

# 데이터 방송 애플리케이션의 효율적 다운로드를 지원하는 리턴 채널 활용 기법

강 승 미<sup>†</sup> · 선 승 상<sup>\*\*</sup> · 엄 영 익<sup>\*\*\*</sup>

## 요 약

기존 데이터 방송 시스템은 수신 단말이 완전한 애플리케이션 데이터를 획득하지 못한 경우 반복적으로 방송 애플리케이션 데이터 전체를 다운로드하는 매커니즘을 가지고 있다. 이 방식은 다운로드 시간이 길어지는 요인이 된다. 이러한 문제점을 개선하기 위해 본 논문은 리턴 채널을 활용하여 누락된 방송 애플리케이션 데이터를 효과적으로 다운로드하는 기법을 제안하고 실험을 통하여 제안된 기법의 비교 성능 평가를 보인다.

키워드 : 데이터 방송, 리턴 채널

## A Scheme for Return Channel Utilization for Efficient Download of Data Broadcasting Applications

Seung-Mi Kang<sup>†</sup> · Seungsang Sun<sup>\*\*</sup> · Young Ik Eom<sup>\*\*\*</sup>

## ABSTRACT

The existing data broadcasting system has the mechanism that the receiving device should download repetitively the entire broadcasting application data until it acquires the whole of the application data. This mechanism takes long time for the download. To solve this problem, this paper proposes a scheme that the device effectively downloads the missing parts of broadcasting application data through the return channel and shows the performance evaluations of the proposed scheme with respect to the experiment.

Key Words : Data Broadcasting, Return Channel

### 1. 서 론

아날로그(analog)로 시작한 방송은 디지털 방송(digital broadcasting)을 거쳐 데이터 방송(data broadcasting)의 시대로 진화하고 있다. 데이터 방송은 데이터를 비디오와 함께 방송하는 것을 의미하고, 애플리케이션 및 이미지, 텍스트 파일 등과 같은 데이터를 디지털화된 영상에 방송 관련 정보나 쇼핑, 뉴스, 증권 등의 응용 프로그램들과 함께 실어 보냄으로써, 시청자가 능동적으로 방송에 참여할 수 있도록 지원한다.

현재 국내에서는 2002년에 한국 디지털 위성방송 스카이라이프(SkyLife)가 DVB-MHP(Digital Video Broadcasting Multimedia Home Platform) 방식으로 양방향 데이터 방송 서비스를 개시했고, 지상파(MBC, KBS, SBS, EBS)를 통해

ACAP-J(Advanced Common Application Platform-Java) 기반의 데이터 방송이 서비스되고 있으며, 케이블 TV에서도 OCAP(OpenCable Application Platform)을 기반으로 한 데이터 방송 서비스를 준비 중이다.

데이터 방송 서비스는 관련된 산업 전반에 걸쳐 새로운 고부가가치를 창출할 것으로 예상되어, 세계 각국에서 서비스 실현을 위한 표준화 및 규격화(이하 "표준화") 작업을 진행해 왔다. 데이터 방송을 수신하기 위해서는 방송 미들웨어를 탑재한 수신 단말이 필요하다. 하지만, 방송 미들웨어는 다양한 표준들이 묶여 표준화가 진행되고 있어서 상호운영성에 문제를 포함하고 있는 상황이다.

데이터 방송 서비스는 브로드캐스팅(broadcasting)이라는 단방향성을 기본적인 특징으로 갖는다. 따라서, 수신 측에서 예기치 못한 상황이 발생해도 송신하는 방송국에서 에러처리를 해줄 수는 없다. 그러나 수신단말은 데이터 방송에서 서비스하는 애플리케이션을 사용하기 위하여 미들웨어와 함께 리턴 채널을 갖게 되기 때문에, 기존 방식에서 지원할 수 없었

† 정 회 원 : 성균관대학교 컴퓨터공학과 석사과정  
\*\* 준 회 원 : 성균관대학교 컴퓨터공학과 박사과정  
\*\*\* 중 심 회 원 : 성균관대학교 정보통신공학부 교수  
논문접수: 2005년 5월 9일, 심사완료: 2005년 9월 23일

던 에러 처리 기능, 누락된 데이터 재전송 기능들에 대한 구현이 가능해졌다.

본 논문의 제 2 장에서는 방송 미들웨어 표준과 배경 기술을 기술하고, 제 3 장에서는 기존 방송 애플리케이션 다운로드 방식에 대한 분석 및 문제점을 지적한다. 제 4 장에서는 방송 애플리케이션 다운로드 속도를 향상시킬 수 있는 알고리즘을 제시하고, 제 5장에서 이에 대한 분석과 더불어 성능 평가 결과를 기술한다. 끝으로 제 6장에서 제안된 기술의 효과와 향후 연구과제를 제시한다.

## 2. 관련 연구

### 2.1 방송 미들웨어 표준

데이터 방송의 전송과 수신 및 표현 기술에 대한 표준들은 DVB-MHP, ACAP, OCAP 등이 있다.

#### 2.1.1 DVB-MHP

DVB-MHP는 디지털 TV와 인터넷 및 웹을 결합하여, 양방향 및 다목적 통신 네트워크에서 사용자와 사업자가 공유할 수 있는 멀티미디어 환경을 제공함으로써, 표준을 준수하여 개발하는 하드웨어는 새로운 서비스가 추가되더라도 계속 사용 가능하게 하는 것을 목적으로 하고 있다. DVB-MHP 표준의 핵심은 하드웨어에 의존하지 않는 JVM(Java Virtual Machine)의 사용이다. 애플리케이션을 실행시키기 위해 MHP API(Application Program Interface)들은 JVM에서 동작하는 여러 인터페이스를 제공한다. 또한 DVB-SI(Service Information), DSM-CC(Digital Storage Media Command and Control) 등의 전송 프로토콜, 애플리케이션에 대한 정보를 제공하기 위한 신호처리 프로토콜, 보안 프로토콜 등을 함께 제공한다. MHP가 포함하는 API는 JavaTV, JMF(Java Media Framework), HAVi-UI(Home Audio/Video Interoperability-User Interface) 등이 있다[1], [2].

#### 2.1.2 OCAP

OCAP은 케이블 방송에서 양방향 서비스를 위한 애플리케이션 제작의 기반이 되는 표준이다. CableLabs는 OpenCable 호환 디지털 케이블 방송 수신기를 위한 데이터 방송을 위해서 미들웨어 표준화 작업을 1999년부터 진행하여, MHP 1.01 표준을 바탕으로 OCAP 1.0을 발표하였다. OpenCable은 OCAP이라는 공통된 미들웨어 플랫폼을 사용함으로써 애플리케이션 소프트웨어와 콘텐츠의 공유를 가능하도록 하고 있다. 또한, 케이블 방송 환경을 고려하기 위하여 OCAP은 MHP, ACAP과는 달리 Monitor Application, Unbound Application, XAIT(eXtended Application Information Table), OOB(Out-Of-Band) Return Channel 등을 포함하고 있다[3], [4].

