논문 17-42-03-10 The Journal of Korean Institute of Communications and Information Sciences '17-03 Vol.42 No.03
https://doi.org/10.7840/kics.2017.42.3.623

클라우드 방송 서비스 플랫폼

김 홍 익◆°, 이 동 익*, 이 종 한*

Cloud Broadcasting Service Platform

Hong-Ik Kim◆°, Dong-Ik Lee*, Jong-Han Lee*

요 약

클라우드(Cloud) 기술이 적용되는 영역은 기술의 발전과 서비스의 다양화로 점차 확대되고 있다. 방송 플랫폼에도 투자 효율화, 운영 효율화, 서비스 경쟁력 확보 등의 측면에서 클라우드 기술이 도입되고 있다. 최근 국내에서는 클라우드 방송 플랫폼이 UI와 서비스에 도입되면서 서비스 경쟁이 가속화 되고 있다. 클라우드 방송 기술은 단말기의 자원과 소프트웨어 구조에 종속성 없이 서비스를 제공하고, 단말기 별로 분리되어 관리되고 있던 UI와 서비스를 서버기반으로 통합하여 운영될 수 있도록 한다. 논문에서는 클라우드 방송 플랫폼에 대해서 설명하고 적용 효과에 대해서 설명한다.

Key Words : Cloud Broadcasting Platform, Eliminating remove Service Dependency, Unified Operation

ABSTRACT

Application fields of cloud technologies have been gradually expanded with development of technology development and diversification of services. Cloud technology is used for investment efficiency, operation efficiency and service competitive advantage in digital broadcasting platform. Recently, Cloud broadcasting platform commercialized for UI(User Interface) and data broadcasting in Korea, and broadcasting service competition becomes fierce. Cloud technology of broadcasting provides remove a service dependency hardware resource and software architecture on STB device, and unified operation of user interface and service using cloud server without legacy separating management of STB types. In this paper, we explain application effects in image based cloud broadcasting service platform.

Ⅰ. 서 론

클라우드(Cloud) 관련 기술이 빠르게 발전하면서 적용되는 분야가 점차 확산되고 있으며 방송 기술과 서비스에도 클라우드 가상화 기술, 범용 서버 기술, 소프트웨어기술 등의 도입이 가속화 되고 있다. 클라우드 기술은 비교적 폐쇄적이었던 방송과 IT분야의 기술경계를 허물고 있으며 상호간의 통합화를 가속화 시키는 기술적 기반이 될 수 있어 많은 개발이 진행되고 있다.[1-4]

방송분야에서 클라우드 기술이 활발하게 개발되어 적용되고 있는 분야는 가상화(Virtualization) 헤드엔드(Head-end), 클라우드 서비스 플랫폼, 백오피스(Back-Office) 크게 3가지 영역으로 구분할 수 있다. 이중에서 클라우드 서비스 플랫폼은 투자와 개발 주기가 상대적으로 짧고 사용자의 직접적인 반응이 조기에 나타나는 UI(User Interface)와 양방향 서비스에 적용하여 기술개발의 속도가 빠르게 진행되고 있다. 클라우드 서비스 플랫폼의 주요 활용분야는 UI/UX, 콘텐츠 저장/관리 서비스, 고사양 게임, 스마트 서비

◆° First and Corresponding Author : CJ HelloVision, Smart Engineering Team, hongik.kim@cj.net, 정회원
* CJ HelloVision, Technical Division {dongik.lee, john.lee1}@cj.net
논문번호 : KICS2016-10-310, Received October 14, 2016; Revised January 2, 2017; Accepted February 8, 2017

스, 멀티 디코딩 서비스 등이 있고 그 적용범위는 점차 증가하고 있다.[5-8]

클라우드 서비스 플랫폼이 적용되면 STB(Set-top box)의 하드웨어 성능에 따라 발생하는 서비스 격차 및 STB의 교체주기와 서비스 주기가 상이하여 발생하는 신규서비스의 확대의 어려움과 신규 고사양 STB에 하향 평준화 된 서비스를 제공하는 문제를 해결할 수 있다. 그리고, STB의 모델별 개발 및 관리 되는 기존 미들웨어(Middleware) 방식의 문제를 해결하여 클라우드 서버를 통해 모든 단말기를 동시에 운영할 수 있어 시장변화에 빠르게 대응할 수 있고 비용절감을 할 수 있는 구조를 가질 수 있다.[9-11]

그러나 지금까지 개발되어왔던 클라우드 서비스 플랫폼 기술들은 대부분 많은 인프라 자원을 사용하는 비디오(Video) 스트리밍 중심으로 발전해 왔기 때문에 클라우드 기술로 인한 장점보다 높은 투자비 장벽 문제로 인해 적용과 서비스 확산에 한계가 있었다.

이러한 문제를 해결하기 위해 기존 미들웨어 중심의 방송서비스 플랫폼보다 인프라 자원을 작게 사용하면서 저사양 STB에서도 안정적으로 동작할 수 있는 이미지(Image) 스트리밍 중심의 클라우드 서비스 플랫폼이 상용화되었다. 클라우드 서비스 플랫폼 기술을 통해 클라우드 기술의 장점을 유지하면서 기존 높은 투자비 장벽을 해결할 수 있게 되어 최근에는 국내 대부분의 방송사업자들이 이미지 방식 혹은 유사한 방식의 클라우드 서비스 플랫폼을 도입했거나 도입할 예정이다.

클라우드 서비스 플랫폼이 도입되면 방송의 UI (User Interface)와 서비스에 많이 적용되고 있으며 그 영역은 확대되고 있다. UI는 화면 전환시간 단축, 효과 다양화, 화질개선, 개발기간 단축, 운영비용 절감 등의 효과가 있었고 사용자의 이용형태 분석도 가능하게 되었다. 서비스는 하드웨어 성능 이슈와 소프트웨어의 제약 때문에 제공하지 못하던 서비스를 빠르고 쉽게 제공할 수 있게 되었다.

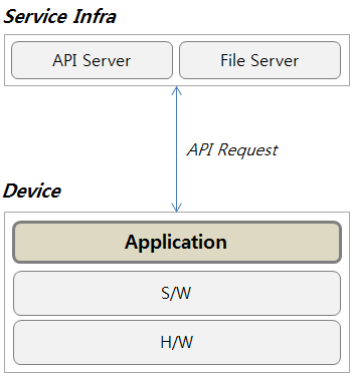
논문에서는 클라우드 서비스 플랫폼에 대한 설명과 플랫폼 적용 후 서비스 개선효과에 대해서 알아본다. 본 논문은 다음과 같은 구성을 가진다. Ⅱ장은 클라우드 방송 서비스 플랫폼에 대한 내용과 이를 기반으로 하는 서비스를 설명한다. Ⅲ장은 클라우드 방송 플랫폼 구성과 적용된 기술 내용에 대해서 설명한다. Ⅳ장은 클라우드 방송 플랫폼 서비스에 대한 내용을 설명하고, Ⅴ장은 클라우드 방송 플랫폼을 적용한 후 성능 평가에 대한 내용을 분석하며, Ⅵ장은 분석 결과를 제시한다.

Ⅱ. 클라우드 방송

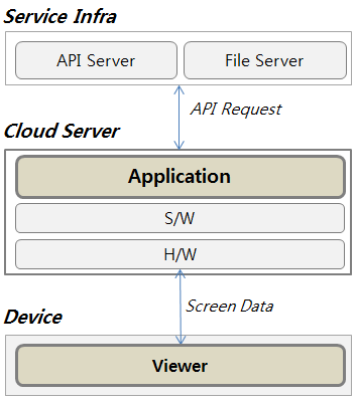
2.1 클라우드 방송 서비스 플랫폼

일반적으로 어플리케이션은 하드웨어 및 소프트웨어 환경에 종속적이다. 어플리케이션이 실행되는 단말기의 CPU, 메모리, 그래픽카드 등 여러 리소스가 활용되므로 하드웨어 리소스의 한계에 따라 여러 제약이 존재한다. 또한, 어플리케이션이 플랫폼의 기본 기능을 사용하기 위해 필요한 '서비스 플랫폼(M/W, Web, JVM, Framework, Android)환경'에서 실행되므로 소프트웨어에서 제공하는 기능에 따라 제약이 존재한다.

클라우드 서비스 방식은 상대적으로 풍부한 하드웨어 및 소프트웨어 리소스를 가진 서버환경을 구축하고 서버에서 어플리케이션을 실행시킨다. 서버에서 어플리케이션 실행 결과를 단말로 전송하면, 단말은 실



(a) Legacy application architecture



(b) Cloud application architecture

그림 1. 클라우드 어플리케이션의 구조 (a) Legacy 어플리케이션 구조 (b) 클라우드 어플리케이션 구조
Fig. 1. Application architecture in cloud broadcasting platform (a) Legacy application architecture (b) Cloud application architecture

행된 결과를 디스플레이 하는 역할만 담당하여 서비스를 제공하는 방식이다.

클라우드 방송 서비스 플랫폼은 이러한 클라우드 방식을 이용하여 방송 서비스를 제공한다.

기존 방송 서비스 플랫폼에서는 STB 등 단말기에 있는 미들웨어가 어플리케이션을 실행하고 동작되던 방식이라 단말기의 하드웨어(Hardware)와 소프트웨어(Software)에 종속적으로 서비스가 제공되는 구조이다. 하지만 클라우드 방송 서비스 플랫폼은 방송 단말기에 아무런 종속성 없이 다양한 방송 서비스를 제공할 수 있도록 클라우드 서버가 서비스를 실행한 후 단말기로 전송하고, 단말기는 단순 화면표시를 하는 서비스 플랫폼 구조를 가지고 있다.

