

HTML5 기반 HTTP 스트리밍 환경에서의 서비스 이동성 연구

최헌회[†], 김근형^{**}

요 약

최근, W3C에서는 웹을 통해서 스트리밍 서비스뿐만 아니라 다양한 멀티미디어 서비스를 제공할 수 있는 기능을 포함한 HTML5에 대한 표준화가 추진 중이다. HTML5 이전의 웹 환경에서는 별도의 플러그인이 설치되어야 스트리밍 서비스를 이용할 수 있으나, HTML5 기반 웹 브라우저에서는 내장된 비디오 코덱을 이용하여 스트리밍 서비스를 직접 이용할 수 있다. 또한, 인터넷 방화벽에 의해 서비스되지 않는 RTP/UDP를 이용한 스트리밍 대신 HTTP/TCP를 이용하여 적응적으로 스트리밍하는 기술이 표준화 중이다. 본 논문에서는 현재 표준이 이루어지고 있는 HTML5 기반 적응적 HTTP 스트리밍 환경에서 스트리밍 서비스 세션 이동성을 보장할 수 있는 방안을 제시한다. 제안된 알고리즘과 구조를 이용하여 다양한 단말들 사이에 미디어 흐름, 미디어 흐름 제어, 세션 제어를 전달할 수 있도록 자바스크립트 언어로 모델링 및 구현하였으며 다양한 실험을 통하여 다양한 단말 간 HTTP 스트리밍 서비스를 효과적으로 전달할 수 있음을 보였다.

A Study on Service Migration in HTML5 based HTTP Streaming Environments

Hun-Hoi Choi[†], Geun-Hyung Kim^{**}

ABSTRACT

Recently, the W3C has developed the HTML 5 standard to provide various multimedia services as well as a web based streaming. HTML5 based web browsers with the built-in video codecs can be used directly for streaming services, though previous web browsers should contain playback plug-in for streaming services. In addition, an adaptive streaming technology using HTTP/TCP has been standardized, since an Internet firewall does not support RTP/UDP streaming protocol but HTTP/TCP streaming protocol. In this paper, we propose the service migration mechanism among various devices in the HTML5-based adaptive HTTP streaming environments. The proposed algorithm and architecture were modeled for migrating media flows, media flow control, and session control and implemented using JavaScript. Finally, we showed that the proposed mechanism was effective to migrate HTTP streaming service among various devices by performing several experiments.

Key words: HTTP Streaming(HTTP 스트리밍), Service Migration(서비스 이동), HTML5 Video (HTML 비디오)

※ 교신저자(Corresponding Author): 김 근형, 주소: 부산광역시 부산진구 가야3동 동의대학교 산학협력관 605호 (614-714), 전화: 051)890-2271, FAX: 051)890-2265, E-mail: geunkim@deu.ac.kr

접수일: 2010년 12월 31일, 수정일: 2011년 5월 25일

완료일: 2011년 6월 13일

[†] 정회원, 동의대학교 디지털미디어공학과

(E-mail: hunhoi@deu.ac.kr)

^{**} 정회원, 동의대학교 영상정보공학과

※ 본 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(No.2010-0025069) 및 한국 전자통신연구원의 지식경제 기술혁신사업(2009-S-026-01, 방송통신융합 다자간 서비스 및 연속성 제어 기술 개발)의 지원으로 수행된 것임.

1. 서 론

압축 기술의 발전으로 디지털 방송 서비스가 보편화되고 네트워크 기술의 발전으로 인터넷을 통한 디지털 방송 콘텐츠의 전달 기술 개발이 꾸준히 진행되고 있다. 또한, 스마트 폰의 보급에 따라 사용자가 언제 어디서나 비디오 콘텐츠를 이용할 수 있게 되면서 비디오 스트리밍 기술 개발이 요구된다. 과거에는 비디오 콘텐츠의 스트리밍을 위해 RTP(Real time Transport Protocol)/RTSP(Real time Streaming Protocol) 등의 다양한 프로토콜이 사용되었다. 그러나 RTP/RTSP 기반 스트리밍 기법은 정의되지 않은 UDP(User Datagram Protocol) 포트 번호를 사용하기 때문에 방화벽을 통과하지 못한다. 이러한 문제점 때문에 HTTP(Hyper Text Transfer Protocol)를 이용한 비디오 콘텐츠의 전달에 대한 관심이 높아졌다 [1]. 또한, RTP/RTSP를 사용하는 전통적인 스트리밍 기술은 고가의 전용 스트리밍 서버와 클라이언트 소프트웨어를 필요로 하지만, HTTP를 이용한 비디오 콘텐츠의 전달 기술은 표준 웹 서버와 웹 캐쉬(cache)를 이용할 수 있다는 장점을 가지고 있다.

W3C(World Wide Web Consortium)에서 표준화 중인 HTML(Hyper Text Markup Language) 5에서는 웹 브라우저에 별도의 플러그인(plug-in)없이 동영상 재생할 수 있도록 비디오 태그와 오디오 태그를 정의한다. 이 때문에 별도의 플러그인을 설치하는 것이 어려운 모바일 단말에서 웹 브라우저를 통해 쉽게 동영상 재생이 가능하게 되었다. 현재, 대부분의 웹 브라우저는 HTML5 기반의 비디오 태그 및 프로그레시브 다운로드(progressive download)를 지원한다. 프로그레시브 다운로드란 웹 브라우저에서 비디오 파일 전체를 다운로드하지 않고도 재생을 시작하는 기술이다. 그러나 이 방식은 파일을 다운로드하면서 재생하기 때문에 하나의 압축률로 압축된 동영상 파일을 다운로드하던 중에 가용 대역폭이 줄어들면 재생이 멈추게 된다.

이러한 문제를 해결하기 위해서 네트워크의 대역폭 변화에 적응적으로 대응하기 위해 같은 비디오 콘텐츠를 여러 압축률로 압축한 파일을 웹 서버에 저장해 두고, 클라이언트에서 네트워크 상태 변화에 따라 전송을 최적화하려는 HTTP 기반 스트리밍 기술이 논의되고 있다[2-5]. 또한, 여러 압축률로 압축

된 파일을 서버에 저장하면서 네트워크의 상태 변화에 신속하게 대응하기 위해서 비디오 콘텐츠를 일정 시간 간격의 작은 파일로 나누어 저장한다. 다양한 유무선 액세스 망을 통한 HTTP 기반 적응형 비디오 스트리밍 기술에 대한 연구와 표준화가 3GPP(3rd Generation Partnership Project) [2], OIPF(Open IPTV Forum)[3], IETF(Internet Engineering Task Force)[4], MPEG(Moving Picture Expert Group)[5]에서 이루어지고 있다. 또한 애플[1], 마이크로소프트[6], 어도비[7]에서 HTTP 기반 적응형 스트리밍 시스템을 개발하여 상용 제품으로 출시하거나 무료 소프트웨어로 제공하고 있다.