#### 2.1.3 ACAP

ACAP은 미국 데이터 방송 표준의 하나로 지상파 데이터 방송과 케이블 데이터 방송의 콘텐츠 호환성을 위해 ATSC

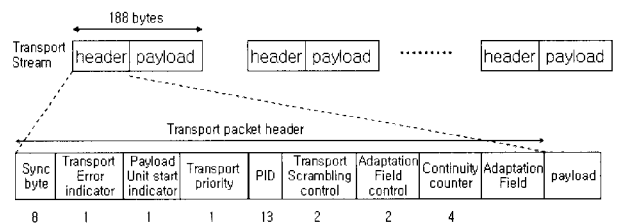
와 CableLabs의 공조로 만들어진 ATSC 표준안이다. ACAP은 ATSC-DASE(Advanced Television Systems Committee-DTV Application Software Environment) 와 OCAP의 단일화된 미들웨어 표준이라고 할 수 있으며, GEM(Globally Executable MHP) 표준을 기반으로 구성된다. ACAP 표준은 아직 Candidate Standard 상태이나, 국내 지상파 방송국에서는 ATSC-DASE 규격을 따르지 않고 ACAP-J 표준을 기반으로 한 데이터 방송을 서비스하고 있다[5], [6].

#### 2.1.4 관련 배경 기술

데이터 방송은 기존의 A/V 중심적인 단방향 서비스에 국한되지 않고, 다채널, 고화질, 다양한 부가서비스를 제공하는 애플리케이션을 제공한다. 데이터 방송의 수신 단말은 방송 스트림과 함께 전송되는 애플리케이션을 수신하여 실행하기 위해서는 전송 계층(transport layer)가 따르는 표준을 만족해야 한다. 따라서 본 절에서는 전송 표준인 MPEG-2 TS (Transport Stream), DSM-CC 표준과 현재 방송 애플리케이션의 다운로드 방식에 대해서 분석한다.

#### 2.1.5 MPEG-2 Transport Stream

MPEG-2는 1994년 ISO/IEC 13818로 규격화된 영상압축 기술 표준이다. 전송 및 분배를 목적으로 오류 가능성이 있는 분야(TV 방송 및 네트워크 전송 등)에서는 전송 스트림(TS, Transport Stream)을 사용하고, DVD와 같은 저장 매체나 오류 가능성이 없는 LAN에서는 프로그램 스트림(PS, Program Stream)을 사용한다. 현재 MPEG-2 TS는 디지털 TV 방송의 지상, 위성 및 케이블 분야를 비롯하여, 컴퓨터 통신을 비롯해서 저장 미디어 분야에서도 표준으로 채택되고 있다[7].



(그림 1) MPEG-2 TS Packet 구조

MPEG-2 TS는 (그림 1)과 같이 188바이트 패킷들로 구성된다. TS의 헤더(header)에서 가장 중요한 것은 패킷 식별자(PID, Packet Identifier)로써, 다중화된 TS를 역다중화하여 ES(Elementary Stream)들을 만들어 내는데 필요한 정보이다. TS가 페이로드(payload)에서 포함하고 있는 데이터들을 종류별로 역다중화하기 위해서, TS 규격에서는 전송될 데이터의 PID 목록을 전달하기 위한 데이터 형식을 별도로 규정하고 있는데, 이를 PSI(Program Specific Information)라고 한다. PSI는 TS에 포함된 데이터들의 PID 목록을 전달하기 위해 PAT(Program Association Table)와 PMT(Program Map Table)를 정의하고 있다. PMT의 elementary stream loop은 Video, Audio 및 DSM-CC 전송 스트림의 PID를 포

함하고 있다[8], [9].

### 2.1.6 DSM-CC

DSM-CC는 MPEG-1 및 MPEG-2 비트 스트림을 다루기 위한 제어 기능 및 함수를 정의하고 있는 프로토콜로서, ISO/IEC 13818의 6장에 정의되어 있다. DSM-CC는 클라이언트와 서버가 포함된 사용자 시스템(User System)과, 서비스와 자원관리자(SRM, Service and Resource Manager)가 포함된 네트워크 시스템으로 구성되며, U2U(User-to-User), U2N(User-to-Network)의 두 인터페이스가 규정된다. 방송 애플리케이션의 전송을 위해서는 DSM-CC U-U 객체 캐러셀(object carousel)과 DSM-CC 데이터 캐러셀(data carousel)을 이용한다. 데이터 캐러셀은 데이터 모듈을 주기적으로 전송하는 메커니즘이며, 객체 캐러셀은 데이터 캐러셀을 통하여 DSM-CC U-U 파일과 계층적 디렉토리 구조를 주기적으로 전송하는 메커니즘이다.

데이터 캐러셀에서 데이터는 모듈단위로 구성되며, 모듈은 하나 혹은 그 이상의 페이로드를 갖는 블록(block)으로 나누어진다. 데이터 캐러셀은 실제 데이터를 담고 있는 DDB(Download Data Block)로 표현된 다운로드 데이터(download data) 메시지와 모듈들의 집합에 대한 정보를 갖는 DSI(Download Server Initiate) 메시지와 DII(Download Info Indication)로 표현된 다운로드 제어(download control) 메시지로 구성된다. DSI는 최상위 제어 메시지이며, 단말에서 애플리케이션을 다운로드 받았을 때 데이터를 가져오기 위해 가장 먼저 찾게 되는 메시지이다. DII는 데이터 캐러셀 안의 모듈들에 대한 정보를 담고 있다[10], [11].

객체 캐러셀은 디렉토리, 파일, 스트림 등의 객체들을 이용해서 방송 서버로부터 수신 단말에게 구조화된 데이터 그룹을 전송하는 것이다. 구조화된 데이터 그룹의 최상위 디렉토리를 서비스 게이트웨이(SGW, Service Gateway)라고 한다 [12].

