
실시간 미디어 전송을 위한 UPnP AV 프레임워크

UPnP AV frameworks that supports advanced streaming

박지윤, Jiyun Park, 이현주, Hyunju Lee, 김상욱, Sangwook Kim.
경북대학교 컴퓨터과학과

요약 본 논문은 UPnP 기반의 홈 네트워크 환경에서 실시간 미디어 전송을 위한 UPnP AV 프레임워크를 설계하고, 디지털 엔터테인먼트 센터를 구현하였다. 가정에서의 오디오, 비디오 콘텐츠의 사용은 날로 증가하고 있으며, 사용자들은 이들 콘텐츠의 실시간 전송 서비스를 요구하고 있다. 기존의 UPnP AV 프레임워크에서의 콘텐츠를 HTTP-GET 스트리밍 방식으로 전송 하였으므로 실시간 서비스를 하기에는 한계가 있었다. 본 연구에서는 RTP/RTSP 스트리밍 방식으로 콘텐츠 전송 서비스가 가능한 확장된 UPnP AV 프레임워크를 구현 하였다. 그 결과 사용자는 실시간 미디어 전송 서비스 이용이 가능하게 되었다.

핵심어: UPnP, HTTP, RTP/RTSP

1. 서론

홈 네트워크 도입은 점차 증가하고 있으며, 사용자 또한 증가하고 있다. 사용자들은 홈 네트워크를 통해 다양한 서비스들을 제공받고 이용하고 있으며, 특히 오디오, 비디오와 같은 멀티미디어 콘텐츠 사용률은 급격히 증가하고 있다[1]. 앞으로의 홈 네트워크에는 사용자 중심의 멀티미디어 서비스가 중심을 이룰 것이며 이를 지원할 수 있는 통합 AV 프레임워크 기술이 요구된다.

홈 네트워크 환경에 연결된 디바이스 간의 멀티미디어 콘텐츠를 공유, 컨트롤 할 수 있게 하는 프로토콜로써 Universal Plug and Play (UPnP)를 UPnP Forum에서 정의하고 발표하였다[2]. UPnP는 UPnP 디바이스 컨트롤 프로토콜로 이들을 인터넷 통신 환경에서 가능하게 하였다.

우리는 홈 네트워크에서 사용할 수 있는 다양한 콘텐츠들 중 오디오, 비디오와 같은 멀티미디어 콘텐츠에 맞는 UPnP AV 프레임워크를 제안하고 구현하였다[3]. 이는 리눅스 환경에서의 구현된 UPnP AV 프레임워크를 기반으로 UPnP 디바이스 제어를 위한 통합 멀티미디어 시스템이다. 사용자는 미디어 서버, 미디어 렌더러, 컨트롤 포인트로 구성된 시스템을 통하여 홈 네트워크 내에 존재하는 AV 콘텐츠를 이용할 수 있으며 디바이스의 UI 화면을 통해 제어, 실행할 수 있다.

기존에 구현된 UPnP AV 프레임워크에서의 디지털 콘텐츠는 HTTP-GET 방식으로 스트리밍 되고 있었다. 그러나 가정 내에서의 콘텐츠 이용은 오디오, 비디오 데이터가 대부분이며 이들 콘텐츠들은 실시간 스트리밍을 필요로 한다. 이를 위해 우리는 구현한 UPnP AV 프레임워크를 RTP/RTSP와 같은 실시간 전송 프로토콜을 이용하여 스트리밍 서비스를 제공하는 확장형 미디어 재생 프레임워크로 발전 시켰다. 그 결과 실시간 미디어 전송이 가능하게 됨으로써 사용자 중심의 디지털 엔터테인먼트 센터를 구현할 수 있게 되었다.

본 논문의 2절에서는 UPnP AV 프레임워크와 실시간 미디어 전송 제어를 위한 프로토콜에 대해서 알아보고, 3절에서는 실시간 미디어 전송을 위한 프레임워크 설계를 UPnP 모듈별로 살펴본다. 제 4절에서는 설계를 바탕으로 구현된 결과를 실행 화면과 함께 살펴보고 5절에서 결론을 맺는다.

2. 관련 연구

UPnP 구조는 TCP/IP를 사용하는 네트워크에서 네트워크로 연결된 디바이스들 사이에 데이터를 전송하고 컨트롤을 가능하게 해준다. UPnP 구조 내에서 실시간 미디어 전송을 위해서는 Real-time Transport Protocol(RTP)와 Real-time

Streaming Protocol(RTSP)와 같은 멀티미디어 콘텐츠 전송을 위한 프로토콜이 필요하다[4].