3GPP에서는 기존 RTP/RTSP를 이용한 전통적인 미디어 전송 방법의 대안으로 적응적인 HTTP 스트리밍 방법을 고려하고 있으며, 이를 위해 미디어 표현 기술(Media Presentation Description, MPD), 미디어 분할 형식(Media Segmentation Format), HTTP 서버, 고객의 행위(client behavior) 등에 대한 규격을 정의하고 있다. OIPF는 3GPP의 HTTP 스트리밍 규격을 바탕으로 OIPF 서비스와 장치에 응용할 수 있도록 HTTP 적응적 스트리밍에 대한 기술 규격을 제정하고 있다. MPEG에서도 DASH(Dynamic Adaptive Streaming over HTTP)라는 표준 제정 작업이 진행되고 있다[8-11]. IETF에서는 어떠한 조각 파일을 어떻게 순차적으로 가져와서 재생하는가를 결정하는데 필요한 정보를 표준화 한다.

차세대 광대역 유무선 네트워크 환경에서 이동하면서 비디오 스트리밍 서비스를 이용하기 위해서는 이동성 관리가 매우 중요하다. 이동성은 크게 단말 이동성, 사용자 이동성, 서비스 이동성으로 분류한다 [12]. 단말 이동성은 단말이 유무선 접속 망에 관계없이 네트워크 접속점을 바꾸어 자유롭게 이동할 수 있도록 하는 것으로 단말이 이동 중에도 서비스를 끊김 없이 사용하도록 보장한다. 사용자 이동성은 사용자가 이동하면서 사용 단말과 접속 네트워크에 관계없이 서비스를 이용할 수 있도록 하는 것이다. 서비스 이동성은 서비스를 이용 중에 사용 단말이 변경되어도 끊김 없이 서비스를 이용할 수 있도록 보장하는 것으로 집 밖에서 모바일 단말로 서비스를 이용하다가 집에 도착하여 TV와 같은 다른 단말로 서비스를 이동시켜 이용하는 것이 한 예이다. 서비스 이동성은 서비스 이용에 가장 적합한 단말로 서비스를

이동시켜서 향상된 서비스 이용 경험을 얻을 수 있고, 더 저렴한 액세스 네트워크에 접속된 단말로 서비스를 이동시켜 통신비용을 절감할 수 있다는 장점을 가진다.

서비스 이동 관련 연구는 VoIP(Voice over Internet Protocol)와 같은 멀티미디어 통신 서비스와 웹 기반 서비스의 이동은 많이 이루어졌으나 스트리밍 서비스에 대한 연구는 많이 이루어지지 않았다. 멀티미디어 통신 서비스는 SIP(Session Initiation Protocol)을 이용하여 서비스 세션을 설정하기 때문에 서비스 이동을 위해 RFC(Request for Comments) 3515에서 규정하는 SIP REFER 요청을 사용하는 방식[13] 및 RFC 3725에서 정의한 3PCC(3rd Party Call Control) 방식[14]과 이를 응용한 여러 방법이 제안되었다. SSIIP[15] 방법은 SIP 프로토콜의 INVITE와 REFER 요청에 Mobility 확장 헤더를 추가하여 여러 단말로 서비스의 분리 및 분리된 서비스를 다시 모으는 방법을 제안하였다. Shacham[16]은 REFER 요청과 3PCC 제어 메커니즘을 통합하여 여러 단말로 서비스를 이동하는 방법과 여러 단말로 분리된 서비스를 통합적으로 관리하기 위해 “가상 단말” 개념을 제안하였다.

웹 기반 서비스 이동을 위해 Song[17]은 웹 브라우저 세션 저장 및 이동 메커니즘을 제안하였다. 이는 한 브라우저에서 다른 브라우저로 서비스 이동을 위해 브라우저 상태 저장 서비스를 정의하여 단말을 변경할 때 먼저 브라우저의 상태 정보를 네트워크의 저장장소에 저장하고 다른 단말의 브라우저에서 저장된 상태 정보를 다운로드하여 이전 서비스 이용 상태를 유지한다. Song의 방법과 달리 Tsai[18]는 네트워크 서버 없이 클라이언트에 부가적인 컴포넌트를 추가하여 서비스를 이동할 단말에 직접 관련 정보를 전달하는 메커니즘을 제안하였다. Schulzrinne[19]는 SIP 프로토콜을 이용하여 웹 브라우저의 화면을 공유하는 방법을 제안하였다. 웹 브라우저의 화면을 다른 사람과 공유하기 위한 방법으로 웹 브라우저의 URL 정보를 공유하고자 하는 단말에게 SIP의 MESSAGE 또는 NOTIFY 요청을 이용하여 전달하는 것을 제시하였다. Munkongpitakkun[20]도 SIP 프로토콜을 이용하여 클라이언트 기반 웹 세션 이동 방안을 제안하였다. 제안한 구조는 SIP 서버, 이동 서비스(migration service), 클라이언트 에이전트로

구성되었다. 클라이언트 에이전트는 SIP 메커니즘을 이용하여 웹 세션 정보를 저장 및 복원하고 다른 클라이언트 에이전트와 교환한다. 이동 서비스는 저장 서비스, 북마크 서비스, 세션 이동 대상 리스트 서비스와 같이 세션 이동과 관련한 서비스를 제공한다. Adeyeye[21]는 [20]과 같이 SIP MESSAGE 요청을 이용하여 웹 세션 정보를 다른 웹 브라우저로 전달하여 웹 서비스 세션을 이동하는 방안을 제안하고 모질라 파이어폭스에서 플러그인 형태로 개발하였다.

스트리밍 서비스의 경우 Más[22]가 SIP 기반 IPTV 시스템에서 IMS(IP Multimedia Subsystem) IPTV 세션을 STB(Set-Top Box)와 PC간에 이동하는 절차를 제안하고 IMS 기반 IPTV 환경에서 구현하였다. RTSP 프로토콜을 사용하는 모바일 IPTV 서비스 제공환경에서 RTSP 프로토콜에 기반한 사용자 이동성 및 서비스 세션 이동성을 보장하는 기술이 개발되었다[23,24]. 그러나 아직까지 HTTP 스트리밍 서비스의 서비스 이동성과 미디어 흐름 제어의 이동성을 보장하는 기술 개발은 이루어지지 않았다.