따라서 (그림 2)에서 보이는 바와 같이 방송 애플리케이션

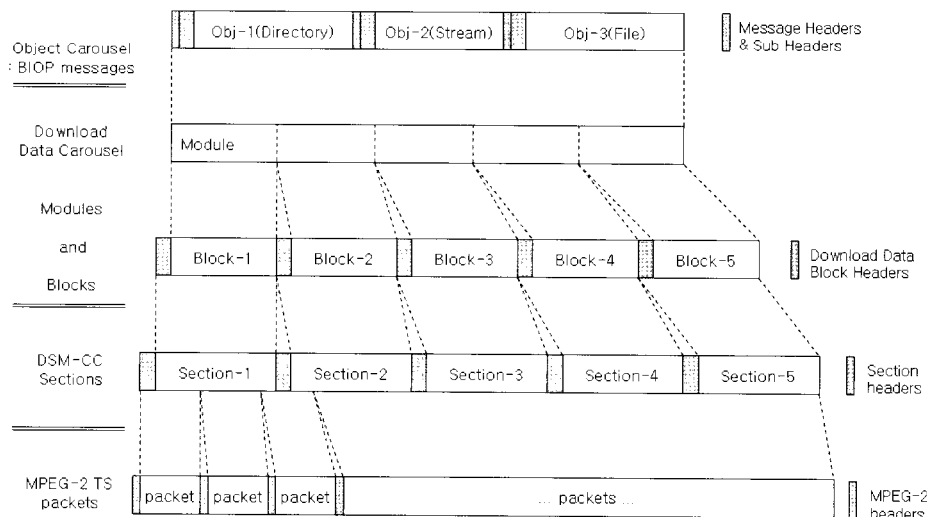
은 객체 캐러셀로 생성되어 데이터 캐러셀과 MPEG-2 섹션에 캡슐화되고, MPEG-2 TS로 패킷화되어 전송된다. 이 MPEG-2 TS를 수신하는 단말에서는 DSM-CC 섹션을 꺼내어 데이터 캐러셀 메시지로 해석하고, 다시 객체 캐러셀화 하는 과정을 거치게 된다. 객체 캐러셀 메시지가 데이터 캐러셀 모듈에 담겨질 때, 데이터 캐러셀의 DSI 메시지에는 서비스 게이트웨이 정보가 포함된다[13].

## 3. 기존 방송 애플리케이션 다운로드 방식 분석 및 문제점

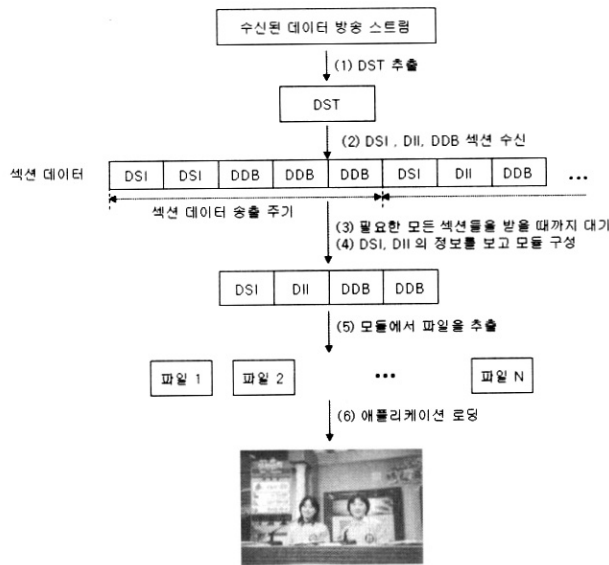
### 3.1 기존 방송 애플리케이션 다운로드 방식

일반적인 데이터 방송에서 애플리케이션을 다운로드하는 첫 번째 과정은 방송 단말에 들어오는 스트림으로부터 가장 먼저 DST(Data Service Table)를 추출해 내는 것이다. 데이터 섹션들은 일정 단위로 스트림에 실리기 때문에, 이후 애플리케이션에서 필요한 파일을 읽기 위하여 각 섹션의 DSI, DII에서 제공하는 루트 정보를 추출하여 DDB를 수신하고 모듈화한다[14], [15].

애플리케이션 다운로드 과정을 도식화한 (그림 3)에서 알 수 있듯이, 하나의 애플리케이션을 실행하기 위해서는 필요한 섹션 데이터 전체가 수신될 때까지 기다려야 한다. 그러나, 송수신 채널에서의 비트 에러, 수신 단말의 CPU 성능, 하드웨어 또는 소프트웨어의 불안정한 동작 등의 이유로 모듈 구성을 위해 필요한 섹션 데이터가 누락되는 경우가 있다. 방송 스트림에는 FEC(Forward Error Correction) 코드가 적용되어 있지만 수신 단말에서 에러를 정정할 수 있는 한계를 넘어서는 경우에는 비트 에러가 발생하게 된다. 방송 스트림에 포함된 캐러셀은 주기적으로 전송되기 때문에 하나의 섹션 데이터를 수신하지 못할 경우, 수신 못한 섹션 데이터는 다음 송출 주기까지 기다려야 한다.



(그림 2) 객체 캐러셀(Object Carousel)과 데이터 캐러셀(Data Carousel)의 구조



(그림 3) 애플리케이션 다운로드 과정

3.2 기존 방송 애플리케이션 다운로드 방식의 문제점

데이터 방송에 실려오는 애플리케이션 데이터는 사용자가 언제 방송 수신을 시작할지 모르기 때문에 순차적이면서 주기적으로 송출된다. 따라서 사용자가 애플리케이션을 다운로드 하기 위해서는 최소한 한 주기 이상의 섹션 데이터 송출을 기다려야 함을 3.1절에서 설명하였다.

방송 애플리케이션을 방송 신호를 통해서 전송하는 것 외에 리턴 채널을 이용해서 전송하는 방식도 논의되고 있다. 방송 신호에 주기적으로 실려오는 애플리케이션과는 다르게 시청자가 원하는 시간에, 원하는 부분에 대해서만 애플리케이션 정보를 수신할 수 있으므로 부분적인 에러로 인해서 오랜 시간을 기다려야 할 필요가 없다. 그러나 리턴 채널을 이용해 전체 애플리케이션을 전송하면 동시 시청자의 수가 늘어날 수록 서버에 걸리는 부하가 크게 늘어난다. 또한 이러한 방식으로 데이터 전송이 이루어질 경우 서버와 리턴 채널에 걸리는 부하가 크게 늘어나 양방향 통신의 정보 전달시간이 늘어나게 되는 문제점을 가지고 있다.