기존 방송 서비스 플랫폼은 단말기에서 모든 서비스가 실행되기 때문에 단말기의 성능에 따라 사용자에게 제공할 수 있는 서비스에 제약이 있으며, 단말기별로 개별 개발, 관리, 운영을 해야 하므로 비용과 시간이 많이 필요한 구조이다. 하지만 클라우드 방송 서비스 플랫폼은 서버가 서비스를 실행하여 단말기로 단순 화면을 전송하는 구조이기 때문에 단말기 성능에 따른 서비스 제한이 없으며 단말기의 종류가 많더라도 서버 작업만으로 동시에 일괄적으로 통합 개발, 관리, 운영이 가능해서 비용과 시간이 많이 절감되는 특징을 가지고 있다.

2.2 클라우드 방송 서비스

클라우드 방송 서비스 플랫폼을 활용할 수 있는 분야로는 UI, High-end 게임(콘솔게임 등), 웹 콘텐츠 서비스, 네트워크 저장장치 기반 서비스(NPVR 등), 인터넷 동영상 서비스, 개방형 앱스토어, 멀티뷰, 멀티 스크린 등이 있고 기술의 발달로 점차 방송에서 제공하기 불가능했던 서비스까지 활용범위가 확대되고 있다.

클라우드 기술이 방송 서비스에 적용되면서 방송 플랫폼과 서비스에 클라우드라는 범용 IT 기술이 적용 가능하게 되어 실질적인 의미에서 방송통신 융합을 할 수 있게 된다. 이는 방송기술의 IT화를 가속화하게 될 것으로 예상된다..

Ⅲ. 클라우드 방송 플랫폼 구성 및 적용

3.1 클라우드 방송 플랫폼 구성

3.1.1 플랫폼 구성

그림 2는 클라우드 방송 플랫폼의 구조를 나타낸다. 클라우드 방송 서비스 플랫폼은 크게 클라우드 서비스 시스템, 클라우드 관리 시스템, 클라우드 모니터링 시스템, 클라우드 클라이언트 4부분으로 구분된다.

클라우드 서비스 시스템은 클라우드 클라이언트로부터 전달되는 사용자의 요청을 클라우드 브라우저(Browser)에 전달하고 클라우드 어플리케이션(Application)의 처리된 결과를 클라우드 엔진(Engine)에서 이미지, 비디오, 웹, 에니메이션 등의 화면 데이터를 생성하여 가입자 단말에 서비스와 UI를 전송하는 역할을 한다. 클라우드 엔진에서 제공하는 클라우드 방식에 따라 이미지(Image) 클라우드, 비디오(Video) 클라우드, 메타데이터(Meta Data) 클라우

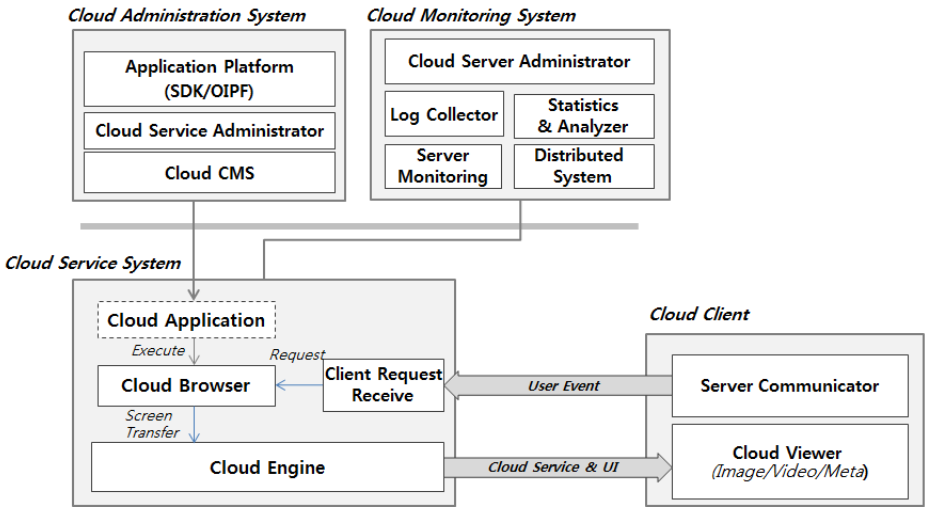


그림 2. 클라우드 방송 플랫폼 구조
Fig. 2. Cloud broadcasting platform architecture

드로 구분된다.

클라우드 관리 시스템은 개발환경 부분과 운영부분으로 구분된다. 개발환경 부분은 개방형 플랫폼에서 개발환경 표준화를 할 수 있는 어플리케이션 플랫폼의 기능을 한다. 다수의 단말 및 서버 인프라의 환경에 따라 다양한 개발이 필요한 기존 환경과 달리 클라우드 환경에서는 표준 어플리케이션 플랫폼 환경인 프레임워크(Framework)를 포함한 SDK(Software Development Kit)와 규격화된 API(Application Program Interface)를 제공함으로써 개방형 클라우드 환경을 지원한다. SDK에 대한 범위는 방송 서버 인프라뿐만 아니라, 클라우드 환경제어, 단말제어까지 포함한 서비스 개발에 필요한 전체 기능을 포함한다.

운영부분은 클라우드 서비스 플랫폼을 통해서 제공되는 콘텐츠와 서비스를 관리와 운영을 한다. 방송 플랫폼 사업자가 구축한 다수의 클라우드 서버 환경에서 서비스 공급자(Service Provider)의 어플리케이션과 콘텐츠의 등록 및 배포기능을 제공한다. 서비스 공급자가 클라우드 서비스 관리 시스템을 통해 어플리케이션과 콘텐츠를 등록하면, 클라우드 CMS(Content Management System)는 서비스의 특성에 따라 클라우드 서비스 시스템에 배포된다.

클라우드 모니터링 시스템은 클라우드 시스템과 서비스를 모니터링 하고 원격 운영을 할 수 있도록 한다. 물리적인 클라우드 서버의 정보가 모니터링 시스템에 등록되면 사용자의 접속 이력 및 서비스 오류 및 장애를 감시한다. 사용자 접속 요청 시 지역별, 서비스별, 단말별로 구축되어 있는 다수의 클라우드 서버에 대한 모니터링 정보를 이용하여 서버 할당이 진행되고 안정적인 시스템이 운영될 수 있도록 접속을 분산한다. 접속로그 및 통계를 이용하여 클라우드 방송 서비스와 UI의 이용 현황을 수집, 통계, 분석 할 수 있다.

클라우드 클라이언트는 사용자 단말에 설치되어 클라우드 서비스 시스템과 연동하여 서비스를 제공 한다. 클라우드 클라이언트는 서비스를 실행하거나 랜더링(Rendering)하는 역할을 하지 않고 단순히 화면에 표시하는 에이전트(Agent) 역할을 수행한다. 기존 클라이언트 어플리케이션과 달리 서버 통신 및 클라우드 뷰어(Viewer) 기능 등 비교적 단순한 기능만을 요구하므로 다수의 단말 환경에서 공통된 서비스가 가능한 환경을 제공한다.

3.1.2 서비스 구성

클라우드 방송 플랫폼을 통해 서비스와 UI를 제공하는 서비스 흐름은 그림 3과 같다.

클라우드 방송 서비스를 위해서는 서버 할당 및 세션 생성하는 과정과 어플리케이션 및 클라우드 서버 할당 하는 과정이 있다.

어플리케이션 및 클라우드 서버 할당은 어플리케이션을 클라우드 서비스 시스템에 미리 배포하거나 설정하여 사용자 요청이 있을 때, 서비스가 가능한 서버를 할당하기 위해 분산시스템에 등록하는 과정이다.

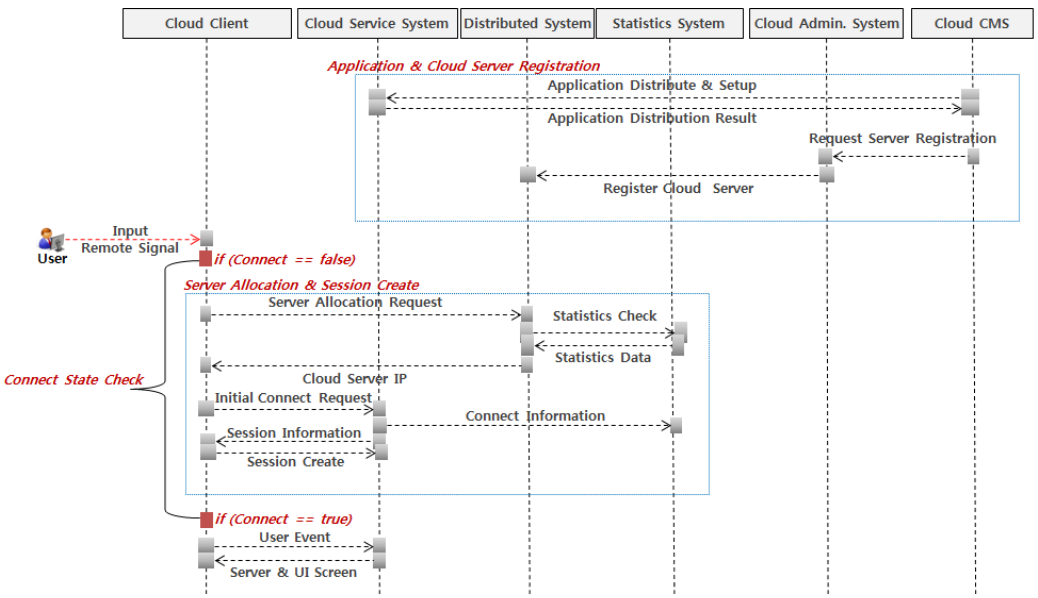


그림 3. 클라우드 방송 플랫폼을 통한 서비스와 UI 서비스 흐름
Fig. 3. Service and UI flow of cloud broadcasting platform

어플리케이션 서버가 클라우드 서비스 시스템으로 어플리케이션을 배포하고 어플리케이션 배포 결과를 받는다. 그리고 어플리케이션 서버는 클라우드 서버 관리시스템으로 클라우드 서버 등록 요청을 하고 클라우드 서버관리 시스템은 분산 시스템으로 클라우드 서버를 등록하게 된다.