별도의 플러그인 없이 비디오 콘텐츠의 재생이 가능한 HTML5 표준을 지원하는 웹브라우저가 PC를 비롯하여 TV와 스마트폰에서 제공되고 있어, 이들 단말간 비디오 스트리밍 서비스의 이동을 위한 기술을 HTML5 기반으로 한번 개발하면 여러 단말에 모두 적용할 수 있다는 장점을 가진다. 기존에 개발한 응용 프로그램 기반 스트리밍 서비스 이동 기술 [23,24]은 각 단말 별로 클라이언트를 개발하여야 하기 때문에 개발의 오버헤드가 증가하였다. 또한, 기존 스트리밍 서비스의 이동성 보장 기술은 SIP 프로토콜과 RTSP 프로토콜 기반으로 제안하고 있으며 단지 미디어 플로우의 이동에 대해서만 고려하고 있다. 본 논문은 최근 관심이 집중되고 있는 HTTP 스트리밍 서비스 제공 환경에서 스트리밍 서비스를 다양한 단말에서 끊김 없이 이용할 수 있도록 하는 이동성 보장 방안과 한 단말(예: TV)에서는 비디오 콘텐츠를 재생하고, 다른 단말(예: 스마트폰)에서 비디오 미디어 흐름을 제어할 수 있도록 제어권의 이동 방안을 연구하였다. 본 논문에서는 HTML5에서 정의한 동영상 관련 API를 이용하여 HTTP 스트리밍 환경에서 단말 간 효율적인 서비스 이동을 할 수 있는 알고리즘을 제안하고 이의 설계 및 구현기술을 설명한다. 제안된 알고리즘과 구조를 이용하여 동종

단말 간 서비스 이동 및 이종 단말 간 서비스 이동 및 미디어 흐름 제어를 이동하는 것을 보였다. 또한 HTTP 프로그레시브 다운로드와 HTTP 스트리밍 환경에서 단말 간 서비스 이동을 하였을 때 지연시간을 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 웹 기반 동영상 스트리밍 기술에 대해서 살펴 본 후, 3장에서는 현재 표준화 단체에서 연구 중인 HTTP 스트리밍 서비스의 세션 이동 기술과 제안한 동종 단말 간 서비스 이동 메커니즘과 이종 단말 간 서비스 이동 메커니즘을 설명하고 실제 구현하여 실험한 결과를 보인다. 마지막으로 4장에서 결론을 기술하였다.

2. 웹 기반 동영상 스트리밍 기술

2.1 HTML5 기반 동영상 스트리밍

기존 웹 브라우저에서 미디어 플레이어를 이용하여 동영상을 재생하려면 외부 플러그인이 필요하다. 그러나 스마트폰과 같은 모바일 단말에서 동영상을 재생하기 위해 플러그인을 설치하는 것은 비효율적이라 웹에서 이미지를 보듯이 멀티미디어 콘텐츠를 재생할 수 있도록 HTML5에서 비디오와 오디오 태그를 정의하였다. HTML5 이전에는 웹페이지에 동영상을 삽입하기 위해서 외부 객체를 넣을 수 있는 오브젝트 태그와 많은 파라미터를 중복 지정했다. 특히, 별도의 응용 프로그램에 미디어 데이터를 넘겨 처리할 경우 예측할 수 없는 상황이 많이 발생한다. 웹 페이지에 플러그인을 삽입하면 웹 브라우저가 플러그인이 차지하는 부분은 블랙박스 처리하기 때문에 플러그인 내부에서 발생하는 이벤트에 대해서는 처리하거나 알리고 하지 않는다. 이러한 특징은

페이지의 레이아웃이 동적으로 바뀌는 구조와 플러그인에 할당된 영역과 페이지 나머지 요소가 겹치는 구조에서는 문제가 발생할 수 있다. HTML5에서는 플러그인을 더 이상 사용하지 않고 비디오와 오디오를 웹의 한 요소로 포함시키는 표준화된 방법을 정의하여 웹 브라우저 자체에서 동영상을 직접 재생하고 보여줄 수 있는 방법을 제공한다. 현재 대부분의 웹 브라우저가 비디오 태그를 지원하고 있다[25].

모든 동영상은 비디오와 오디오로 구성되며 비디오와 오디오의 요소를 하나로 묶어놓은 것을 비디오 컨테이너 파일이라 한다. 비디오 컨테이너 파일은 비디오, 오디오, 데이터 등의 트랙으로 구성되며 일반적으로 상호 연관되어 있다. 또한 각 트랙은 비디오의 화면비율 또는 오디오의 언어정보와 같은 메타데이터를 저장하며, 컨테이너 역시 동영상의 제목과 표지 디자인 등의 메타데이터를 포함한다. 대표적인 비디오 컨테이너 포맷의 종류는 다음 표 1과 같다.

HTML5 표준은 웹 페이지에 동영상 삽입을 위해 비디오 태그를 사용할 뿐 코덱과 컨테이너 포맷을 정의하지 않았다. 대표적으로 고려되는 비디오 코덱은 H.264와 테오라 (Theora), VP8이고 오디오 코덱은 MP3, AAC(Advanced Audio Encoding), 보비스 (Vorbis) 코덱이다. 현재 웹 브라우저와 모바일 기기에서 지원하는 코덱 및 비디오 컨테이너의 조합은 다음 표 2와 같다. 표 2에서 볼 수 있듯이 모든 HTML5 지원 웹 브라우저에서 모두 지원되는 코덱과 컨테이너 조합은 없다.

2.2 HTTP 기반 적응적 스트리밍

지금까지 인터넷의 동영상 스트리밍은 전용 스트리밍 서버와 스트리밍 전송 프로토콜인 RTP/RTCP/

표 1. 대표적인 비디오 컨테이너 파일 포맷.

포맷	설명
MPEG-4 (mp4, m4v)	애플의 쿼타임 컨테이너(.mov)를 기반으로 하며 MPEG-4 part14로 규정된 파일 포맷
Flash video (*.flv)	어도비사의 플래시 플레이어에서만 재생할 수 있는 파일 포맷
Ogg (*.ogg)	특허권에 제약을 받지 않는 공개 소스 파일 포맷으로 보비스나 플렉과 같은 오디오 코덱과 비디오 코덱으로 압축된 스트림을 저장하는 파일 포맷
WebM (*.webm)	2010년 구글 I/O 개발자 컨퍼런스에서 발표된 공개 소스 방식의 파일 포맷
AVI (*.avi)	윈도우 운영체제에서 오디오와 비디오를 합쳐 하나의 동영상으로 지원하는 동영상 파일 포맷으로 파일의 용량이 크기 때문에 실시간 영상에는 적합하지 않음

표 2. 웹 브라우저 별 지원 코덱 및 컨테이너 조합.