2.1 UPnP AV Framework

가정에서의 디지털 엔터테인먼트 서비스는 맥내에 산재하여 존재하는 멀티미디어 콘텐츠들을 홈 네트워크 망을 통하여 집안의 곳곳에서 서비스를 받는 기술이다. 가정에 존재하는 다양한 A/V 디바이스 간에 콘텐츠를 공유하여 멀티미디어 서비스를 받기 위해서는 장치들 간 상호 운용성이 중요하다. UPnP AV 프레임워크는 현재 가정에서 디지털 콘텐츠를 향유하기 위한 미들웨어 기술로서, 가정 내에 존재하는 인텔리전트 가전(intelligent appliances), 무선 디바이스, PC 간에 peer-to-peer 네트워크 연결을 위한 표준 소프트웨어 기술인 UPnP(Universal Plug and Play)에 기반하여 A/V 디바이스 간 멀티미디어 서비스를 제공한다.

UPnP는 다양한 미디어 및 복수의 벤더가 존재하는 홈 네트워크 환경에서 운영체제, 언어 및 하드웨어에 독립적인 서비스 환경을 제공하는 미들웨어이다. 기존의 IP 네트워크와 HTTP 등의 표준 인터넷 프로토콜을 사용하여 홈 네트워크 상의 기기들이 제공하는 서비스를 자동으로 발견하고, 또한 사용할 수 있도록 한다. UPnP는 단순하고 유연하며 표준에 기반한 peer-to-peer 방식의 연결을 제공한다. 그러므로 사용자는 단지 장치를 네트워크에 연결시키기만 하면 네트워크 상에 연결된 기존 디바이스들이 자동으로 새로 추가된 장치를 발견한다. 디바이스가 네트워크에 연결되면 해당 디바이스를 제어하거나 다른 디바이스가 가진 서비스를 검색할 수 있다.

UPnP 포럼에서는 홈 네트워크 내에서 디지털 콘텐츠를 분산하여 공유하기 위한 미들웨어 기술로 UPnP AV 프레임워크를 정의하였으며, 이를 통해 TV, VCR, DVD 플레이어, 스테레오 시스템, MP3 플레이어와 PC와 같은 CE(Consumer Electronic) 디바이스를 타깃으로 하여 일련의 UPnP 디바이스와 서비스 템플릿을 제공한다.

네트워크가 구성되지 않은 이전 홈 엔터테인먼트 환경에서 가정 내 A/V 디바이스들은 RCA 케이블과 같은 전용 케이블을 사용하여 상호 연동한다. 이때, UPnP AV 프레임워크는 디바이스 간에 상호 연동하기 위하여 전용 아날로그 케이블 대신에 디지털 네트워크를 사용할 수 있게 한다. UPnP AV 프레임워크에서는 HTTP, XML, SOAP, SSDP와 GENA를 포함하는 인터넷 프로토콜의 사용으로 peer-to-peer IP 네트워크를 통해 디지털 콘텐츠를 전송한다. 이러한 peer-to-peer 네트워크 구조는 각 디바이스가 네트워크 상에서 서버 또는 클라이언트 디바이스 모두의 역할을 할 수 있다. 한 예로 네트워크 상의 다른 장치에 콘텐츠를 제공하면서 다른 장치에 저장되어 있는 콘텐츠를 전송 받아 재생할 수 있는 PVR을 들 수 있다. 이를 통해 디지털 홈 네트워크 전반에 분산된 A/V 콘텐츠를 CE 디바이스에서 볼 수 있다.

UPnP AV 프레임워크의 대표적인 장점으로는 다음과 같다. UPnP AV는 분산 환경하에서 개방형 네트워크 프레임워크이며, 표준 TCP/IP를 사용함으로써, 기존의 네트워크와의 접근에 유리하다. 또한 이미 사용되는 XML, HTTP, SOAP 등의 기존 프로토콜을 사용함으로써, 비용의 절감과 타 네트

워크와의 연동에 유리하다. 그리고 특정 OS, 프로그래밍 언어, 미디어 콘텐츠 접근 기술에 독립적이므로, 제품 개발자와 서비스 제공자 모두에게 유리하다.

