA Inspection Tool, the user can easily identify the content information, as well as, see which renderers or players in the home network are capable of playing the selected content. It can be seen that for the selected “Penguins” content, 2 renderers (LG TV and Galaxy Tab) in the home network are able to process it. For the selected “Macarons” content, only 1 renderer (Galaxy Tab) is able to process it. Therefore, by using the tool the user can identify compatibility and interoperability issues easily.



Lighthouse (JPG)
Macarons (PNG)
Niece_JiA1 (PNG)
Niece_JiA2 (PNG)
Penguins (JPG)
Tulips (JPG)
Mobile Tsunami

Penguins
2008/2/18
1024x768
759kB
JPG
Renderers:
LG TV
Galaxy Tab

Jellyfish (JPG)
Koala (JPG)
Lighthouse (JPG)
Macarons (PNG)
Niece_JiA1 (PNG)
Niece_JiA2 (PNG)
Penguins (JPG)

Macarons
2014/5/29
865x706
605kB
PNG
Renderers:
Galaxy Tab

Fig. 6. Compatible content-device identification

In order to show the usability of the tool in real-time environment, **Fig. 7** shows the performance of the tool implemented in terms of processing time. The measurements were taken from a PC with Intel Core i7-2637M CPU @ 1.70 GHz and 4.00 GB of RAM, running 64-bit Windows OS. The traffic segment was captured during music streaming from one device to another. The commands and events portion includes device discovery and identification and service and content information gathering. The data is parsed and the relevant information is saved to the database for analysis. A particular music file is then selected for out-of-band data transfer.

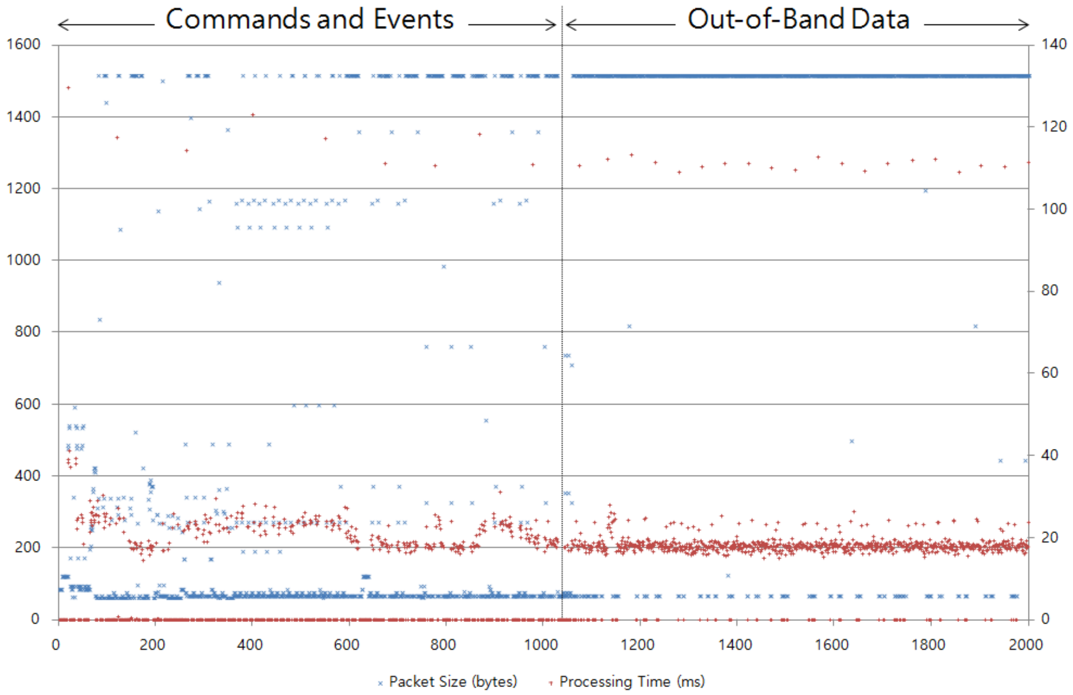


Fig. 7. Performance of the DLNA inspection tool

The maximum size of the data ranges up to 1,514 bytes which is the maximum length of an Ethernet packet. The processing of events and commands takes approximately 25 ms. Processing times close to 0 ms is an indication that the packet has been dropped due to it not being a DLNA relevant data. Processing of bulk multimedia traffic (i.e. actual payload data of streaming music) or out-of-band data takes approximately 20 ms. This is shown starting around at sample 1,000. Most of the processing time incurred at this time is for database write operation.