코덱/컨테이너	IE	파이어폭스	사파리	크롬	오페라	아이폰	안드로이드
테오라+보비스+Ogg		3.5+		5.0+	10+		
H.264+ AAC+MP4	9+		3.0+	5.0+		3.0+	2.0+
WebM		4.0+		8.0+	10.6+		

RTSP 등을 사용하는 전통적인 스트리밍 방법과 동영상 파일의 일부를 클라이언트에 다운로드하며 재생을 하는 HTTP 프로그레시브 다운로드 방법이 사용되었다. HTTP 프로그레시브 다운로드 방법은 전용의 스트리밍 서버를 사용하지 않고 웹 서버만으로 서비스가 가능하다는 장점을 가지나, 스트리밍 파일을 요청하면 파일 모두를 다운로드하기 때문에 다운로드 중 재생을 중지하는 경우 대역폭 낭비가 발생할 수 있으며, 네트워크 상태 변화에 적응적으로 동작하지 못한다. HTTP 기반 적응적 스트리밍 기술은 네트워크의 상태에 따라 적당한 압축률로 압축된 스트림 데이터로 스위칭이 가능하도록 하기 위해서 하나의 콘텐츠를 짧은 시간 간격으로 나누어 다양한 압축률로 압축한 파일들을 웹 서버에 저장한다. 클라이언트는 네트워크 상태를 파악하여 여러 파일 중 적합한 파일의 전송을 요청한다.

HTTP 기반 스트리밍 구조의 구성 요소는 다음 그림 1과 같이 스트리밍 콘텐츠를 생성하는 인코더와 인코딩된 스트리밍 데이터를 일정한 시간 간격의 작은 조각(chunk)으로 나누는 세그멘터(segmenter), 스트리밍 미디어를 재생하는 클라이언트(웹 브라우저 또는 응용 프로그램), 콘텐츠의 분배를 담당하는 콘텐츠 분배기로 구성된다.

HTTP 기반 적응적 스트리밍 기술은 HTTP 프로그레시브 다운로드 기술에 기반하고 있으며 재생을 위한 스트리밍 파일을 연속적인 작은 조각으로 다운로드하여 적응적인 실시간 스트리밍과 적은 비용으로 콘텐츠 캐시의 구현을 가능하게 한다. 또한, 한

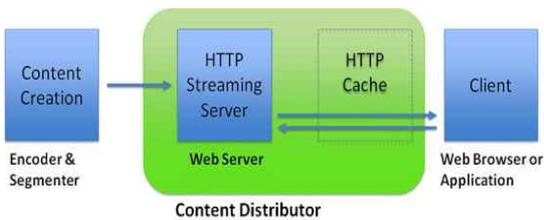


그림 1. HTTP 기반 스트리밍 구성요소

콘텐츠에 대해 그림 2와 같이 여러 압축률로 인코딩된 스트림을 제공하여 네트워크 상태와 단말의 능력에 적합한 스트리밍이 가능하며 네트워크 상태와 단말의 프로세스 부하의 변경에 따라 단말이 다양한 압축률의 스트리밍 데이터간 스위칭이 가능하다.

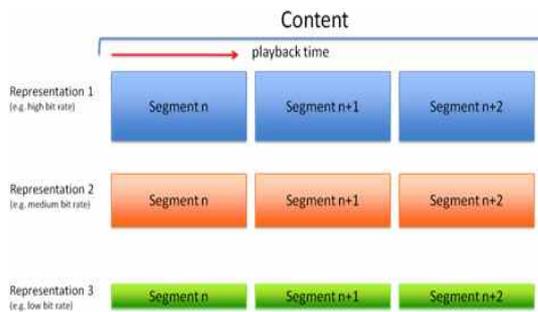


그림 2. HTTP 스트리밍을 위한 세그먼트 및 다중 압축률 제공 구조

3. HTTP 기반 스트리밍 서비스의 단말 간 이동 기술

본 장에서는 OIPF의 서비스 이동 기술과 단말 간 서비스 이동성을 제공할 때 고려해야 할 사항을 살펴보고 구현한 HTTP 스트리밍 서비스 세션 이동 메커니즘을 설명한다.

3.1 OIPF의 서비스 이동 기술

OIPF 표준 단체는 단말 간 서비스 이동과 관련하여 서 릴리즈 2의 서비스 및 플랫폼 요구사항을 2008년 12월에 발표한 이후 2010년 9월에 관련 표준을 발표하였다. OIPF는 단말 간 서비스 이동을 세션 연속성(session continuity)과 세션 이동성(session portability)으로 정의한다. 그중 세션 연속성을 보장하기 위해서 세션 이동(session transfer)과 세션 복제(session replication) 기능을 정의한다. 세션 이동 기능은 사용자가 현재 스트리밍 중인 세션을 다른 단말

로 이동하는 것으로 세션 이동을 통해 사용자는 목표 단말에서 같은 콘텐츠의 재생을 연속적으로 시작할 수 있도록 한다. 세션 복제 기능은 사용자가 다른 단말에서 동일한 콘텐츠를 같은 시점부터 재생할 수 있도록 하는 것으로 세션이 이동된 후에도 이전 단말에서 세션을 유지한다. 세션 복제가 이루어진 후, 이전 단말과 목표 단말은 독립적인 세션을 가지게 된다.

세션 이동 및 복제 기능은 사용자가 어떠한 단말에서 세션 이동 및 복제를 요청하는가에 따라서 푸시 모드와 풀 모드로 분류한다. 푸시 모드는 사용자가 현재 사용하고 있는 단말에서 목표 단말로 세션을 이동을 요청하는 것이며, 풀 모드는 사용자가 목표 단말에서 세션 이동 및 복제를 할 세션을 선택하여 가져오거나 복제를 요청하는 것에 해당된다.

3.2 단말 간 서비스 연속성 보장을 위한 고려 사항

서비스 연속성을 보장하기 위해서는 보안적인 세션 이동 메커니즘, 서비스 이동에 필요한 정보의 정의 및 관련 정보의 추출 방법, 단말의 디스플레이 속성, 접속 네트워크의 대역폭, 서비스를 이동시킬 목표 단말의 탐색 방법들이 고려되어야 한다.

단말 간 세션을 주고받을 때 문제점은 개인 정보 유출과 권한이 없는 사용자가 콘텐츠를 재생하는 것이다. 보안이 요구되는 정보는 아이디/비밀번호와 같은 개인정보 뿐만 아니라 서비스 이동과 관련한 사생활 관련 정보(예, 재생중인 콘텐츠 정보, 이동 단말 정보, 생활 패턴 등)이며, 서비스 세션 이동 시 세션 이동의 범위, 이동에 대한 권한 등을 설정할 수 있어야 한다.

콘텐츠를 연속적으로 재생할 수 있도록 서비스를 이동하기 위해 필수적으로 전달되어야 하는 정보는 콘텐츠에 대한 주소와 단말이 스위칭 될 때의 콘텐츠의 재생 시점이다. 이 외에 부가적인 정보로 콘텐츠의 타이틀, 설명, 세션을 이동시킨 사용자, 사용자가 전달하려는 메시지 등을 고려할 수 있다. 사용자 단말에서 세션 전달에 필요한 정보의 추출 기능이 구현되어야 한다.