서버 할당 및 세션 생성은 사용자 입력이나 요청에 의해 클라우드 클라이언트가 현재 접속 여부를 확인하고 접속이 되어 있지 않으면 분산시스템에 서버할당 요청을 한다. 분산시스템은 사용자의 서버 할당요청에 따라 서비스가 가능한 서버를 확인하고, 통계시스템에 조회하여 최적화된 클라우드 서버 IP를 클라우드 클라이언트에 할당한다. 그리고 클라우드 클라이언트는 클라우드 서비스 시스템으로 접속요청을 하고 세션 정보를 전달받아 세션을 생성하게 된다. 세션이 만들어진 이후에는 클라우드 클라이언트는 사용자의 요청을 클라우드 서비스 시스템으로 전달하게 되고 클라우드 서비스 시스템에서 만들어진 서비스 및 UI 정보를 받아 화면에 표시하게 된다.

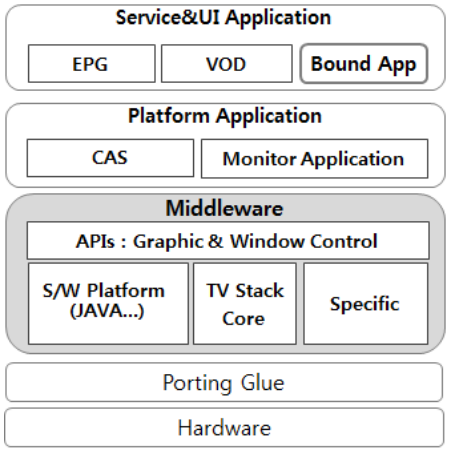
3.2 클라우드 방송 미들웨어 구성

3.2.1 STB의 소프트웨어 구조

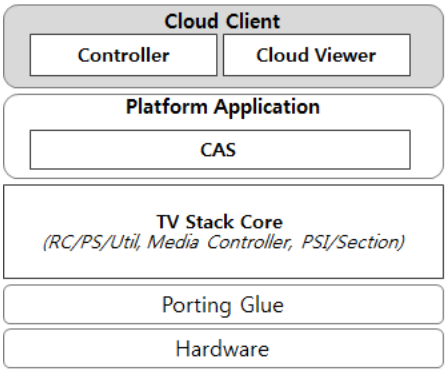
기존 방송 플랫폼에서 방송 단말기(STB 등)의 미들웨어 역할은 방송 서비스 제공을 위해 필요한 어플리케이션을 실행하고 관리하는 것이다. 기존 방송 플랫폼에서는 모든 어플리케이션이 단말에서 실행되고 관리되기 때문에 미들웨어에 종속적으로 서비스를 제공해야 하고 미들웨어 없이는 방송 서비스를 제공할 수 없는 구조였다. 하지만 클라우드 방송 플랫폼에서는 어플리케이션의 실행과 관리가 모두 클라우드 서비스 서버에서 되기 때문에 기존과 동일한 역할을 하는 단말기의 미들웨어는 필요가 없게 된다.

클라우드 방송 플랫폼을 이용하여 서비스와 UI를 제공하게 되면 STB을 포함한 가입자 단말기의 소프트웨어 구조가 달라지게 되는데 그 구조는 그림 4와 같다.

클라우드 방송 플랫폼에서 단말기의 소프트웨어는 기존 단말기 소프트웨어 구성에서 어플리케이션을 실행하고 관리하기 위해서 JVM(Java Virtual Machine)을 포함한 소프트웨어 플랫폼(Platform), 그래픽과 윈도우 관리를 위해 필요한 API, 단말기에서 자체적으로 실행되는 어플리케이션(Bound App 등)이 없어지게 된다. 이러한 기능들은 클라우드 클라이언트(Client)가 모두 수행하게 되고 클라우드 클라이언트



(a) Legacy STB 소프트웨어 구조



(b) 클라우드 STB 소프트웨어 구조

그림 4. 클라우드 방송 플랫폼의 단말기 소프트웨어 구조 (a) Legacy STB 소프트웨어 구조 (b) 클라우드 STB 소프트웨어 구조
Fig. 4. Software architecture of STB in cloud broadcasting platform (a) Legacy STB software architecture (b) Cloud STB software architecture

내에 있는 콘트롤러(Controller)는 채널전환 등 기존 EPG(Electronic Program Guide)가 하는 역할과 STB 자원을 관리 하는 역할, VOD 클라이언트의 역할도 같이 수행할 수 있다.

클라우드 단말기 소프트웨어는 구조가 간단하고 소프트웨어가 수행하는 역할도 상대적으로 줄어들기 때문에 단말기의 자원을 상대적으로 작게 사용하여도 단말기의 소프투웨어 구조와 하드웨어 성능의 종속성 없이 서비스의 제공이 가능하고 서비스의 안정성도 높아진다. 그리고, TV Stack은 범용으로 활용 가능해서 JVM과 소프트웨어 플랫폼 기능 때문에 지불해 왔던 미들웨어 라이센스 비용에 대해서 자유롭게 하는 구성이 가능하다.

3.2.2 서비스 흐름

클라우드 STB 소프트웨어 구조에서 단말기가 서비스와 UI가 실행하는 방식은 그림 5와 같다.

사용자가 리모콘을 입력하여 단말기의 하드웨어를 통해 TV Stack 코어(Core)에 이벤트가 전달되면, TV Stack 코어는 클라우드 클라이언트 내에 있는 콘트롤러(Controller)에 이벤트를 전달한다. 콘트롤러는 전달된 이벤트를 분석하여 키 값을 획득하고 클라우드 서버에 접속하여 클라우드 브라우저(Browser)에 키 값을 전달하게 된다. 클라우드 브라우저는 어플리케이션에 해당 값을 전달하여 어플리케이션에서 실행된 결과를 수신하게 된다. 이때, 어플리케이션의 처리 결과에 따라 채널전환, 화면변경, 인증요청 등 단말제어가 필요할 경우 제어 명령을 클라우드 브라우저가 콘트롤러에 단말 제어 요청을 보낸다. 콘트롤러는 단말 제어 요청을 TV Stack 코어를 통해 하드웨어에게 전달하여 단말을 제어한다. 어플리케이션 처리결과에 따라 서비스 화면전송이 필요할 경우 클라우드 브라우저는 클라우드 데이터를 이미지, 비디오, 메타 데이터 형태로 생성하여 클라우드 클라이언트에 전송한다. 클라우드 클라이언트는 클라우드 브라우저로 부터 받은 클라우드 데이터를 화면에 표시한다.

3.3 클라우드 방송 서비스 기술

클라우드 방송 플랫폼을 활용하여 서비스할 수 있는 방식은 이미지(Image) 방식, 비디오(Video) 방식, 메타데이터(Metadata) 방식 등 크게 3가지로 구분된다. 각 방식은 클라우드 서비스 시스템에서 사용자의 요청에 따라 화면을 구성하는 데이터를 만드는 방식에 따라 구분된다.

3.3.1 이미지 클라우드

그림 6는 이미지 클라우드의 서비스 구조이다. 이미지 클라우드는 사용자 입력/요청에 따라 실행한 클라우드 어플리케이션 화면을 '화면캡처'를 통해 이미지 데이터(Image Data)를 생성하고 생성된 이미지를 클라우드 클라이언트인 뷰어(Viewer)에 전송하여 서비스를 제공하는 방식이다. 실시간 방송 등 오디오와 비디오 스트림(Stream)외에 서비스를 구성하는 화면에 필요한 데이터를 방송 서비스 인프라를 통해 직접 수신 받아, 클라우드 서버에서 렌더링을 실행하며 GIF, JPG, PNG 등 일반적으로 사용하는 이미지의 다양한 포맷을 지원하도록 구성된다.

이미지 클라우드는 UI와 같은 빠른 화면전환이 필요하지만 화면 내에서는 움직임이 많지 않은 서비스에 효율적으로 활용할 수 있다.