4. 리턴 채널을 활용한 방송 애플리케이션 다운로드 속도 향상 기법

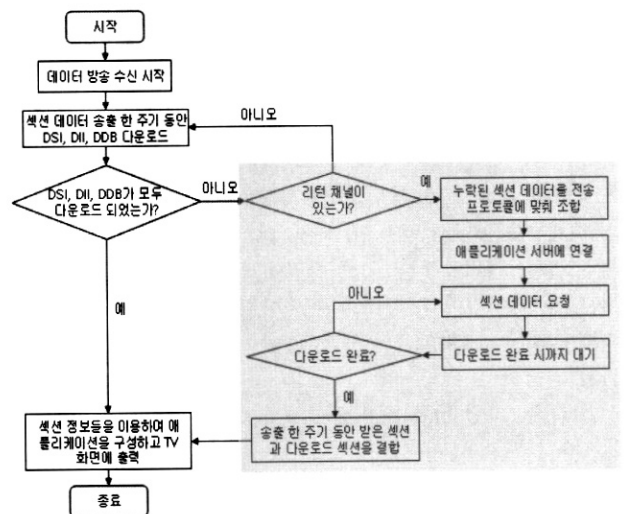
데이터 방송을 지원하는 단말은 양방향 서비스를 가능하게 하기 위하여 네트워크 리턴 채널을 가지고 있고, 데이터 방송 스트림은 다양한 제어 메커니즘을 제공한다. 이를 기반으로 본 장에서는 본 논문에서 제안하는 방송 애플리케이션 다운로드 속도를 향상시키는 알고리즘을 기술한다.

4.1 제안 알고리즘

본 논문에서 제시하는 방송 애플리케이션 다운로드 알고리즘은 데이터 방송 수신 시 애플리케이션 다운로드를 위해서 데이터 방송 스트림과 리턴 채널을 효율적으로 이용함으

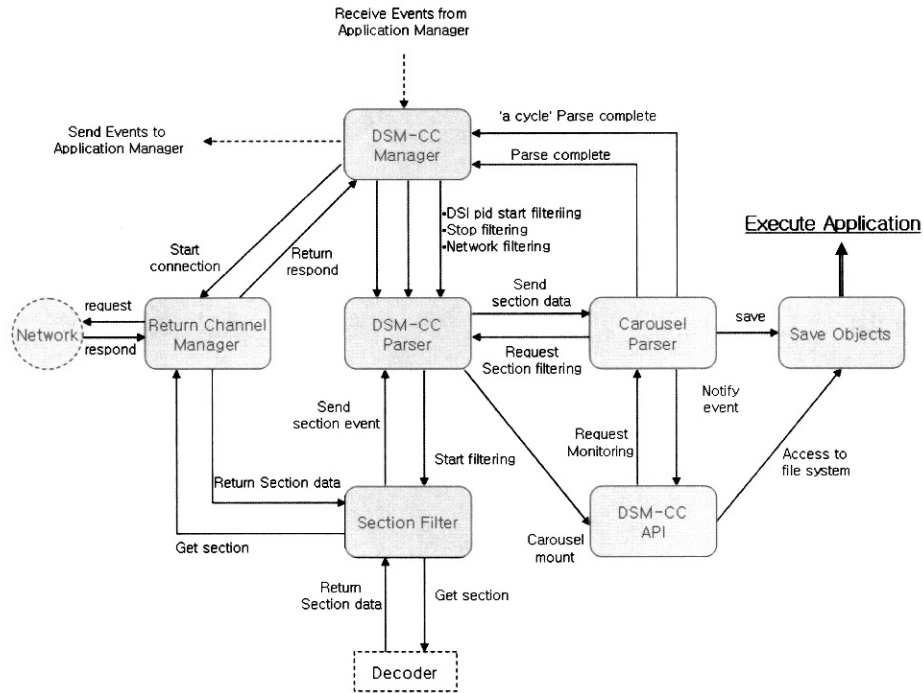
로써 애플리케이션 다운로드 시간을 줄이고, 방송 서버와 리턴 채널에 걸리는 부하를 최대한 줄이는 것에 목적이 있다.

제안 알고리즘은 데이터 방송 수신 단말에서 다음과 같이 동작한다. 데이터 방송 수신 단말은 시청자가 데이터 방송 수신을 위해서 전원을 켜거나 채널을 변경하면 방송 신호로부터 애플리케이션을 다운로드 한다. 방송 수신 단말은 한 주기 동안의 섹션 데이터 송출을 다운로드한 후 누락된 섹션 데이터가 있는지를 검사한다. 누락된 섹션 데이터가 없는 경우, 수신 단말은 수신된 섹션 데이터를 모듈화하고 애플리케이션 파일들을 추출하여 TV 화면에 보여준다. 누락된 섹션 데이터가 있는 경우, 수신 단말은 다음 주기를 다시 기다리지 않고 리턴 채널이 존재하는지 검사한다. 리턴 채널이 있는 경우, 수신 단말은 리턴 채널을 이용해서 방송국의 서버에 접속하여 누락된 섹션 데이터만을 받아온다. 수신 단말은 리턴 채널을 통한 데이터 수신을 완료하면, 기존의 방송 신호로부터 수신된 섹션 데이터와 조합하여 애플리케이션 파일들을 추출한다. 수신 단말은 애플리케이션 파일의 추출이 완료되면 이를 구동시켜 TV 화면에 그래픽으로 나타낸다. (그림 4)는 이와 같은 알고리즘으로 동작하는 애플리케이션 다운로드 방식을 도시한 것이다.



(그림 4) 제안하는 알고리즘의 동작

수신 단말에서는 DSI와 DII의 정보를 통해 전체 모듈을 구성하는 DDB의 정보에 대해 알 수 있으므로, 섹션 데이터 송출 한 주기가 지난 뒤에 어떤 DDB 데이터가 누락되었는지 파악할 수 있다. 수신 단말은 네트워크 프로토콜을 통해 애플리케이션 서버와 연결을 맺고, 누락된 DDB의 ID로 데이터 다운로드를 요청한다. 애플리케이션 서버는 요청이 들어온 DDB 데이터를 정해진 포맷에 맞추어 단말로 전송한다. 단말은 요청한 데이터와 수신한 데이터가 동일인지 확인하여, 동일하면 네트워크 연결을 종료하고 다르다면 데이터를 다시 요청한다. 다운로드가 끝나면 단말은 애플리케이션 서버로부터 받은 DDB 데이터와 방송 신호로부터 받은 DDB 데이터를 조합하여 애플리케이션 파일들을 추출하고 실행시킨다.



(그림 5) 제안 알고리즘의 클래스 다이어그램

제안하는 알고리즘을 적용한 애플리케이션 다운로드 과정을 클래스 다이어그램으로 표현하면 (그림 5)와 같이 나타낼 수 있다. 애플리케이션의 라이프사이클(lifecycle)을 관장하는 Application Manager(이하 “AM”)가 DSM-CC 프로토콜을 통하여 디코더(Decoder)로 들어오는 섹션 데이터를 파싱하고 조합하여 애플리케이션을 실행시킨다. DSM-CC Manager는 AM과의 이벤트 통신을 처리하고, 파서를 구동시키고 제어하는 역할을 한다. DSM-CC Manager로부터 DSId PID를 받아 필터링(filtering)을 시작하라는 메시지를 받은 DSM-CC Parser는 실제 섹션 데이터를 얻어오기 위한 필터링을 수행하게 된다. DSM-CC Parser는 송출 한 주기 동안 섹션 필터링을 수행하고 저장하며, 한 주기가 지나면 누락된 섹션이 있는지 여부를 Carousel Parser를 통해 DSM-CC Manager에게 알린다. Carousel Parser는 DII와 DDB에 대한 모니터링을 담당하게 되므로 누락된 섹션에 대한 정보를 파악할 수 있다. DSM-CC Manager는 누락된 섹션이 있으면 어떤 섹션이 누락되었는지 파악하여 Return Channel Manager(이하 “RCM”)를 구동시킨다. 애플리케이션 서버와 연결되면 RCM은 상태를 DSM-CC Manager에게 알리고, DSM-CC Manager는 DSM-CC Parser에게 네트워크 필터링을 시작하라고 알려주게 된다.