UPnP AV는 홈 네트워크에서 디바이스를 제어하는 모델을 정의한다. 이러한 모델을 통해 사용자는 디지털 콘텐츠를 다운로드하고 저장해둔 PC를 콘텐츠 서버로 사용할 수 있다. 콘텐츠는 오디오, 비디오뿐만 아니라 텍스트, 그래픽, 사진 등과 같은 다양한 콘텐츠 포맷도 사용 가능하다. 이러한 콘텐츠를 UPnP AV 프레임워크 내에 존재하는 A/V 디바이스로 전송하여 재생한다. UPnP AV 프레임워크는 홈 네트워크 내에 존재하는 디바이스 사이에 멀티미디어 콘텐츠를 공유하여 사용하기 위한 미들웨어로서 미디어 서버, 미디어 렌더러와 컨트롤 포인트 모듈로 구성된다.

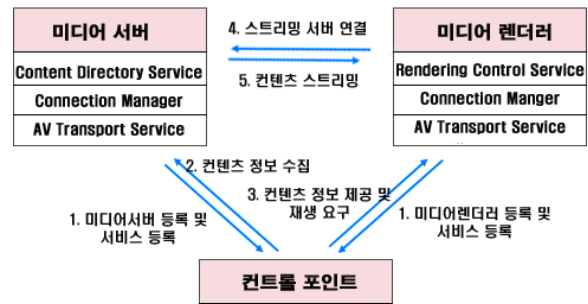


그림 1. UPnP AV 프레임워크 구조 및 동작 흐름

UPnP AV는 미디어 서버와 미디어 렌더러, 두 가지 디바이스 타입을 정의한다. 미디어 서버는 멀티미디어 콘텐츠를 저장하고 있거나 콘텐츠에 대한 접근 능력을 가지고 있으며, 네트워크 내에 다른 UPnP AV 디바이스에게 전송한다. 미디어 렌더러는 UPnP 미디어 서버를 통해 전송 받은 외부 콘텐츠를 로컬 하드웨어에서 재생하여 사용자에게 제공한다. 컨트롤 포인트는 사용자에게 콘텐츠 서비스를 제공하기 위하여 네트워크 내에 존재하는 미디어 서버와 미디어 렌더러를 검색하여 서버에 저장된 콘텐츠 목록을 사용자에게 제공하며, 미디어 렌더러의 콘텐츠 재생을 제어한다. 미디어 서버의 예로는 VCR, CD/DVD 플레이어, 카메라, 캠코더, PC, 셋탑 등이 있으며, 미디어 렌더러에는 TV, 스테레오, 스피커 등이 있다.

UPnP AV 프레임워크에서 컨트롤 포인트와 각각의 디바이스들은 서로 분리되어 상호 작용한다. 이들 디바이스 간의 직접적인 연결은 없고, 컨트롤 포인트를 통해 연결되고 제어를 받는다. 미디어 서버와 미디어 렌더러는 컨트롤 포인트의 요구에 따라 각 디바이스에 정의된 UPnP 액션을 실행함으로써 이들 두 디바이스 간에 직접 콘텐츠 스트림을 전송하고, 그 결과를 컨트롤 포인트에 알린다.

일반적인 UPnP에서는 컨트롤 포인트와 각 디바이스들은 분리되어 상호 작용을 하며 컨트롤 포인트와 UPnP 디바이스 간에 제어 명령과 통신만 존재한다. 그러나 이와 달리, UPnP AV에서는 A/V 디바이스 사이에 직접 데이터 교환이 이뤄진다. 즉, 사실상의 콘텐츠 전송은 컨트롤 포인트와 미디어 렌더러 사이에 있는 것이 아니라, 미디어 서버와 미디어 렌더러 사이에 있다. 그리고 컨트롤 포인트는 실제 전송은 담당

하지 않고 두 장치 사이의 동기화와 조정만을 담당하고, 실제 A/V 데이터는 HTTP 등의 전송 프로토콜을 이용하여 소스(source)와 싱크(sink) 장치 사이에서 직접 발생한다. 이때, 미디어 렌더러와 미디어 서버 간에 상호작용은 Non-UPnP 환경(Out-of-band)의 프로토콜로도 가능하다. 이러한 구조의 경우, 각각의 디바이스(미디어 렌더러)들을 네트워크 내에서는 하나의 디바이스(미디어 서버)로 인식시킬 수 있다.