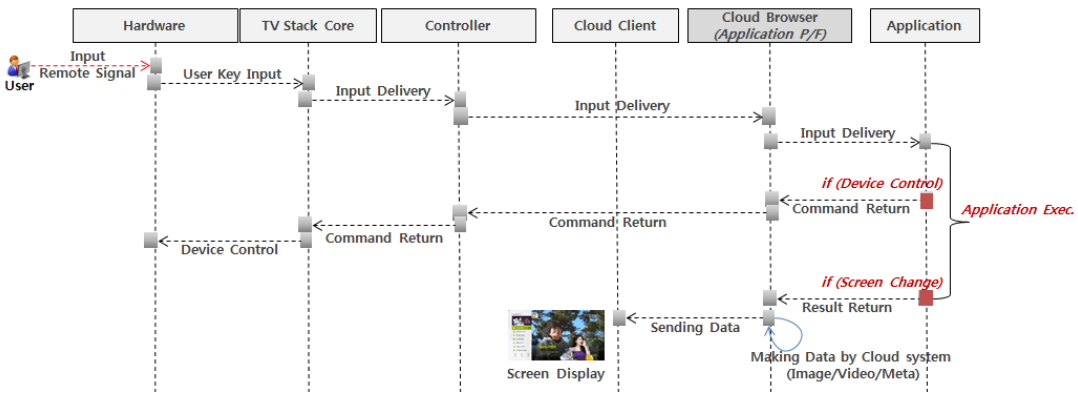


그림 5. 클라우드 STB 소프트웨어 구조에서 서비스와 UI의 흐름
Fig. 5. Service and UI flow of Cloud STB software architecture

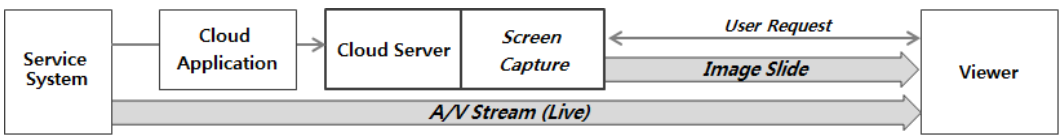


그림 6. 이미지 클라우드의 서비스 구조
Fig. 6. Service architecture of image cloud

3.3.2 비디오 클라우드

그림 7은 비디오 클라우드의 서비스 구조이다. 비디오 클라우드는 사용자 입력/요청에 따라 실행한 클라우드 어플리케이션 화면을 "화면 인코딩"을 통해 비디오 스트림으로 만들어서 클라우드 클라이언트인 플레이어에 전송하여 서비스를 제공하는 방식이다. 클라이언트는 단순히 오디오와 비디오를 재생 가능한 Player를 통해 모든 데이터를 수신하게 되므로, 서비스 제공에 필요한 단말의 리소스가 최소화 된다. 비디오 클라우드는 단말기의 리소스와 환경에 제약 없이 서비스를 구현할 수 있기 때문에 단말기의 리소스 자원이 많이 필요하거나 서비스 환경 설정이 어려운 서비스에 효율적으로 활용할 수 있다.

3.3.3 메타데이터 클라우드

그림 8은 메타데이터 클라우드의 서비스 구조이다. 메타데이터 클라우드는 서버에서 클라우드 어플리케이션 구현을 위해 필요한 메타데이터를 생성하여 만들어진 메타데이터를 클라우드 클라이언트인 렌드러로 전송하여 서비스를 제공하는 방식이다. 메타데이터는 클라우드 어플리케이션을 실행한 결과를 서버에서 분석하여 만들어진다. 클라우드 클라이언트는 전송된 메타데이터를 분석하고 화면 구성에 필요한 이미지(Image), 데이터(Data), 링크(Link) 등을 받아서 화면을 구성한다.

메타데이터 클라우드는 단말기 자원의 활용을 최대화 하여 애니메이션, 그래픽 효과 등 다양한 UI 효과를 나타내는데 효율적으로 활용할 수 있다.

3.4 클라우드 방송 플랫폼 효율화 기술

클라우드 방송 플랫폼 구성할 때, 클라우드 서버의 성능 중 동시 접속율(CCU : Concurrent Connected User)과 화면 전송지연(Latency)은 서버 하드웨어 수량과 관련되어 매우 중요한 요소이다. 클라우드 서버의 효율성을 높이기 위해 여러 처리 방식을 적용 하였으며, 방송 플랫폼 및 서비스에 적합한 구조의 클라우드 방송 플랫폼을 구축하였다.

3.4.1 동접율 효율화

그림 9는 클라우드 방송 플랫폼의 동접율을 개선하기 위한 구조를 나타낸다. 클라우드 서버가 클라이언트의 요청이 있을 때 마다, 어플리케이션 실행 후 화면을 전송하게 되면 서버 리소스 사용에 따른 성능부하와 화면 전송속도 지연이 발생한다. 따라서, 서비스 화면 중 사용자 공통 화면(홈메뉴, 프로모션, VOD카테고리 등)은 클라우드 데이터를 미리 생성하여 클라이언트 요청 시 바로 전송하는 프리프로세싱(Pre-Processing) 방식을 이용하여 서버 효율성을 높일 수 있으며, 클라우드 서버가 동시에 처리 가능한 클라이언트 수가 증가된다. 특히, 클라우드 서버 초기 접속에 필요한 서버 할당 및 어플리케이션 실행에 대한 지연시간이 최소화 되어 서비스 품질을 강화 할 수 있다. 프리프로세싱 처리를 통해 클라우드 클라이언트의 서비스 요청에 대한 서비스 인프라와 통신이 최소화 되므로 네트워크 트래픽 및 지연이 감소된다. 이러한 장점은 주요 서비스 인프라의 최대 접속 용량이 감소되어 인프라 서버 수 뿐만 아니라 네트워크 전송 용량에 대한 운영비용 절감이 가능하다.

3.4.2 트래픽 효율화

클라우드 서버는 서비스를 위한 데이터 획득과 서비스 화면을 전송하기 위해서 네트워크로 통신한다. 따라

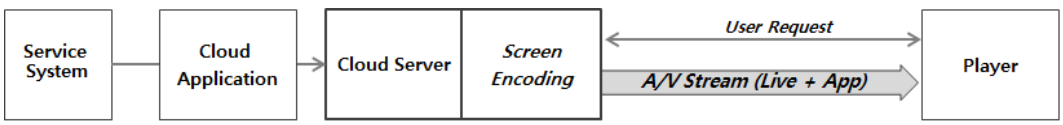


그림 7. 비디오 클라우드의 서비스 구조
Fig. 7. Service architecture of video cloud

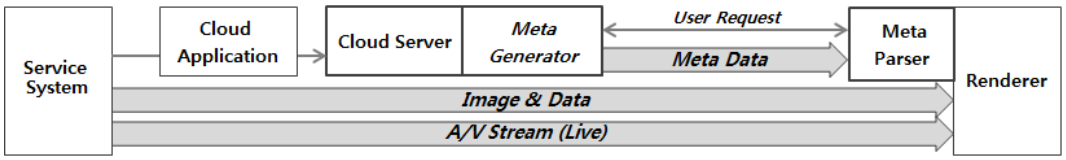


그림 8. 메타데이터 클라우드의 서비스 구조
Fig. 8. Service architecture of metadata cloud

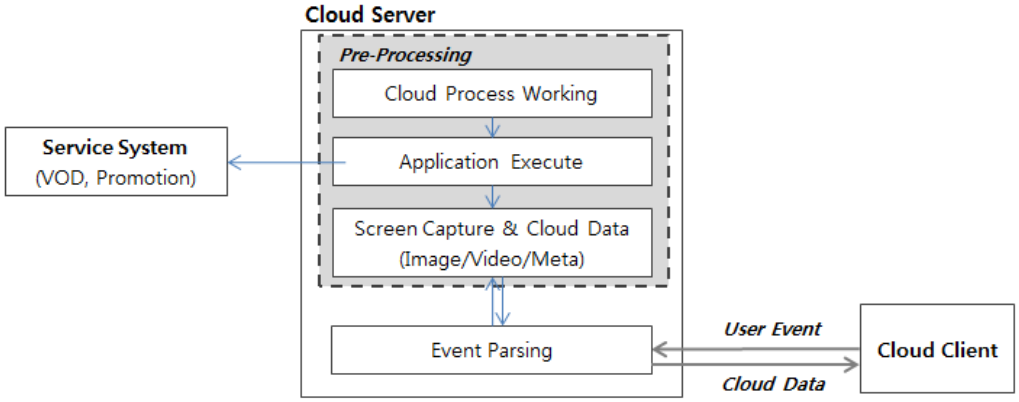


그림 9. 클라우드 방송 플랫폼 동접율 효율화 구조
Fig. 9. Concurrent rate efficiency architecture in cloud broadcasting platform

서 플랫폼을 효율화 하기 위해서는 데이터를 통신하는 각 구간별로 트래픽 효율화가 필요하다. 그림 10은 클라우드 서버의 각 트래픽 효율화에 대한 내용을 나타낸다.

(1) 전송 Data 캐싱

Cloud 클라이언트에서 Cloud Application으로 요청이 전달되면, Cloud Application은 사용자 요청을 분석하여 기존에 전송한 데이터가 있었는지 캐싱 Data를 확인한다. 만약 캐싱 Data가 존재 하면 Cloud 서버는 즉시 Cloud Data를 획득하여 전송하고, 존재하지 않으면 Application 실행 후 Data를 생성하고 클라이언트에게 Clcoud Data를 전송한다.

(2) URL 캐싱

Cloud 서비스는 Cloud 서버가 직접 '서비스 인프라 시스템(VOD, 편성 시스템)'과 통신하는 구조이므로, 서비스 인프라 시스템과 Cloud 서버간 통신은 네트워크 환경에 많은 영향을 받게 된다. 이러한 구조를 보안하기 위해 Cloud 서버에 URL 캐싱을 적용하여 인프라 시스템의 부하와 응답지연을 최소화 하고 Cloud 서비스 품질을 강화할 수 있다.

Cloud 클라이언트에서 Cloud Application으로 요청이 전달되면, Cloud Application은 URL 캐싱 목록을 조사한다. 만약 목록에 캐싱된 Data가 없을 경우 서비스 인프라에 직접 요청하고 결과를 캐싱하여 Application에 전달한다. 만약 캐싱 Data가 존재하면 Application은 캐싱 Data를 이용하여 결과를 실행한다.