RCM은 제안 알고리즘의 핵심이 되는 부분으로써, 다음과 같은 역할을 하게 된다.

- 네트워크 리소스 사용 가능 여부 판단.
- 애플리케이션 서버와의 연결 관리.
- DSM-CC Manager와의 이벤트 통신.
- 네트워크를 통한 섹션 필터링.
- 네트워크를 통해 수신된 섹션 데이터가 요청한 데이터

와 다를 경우, 재시도할 것인지 지상파를 통해 수신할 것인지 여부 판단.

- 수신 데이터의 에러 처리.

송출 한 주기 동안 지상파를 통해 수신하여 저장해놓은 섹션 데이터와 네트워크를 통한 섹션 데이터를 결합하고 나면, Carousel Parser가 DSM-CC Manager에게 ‘parse complete’ 이벤트를 보낸다. DSM-CC Manager는 AM에게 다시 통보함으로써 AM이 애플리케이션의 라이프 사이클을 관리하여 애플리케이션을 실행할 수 있도록 해준다.

#### 4.2 알고리즘 분석

이번 절에서는 4.1절에서 제시한 알고리즘의 성능을 수학적으로 분석한다. 4.2.1에서는 분석을 위한 가정 및 변수를 정의하고, 4.2.2에서는 기존의 알고리즘의 성능을 수학적으로 분석하며, 4.2.3에서는 본 논문에서 제시한 알고리즘의 성능을 수식적으로 분석한다. 4.2.4에서는 기존의 알고리즘에서의 성능과 제시한 알고리즘의 성능을 비교한다.

##### 4.2.1 가정 및 변수 정의

알고리즘 분석은 다음과 같은 가정을 두고 이루어진다.

- 방송 데이터의 수신은 애플리케이션 수신을 위해서 필요한 초기 데이터를 수신한 후 모든 데이터를 받기까지의 시간으로 정의한다(모든 DDB를 수신하기까지 걸리는 시간).
- 각각의 DDB의 수신 확률은 서로 독립적이다.
- 네트워크를 사용하여 DDB를 수신할 때, DDB의 전송시간은 DDB의 길이와 네트워크 전송율(transmission rate), 단말과 서버 사이의 Round Trip Time으로 결정된다.

알고리즘에 사용되는 변수들과 각각의 정의는 다음과 같다.

- $T$  : 캐러셀 한 주기 (같은 DDB가 반복되어서 들어오기까지 걸리는 시간)
- $N$  : 캐러셀 한 주기 동안 전송되는 DDB 수
- $p$  : 하나의 DDB에 대해서 수신 실패할 확률
- $1-p$  : 하나의 DDB에 대해서 수신 성공할 확률
- $P_N$  : N개의 DDB를 모두 전송받기 위해서 기다려야 하는 주기 수(Random Variable)
- $E[P_N]$  : N개의 DDB를 받기 위해서 기다려야 하는 평균 캐러셀 주기 수
- $E[P_T]$  : 기존의 알고리즘을 사용했을 때, 모든 DDB를 받기까지 걸리는 평균 시간
- $E[\tilde{P}_T]$  : 제안한 알고리즘을 사용했을 때, 모든 DDB를 받기까지 걸리는 평균 시간
- $Tr$  : 단말이 사용하는 네트워크의 Data Rate(Transmission Rate)
- $RTT$  : 단말과 DDB 서버 사이의 네트워크 Round Trip Time
- $L$  : DDB 하나의 길이 (bits)

#### 4.2.2 기존 방송 데이터 수신 알고리즘 분석

Random Variable  $X$ 를 캐러셀 한 주기 동안 수신하지 못하는 DDB의 개수라고 정의한다. 하나의 DDB를 수신하지 못하게 되는 경우의 수는 수신 실패 확률  $p$ 를 확률 값으로 가지는 베르누이 분포(Bernoulli distribution)에 따라서 정의될 수 있으므로 Random Variable  $X$ 는 이항 분포(Binomial distribution)에 의해서 표현될 수 있다. 따라서 한 주기 동안 수신되지 못하는 DDB의 개수는  $X$ 의 평균값  $E[X]$ 로 표현된다. 이산 분포 정의에 따라서  $E[X]$ 는 다음 수식과 같이 표현할 수 있다.

$$E[X] = Np \tag{1}$$

다음은 모든 DDB를 받기 위해서 단말이 캐러셀 몇 주기 정도를 기다려야 하는지가 정의되어야 하므로,  $N$ 개의 DDB가 있을 경우,  $N$ 개의 DDB를 모두 받는데 걸리는 캐러셀 주기의 평균 수를  $E[P_N]$ 으로 정의한다. 첫 번째 주기에서 빠질 수 있는 DDB의 개수를  $k$ 라 하며,  $k$ 개의 DDB를 받기 위해 기다려야 하는 평균 캐러셀 주기 수를  $E[P_k]$ 로 정의한다.  $E[P_N]$ 은 다음과 같은 식으로 표현될 수 있다.

$$E[P_N] = \text{Prob}(\text{첫번째 주기에서 모든 DDB를 받을 확률}) * 1 + \sum_{k=1}^N \text{Prob}(\text{첫번째 주기에서 DDB를 } k \text{ 개 못 받을 확률}) * (1 + E[P_k])$$

이를 통해  $E[P_N]$ 은 다음과 같이 정리하여 표현할 수 있다.

$$E[P_N] = (1-p)^N * 1 + \sum_{k=1}^N \binom{N}{k} (1-p)^{N-k} p^k (1 + E[P_k]),$$

$$E[P_N] = 1 + \sum_{k=1}^N \binom{N}{k} (1-p)^{N-k} p^k E[P_k]$$

위 식을  $E[P_N]$ 에 관해서 정리하면 다음 수식 (2)와 같다.

$$E[P_N] = \frac{1}{1-p^N} \left\{ 1 + \sum_{k=1}^{N-1} \binom{N}{k} (1-p)^{N-k} p^k E[P_k] \right\} \tag{2}$$