UPnP AV 프레임워크에서는 세가지 핵심 모듈, 즉, 미디어 서버, 미디어 렌더러와 컨트롤 포인트가 정의되어 있지만, 실제 디바이스에 탑재할 때는 디바이스의 특성과 성능에 따라 관련 모듈을 조합하여 구현할 수 있다. 다시 말해, 세가지의 구성요소는 각각의 전용 디바이스에 탑재되어 구현될 수도 있지만, 디바이스의 특성에 따라 미디어 렌더러와 컨트롤 포인트가 하나의 디바이스에 탑재될 수도 있고, 미디어 서버와 컨트롤 포인트가 하나의 디바이스에 탑재될 수도 있다. 이는 디바이스의 성능과 사용의 편리성을 고려하여 설계 및 구현하게 된다. 만약, 미디어 렌더러와 컨트롤 포인트를 동일한 디바이스에 탑재한다면, 사용자가 동일한 디바이스에서 컨트롤을 렌더링하고, 동시에 제어할 수 있게 된다.

2.2 실시간 프로토콜

어플리케이션 층에서의 다양한 인터넷 프로토콜 중 AV와 같은 멀티미디어 데이터 전송을 위한 것으로는 Real-time Transport Protocol(RTP)와 Real-time Streaming Protocol(RTSP)이 있다. 이들은 데이터의 실시간 전송을 보장하지는 않지만, 가장 효율적으로 전송한다.

RTP는 인터넷 위에서 AV 데이터 전송을 위한 표준화된 패킷 규약으로 RFC 1889로 1996년에 규정되었다. TCP나 UDP에 종속적이지는 않지만, RTP Control Protocol(RTCP)와 함께 UDP를 이용하며, RTP를 이용하는 어플리케이션은 패킷 손실 보다는 실시간 전송을 더 중시 여기므로 TCP보다 UDP에 더 적합하다. RTP로 데이터를 지연 없이 이용할 수 있지만, 혼잡 및 흐름제어를 지원하지 않기에 QoS를 보장하지 못하는 단점이 있다. 이러한 RTP는 화상회의와 같은 비디오 데이터 전송-스트리밍 미디어 시스템-에 이용되며, VoIP의 기본이 되고 있다.

RTSP는 스트리밍 미디어 시스템을 위한 프로토콜로 RFC 2326으로 1998년에 규정되었다. 이는 클라이언트가 스트리밍 미디어 서버를 원격으로 컨트롤 하게 해주는 프로토콜이며, RTSP도 H.323과 마찬가지로 멀티미디어 콘텐츠 패킷 포맷을 지정하기 위해 RTP를 사용한다. H.323이 소규모 그룹간의 화상회의를 위해 설계된 데 반해, RTSP는 대규모 그룹들에게 AV 데이터를 효율적으로 브로드 캐스트하기 위한 목적으로 설계되었다. 기본적으로 RTSP는 AV와 같은 시간적으로 동기화된 스트림을 생성, 제어한다. 그러나 전형적으로 연속 매체 자체는 전송하지 않고 멀티미디어 서버를 위한 네트워크 제어 역할만 수행한다.

RTSP는 기본적으로 HTTP와 비슷한 syntax와 operation을 가지나, 서버와 클라이언트 모두가 요구(request)를 보내고 응답(response)를 받을 수 있으며, 초기 상태, 준비 상태, 재생 상태 등의 프로토콜 상태(state)를 가진다. 또한 HTTP는 메서드로 GET, POST만 있지만,

RTSP는 표 1과 같이 다양한 메서드를 가진다.

표 1. RTSP의 메서드

메서드	기능
OPTIONS	가능한 메서드의 종류를 반환한다.
SETUP	세션을 연다
DESCRIBE	미디어 객체의 description 을 얻는다.
ANNOUNCE	미디어 객체의 description 을 변경한다.
PLAY	재생을 시작한다.
RECORD	기록을 시작한다.
REDIRECT	새로운 서버로 재연결한다.
PAUSE	전송을 멈춘다.
TEARDOWN	세션을 닫는다.

3. 실시간 미디어 전송

기존의 UPnP AV 프레임워크는 미디어 콘텐츠를 스트리밍하는 방식으로 기존에 지원되는 HTTP-GET 방식을 사용하였다. 본 논문에서는 AV 콘텐츠의 실시간 스트리밍을 위하여 기존의 HTTP 외에 RTSP 스트리밍 방식을 지원하기 위한 미디어 재생 프레임워크를 UPnP AV 프레임워크와 연동하여 설계하고 구현하였다.