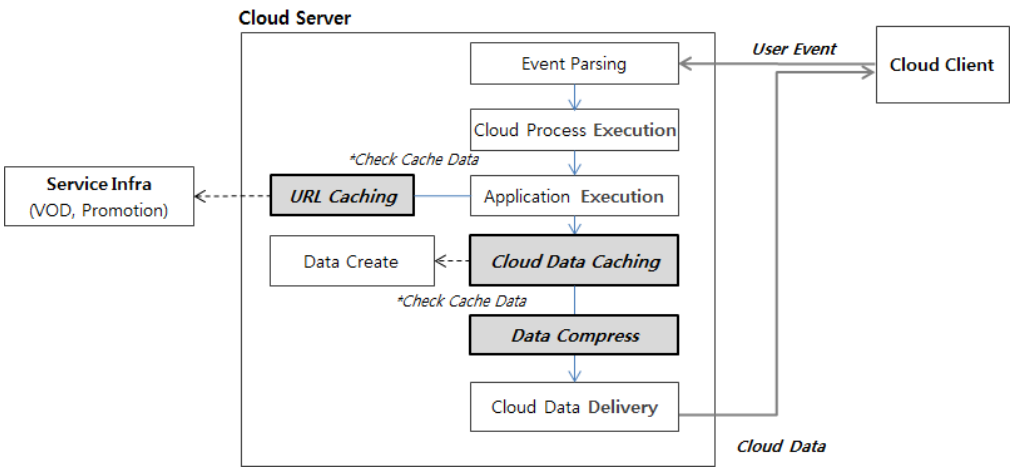


그림 10. 클라우드 서버의 트래픽 효율화
Fig. 10. Traffic efficiency architecture in cloud serve

(3) Cloud Data 압축

Cloud Data 압축은 Cloud Data 방식(Image/Video/Mata)에 따른 포맷 압축과 전송 프로토콜 압축을 이용한다. 이미지의 경우 JPG/PNG 압축, Video는 H.264, Meta의 경우 zip등의 압축방식이 가능하며, 클라이언트에서는 각 압축 방식을 지원한다.

3.4.3 네트워트 분산

그림 11은 케이블 방송 플랫폼에 클라우드 방송 시스템을 분산형 구조로 적용한 내용을 나타낸다.

안정적인 클라우드 방송 시스템을 구성하기 위해서는 케이블 방송의 지역별 네트워크 특성을 고려한 서비스 분산 구조가 적합하다. 방송 시스템의 구성상 각 SO(System Operator)의 네트워크 구성은 DMC(Digital Media Center) 집중구조로 구성되어 있으며, 백본(Backbone) 네트워크에 대용량 트래픽 수용이 필요하다.

이러한 구조에서 클라우드 서버를 지역별로 분산 구축하여 단말의 요청을 지역 SO내에서 처리 가능하도록 하여 DMC로 집중되어 있는 네트워크가 분산되며 안정적인 네트워크 운영을 할 수 있다. 네트워크 분산 구조를 이용하면 클라우드 서버 증설만으로 SO지역 및 단말확대에 따른 서비스 확장이 보다 쉽게 적용 가능하다.

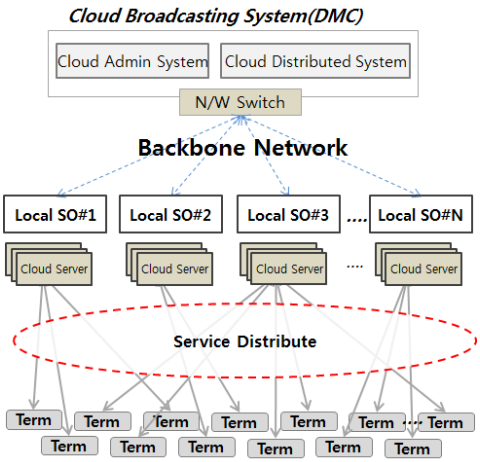


그림 11. 클라우드 방송 시스템의 분산형 구조
Fig. 11. Distribution architecture of cloud broadcasting system

3.4.4 서버 및 시스템 분산

클라우드 방송은 서비스의 실행을 서버 환경에 의존하므로 보다 높은 시스템 안정성이 요구된다. 클라우드 분산 시스템은 단말이 서버할당을 요청할 때, 각 SO의 클라우드 서버 동시접속 현황 및 프로세스(Process)를 감시하여 서버를 할당 하고, 서비스 접속 종료 시 프로세스를 종료하고 서버 반환을 요청한다. 만약 트래픽 증가 혹은 서버 및 네트워크 장애가 발생하여 해당 SO의 서비스 가능한 서버가 없을 경우, 타 지역 SO의 서버를 할당한다. 이러한 높은 클러스터 구조는 시스템의 부하 분산뿐만 아니라 시스템 및 네트워크 장애 발생 시 매우 높은 서비스 안정성을 확보할 수 있다.

3.5 클라우드 방송 서비스 개발 기술

그림 12는 클라우드 방송 서비스 플랫폼의 구성을 나타낸다. 플랫폼은 클라우드 엔진(Engine) 부분, 클라우드 브라우저(Browser) 부분, 브라우저 플러그인(Plug-in), 프레임워크(Framework) 부분으로 구분된다.

클라우드 엔진 부분은 클라우드 데이터를 생성하고 단말에 전송하는 역할을 담당하며, 클라우드 Browser의 처리 결과에 따라 지연을 최소화하여 화면 데이터를 클라이언트에 전송하는 역할을 담당한다. 클라우드 브라우저는 어플리케이션의 실행을 담당하며, 처리 결과에 따라 화면을 렌더링하고 어플리케이션 요청사항을 클라우드 엔진에 전달한다. 브라우저 플러그인 부분은 클라우드 브라우저 환경에서 지원되는 플러그인을 통해 어플리케이션 실행에 필요한 주요 환경을 제공한다. 마지막 프레임워크 부분은 클라우드 어플리케이션 실행에 필요한 다양한 서비스 환경에 대한 표준화된 인터페이스를 제공한다.

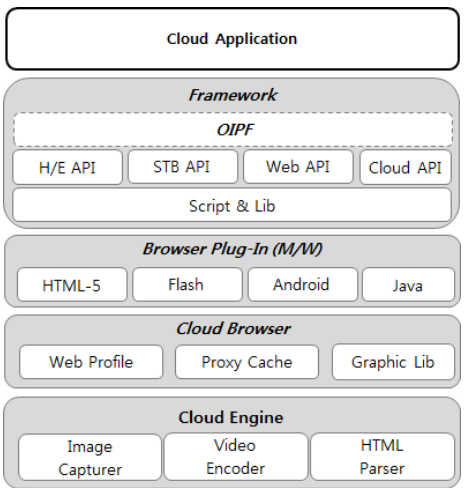


그림 12. 클라우드 방송 서비스 플랫폼 구성
Fig. 12. Cloud broadcasting service platform architecture

3.5.1 서비스 개발 플랫폼

클라우드 어플리케이션 개발은 클라우드 브라우져에서 지원 가능한 다양한 프로파일(Profile) 및 플러그인(Plug-In) 환경이 제공되며, 서버 환경(Server-Side)에서의 개발이 가능하다. HTML-5, Flash 등 Web 환경뿐만 아니라 플러그 인의 확장에 따라 기존에 단말에서 제공하던 미들웨어 인 Android, JAVA 등 환경도 제공한다. 클라우드 브라우져를 통한 단말 가상화 환경이 제공되어 어플리케이션 실행 결과를 단말 없이 확인 가능하다.

3.5.2 서비스 표준 개발 환경

기존 미들웨어 환경에서 어플리케이션 개발은 수많은 방송 인프라 및 인터페이스에 대한 분석과 다양한 통신 프로토콜을 활용하여 개발하였으나, 클라우드 환경에서 서비스 개발은 표준화된 인터페이스 제공이 가능하다. 기존 단말 환경에서 제공되던 플랫폼환경(방송, IP, CAS 등)뿐만 아니라, 클라우드 클라이언트와 통신을 위한 환경을 제공하기 위해서 프레임워크와 SDK, Lib 환경을 통해 표준 개발환경을 제공한다.

클라우드 어플리케이션 개발을 위한 표준화된 프레임워크는 방송인프라와 통신하기 위한 헤드엔드 API 부분과 STB 제어를 위한 STB API 부분, 브라우져 환경에서 사용하기 위한 Web API부분, 클라우드 서비스 제공을 위한 클라우드 API로 구성되며, 클라우드 브라우져와 호환성을 위해 스크립트(Script) 및 라이브러리(Library) 형태로 제공된다.

Ⅳ. 클라우드 방송 플랫폼 서비스

4.1 방송 서비스 UI

기존 레가시(Legacy) UI의 경우 사용자 서비스 흐름에 따라 단말 메모리상에 로딩이 완료되면 포커스 이동 및 단순페이지 이동에 따른 속도는 상대적으로 빠르다. 하지만 네트워크 통신과 대용량 데이터 처리에 대한 단말 리소스 부하로 인한 로딩은 상대적으로 매우 느리다. 이러한 환경에서 대부분의 서비스 공급자들이 제공하는 UI는 실시간 및 양방향서비스를 위해 매우 느린 로딩 시간을 감수하거나 고사양의 단말에 대한 투자를 지속적으로 진행하고 있다.

클라우드 UI의 경우 사용자 요청에 따라 클라우드 서버에 서비스 화면을 요청하게 되고, 처리된 결과만을 단말에 전송하게 된다. 이러한 구조는 네트워크 통신 구간 단순화와 송수신 데이터의 균일화를 통해 빠르고 동일한 속도의 서비스를 제공할 수 있다. 그림 13은 클라우드 UI를 이용한 VOD 서비스 흐름을 나타낸다.