서비스를 주고받는 단말의 디스플레이 크기와 접속 가능한 네트워크가 다르기 때문에 서비스를 이동할 때 단말의 디스플레이 크기와 네트워크 대역폭에 따른 적합한 동영상을 전달받아 사용자의 서비스 경험을 극대화하는 것도 필요하다. 현재까지의 서비스

이동은 오디오와 비디오의 화면을 옮기는 것을 목표로 하지만 앞으로의 웹 기반 비디오 서비스는 비디오 콘텐츠뿐만 아니라 다양한 부가적인 데이터 정보가 함께 서비스될 것이다. 이러한 부가적인 데이터를 TV 화면에 같이 보여주면 동영상도 못보고 데이터 정보도 읽기 어려운 문제점이 생긴다. 사용자의 서비스 경험을 극대화하기 위해서는 데이터 정보는 적절하게 서비스할 수 있는 단말로 동영상은 대형 TV로 오디오는 맥내 오디오 시스템으로 보내는 서비스 이동을 보장할 수 있는 기술이 필요하다. 그리고 콘텐츠의 이동뿐만 아니라 스마트폰을 이용하여 콘텐츠의 재생을 제어할 수 있는 기능도 필요하다. 단말의 제어 기능은 재생/정지 기능 이외에 볼륨 조절, 음 소거, 일정 시간 앞으로 뒤로, 디스플레이 화면의 확대/축소, 전체 화면 기능 등 많은 제어 기능을 제공할 수 있어야 한다.

HTML5 기반 비디오 스트리밍 서비스는 웹 브라우저 기반의 동영상을 재생할 수 있어야 한다. 그러나 현재까지 HTML5 기반 웹 브라우저에서 지원하는 비디오/오디오 코덱 정보가 모두 다르기 때문에 코덱 표준이 정해지기 전 까지 각 웹 브라우저에서 지원하는 콘텐츠로의 변환이 필요하다. 즉, 비디오 콘텐츠 간 트랜스코딩을 해주는 기능이 필요하다.

본 논문에서는 같은 시스템 특성을 갖는 동종 단말 간과 시스템 특성이 다른 이종 단말 간 동영상 서비스를 이동시키기 위한 방안을 설계하고 맥북 간, iPhone 간, 맥북과 iPhone 간 서비스 이동을 구현하였으며, 맥북에서 재생되는 것을 iPhone 단말을 사용하여 제어하는 기술도 개발내용을 설명한다.

3.3 단말 간 HTTP 스트리밍 서비스 세션 이동 메커니즘

3.3.1 동종 단말 간 HTTP 스트리밍 서비스 세션 이동 메커니즘

동종 단말 간 서비스 이동은 특정 단말에서 이용 중인 스트리밍 서비스를 동종의 다른 단말을 이용하여 같은 서비스를 연속적으로 이용하게 하는 것을 뜻한다. 단말 간 서비스 이동 시 전달해야 할 정보와 세션 제어 서버에서 사용자의 편의를 위해 관리해야 할 정보가 존재한다. 그림 3은 세션 이동 시 필수적인 정보를 보여준다. 중앙의 로그인 정보는 ID와 비밀번호 등 개인 정보를 저장하며, 이는 세션 정보를 주고받을 때 사용자 식별과 개인 세션정보의 보호를 위해

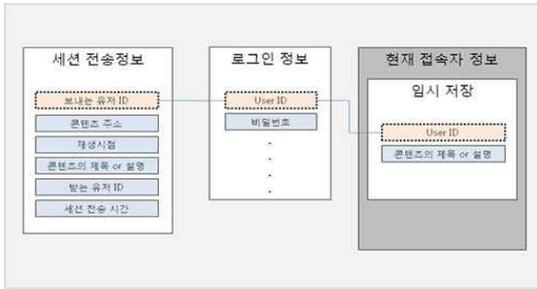


그림 3. 세션 이동에 사용되는 정보

필요하다. 또한 사용자가 오프라인인 경우 자신에게 전달된 세션 정보를 관리하기 위해 로그인 정보가 필요하다. 왼쪽의 정보는 실제 세션 전달에 관련된 정보로 사용자가 서비스 세션 이동을 할 때 보내는 사용자 ID, 콘텐츠의 주소, 현재 재생 시점, 콘텐츠 이름, 세션을 받는 사용자 ID, 전달 시간 등이 저장된다. 오른쪽의 현재 접속자 정보는 현재 로그인된 사용자 ID와 서비스 중인 콘텐츠를 저장한다. 현재 사용자가 다른 사용자의 재생 중인 콘텐츠를 확인하여 전달을 요청하는 풀(pull) 모드로 구현하기 위해 사용된다.

예로, 사용자A가 사용자B로 서비스 이동이 된다고 할 때, 사용자A가 현재 재생하고 있던 콘텐츠의 주소와 현재 재생하고 있는 시점을 사용자B에게 전달해야 한다. 서비스 세션 이동을 구현한 시스템에서 세션 전달 시 세션 정보의 저장은 네트워크 중앙의 서버에 저장하여 단말 간 서비스 이동 시 참조할 수 있도록 한다. 사용자A가 사용자B로 서비스 이동하기 위해서는 사용자A는 현재 세션 정보를 세션관리 서버에 저장한다. 네트워크 서버는 사용자 A가 보낸 세션 이동 정보를 기반으로 사용자B를 찾는다. 사용자B가 로그인 되어 있으면 사용자A의 세션 이동 요청을 사용자B에게 전달한다. 세션관리 서버는 사용자B가 로그인되지 않았다면 이동 정보를 저장해 두었다가 사용자B가 로그인하였을 때 알려준다. 사용자B에게 세션 이동 요청이 있어도 사용자B의



그림 4. 서비스세션 전달 절차

승낙이 없으면 세션 이동이 되지 않도록 하여 악의적인 세션 이동 요청을 필터링 한다. 다음과 같은 절차에 의해 서비스의 이동이 끊임 없이 이루어진다. 그림 4는 세션 이동 절차를 나타낸다.

서비스 이동 방법에는 사용자A가 요청하여 사용자A가 이용하던 서비스를 사용자B에게 이동하는 방법과 사용자B의 요청으로 사용자A가 이용하던 서비스를 이동 요청하는 방법으로 구별된다. 첫 번째 방법은 서비스 이동의 푸시 모드라 하고 두 번째 것은 서비스 이동의 풀 모드라고 한다. 그림 5는 푸시 모드의 서비스 이동 방법의 실제 구현 방법을 나타낸다. 푸시 모드의 서비스 이동 절차는 사용자A가 서비스를 이동하고자 하는 대상으로 사용자B를 선택하는 것으로 부터 시작된다. 사용자A는 현재 자신의 단말에서 재생 중인 콘텐츠의 주소와 현재 재생 시간을 추출한 후 세션 전달을 위해 네트워크에 존재하는 데이터베이스에 저장한다. 웹 브라우저와 웹 서버간의 통신은 HTML5의 웹 소켓을 사용하지 않으면, 웹 서버에서 웹 브라우저로 서비스 이동 요청을 알려주기(notify) 위해서 AJAX(Asynchronous JavaScript and XML)와 같은 부가적인 기술이 필요하다. 그래서 본 논문의 구현에서는 서비스 이동의 가능성을 보기 위해서 웹 소켓을 사용하는 대신 각 단말에서 데이터베이스의 정보를 폴링(polling)하도록 설계하였다. 즉, 사용자B의 단말에서 데이터베이스의 정보를 폴링하여 자신에게 전달된 서비스가 있는지 확인한다. 그래서 자신에게 전달된 서비스 이동 요청이 있으면 세션 정보를 이용하여 콘텐츠를 가져와 재생하는 구조로 되어 있다.