3.1 모듈별 상세 설계

디지털 엔터테인먼트 센터의 전체적인 구성에서 메시지가 전달되는 채널은 3가지가 존재한다. 컨트롤 포인트에서 미디어 서버와 미디어 렌더러에게 전달되는 채널, 그리고 콘텐츠 스트리밍을 위한 미디어 서버와 미디어 렌더러와의 메시지 전달 채널이다.

컨트롤 포인트는 미디어 서버와 미디어 렌더러에게 사용자의 직접적인 요청 메시지를 주로 전달한다. 그리고 컨트롤 포인트는 각 구성 요소에서 발생된 이벤트 메시지를 주로 받는다. 메시지를 전달할 때는 SOAP을 사용하며, 미디어 서버에게는 콘텐츠의 탐색, 검색, 스트리밍 요청 등의 메시지를 전달하고 미디어 렌더러에게는 콘텐츠 재생과 관련된 메시지(play, stop, pause, mute, volume up/down, position jump 등)와 사용자가 스트리밍 받기를 원하는 미디어의 위치 정보 등을 넘겨준다.

미디어 서버는 미디어 렌더러와 HTTP-GET 방식 또는 RTSP 방식으로 미디어를 스트리밍하며, 미디어 렌더러는 동영상 재생하는 플레이어와 이미지 뷰어로 구성되어 있고, 컨트롤 포인트에게서 전달 받은 콘텐츠의 위치에 따라 미디어 서버로 재생을 위한 스트리밍 요청 메시지를 전달한다. 그리고 미디어 서버와 렌더러는 각각의 구성 요소에서 발생

되는 이벤트 메시지를 컨트롤 포인트에게 전달하여 사용자가 현재 재생되는 콘텐츠의 부가적인 정보나 시스템 자체의 메시지를 쉽게 볼 수 있도록 한다. 그리고 셋탑에서는 미디어 렌더러와 컨트롤 포인트 두 모듈이 모두 필요하므로, 통합 모듈은 사용자가 쉽게 이용할 수 있는 인터페이스를 구성하였다.

다음 그림은 실시간 미디어 전송을 위해 구현된 디지털 엔터테인먼트 센터의 UPnP AV 프레임워크의 전체 구조이다.