4.1.1 서비스 구성

(1) VOD 서비스

사용자가 리모콘을 이용하여 키를 입력하면, STB은 사용자 키 입력을 클라우드 서버에 전달한다. 클라우드 서버는 클라우드 UI 어플리케이션을 실행하고 사용자가 입력한 키 값을 전달한다. UI 어플리케이션은 서비스 시나리오에 따라 키 입력을 처리하고 VOD 서버와 통신하여 VOD 편성 정보 및 에셋(Asset) 정보를 획득한다.

편성 정보에 따라 화면 구성에 필요한 포스터 등 화면 구성에 필요한 모든 파일 정보를 UI 어플리케이션에서 수신 받아 화면을 구성하고 최종 처리된 결과를 STB에 전송한다. UI 어플리케이션의 시나리오에 따라서 VOD 재생이 필요할 경우 STB은 클라우드 서버에서 수신 받은 VOD 재생 정보를 이용하여 VOD 스트림 서버에게 VOD 재생을 요청하여 영상을 재생한다.

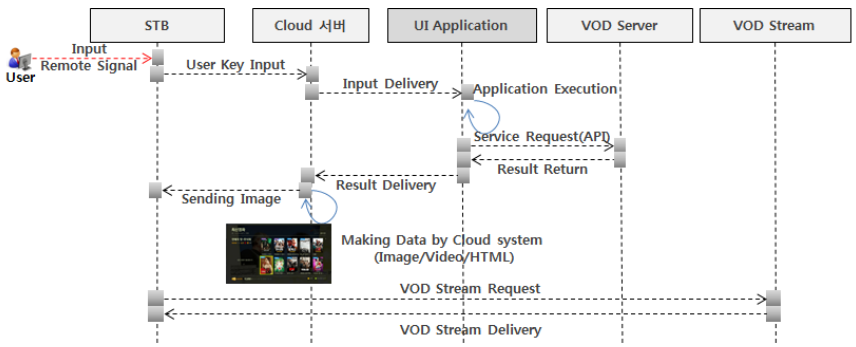


그림 13. 클라우드 UI를 이용한 VOD 서비스 흐름
Fig. 13. VOD service flow using Cloud UI

(2) 결제 서비스

사용자의 인증 정보 및 결제 정보는 STB에 저장되어 있고, 서비스 어플리케이션은 서버에서 실행되는 클라우드 서비스의 특성에 따라 결제를 위한 인증흐름은 클라우드 서버에서 STB으로 요청 해야 하며, 결제 요청은 클라우드 서버에서 진행된다. 인증을 위한 개인정보를 서버에 전송해야 하므로 인증정보 전송 및 결제 요청 시 클라우드 서버에 보안 및 암호화 방식 적용이 필요하다.

그림 14는 클라우드 방송 플랫폼을 활용한 결제 서비스 흐름을 나타낸다. 가입자가 서비스를 개통하면 프로비저닝서버(PVS)는 가입자 정보를 CAS(Conditional Access System)를 통해 STB에 전송하고, STB은 인증에 필요한 사용자 정보를 생성한다.

사용자가 서비스 이용 중 콘텐츠를 선택하여 결제가 필요할 경우 UI 어플리케이션은 STB에 인증을 위한 인증키 입력을 요청한다. STB은 사용자의 키 입력을 UI 어플리케이션에 전달하고 UI 어플리케이션은 전송해 오는 인증정보를 저장한다. 인증 키에 대한 입력이 완료되면 UI 어플리케이션 인증정보를 이용하여 STB에 입력된 인증 정보를 전송하여 승인과정을 진행한다.

STB을 통해 사용자 인증이 완료되면, UI 어플리케이션은 결제 서버에게 결제를 요청하고, 처리 결과를 STB에 전송한다.

4.2 데이터 방송

기존 데이터 방송은 실행 및 화면 전환 간 로딩시간 지연과 메모리, 그래픽의 제한으로 인한 낮은 품질 등의 많은 단점으로 인해 서비스 활성화에는 한계가 있다. 클라우드 방식을 이용하여 게임, 노래방, 키즈 등 기존의 데이터 방송 서비스를 고품질, 고사양으로 제공할 수 있다.

특히, 비디오 클라우드 방식을 이용하여 고사양 게임, 스마트 서비스 등의 독립형 서비스를 제공하여 데이터 방송 서비스를 강화할 수 있다.

4.2.1 독립형 서비스

실시간 채널 없이 전체 화면이 서비스로 구성되는 독립형 서비스는 비디오 클라우드 방식을 이용하여

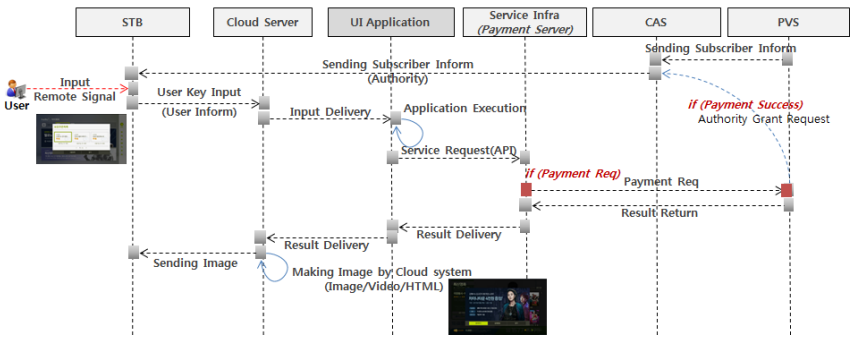


그림 14. 클라우드 방송 플랫폼에서 결제 서비스 흐름
Fig. 14. Payment service flow in cloud broadcasting platform

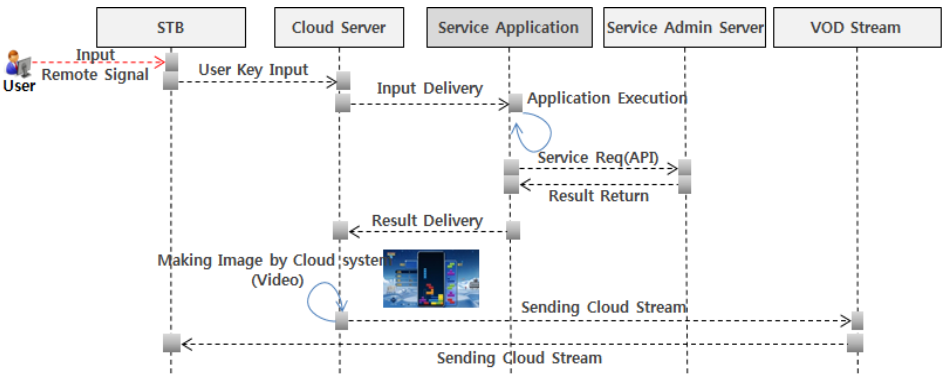


그림 15. 클라우드 방송 플랫폼에서 독립형 데이터방송 서비스 흐름
Fig. 15. Independent data broadcasting service flow in cloud broadcasting platform

STB의 하드웨어와 소프트웨어의 제약 없이 서비스 제공이 가능하다. 서비스 제공에 필요한 API 및 콘텐츠를 클라우드 서버환경에서 직접 제공하므로 STB의 환경과 상관없이 다양한 서비스를 구성하여 서비스를 제공할 수 있다. 그림 15는 클라우드 방송 플랫폼에서 독립형 데이터방송의 서비스 흐름을 나타낸다.

사용자가 서비스 실행을 요청하면, 클라우드 서버는 사용자가 요청한 서비스 어플리케이션을 실행한다. 서비스 어플리케이션은 서비스 제공에 필요한 API를 제공하는 서비스 관리서버와 통신하여 서비스 구성에 필요한 콘텐츠를 이용하여 화면을 구성한다. 클라우드 서버는 서비스 어플리케이션의 실행된 결과를 실시간으로 Video 클라우드로 변환하여 VOD 스트림 서버에게 전송하고, STB은 VOD 스트림 서버를 통해 실행 결과를 수신 받아 사용자에게 제공한다.

4.3 확장형 서비스

클라우드 방식을 이용하여 채널과 관련된 다양한 화면을 생성하여 전송할 수 있다. 비디오 클라우드 방식을 이용하여 직접 채널을 구성하여 전송하거나, 채널과 관련된 부가 정보를 실시간으로 동기화 하여 이미지 방식으로 전송하여 기존 부가서비스 방식을 확장하여 다양한 클라우드 확장형 서비스를 구현하였다.

4.3.1 서비스 구성

(1) 멀티뷰

클라우드 서버가 직접 다수의 채널을 수신 받아 멀티뷰 화면을 구성하고 STB에 전송하는 방식으로 멀티뷰 서비스 제공이 가능하다. 클라우드 서버의 효율성을 높이기 위해 비디오 클라우드를 이용하여 멀티뷰 채널을 송출하고 이미지 클라우드를 통해 채널을 선택할 수 있도록 멀티뷰 서비스 UI를 제공하였다.

그림 16은 클라우드 방송 플랫폼에서 멀티뷰 서비스의 흐름을 나타낸다. 멀티뷰 채널생성은 서비스 관리서버에 설정에 따라, 다수의 실시간 채널을 직접 클라우드 서버가 수신 받고 다양한 레이아웃으로 구성하여 멀티뷰 화면을 생성한다. 생성된 멀티뷰 화면은 클라우드 서버를 통해 비디오 클라우드 방식을 이용하여 실시간 채널 헤드엔드로 전송한다.