위 식에서,  $N$ 이 그리 크지 않은 정수로 가정하고,  $p$ 를 작은 실수로 가정하면,  $p^4$ 이하 항은 무시할 정도로 작으므로, 수식 (3)과 같이 단순화 시켜서 표현할 수 있다.

$$E[P_N] \approx \frac{1}{1-p^N} \left\{ 1 + \sum_{k=1}^3 \binom{N}{k} (1-p)^{N-k} p^k E[P_k] \right\}, N \geq 4$$

$$E[P_N] \approx \frac{1}{1-p^N} \left\{ 1 + \binom{N}{1} (1-p)^{N-1} p E[P_1] + \binom{N}{2} (1-p)^{N-2} p^2 E[P_2] + \binom{N}{3} (1-p)^{N-3} p^3 E[P_3] \right\} \tag{3}$$

위 식에서  $E[P_1], E[P_2], E[P_3]$ 의 값은 (2)식에 대입하여 다음과 같이 구할 수 있다.

$$E[P_1] = \frac{1}{1-p}, E[P_2] = \frac{1+2p}{1-p^2}, E[P_3] = \frac{1}{1-p^3} \left( 1+3p + \frac{3p^3}{1+p} \right)$$

(3)식에 대입하여 정리하면  $E[P_N]$ 은 다음과 같이 단순화 시켜서 표현할 수 있다.

$$E[P_N] \approx \frac{1}{1-p^N} \left\{ 1 + N(1-p)^{N-2} p + \frac{N(N-1)}{2} (1-p)^{N-3} p^2 \frac{1+2p}{1+p} + \frac{N(N-1)(N-2)}{6} (1-p)^{N-4} p^3 \left( \frac{1}{1+p^2+p^3} (1+3p + \frac{3p^3}{1+p}) \right) \right\} \tag{4}$$

DDB를 모두 받는데 걸리는 평균 시간을  $E[P_T]$ 로 표현하면 수식(5)와 같이 표현할 수 있다.

$$E[P_T] = E[P_N] * T \tag{5}$$

#### 4.2.3 제안한 알고리즘 분석

제안한 알고리즘에서는 한 주기 동안 DDB를 수신한 후에, 한 주기 동안 수신하지 못한 DDB에 대해서는 네트워크를 통해서 전송 받는 것으로 정의하고 있다. 따라서 제안한 알고리즘을 사용할 경우 전체 DDB를 수신하는데 걸리는 시간은 캐

리셀 한 주기 시간에 남은 DDB를 네트워크를 통해 받는 시간을 더하여 계산할 수 있다. 남은 DDB의 평균 개수는  $E[X]$ 로 결정된다. 네트워크를 통해서 받는 시간은 수신하고자 하는 전체 데이터 길이를  $Tr$ 로 나눈 후, 서버와 단말간의  $RTT$ 를 더한 값으로 정의가 가능하다. 따라서 전체 DDB 수신을 하기 위해서 필요한 시간의 평균 값은 다음 식 (6)으로 표현할 수 있다.

$$E[\tilde{P}_T] = T + \frac{E[X]*L}{Tr} + RTT \quad (6)$$

그러므로 제안한 알고리즘의 성능 향상 효과는 (5)식과 (6)식을 비교하여 표현할 수 있다.

$$Performance\_increase = \left( \frac{(5)}{(6)} * 100 - 100 \right) \%$$

$$= \left[ \left( \frac{E[P_N]*T}{T + \frac{E[X]*L}{Tr} + RTT} \right) * 100 - 100 \right] \% \quad (7)$$

여기서  $E[X]$ 는 (1)식을 통해서 표현되며,  $E[P_N]$ 은 (4)식을 통해서 표현된다.

#### 4.2.4 제안한 알고리즘의 성능 향상 효과 분석

4.2.3절에서 알고리즘 분석식을 전개하는 과정에서 알고리즘 분석식 단순화를 위해서 ‘ $p$ 의 값이 매우 작아 하나의 DDB를 잃을 확률이 매우 작다’는 가정 하에 알고리즘 분석을 하였다. 이러한 가정 하에 분석을 한 이유는, 실제로 들어오는 방송 애플리케이션에서 수신 단말이 DDB를 놓치는 확률을 조사해 보았을 때, 하나의 DDB를 수신 못할 확률이 0.01 이하로 작은 값을 가졌고, 한 주기의 캐리셀동안 DDB 수도 300개 이하임을 확인할 수 있었다. 따라서 한 주기 동안 DDB 4개 이상을 잃는 확률은 전체 확률에서 무시할 수 있을 정도로 작다고 가정할 수 있었고, 이러한 가정을 알고리즘 분석식 단순화를 위해서 사용할 수 있었다. 이번 절에서는 <표 1>에 정의한 변수들을 가지는 방송 환경 안에서 제안하는 알고리즘의 성능을 계산했을 경우의 성능 향상 효과를 계산하였다. 변수를 수식 (7)에 적용한 후  $p$ 의 값을  $10^{-5}$ 부터  $10^{-2}$ 까지 변화시키면서 성능 비교를 했을 때의 결과는 <표 2>와 같이 나타난다.

리턴 채널의 속도가 10kbps인 지속 환경을 고려했음에도 불구하고 제안한 알고리즘에서 DDB 수신 실패 확률이 커짐에 따라 탁월한 성능 향상 효과를 보임을 확인할 수 있다. 제안 알고리즘을 적용했을 경우, 전체 DDB를 받는 데 걸린 시간이 DDB 수신 실패 확률에 크게 영향을 받지 않고 거의 일정한 반면, 기존 알고리즘의 경우는 DDB 수신 실패 확률이 커짐에 따라 전체 DDB 수신 시간이 크게 늘어나게 됨을 확인할 수 있다.

<표 1> 변수 값

	$T$	$N$	$p$	$Tr$	$RTT$	$L$
값	30 초	200 개	가변 변수	10 kbps	0.01 초	4096 bits

<표 2> DDB 수신 실패 확률에 따른 성능 비교표

실패확률(p)	$10^{-5}$	$5 \times 10^{-5}$	$10^{-4}$	$5 \times 10^{-4}$	$10^{-3}$	$5 \times 10^{-3}$	$10^{-2}$
DDB수신시간(기존)	30.06초	30.30초	30.59초	32.86초	35.45초	48.59초	52.19초
DDB수신시간(제안)	30.01초	30.01초	30.02초	30.05초	30.09초	30.74초	30.81초
성능향상효과	0.2%	0.9%	1.9%	9.34%	17.80%	59.74%	69.37%

## 5. 실험 및 분석

이번 절은 본 논문에서는 제안 알고리즘을 지상파 ACAP 미들웨어의 데이터 다운로드 부분에 적용하여 실험한 내용과 분석에 대해서 기술한다.