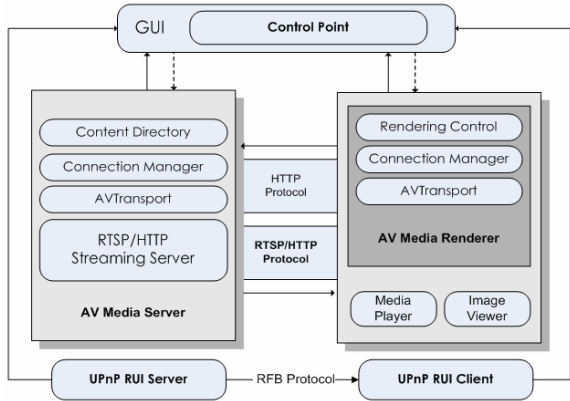


그림 2. RTSP를 이용한 UPnP AV 프레임워크

3.2 실시간 스트리밍 지원

미디어 렌더러의 플레이어인 MPlayer에서 RTSP 방식으로 콘텐츠를 스트리밍 하기 위해 Live555 Streaming Media의 Library를 이용한다[5]. 이는 Unix, Window, QNX 등 다양한 운영체제 상에서 MPEG, H.263, JPEG Video 등 다양한 코덱을 지원하는 RTP/RTCP, RTSP를 이용한 멀티미디어 스트리밍을 위한 라이브러리이다. 다음은 비디오 콘텐츠를 RTSP로 스트리밍 할 때의 메시지 화면이며 Live555 library를 사용하여 RTSP로 스트리밍하고 있음을 확인 할 수 있다.

```

root@localhost:~/work/setting/freevo-1.5.3
Playing rtsp://155.230.209.12:554/Qzpc3RyZWZtaW5nXFRFU1QxAA==/anyclubpart2.mp4
STREAM_RTSP, URL: rtsp://155.230.209.12:554/Qzpc3RyZWZtaW5nXFRFU1QxAA==/anyclubpart2.mp4
Resolving 155.230.209.12 for AF_INET6...
Couldn't resolve name for AF_INET6: 155.230.209.12
Connecting to server 155.230.209.12[155.230.209.12]: 554...
rtsp_session: Not a Real server. Server type is 'DSS/5.5.1 (Build/489.8; Platform/Win32; Release/Darwin; )'.
STREAM_LIVE555, URL: rtsp://155.230.209.12:554/Qzpc3RyZWZtaW5nXFRFU1QxAA==/anyclubpart2.mp4
Stream not seekable!
Initiated "video/MP4V-ES" RTP subsession on port 32988
Initiated "audio/MPA" RTP subsession on port 32990
Initiated "video/MP4V-ES" RTP subsession on port 32992
Initiated "audio/MPA" RTP subsession on port 32994
demux_rtp: Guesseed the video frame rate as 30 frames-per-second.
(If this is wrong, use the "-fps cframe-rate" option instead.)
WARNING: Video stream header 0 redefined.
demux_rtp: Guesseed the video frame rate as 30 frames-per-second.
(If this is wrong, use the "-fps cframe-rate" option instead.)
WARNING: Audio stream header 0 redefined.
VIDEO: [mp4v] 0x0 0bpp 30.000 fps 0.0 kbps ( 0.0 kbyte/s)
-----
Opening audio decoder: [mp3lib] MPEG layer-2, layer-3
AUDIO: 44100 Hz, 2 ch, s16le, 192.0 kbit/13.61% (ratio: 24000->176400)
Selected audio codec: [mp3] afm: mp3lib (mp3lib MPEG layer-2, layer-3)
-----
Opening video decoder: [ffmpeg] FFmpeg's libavcodec codec family
Selected video codec: [ffodivx] vfm: ffmpeg (FFmpeg MPEG-4)
-----

```

그림 1. RTSP 스트리밍

컨트롤 포인트는 AV 콘텐츠의 URI를 미디어 렌더러에게 넘겨주기 위해 미디어 서버에게 DIDL (Digital Item Declaration Language)로 요청한다. 미디어 렌더러는 RTSP 또는 HTTP를 이용하여 스트리밍 할 수 있는데, AV 콘텐츠

에 적합한 RTSP 스트리밍을 우선으로 하며, 컨트롤이 이를 지원하지 않을 시에는 HTTP를 이용한다. 다음은 콘텐츠 종류에 따라 지원하는 프로토콜이 서로 다를 수 있다.

```

root@localhost:~/work/setting/freevo-1.5.3
VIDEO ITEM 0 : 7F9293CE-40C6-48c2-B15E-AED2B5014DA0
VIDEO ITEM 1 : 3
VIDEO ITEM 2 : 31
VIDEO ITEM 3 : movie
VIDEO ITEM 4 : object.item.videoItem
VIDEO ITEM 5 : 4966217
VIDEO ITEM 6 : rtsp://155.230.209.12:554/Qzpc3RyZWZtaW5nXFRFU1QxAA==/movie.mp4
VIDEO ITEM 0 : 7F9293CE-40C6-48c2-B15E-AED2B5014DA0
VIDEO ITEM 1 : 3
VIDEO ITEM 2 : 34
VIDEO ITEM 3 : QAF
VIDEO ITEM 4 : object.item.videoItem
VIDEO ITEM 5 : 181524496
VIDEO ITEM 6 : http://155.230.209.12:8100/Qzpc3RyZWZtaW5nXFRFU1QxAA==/QAF.mpg

```

그림 3. 콘텐츠 URI 확인

4. 구현 및 실행

디지털 엔터테인먼트 센터에서 미디어 렌더러는 컨트롤 포인트를 통해 전달받은 AV 콘텐츠를 각각의 타입에 맞는 렌더러로 전달하고 이들의 속성을 제어한다. 미디어 렌더러는 다양한 데이터 포맷과 전송 프로토콜을 지원할 수 있어야 하므로 AV 콘텐츠는 MPlayer를 연동시키고, 이미지는 이미지 뷰어를 이용한다. 아래 표에서와 같이 구현된 디지털 엔터테인먼트 센터에서는 다양한 종류의 콘텐츠를 지원하고 있다.