멀티뷰 서비스 UI는 사용자가 클라우드 서버를 통해 멀티뷰 서비스 실행을 요청하면 멀티뷰 어플리케이션이 실행되고 어플리케이션은 서비스 관리서버를 통해 채널 접속 정보 및 부가 서비스 정보를 수신 받아 멀티뷰 서비스 UI 화면을 구성한다. 멀티뷰 서비스 UI는 STB에 이미지 클라우드 방식으로 전송되고 멀티뷰 채널을 이용하기 위한 채널 정보도 함께 STB에 전송한다. STB은 멀티뷰 서비스 UI 화면과 멀티뷰 채널을 동시에 사용자에게 제공하여 멀티뷰 서비스를 제공한다.

(2) 채널 연동형 서비스

연동형 데이터 방송 서비스를 제공하기 위해서는 채널의 별도 대역폭을 점유할 뿐만 아니라, 실시간 채널과 동기화되어 서비스하기 불가능하다. 클라우드 방식을 이용한 채널 연동형 서비스는 클라우드 서버에서 채널과 동기화된 데이터를 직접 수신 받아 각 채널 별로 다양한 서비스를 구성하여 서비스를 제공한다.

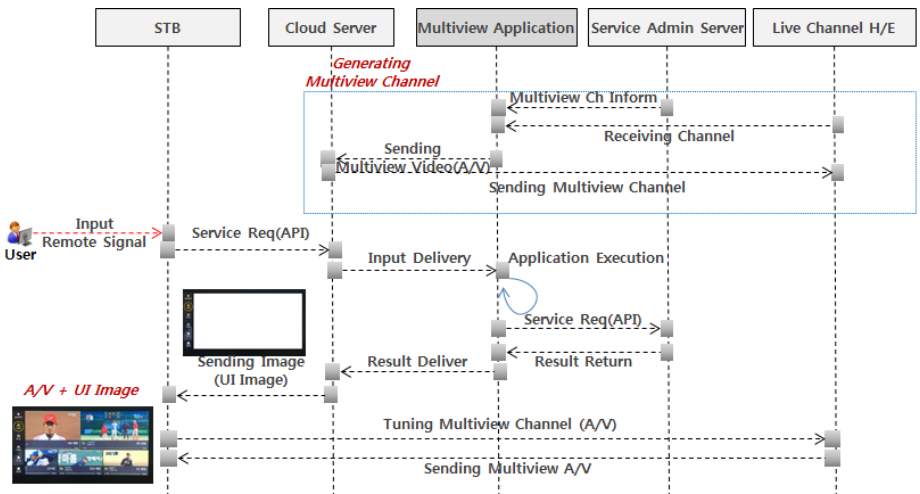


그림 16. 클라우드 방송 플랫폼에서 멀티뷰 서비스 흐름
Fig. 16. Multi-view service flow in cloud broadcasting platform

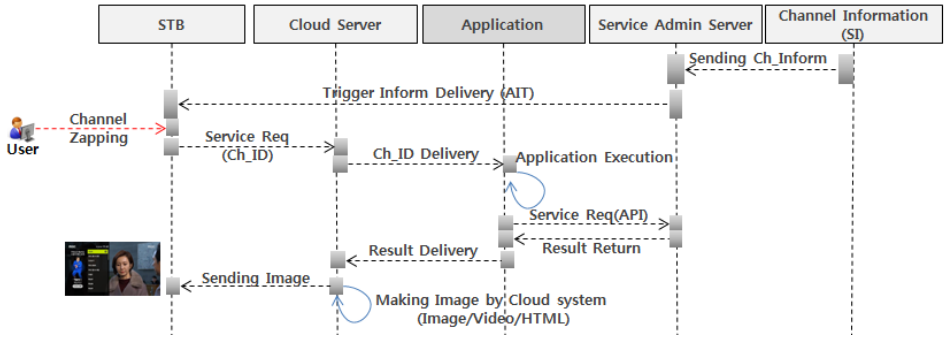


그림 17. 클라우드 방송 플랫폼에서 채널 연동형 서비스 흐름
Fig. 17. Program related service flow in cloud broadcasting platform

그림 17은 클라우드 방송 플랫폼에서 채널 연동형 서비스의 흐름을 나타낸다. 서비스 관리서버는 STB이 수신 받고 있는 SI(System Information) 정보를 수신을 받아 채널과 동기화된 부가 정보를 실시간으로 생성하고, 서비스 대상 채널을 구성하여 STB에게 정보를 전송한다.

STB은 사용자가 전환 요청한 채널이 연동형 서비스 대상채널일 경우, 클라우드 서버에게 해당 채널ID를 서버에게 전달한다. 클라우드 서버는 연동형 어플리케이션에게 채널ID를 전달하여 채널에 설정된 부가 정보를 이용하여 서비스를 구성하고 실행된 결과 화면을 생성하여 STB에 전달 한다.

V. 클라우드 방송 플랫폼 분석 평가

클라우드 방송 플랫폼을 상용화 하여 서비스와 UI를 변경하였고, 이를 통해 품질과 성능측면에서 기존 방송서비스 보다 많은 개선이 확인되었다. 클라우드 방송 플랫폼에서 개선된 품질과 성능을 비교분석하기 위해서 환경구성을 하여 측정하였다. 방송서비스를 위해 필요한 헤드엔드와 STB의 하드웨어는 동일하게 하였고, 클라우드 방송 환경은 클라우드 방송 플랫폼을 정합하여 STB의 소프트웨어 구조를 클라우드 STB 소프트웨어 구조로 변경하였다.

방송플랫폼은 오픈케이블(OpenCable) 기반의 케이블 방송 헤드엔드이다. STB의 칩셋은 BCM7229 (1345DMIPS), OS는 Linux, Middleware는 OCAP1.0, Cable Modem은 DOCSIS 2.0, RAM은 256MB, Flash는 128MB이다. 클라우드 방송 플랫폼은 이미지 클라우드 방식으로 서버와 STB에 적용이며, 클라우드를 적용한 UI와 서비스는 국내 케이블 방송사가 상용화하여 250만 이상의 가입자가 사용하고 있는 HD STB의 UI와 서비스이다.

5.1 서비스 제공 속도

기존 디지털 방송 플랫폼(Legacy)에서 UI는 미들웨어가 서버로부터 여러 이미지들과 데이터를 받아서 UI를 구성하여 표현하게 된다. 이러한 구조에서는 미들웨어가 여러 이미지들과 데이터를 수신하는 시간과 이들을 조합하여 화면으로 만드는 시간이 필요하다. 하지만, 클라우드 방송 플랫폼을 UI에 활용하게 되면 클라우드 서버에서 처리된 결과 화면만 단말기가 받아서 나타내기 때문에 서비스 구조를 단순화 하고 표준화 할 수 있어 빠르고 일정한 속도로 서비스를 제공할 수 있다. 클라우드 플랫폼 적용 후 전반적인 서비스 제공 속도가 향상되었다. 홈메뉴 및 VOD서비스를 제공하는 UI영역뿐만 아니라 독립형 서비스와 채널 연동형 등 데이터 방송 서비스도 진입시 발생하는 로딩과 페이지전환에 따른 시간이 감소되었으며, 전체적인 모든 서비스 영역의 품질이 향상되었다.

표 1은 서비스 유형에 따른 클라우드 서비스 제공 속도 측정 결과이다. 클라우드 적용 후 UI 반응속도가

표 1. 방송 서비스 유형에 따른 클라우드 서비스 제공 속도
Table 1. Comparison of average service latency in broadcasting service

Service Type	Average service latency (sec)	
	Legacy platform	Cloud platform
Home promotion	1.1	0.6
VOD UI & Menu	1.2	0.5
EPG Menu, Program Guide	1.3	0.6
Game, Education, Kids	2.2	0.7
Channel VOD, VOD Trailer	1.5	0.8
Channel Trigger	1.5	0.8
Search, Coupon, Event Portal	1.4	0.7
Average latency	1.5	0.7

향상되었다. 클라우드 플랫폼 적용 이후 서비스 특성에 상관없이 균일한 반응속도를 보인다. 독립형 데이터 방송의 경우 다양한 서비스 유형에 따라 2초 이상 반응시간이 소요되었으나 균일화 적용 후 3배 이상 빠른 속도를 제공한다. 특히, 채널 독립형 서비스의 경우 채널 전환 간 서비스 화면이 빠르게 노출되어야 하며 A/V 전환 속도와 비슷한 수준으로 제공되어야 한다. Cloud플랫폼 적용 결과 채널 전환 속도보다 빠르게 화면이 노출되고 있으며 다양한 서비스 확대가 가능하다.

5.2 UI 서비스 지연속도(Latency)

클라우드 서비스 제공 속도 중 홈메뉴와 VOD서비스를 제공하는 UI서비스에 대해 아래와 같이 측정하였다. UI 지연속도 측정 시나리오는 사용자가 가장 많이 사용하는 VOD 서비스 진입 후 데이터 전송량이 가장 많은 VOD 포스터 리스트 노출을 위주로 진행하였다. 홈 메뉴를 호출하여 VOD 서비스 선택하기 위해 메뉴 이동을 2차례 이동한 후 VOD 서비스의 초기화면으로 진입하여 카테고리 리스트를 노출한다. 카테고리를 2차례 이동하여 VOD 포스터 10장이 동시에 구성되는 화면을 진입하여 포스터 10장이 노출되는 다음 페이지로 이동 후 VOD선택하여 VOD의 상세화면을 나타나는 개별 시간을 측정한다.