It can be seen that at certain points processing time reaches over 100 ms. These are related to database access and management. The accumulation of such processing time creates delays which may lead to potential IP packet loss. In order to not lose any IP data traffic, internal buffer management is performed. To show the effects of database access and processing, performance measurements have also been made with database access disabled. Therefore, in this second case pure DLNA information processing time is captured.

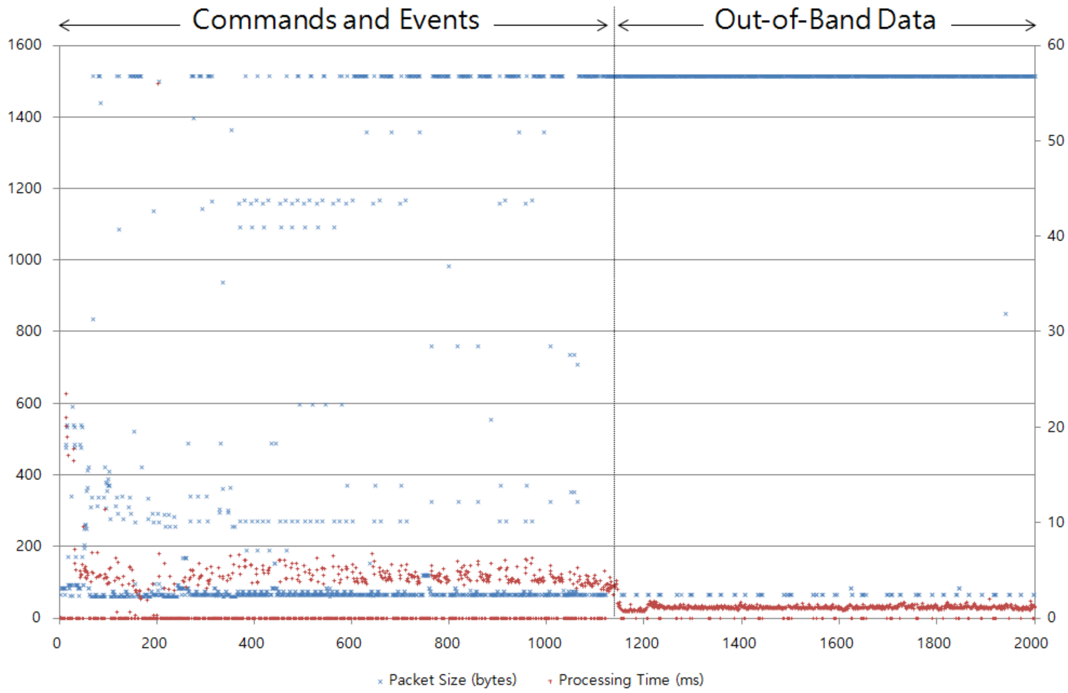


Fig. 8. Performance of the tool without database access

Fig. 8 shows the performance of the tool with database access disabled. It can be seen that the data processing time is reduced to an average of approximately 3 ms compared with the full database access version which measured about 25 ms. The processing of commands and events takes approximately 5 ms, which is shown in the first half of the graph up to sample 1,150. The processing of bulk multimedia or out-of-band data takes approximately 1 ms, which is shown in the latter half of the graph starting after sample 1,150. Processing of bulk multimedia takes less time, since active data parsing or processing does not take place at this point, and the packet is bypassed once identified as multimedia data. Therefore, it can be seen that relatively large processing time is incurred in database access. However, overall, the delay is not significant to affect real-time analysis.

5. Conclusion

In this paper, the design and implementation of a DLNA protocol inspection tool is presented in order to identify compatibility issues between devices and problems in the network. The tool monitors the network, captures DLNA relevant traffic, and performs analysis of the protocols used by DLNA. The tool discovers and identifies DLNA devices on the network and creates a topology of services and media contents provided.

Based on the media type classification information provided by the tool, the user is able to identify device compatibility. Since the tool also captures all IP traffic relevant to DLNA, networking or protocol problems can also be identified. Currently, protocol errors need to be analyzed by manually tracing the data captured. Therefore, as a next step, we intend to provide protocol flow graphs in order to automate and facilitate analysis of traffic flow between devices.