표 2. 지원 가능한 콘텐츠 포맷

표 2. 지원 가능한 콘텐츠 포맷	
Audio	mp3, ogg, wav, m4a, wma, aac, ac3
Video	avi, mpg, mpeg, rm, ogm, vob, asf, mp4, viv, mov, ts
Image	jpg, gif, png, jpeg, bmp, tiff, psd

4.1 렌더링 모듈과의 연결

미디어 렌더러와 렌더링 모듈인 MPlayer, 이미지 뷰어와의 통신은 FIFO를 통해 이뤄진다. 컨트롤 포인트가 미디어 렌더러로 콘텐츠에 대한 URI를 전달하면 이는 FIFO에 저장되고, MPlayer나 이미지 뷰어로 전달된다. MPlayer와 이미지 뷰어는 URI를 이용하여 웹 서버에 접속하여, 미디어 렌더러로부터 받는 재생 제어 정보에 따라 콘텐츠를 스트리밍한다. 재생 제어 정보는 미디어 렌더러에서 렌더링 모듈로, 렌더링 모듈에서 미디어 렌더러로의 양방향 통신을 지원한다.

다음 그림은 미디어 렌더러와 렌더링 모듈인 MPlayer, 이미지 뷰어와 웹서버와의 관계를 나타낸다. 웹 서버는 콘텐츠의 특성에 맞게 RTSP나 HTTP를 통해 MPlayer나 이미지 뷰어로 스트리밍 한다.

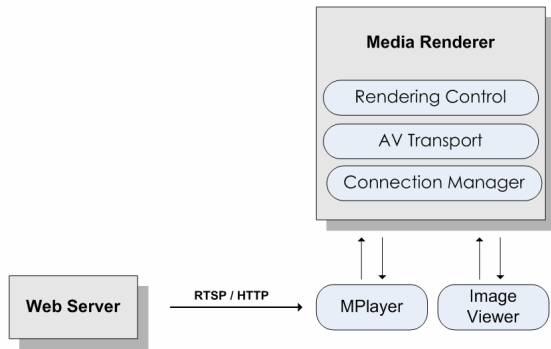


그림 4. 미디어 렌더러와 모듈간의 관계

비디오를 위한 렌더링 모듈인 MPlayer를 실행시키기 위해 컨트롤 포인트는 SetURI()에서 콘텐츠의 DeviceUDN과 CurrentURI를 미디어 렌더러로 넘겨준다. 미디어 렌더러는 UPnP AVTransport_SetAVTransportURI()에서 넘겨받은 URI를 MPlayer로 실행시킨다. 이때 MPlayer의 Full screen 또는 Screen size와 같은 속성을 설정하면서 .

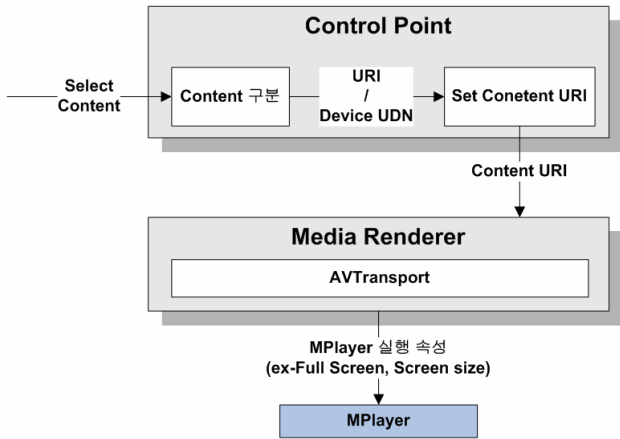


그림 5. MPlayer에서의 비디오 실행 다이어그램

4.2 실행

디지털 엔터테인먼트 센터의 전체 실행 구조는 다음과 같다. UPnP 디바이스 아키텍처 기반으로 컨트롤 포인트는 미디어 서버와 미디어 렌더러를 자동으로 발견하고 정의된 서비스를 호출하여 다양한 AV 디바이스를 제어할 수 있다. 사용자는 사용자 인터페이스 환경에서 원하는 서비스 메뉴를 선택하게 된다. 엔터테인먼트 시스템의 메인 메뉴 중 Share Remote AV 목록을 선택하게 되면 내부적으로 컨트롤 포인트는 현재 검색된 미디어 서버를 디렉토리 형태로 보여준다. 하나의 미디어 서버를 아이템으로 선택하고 컨트롤 포인트는 콘텐츠 디렉토리 서비스의 브라우즈 액션을 전송한다. 미디어 서버는 가장 최상위 디렉토리의 구조를 DIDL 문서 형태로 컨트롤 포인트에 리턴하게 되고 컨트롤 포인트는 필요한 정보를 파싱하여 플러그인 인터페이스로 넘겨준다. 즉 미디어 서버의 디렉토리 구조 정보를 컨트롤 포인트가 획득하여 가상의 파일 시스템으로 동일하게 생성한다. 생성된 디렉토리 and 콘텐츠는 메뉴 시스템의 아이템 모듈에서 관리한다. 컨

텐츠는 비디오, 오디오, 이미지로 분류되고 각 콘텐츠는 컨트롤 포인트를 이용하여 미디어 렌더러로 전달되어 재생된다.