표 2는 기존 방송 플랫폼과 클라우드 방송 플랫폼에서 UI의 평균전환 시간을 비교한 내용이다.

클라우드 방송 플랫폼에서 UI 전환시간은 평균 약 0.5초였고, 기존 레가시 미들웨어 방식에서의 평균 UI 전환시간은 평균 약 1.4초 보다 약 65% 줄어들었다. 그리고, 클라우드 플랫폼 UI는 기존 미들웨어 방식의 UI와 달리 서비스 형태에 관계없이 균일할 속도를 나타낸다는 특징이 있다.

5.3 UI 데이터 소비(Data Consumption)

레가시(Legacy) 방송 플랫폼에서 UI구성을 위해 필요한 데이터는 STB의 메모리에 저장시켜 이용하거나 혹은 서버로부터 필요한 데이터를 받아서 사용한다. STB의 메모리에 저장되어 있는 데이터는 별도의 네트워크 연결 없이 사용할 수 있지만, 서버로부터 필요한 데이터의 수신이 필요할 때에는 네트워크 통해 데이터를 수신한다. 수신해야 하는 데이터량이 많을수록 UI 지연시간이 증가하게 된다.

표 3은 기존 방송 플랫폼과 클라우드 방송 플랫폼에서 사용하는 평균 데이터 소비량을 비교한 내용이다. 측정 시나리오는 UI 지연속도 방식과 동일하다.

기존 레가시 방식에서는 STB이 미리 수신 받아 저장하고 있는 데이터를 노출하는 화면에서는 네트워크 데이터를 수신 받지 않는다. 홈 메뉴 화면은 레가시 방식에서는 미리 멀티캐스트 방식으로 STB에 저장된 데이터를 사용하므로, 전송 데이터가 없다. 또한 화면 내에서 포커스 이동의 경우 네트워크 데이터가 필요 없다. 하지만 화면 초기 진입시에 화면 구성에 필요한 전체 데이터를 한번에 수신 받아야 하므로 화면 전환에는 대용량 네트워크 데이터를 수신 받는다.

클라우드 방송 플랫폼에서 UI를 위해 사용하는 네트워크 데이터는 평균 약 112,104Byte였고, 레가시 미들웨어 방식에서의 사용하는 평균 네트워크 데이터는

표 2. UI 평균전환 시간 비교
Table 2. Comparison of average UI transition latency

UI	Average UI transition latency(sec)	
	Legacy platform	Cloud platform
Home menu exposure	0.5	0.4
Home menu navigation 1	0.6	0.4
Home menu navigation 2	0.5	0.5
VOD category exposure	2	0.6
VOD category navigation1	0.6	0.5
VOD category navigation2	0.7	0.5
VOD post list exposure1	3.2	0.7
VOD post list exposure2	3.3	0.5
VOD detail information	1.2	0.7
Average latency	1.4	0.5

표 3. UI의 평균 데이터 소비량 비교
Table 3. Comparison of average UI data consumption

UI	Average data consumption(Byte)	
	Legacy platform	Cloud platform
Home menu exposure	0	11,399
Home menu navigation 1	0	9,909
Home menu navigation 2	0	12,180
VOD category exposure	594,265	70,631
VOD category navigation 1	0	9,909
VOD category navigation 2	0	8,180
VOD post list 1	147,150	113,103
VOD post list 2	145,481	109,110
VOD detail information	112,042	90,218
Average data consumption	112,104	48,293

평균 약 48,293Byte 보다 약 57% 줄어들었다.

레가시 방식에서는 UI를 구성하는 여러 이미지와 데이터를 분리해서 전송하여 단말기의 미들웨어가 조합하는 구조이지만, 클라우드 방송 플랫폼에서 UI 화면 전체나 변경된 부분을 하나의 이미지로 전송하기 때문에 데이터량을 줄어들게 된다.

Ⅵ. 결 론

지금까지 방송기술은 서비스의 안정성과 보안성이 강조되어 왔기 때문에 보수적이고 폐쇄적인 구조를 가지고 있었다. 하지만 클라우드 기술의 발달로 인해 방송에도 적용되면서 이러한 특징들이 변화되고 있다. 방송 서비스 플랫폼에 클라우드 기술이 적용되면 하드웨어와 소프트웨어 구성에 따른 서비스의 제한이 없어지고 서비스 개발 및 운영에 대한 효율성이 개선되는 효과를 가지게 된다. 하지만, 지금까지 방송에는 비디오 중심의 클라우드 기술이 많이 소개 되었지만 이는 높은 투자 비용문제로 인해 확산에 문제가 있었다.

이러한 문제를 해결하기 위해 이미지 중심의 클라우드 서비스 플랫폼을 상용화 하여 클라우드 기술이 많은 방송사업자들에게 확산 및 안정화 하는데 기여하였다. 클라우드 서비스 플랫폼은 UI와 서비스 적용되어 화면전환속도 개선, 네트워크 데이터 감소, UI 화질 개선, 서비스 이용율 개선, 장애율 감소, 개발 및 운영 프로세스 효율화, 서비스 진입장벽 개선, 사용 이력 분석 등의 효과가 나타났다.

클라우드 서비스 플랫폼은 서버 효율성 개선, 방식 및 기능의 다양화, 스크린 확대, 관리 및 운영 플랫폼으로 지속적인 개발을 하고자 한다.

References

[1] Y. Cho, J. M. Seok, and D. Y. Suh, “3D-based monitoring system and cloud computing for panoramic video service,” *J. KICS*, vol. 39, no. 9, pp. 590-597, Sept. 2014.

[2] J. Kim, S.-I. Park, and H. M. Kim, “Initial timing acquisition algorithm for terrestrial cloud transmission systems,” *J. KICS*, vol. 39, no. 9, pp. 870-879, Sept. 2014.

[3] M. A, V. Nayak, S. Alur, and K. Gudi, “Virtualization strategy to implement cloud-based IPTV system and to optimize cloud resources,” *Int. J. Advanced Res.*, vol. 2, no. 6, pp. 675-682, 2014.

[4] S. K. Gundala, V. P. Goranthala, and Mohd. F. A. Parvez, “Delivering IPTV service through VSTB to optimize cloud resources,” *IJCSEE*, vol. 4, no. 4, pp. 19-23, Oct. 2014.

[5] V. Aggarwal, V. Gopalakrishnan, R. Jana, K. K. Ramakrishnan, and V. Vaishampayan, “Exploiting virtualization for delivering cloud-based IPTV services,” in *Proc. IEEE INFOCOM(mini-conference)*, Shanghai, Apr. 2011.

[6] R. Neill, L. P. Carloni, A. Shabarshin, V. Sigaev, and S. Tcherepanov, “Embedded processor virtualiza- tion for broadband Grid computing,” *IEEE Computer Soc.*, pp. 145-156, 2011.

[7] H.-I. Kim and S.-K. Park, “A hybrid video-on-demand boradcasting and receiving scheme of harmonic and staggered schemes,” *IEEE Trans. Broadcasting*, vol. 54, no. 4, pp. 771-778, Dec. 2008.

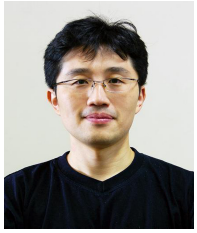
[8] Y. Choi, Y. Lim, and J. Park, “Reinforcement learning approach for resource allocation in cloud computing,” *J. KICS*, vol. 40, no. 4, pp. 653-658, Apr. 2014.

[9] H.-I. Kim and S.-K. Park, “Staircase staggered data broadcasting and receiving scheme for popular video services,” *J. KICS*, vol. 31, no. 10, pp. 919-928, Oct. 2006.

[10] M. Alicherry and T. V. Lakshman, “Network aware resource allocation in distributed clouds,” in *Proc. IEEE INFOCOM*, pp. 963-971, Orlando, USA, Mar. 2012.

[11] S. T. Maguluri, R. Srikant, and L. Ying, “Stochastic models of load balancing and scheduling in cloud computing clusters,” in *Proc. IEEE INFOCOM*, pp. 702-710, Orlando, USA, Mar. 2012.

김 홍 익 (Hong-Ik Kim)



1996년 2월 : 한국항공대학교 전자공학과 공학사
2003년 2월 : 한양대학교 전자통신전파공학과 공학석사
2007년 2월 : 한양대학교 전자통신전파공학과 공학박사
1996년 3월~2000년 8월 : 삼성항공 전산팀
2007년 1월~현재 : CJ헬로비전 기술실
<관심분야> Digital Signal Processing, Cloud Service Platform, Bigdata Analysis, Multimedia Platform, Neural Network, Pattern Recognition

이 종 한 (Jong-Han Lee)

1994년 2월 : 성균관대학교 정보공학과 학사
2009년 2월 : 연세대학교 경영대학원 석사
2013년 9월 : 고려대학교 정보보호대학원 박사수료
1994년~2001년 : 데이콤
2001년 : LG전자
2002년~2003년 : 알티캐스트 이사
2003년~2013년 : NDS 지사장
2013년~현재 : CJ헬로비전 기술실장
<관심분야> 디지털 방송, CAS, 서비스 보안

이 동 익 (Dong-Ik Lee)

2006년 2월 : 서울과학기술대학교 컴퓨터공학과 학사
2007년 1월~2011년 11월 : 아카넷TV 기술연구소
2011년 12월~2013년 7월 : 데이콤 멀티미디어 미디어기술팀
2013년 7월~현재 : CJ헬로비전 스마트기술팀
<관심분야> 전자공학, 통신공학, 광통신 공학