References

[1] DLNA, “DLNA for HD Video Streaming in Home Networking Environments,” DLNA Whitepaper, May 2011.

[2] DLNA, “DLNA Guidelines Part 1: Architectures and Protocols,” December, 2011.

[3] UPnP Forum, “UPnP Device Architecture 1.0,” October, 2008.

[4] UPnP Forum, “UPnP AV Architecture,” December, 2010.

[5] F. Matsubara, T. Hanada, S. Imai, S. Miura, and S. Akatsu, “Networked Device Capability And Content Media Format Matching Scheme For Multimedia Access,” *IEEE Transactions on Consumer Electronics*, vol. 53, no. 1, pp. 145-149, February, 2007. Article (CrossRef Link)

[6] S. Baek, H. Si, S. Saruwatari, H. Morikawa, J. Hjelm, and T. Oda, “Anomaly Detection from the Operation History of DLNA Devices”, *2010 Second International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 390-393, June 2010. Article (CrossRef Link)

[7] K. Tasaka, N. Imai, M. Isomura, and K. Yoshihara, “Implementation and Evaluation of a Collaborative Content Play Method in a Home Network”, *2010 14th International Conference on Intelligence in Next Generation Networks (ICIN)*, pp. 1-6, October 2010. Article (CrossRef Link)

[8] F. Risso and L. Degioanni, “An Architecture for High Performance Network Analysis”, in *Proc. of the 6th IEEE Symposium on Computers and Communications (ISCC 2001)*, pp. 686-693, July 2001. Article (CrossRef Link)

[9] S. Saruwatari, J. Hjelm, T. Oda, and H. Morikawa, “A System for Logging Operation Histories of DLNA Devices by Combining ARP Spoofing and SSDP”, *IEEE International Conference on Consumer Electronics (ICCE)*, pp. 233-234, January 2011. Article (CrossRef Link)

Se-Ho Park is a senior researcher at the Contents Convergence Research Center, Korea Electronics Technology Institute (KETI), Seoul, Korea. Before joining KETI in 2005, he was with I&C Technology and Samsung Electronics, where he worked in projects relevant to SoC (System on Chip) for digital broadcasting and wireless networks. He received his B.S. and M.S. degrees in electrical engineering from Kyungpook National University in 1998 and 2000, respectively. Currently, He is doing his Ph.D. at Kyungpook National University. Daegu, Korea. His current research interests are in the areas of digital broadcasting and wireless, mobile networks.

Yong-Suk Park is a senior researcher at the Contents Convergence Research Center, Korea Electronics Technology Institute (KETI), Seoul, Korea. Before joining KETI in 2003, he was with I&C Technology and Samsung S1, where he worked in projects relevant to wireless networks and system integration. He received his B.S. and M.S. degrees in electrical and computer engineering from Carnegie Mellon University in 1997 and 1998, respectively. He is currently working towards a Ph.D. degree in the School of Electrical & Electronic Engineering from Yonsei University, Seoul, Korea. His current research interests are in the areas of media sharing and contents delivery networks.

Jeong-Wook Seo received the B.S. and M.S. degrees from the Department of Telecommunication and Information Engineering, Korea Aerospace University, Gyeonggi, Korea, in 1999 and 2001, respectively, and the Ph.D. degree from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea, in 2010. He is currently an Assistant Professor at the Department of Information and Communication Engineering, Namseoul University, Chungcheongnam-do, Korea. From 2001 to 2013, he was with Network Convergence Research Center in Korea Electronics Technology Institute, Seoul, Korea. His research interests include statistical signal processing for communications and networking, protocol design for Machine-to-Machine/Internet of Things, and software development of next-generation broadcasting and communication systems.

Jun-Rim Choi received the B.S. degree from Yonsei University, Seoul, Korea, in 1986, the M.S. degree from Cornell University, Ithaca, NY, in 1988, and the Ph.D. degree from University of Minnesota, Twin Cities, in 1991, all in electrical engineering. His current interest is in the design of high-speed, low-power digital circuits for DSP cores, digital filters, and numerical processors. Since 1997, he has been a Professor at Kyungpook National University, Taegu, Korea.