그림 6은 풀 모드의 세션 이동 구조로 사용자B가 서비스 이동 요청을 서비스를 이용하고 있는 사용자

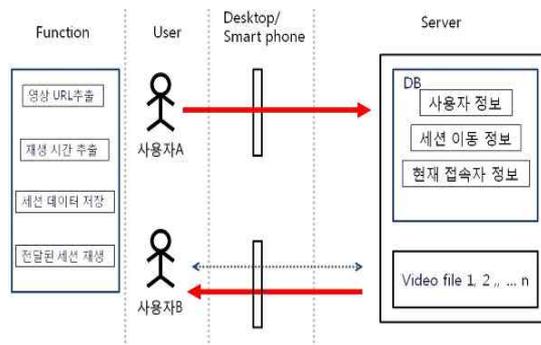


그림 5. 푸시 모드 세션 이동 메커니즘

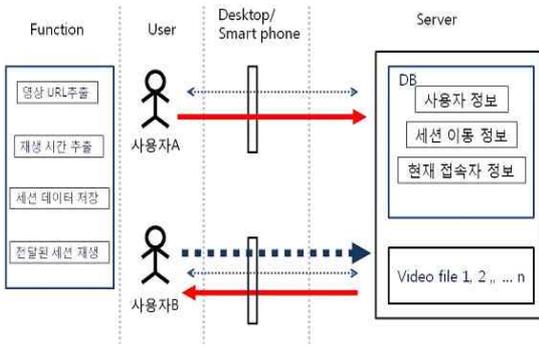


그림 6. 풀 모드 세션 이동 메커니즘

A에게 전달한다. 서비스 이동 요청에 대해 사용자A가 허락하면 푸시 모드와 같이 사용자A로부터 시작되는 세션 이동 절차가 시작된다.

그림 7은 스마트폰 간 서비스 세션 이동 모습이다. HTTP 스트리밍방식 파일을 재생하였으며 재생되는 비디오의 콘텐츠의 압축률을 표시하기 위해 미리 동영상에 압축률을 표시해 두어 비디오가 대역폭에 따라 변경됨을 확인하였다. 그림 7의 (a)는 사용자A가 임의의 콘텐츠를 재생하고 있는 모습이고, 그림 7의 (b)는 세션 전달받을 사용자를 선택 후 서비스



그림 7. 스마트폰 간 서비스 세션 이동 모습

이동 요청 메시지의 전송이 완료된 모습이다. 그림 7의 (c)는 사용자B가 임의의 콘텐츠를 재생하고 있던 중 사용자A로부터의 서비스 세션 이동 요청을 보이는 화면이다. 이 때 네트워크 서버의 IP 주소가 http://203.241.205.126이라 알림 창에 그 주소 값이 표시되었다. 그림 7의 (c)에서 사용자B가 승낙을 하면 그림 7의 (d)와 같이 사용자A가 전달한 세션 정보로부터 콘텐츠 재생을 시작하는 화면이 나타난다.

그림 8은 PC 간의 사파리 웹 브라우저를 사용하여 세션 이동하는 모습이다. 그림 7의 스마트폰 간의 세션 이동 순서와 제공되는 기능은 같다. 그림 8의 (a)는 사용자A가 임의의 콘텐츠를 재생하고 있는 모습이고, 그림 8의 (b)는 세션 전달받을 사용자를 선택 후 서비스 이동 요청 메시지의 전송이 완료된 모습이다. 그림 8의 (c)는 사용자B가 임의의 콘텐츠를 재생하고 있던 중 사용자A로부터의 서비스 세션 이동 요청을 보이는 화면이다. 그림 8의 (c)에서 사용자B가 승낙을 하면 그림 (d)와 같이 사용자A가 전달한 세션 정보로부터 콘텐츠 재생을 시작하는 화면이 나타난다. PC는 스마트폰에 비해 프로세싱 성능이 우수하고 유선 네트워크를 사용하기 때문에 스마트폰 간의 세션 이동보다 세션 정보 전달 시간이나 전달 받은 콘텐츠를 재생하기까지 로딩시간이 적게 걸렸다. 세션이 전달되는 시점의 화면과 세션을 전달받아 재생이 이루어지는 시점의 화면이 되는데, 이는 HTTP 스트리밍을 위해 10초 단위의 작은 단위로 파일을 저장하고 있기 때문에 스위칭이 10초 단위로 이루어지기 때문이다.

본 논문에서 서비스 이동을 위해서 사용한 언어는

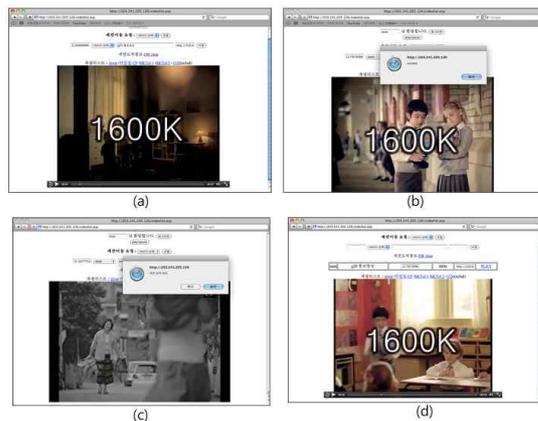


그림 8. PC 간 서비스 세션 이동 모습

표 3. 비디오 객체 선언 코드.

```
<video controls="controls" width="720"
height="480" />
```

자바스크립트(JavaScript)이며 서비스 세션의 세션 정보와 제어를 위해서는 HTML5의 비디오 태그를 이용한다. W3C에서 표준화하고 있는 HTML5의 비디오 태그는 대부분의 웹 브라우저에서 지원하고 있으며, 웹 스크립트 언어로 모바일 단말의 웹 브라우저에서도 지원되고 있기 때문에 적응적 HTTP 기반 스트리밍 서비스의 서비스 이동에 적합한 기술이라 판단된다. 실제 구현한 코드 중 서비스 이동을 위해 필요한 비디오 객체는 표 3과 같이 정의한다.