### 5.1 실험 환경 및 시스템 구성

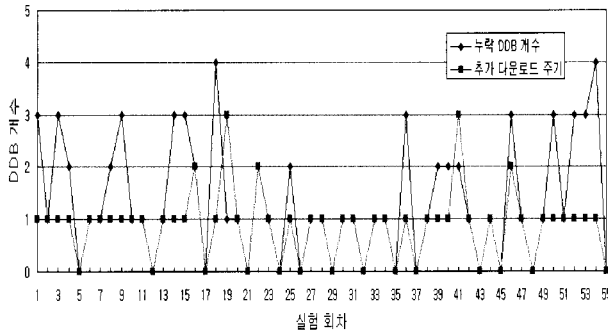
현재 단말에 구현된 ACAP 미들웨어는 자바 언어로 개발되었으며, 각 지상파 방송국에서 시험방송중인 ACAP 데이터 방송을 수신하여 애플리케이션을 실행할 수 있는 수준까지 구현되어 있다. 따라서, 본 논문에서는 개선 알고리즘의 성능 평가를 하기 위해서 ACAP 미들웨어의 데이터 다운로드 부분에 적용하여 소스를 수정하였다.

미들웨어가 구현된 단말에서는 스트림 생성기에서 송출하는 방송을 수신하여, 한 주기 동안 애플리케이션의 섹션 데이터를 저장하고, 누락된 DDB에 대해 데이터 서버에 요청하여 네트워크를 통해 받아온다. 리턴 채널을 통해 받는 DDB는 전송 계층 윗단에서는 ‘Error free’라고 가정한다. 지상파를 통해 받아놓은 데이터와 리턴 채널을 통해 받아놓은 데이터를 합쳐서 모듈화하고 애플리케이션을 구동시킬 수 있다. DDB 단위로 저장해 놓고 요청한 데이터만 송출해주는 장비를 구할 수 없기 때문에, 실험을 위해 데이터 서버에 소켓 프로그램을 구현하여 요청한 단말로 DDB를 보낼 수 있도록 하였다.

본 논문에서의 실험은 한 애플리케이션으로 고정하기 위해서 EBS에서 방송하는 실제 방송을 저장한 뒤 스트림 생성기에 넣어 송출하는 형태로 진행하였고, 단말에서 미들웨어를 구동하여 애플리케이션 다운로드 시간을 측정하였다. DDB를 모두 다운로드 한 뒤 데이터 프로세싱하는 시간은 지상파를 통해서 받거나 리턴 채널을 통해서 받더라도 같기 때문에 고려하지 않는다. 실제 데이터는 DDB에 포함되어 있으므로, 시간 측정은 DDB를 모두 수신한 시간으로 하였다. 시간은 자바의 API를 사용하여 밀리 세컨드(ms)단위로 측정하였으며, 한 주기 동안 다운받는 시간과 전체 다운로드 시간을 측정하였고, 총 55회 측정하였다.

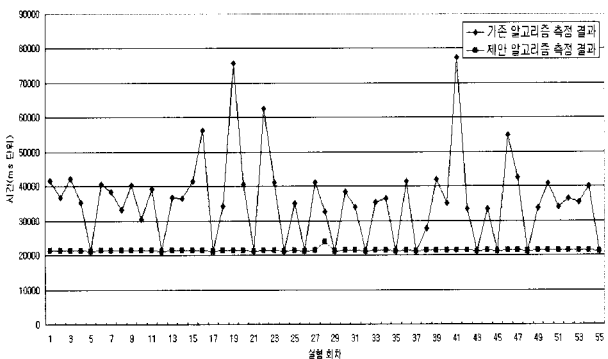
5.2 실험 결과

누락되는 DDB의 개수와 다운로드 주기를 측정된 결과는 (그림 6)과 같다.



(그림 6) 다운로드 주기와 누락 DDB 개수

기존의 알고리즘을 사용했을 경우 데이터 다운로드 시간과 제안한 알고리즘을 사용했을 경우 데이터 다운로드 시간을 실험하여, 그 결과를 실험 회차에 따른 그래프로 나타내면 (그림 7)과 같다.



(그림 7) 실험 결과 그래프

기존의 알고리즘을 사용했을 경우의 데이터 다운로드 평균 시간은 35.7초가 걸렸고, 제안한 알고리즘을 사용했을 경우의 평균 시간은 21.3초가 걸렸음을 확인할 수 있다. 기존의 알고리즘을 사용했을 경우가 제안한 알고리즘을 사용했을 경우보다 1.67배 더 시간이 걸렸다. 따라서 제안한 알고리즘이 67%의 시간 성능 향상 효과를 보임을 확인할 수 있다.

5.3 분석 및 검증

(그림 6)을 보면 각 회차별로 누락된 DDB의 개수와 한 주기 이상 다운로드 된 주기를 표시하고 있다. DDB가 하나라도 누락되면 한 주기 이상 다운로드 되지 않지만, 누락된 DDB 개수가 많을수록 애플리케이션 다운로드 주기를 여러 번 반복하는 상관관계는 존재하지 않음을 알 수 있다. 누락된 DDB 개수가 실험회차에 따른 반복성이나 일정한 증감의 형태를 갖지 않는 것으로 보아, DDB의 누락이 랜덤하게 발생함을 알 수 있다.

(그림 7)의 실험결과를 보면 기존 알고리즘의 데이터 다운

로드 시간 측정 결과가 고르지 않게 분포되어 있음을 확인할 수 있다. 이는 DSM-CC 표준에 따른 DDB의 전송이 주기적으로 반복되기 때문에, 전송 도중 에러가 발생하거나 수신 단말의 선택 필터링 처리가 지연될 경우 DDB가 손실되기 때문이다. 예를 들어 6, 12, 18차등은 단 한 개의 DDB 누락이 없이 한 주기 안에 250개의 모든 DDB를 수신하고 있음을 확인할 수 있는 반면에, 대부분의 실험 회차에서는 첫 주기에서 한 개 이상의 DDB를 놓쳐서 한 주기 이상을 DDB 수신을 위해서 반복하고 있음을 볼 수 있다. 특히 17, 20, 23, 42, 47 회차를 보면 다른 실험 회차보다 DDB 수신을 위해 걸린 시간이 훨씬 크게 표현되었음을 확인할 수 있다. 그 이유는 첫 번째 주기에서 놓친 DDB를 두 번째 주기에서도 수신하지 못하고 2주기 이상 반복되었기 때문이다.