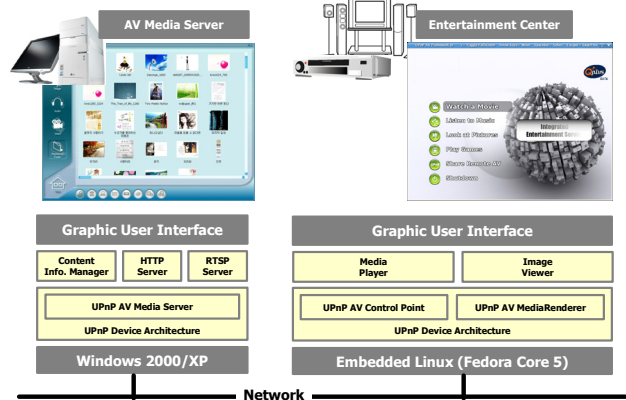


그림 5. 전체 실행 구조

다음 실행 화면은 미디어 서버내의 콘텐츠 중 .ts 파일을 재생한 것이다. 미디어 서버내의 검색된 콘텐츠 디렉토리는 폴더로 보여주며, 디렉토리 내의 콘텐츠들 중 원하는 것을 선택해 실행하면 된다.



그림 6. MPlayer에서의 비디오 실행 다이어그램

5. 결론

본 논문에서는 UPnP AV 프레임워크에서 기존의 HTTP-GET 스트리밍 방식외에 RTSP를 지원함으로써, 실시간 미디어 전송이 가능한 디지털 엔터테인먼트 센터를 설계하고 구현하였다. 임베디드 리눅스 환경에서의 디지털 엔터테인먼트 센터를 개발하기 위하여 Freevo를 기본으로 이용하였으

며, 다양한 AV 콘텐츠 재생을 위한 미디어 재생기는 MPlayer를 이용하였다. 이러한 디지털 엔터테인먼트 센터를 통해 사용자는 미디어 콘텐츠를 실시간으로 이용할 수 있게 되었다.

추가로 진행되고 있는 연구과제는 UPnP AV 프레임워크 내에서 다양한 미디어 렌더러를 제공하도록 한다. 현 프레임 워크에서는 미디어 렌더러가 PC 하나만 이용 할 수 있으나, 홈 내에는 PC, TV, PDA 등 다양한 미디어 렌더러가 존재하고 사용자는 선택한 콘텐츠를 원하는 렌더러에서 이용하기를 원할 것이다. 이를 위해 하나의 콘텐츠에 대해 멀티 렌더러 서비스를 제공함으로써 사용자 중심의 안정적인 디지털 엔터테인먼트 센터를 구현할 수 있을 것이다.

참고문헌

- [1] 이영진. “홈 네트워크 시장 활성화 저해 요인과 주요 갈등”. 정보통신정책. 제 18 권. 제 5 호. pp.19~31. March 2006.
- [2] UPnP Forum. <http://www.upnp.org>
- [3] Jiyun Park, Hyunju Lee, Sangok Kim, Sangwook Kim, “UPnP AV Framework for real-time AV streaming”, Proceedings of the 33th Korea Computer Congress 2006, Vol. 33, No. 2(D), pp.179-182, Oct 2006
- [4] RTSP. <http://www.rtsp.org>
- [5] live555 Library. <http://live555.com/mplayer>
- [6] Universal Plug and Play. <http://www.microsoft.com/whdc/device/netattach/upnp/default.msp>
- [7] 오재학, 차호정, 최영근, "스트리밍 미디어 캐쉬 서버를 위한 RTSP/RTP 스트림 제어 기법", 정보과학회논문지, Vol.9, No.3, pp. 254-255, June 2003
- [8] Schulzrinne. H., “Real Time Streaming Protocol”, RFC 2326, April 1998