비디오 재생을 위해 표 3에서 만든 비디오 객체를 사용하며, 서비스 이동 된 콘텐츠를 재생을 위해 별도의 콘텐츠 입력함수를 표 4와 같이 만들어야 한다. 표 4의 myVideo 변수의 src 값에 재생할 콘텐츠 주소를 저장하고 콘텐츠에 대한 설명과 콘텐츠 주소를 텍스트 박스에 출력하고 재생 중 현재 재생 시간을 표시하기 위해서 cur_time() 함수를 주기적으로 호출한다. 현재 재생 중인 비디오에 대한 정보를 저장하기 위해 window.open() 함수를 사용한다.

콘텐츠의 재생이 시작되면 향후 사용자가 서비스 이동에 사용할 재생 중인 콘텐츠의 현재 재생 시간을 표시하며, 표 5는 현재 재생 시점을 추출하기 위한

함수이다. 구현한 시스템에서 다음의 함수를 이용하여 텍스트 상자에 현재 재생 시점을 출력한다.

표 6은 그림 7에서 실험한 결과에서 mp4와 m3u8 파일을 사용하였으며 세션 이동 후 재생시작 시간과 시점 옮기는 시간에 있어 두 파일이 상대적인 결과를 나타냈다. HTTP 스트리밍 방식은 브라우저가 m3u8을 분석하여 대역폭과 네트워크 상태에 맞는 미디어 파일을 요청하기 때문에 mp4에 비해 버퍼링 시간이 오래 걸린다. 이동 시점의 정확도는 HTTP 스트리밍의 경우 파일의 조각의 단위인 10초 단위로 시점이 이동 하였으며, HTTP 프로그레시브 다운로드의 경우 전달자가 보낸 정보를 토대로 정확한 이동 시점을 나타냈다.

3.3.2 이종 단말 간 HTTP 스트리밍 서비스 세션 이동 메커니즘

이종 단말 간 서비스 세션 이동은 맥내에 존재할 수 있는 다양한 종류의 단말을 대상으로 한다. 이동되어야 할 정보나 구조는 동종 단말 간 세션 이동과 같지만 이종 단말의 경우 전달하는 단말과 전달받는 각 단말의 디스플레이의 크기와 속성, 유선이나 무선과 같은 네트워크 접속 방법을 고려해야 한다.

단말의 디스플레이가 다른 경우 단말의 속성을 고려하기 위해서 여러 단말을 위해 압축한 콘텐츠 중 서비스를 이동할 단말에 적합한 것을 전달하여 재생

표 4. 콘텐츠 입력 함수

```
function my_univ() {
var myVideo = document.getElementsByTagName('video')[0]; // 비디오 객체 추출
myVideo.src = 'HTTP://220.68.5.118/univ.m3u8'; // 재생할 비디오 콘텐츠 정보 지정
document.getElementById('txt_name').value = '대학 홍보영상'; // 화면에 설명 출력
document.getElementById('txt_src').value = myVideo.src; // 화면에 url 출력
setTimeout('cur_time()', 1000); // 현재 재생 시점 출력 함수 호출
win = window.open('session_temp7.asp','AddEmployee','30','50') // 재생 윈도우 생성
}
```

표 5. 현재 재생 시점 추출 함수

```
function cur_time() {
var myVideo = document.getElementsByTagName('video')[0]; // 비디오 객체 추출
document.getElementById('txt_time').value = myVideo.currentTime; //비디오 객체의 현재 재생 시점 값을 화면에 출력
setTimeout('cur_time()',10); // 주기적으로 cur_time() 함수 호출
}
```

표 6. 스마트폰에서 mp4와 m3u8의 세션 이동 비교.

	재생 요청 후 재생까지 걸리는 시간	시간 옮긴 후 재생까지 걸리는 시간	이동 시점 정확도
HTTP 스트리밍 (m3u8)	6-8초	2-3초	10초 단위
HTTP 프로그레시브 다운로드(mp4)	2-4초	1-2초	정확

되도록 한다. 예로, 스마트폰용 콘텐츠와 TV용 콘텐츠를 각각 만들어 각 단말에 맞는 콘텐츠를 전송하여 해결한다. 그림 9는 스마트폰에서 PC로 세션을 이동하는 모습이다. 스마트폰에서는 저해상도 화질인 512k용 콘텐츠가 재생되었고, 세션 이동 후 PC에서는 고해상도인 1600k용 콘텐츠가 재생되었다. 이처럼 세션 이동함에 있어 효율성을 위해 각각의 단말에 맞는 콘텐츠의 전달이 필요하다.

또한 이종 단말 간 서비스를 이용할 경우 다양한 접속 네트워크와 단말의 성능을 고려하여야 한다. 즉, 접속 네트워크의 대역폭과 단말 성능에 적합한 콘텐츠를 재생하여 사용자가 끊김 없이 콘텐츠를 감상할 수 있어야 한다. HTTP 스트리밍 구조에서는 클라이언트에서 네트워크 상태 변화에 적합한 콘텐츠를 요청하게 된다.



그림 9. 스마트폰에서 PC로 서비스 세션 이동 되어 콘텐츠가 변경된 모습

3.3.3 제어세션 이동 메커니즘

이종 단말 간 서비스 세션 이동 시 서비스 제어권 이동에 대해서 고려해야 한다. 서비스 제어권을 전달 받은 단말에서 비디오와 오디오 서비스의 재생을 제어하는 것이다. 그림 10은 스마트폰을 이용하여 PC에서 재생 중인 콘텐츠를 제어하는 모습이다. 스마트폰에서 콘텐츠를 재생하고 있다가 비디오와 오디오 서비스만 PC로 전달하고 서비스의 재생 또는 정지



그림 10. 스마트폰을 이용해 PC에서 재생 중인 콘텐츠를 제어하는 모습

기능은 스마트폰에서 제어한다. 본 연구에서는 제어 기능을 가지는 단말에서 콘텐츠에서 재생되는 단말로 네트워크의 서버를 통해서 제어 메시지를 전달하는 구조로 구현되었다.

단말 간 서비스 이동 후 사용할 수 있는 제어기능은 표 7과 같이 구현하였다. 세션 이동 페이지에 미리 재생/정지 기능을 입력해놓고 제어권을 가진 단말기에서 메시지 "pp"를 전달한다. refresh()함수를 일정 시간 간격으로 호출하여 제어메시지를 받는 단말기는 PlayPause()함수를 실행해 콘텐츠의 재생 또는 정지 기능을 실행한다.

4. 결 론

본 논문에서는 HTTP 스트리밍 환경에서 효율적으로 서비스를 이동을 할 수 있는 알고리즘을 제안하고, 이를 HTML5와 을 이용하여 설계, 구현하였다. 제안된 알고리즘과 구조를 이용하여 웹 브라우저가 탑재된 다양한 단말들 사이에 미디어 흐름, 미디어 흐름 제어, 세션 제어를 전달할 수 있으며, HTTP를 이용하여 HTTP 스트리밍 세션을 HTTP 프로토콜을 이용하여 전달할 수 있다. 제안된 미디어 스트리밍 서비스 전달 시스템은 HTML5에서 정의한 동영상 관련 API를 이용하여 모델링 및 구현이 되었다.

표 7. 단말 간 콘텐츠 제어 기능 함수

<pre>function refresh() { if("pp" == "<%=rs3("name")%>"){ PlayPause() // PlayPause 함수 호출 } }</pre>
<pre>function PlayPause() { var myVideo = window.parent.document.getElementsByTagName('video')[0]; if (myVideo.paused) myVideo.play(); // 일시정지 상태에서 "pp" 메시지를 전달 받은 경우 재생 호출 else myVideo.pause(); // 재생 상태에서 "pp" 메시지를 전달 받은 경우 일시 정지 함수 호출 }</pre>

다양한 경우에 대한 실험을 통하여 제안된 알고리즘과 구조가 다양한 장치들 사이에 HTTP 스트리밍 서비스 세션을 효과적으로 전달할 수 있음을 보였다. 동일한 종류의 단말 장치들 사이의 HTTP 스트리밍 서비스 세션 이동에 대한 실험과 이종 단말 간 HTTP 스트리밍 서비스 세션 이동에 대한 실험을 수행하고 그 결과를 실험 결과에서 보였다. 현재 미디어 스트리밍 기술을 개발한 회사들은 미디어 스트리밍의 서비스 이동을 아직 지원하지 않으며, 현재 3GPP와 OIPF에서 활발히 표준화 작업이 진행 중이다. 제안된 알고리즘과 구조는 사용자가 편리하게 미디어 스트리밍 서비스를 다양한 단말 장치들 간에 전달할 수 있어 추후 사용될 수 있을 것으로 기대된다.

앞으로 HTTP 기반의 서비스의 이동을 통합적으로 관리할 수 있는 프레임워크에 대한 연구와 스트리밍의 초기 로딩 시간 및 원하는 시점에서의 서비스 스위칭이 일어날 수 있도록 파일 포맷에 대한 연구를 진행할 계획이다.

참 고 문 헌

- [1] Apple, "HTTP Live Streaming Overview," 2010.
- [2] 3GPP, TS 26.234 V9.4.0, "Transparent end-to-end Packet-Switched Streaming Service (PSS); Protocols and codecs(Release 9)," 2010.
- [3] OIPF, "HTTP Adaptive Streaming (v. 2)," 2010.
- [4] <http://www.ietf.org>
- [5] <http://mpeg.chiariglione.org>
- [6] Microsoft Corporation, "IIS Smooth Streaming Technical Overview," 2009.
- [7] Adobe, "Using ADOBE HTTP DYNAMIC STREAMING," 2010.
- [8] ISO/IEC JTC1/SC29/WG11, "HTTP Streaming of MPEG Media Context and Objectives (N11337)," 2010.
- [9] ISO/IEC JTC1/SC29/WG11, "Call for Proposals on HTTP Streaming of MPEG Media (N11338)," 2010.
- [10] ISO/IEC JTC1/SC29/WG11, "Use Cases for HTTP Streaming of MPEG Media (N11339)," 2010.
- [11] ISO/IEC JTC1/SC29/WG11, "Requirements on HTTP Streaming of MPEG Media (N11340)," 2010.
- [12] ITU-T Rec. Q.1706/Y.2801, "Mobility Management Requirements for NGN," 2006.
- [13] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method," IETF RFC 3515, 2003.
- [14] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, "Best Current Practices for Third Party Call Control(3pcc) in the Session Initiation Protocol(SIP)," IETF RFC 3725, 2004.
- [15] M.-X. Chen, C.-J. Peng, and R.-H. Hwang, "SSIP: Split a SIP Session Over Multiple Devices," *Journal of Computer Standards and Interfaces*, Vol.29, No.5, pp. 531-545, 2007.

[16] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer, "Session Initiation Protocol (SIP) Session Mobility," IETF RFC 5631, 2009.

[17] H. Song, H.-H. Chu, and S. Kurakake, "Browser Session Preservation and Migration," in Proc. WWW 2002, pp. 7-11, 2002.

[18] Y.-H. Tsail, J.-J. Chen, T.-W. Kuo, and C.-S. Shih, "Client and Server Mobility for Web Applications," IIWAS 2004, 2004.

[19] H. Schulzrinne, "Use SIP MESSAGE Method for Shared web Browsing," draft-wu-sipping-webshare-00.txt, 2001.

[20] W. Munkongpitakkun, S. Kamphiwong, and S. Sae-Wong, "Enhanced Web Session Mobility Based on SIP," in Proc. of the 4th Intl. Conf. on Mobile Technology, Applications and Systems, 2007.

[21] Michael O. Adeyeye, N. Ventura, and D. Humphrey, "Mapping Third Party Call Control and Session Handoff in SIP Mobility to Content Sharing and Session Handoff in the Web Browsing Context," WCNC 2009, pp. 1-6, 2009.

[22] I. Más, V. Berggren, R. Jana, J. Murray, and C. W. Rice, "IPTV Session Mobility," in Proc CHINACOM2008, pp. 903-909, 2008.

[23] 장지원, 김근형, "사용자와 서비스 세션 이동성 보장을 위한 모바일 IPTV 서비스 플랫폼 개발," 한국디지털콘텐츠학회 논문지, Vol.10, No.1, pp. 87-96, 2009.

[24] G.-H. Kim and B.-H. Lee, "Seamless Streaming Service Session Migration Support Architecture for Heterogeneous Devices," LNCS 6294, pp. 473-484, 2010.

[25] 현동석, 강유훈 역, 구글 개발자가 들려주는 HTML5 활용, pp. 129-173, 2010.

[26] 박명우, 김연석, 이경호, "모바일 환경에서 웹 서비스 이동," 한국멀티미디어학회 논문지, Vol. 11, No.8, pp. 1179-1193, 2008.



최 현 회

2004년 3월~2011년 2월 동의대학교 영상정보공학과 학사
 2011년 3월~현재 동의대학교 디지털미디어공학과 석사과정

관심분야: 멀티미디어 스트리밍, 웹 기반 서비스 이동



김 근 형

1982년 3월~1986년 2월 서강대학교 전자공학과 학사
 1986년 3월~1988년 2월 서강대학교 전자공학과 석사
 2000년 3월~2005년 2월 포항공과대학교 컴퓨터공학과 박사

1988년~1990년 LS 산전 연구소 연구원
 1990년~1993년 삼성종합기술원 선임연구원
 1996년~1997년 NIST Guest Researcher
 1993년~2007년 8월 (주) KT BcN 본부 수석연구원
 2007년 9월~현재 동의대학교 영상정보공학과 조교수
 관심분야: Mobile IPTV, 웹 기반 서비스 이동, P4P 스트리밍, Content Centric Network