또한 (그림 7)에서 개선한 알고리즘을 사용했을 경우는 전체 DDB 수신 시간이 거의 일정하게 나왔음을 확인할 수 있다. 이는 첫 번째 주기에서 잃은 DDB의 개수에 관계없이 첫 번째 주기가 끝났을 때 받지 못한 DDB를 모두 일정한 전송율을 가지는 네트워크를 통해서 다운로드 받았기 때문이다. 또한 전송율에 비해서 DDB의 길이가 매우 작으므로 첫 번째 주기에서 수신 못한 DDB 개수에 관계없이 네트워크를 통해 나머지 DDB를 받는 시간은 거의 걸리지 않았을 것이므로 제안한 알고리즘을 이용한 실험 측정 결과가 거의 DDB 수신한 주기와 같은 값으로 고르게 나왔음을 예상할 수 있다.

5.2절의 실험결과를 토대로 성능 향상 효과를 측정하기 위해 4.2.3절의 알고리즘 분석식 (7)번에 대입하는 값은 <표 3>과 같다.

<표 3> 실험 값

	$T$	$N$	$P$	$Tr$	$RTT$	$L$
값	21.3 초	250 개	0.005357	100 Mbps	0 초	4096 bits

$T$ 와  $N$ ,  $L$ 은 실제 들어오는 데이터를 측정해서 구한 값이다. 위의 실험환경에서 조사했을 때, EBS 애플리케이션 구성을 위해서 들어오는 DDB의 개수는 250개였고, 250개가 들어오는 한 주기의 시간은 21.3초임을 측정을 통해서 파악할 수 있었다. 또한 다운받은 데이터 분석을 통해 하나의 DDB는 4096 bits으로 이루어진 데이터 임을 알 수 있었다. 수신 실패확률  $P$ 는 실험 55회 측정 시 수신 실패한 DDB 수를 전체 들어온 DDB 개수로 나눔으로써 계산하였다. 네트워크의  $Tr$ 은 실제로 구현을 하기 위해 구성된 100Mbps 이더넷 LAN의 속도로 결정하여 사용하였고,  $RTT$ 는 로컬 네트워크 상에서 측정된 값이므로 거의 0에 가까운 값이므로 0으로 두고 계산하였다.

<표 3>의 값을 알고리즘 분석식 (7)번에 대입하여 성능 향상 효과를 구해보면 67.0%가 나온다. <표 1>에 정의한 변수 값에서  $P$ 의 값을 실제 실험을 통해 구한 0.005357로 변경하여 알고리즘 분석식 (7)번에 대입하여 성능 향상 효과를 구해보면 69.9%가 나온다. 이를 통해 실험 결과가 분석 시 예상했던 값과 거의 유사하게 측정되었음을 알 수 있다.



## 6. 결론 및 향후 과제

본 논문에서 리턴 채널을 활용한 애플리케이션 다운로드 향상 알고리즘을 제시하였고, 알고리즘을 ACAP 미들웨어에 구현 탑재하여 실험을 통해 기존 알고리즘에 비해 50% 이상 성능 향상 효과가 있음을 확인하였다.

본 논문에서 실험에 사용한 방송 스트림은 스트림 생성기를 통한 정류화된 스트림이었다. 정류화된 스트림도 DSM-CC 전송 표준을 따르므로 수신 단말에서 섹션 데이터 처리에 따른 지연이 발생할 경우 DDB를 놓칠 수 있다. 그러나 송수신 채널의 비트 에러로 인한 DDB의 손실은 발생하지 않으므로, 실제 환경보다 DDB 하나의 수신 실패 확률은 같거나 작다. 따라서 수신 단말이 실제 지상파를 통해서 방송 데이터를 수신하는 경우 실험한 케이스보다 더 큰 DDB 수신 실패 확률을 가질 것으로 예상할 수 있다. 이 경우 첫 번째 주기에서 수신하지 못하는 DDB의 수는 더욱 늘어날 것이고, 이로 인해서 기존의 알고리즘을 사용했을 경우 전체 DDB를 수신하는 시간은 더욱 늘어날 것이다. 그러나 제안한 알고리즘은 첫 번째 주기에서 수신하지 못하는 DDB의 수에 크게 영향을 받지 않으므로 DDB 수신 실패 확률이 커졌다 하더라도 성능 저하가 거의 없을 것이고, 기존 알고리즘에 비해 성능 향상 효과는 더욱 클 것으로 예상할 수 있다.

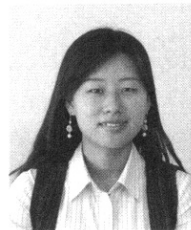
본 논문에서 제안한 알고리즘은 현재 한국에서 시험 방송되고 있는 데이터 방송 시스템 방식을 더욱 개선하기 위해서 제안이다. 현재 한국에서 시험 방송되고 있는 데이터 방송 시스템에서 애플리케이션은 하나의 캐리셀을 통해서만 들어온다. 그러나 국제 표준인 MHP에서는 하나의 방송 애플리케이션이 여러 개의 캐리셀을 통해서도 들어올 수 있음을 명시하고 있다. 따라서 이 경우에 본 논문에서 제안한 알고리즘을 적용하기 위해서는 4.1절에서 제시한 알고리즘을 그대로는 적용할 수 없고, 수정이 필요할 것이다. 여러 개의 캐리셀을 통해서 데이터가 들어오는 시스템에 대해서 이를 적용하려면 수신 단말에서는 각각의 캐리셀에 대해서 4.1절에서 제시한 알고리즘을 적용하는 방식으로 구현하면 될 것이다. 다만 하나의 캐리셀에 적용할 때 보다 여러 개의 캐리셀에 적용할 경우의 성능 향상 효과는 감소할 수 있지만, 여전히 기존의 알고리즘을 사용했을 경우 보다는 좋은 성능 향상 효과를 기대할 수 있을 것으로 예상된다.

따라서, 향후 연구과제로서 본 연구에서 구현된 알고리즘을 여러 개의 캐리셀에 적용하여 성능평가를 통해 제안된 알고리즘의 우수성을 검증하고, 새롭게 도출될 수 있는 문제점과 개선점을 찾아 극복하는 과정이 이루어져야 한다.

## 참고 문헌

- [1] Digital Video Broadcasting(DVB) ; Multimedia Home Platform (MHP) Specification 1.1.1, Jun., 2003.
- [2] The MHP Tutorial, <http://www.mhp-interactive.org/>
- [3] OpenCable Application Platform (OCAP) Specification : OC-SP-OCAP1.0-I14-050119, 19 Jan., 2005.
- [4] 류주현, 디지털 케이블 방송 표준 및 미들웨어 기술, 한국정보처리학회 학회지 11권 5호., 23-31, 2004년 9월.
- [5] ATSC Candidate Standard CS/101A : Advanced Common Application Platform (ACAP), Feb., 2004.
- [6] Digital Video Broadcasting ; Globally Executable MHP(GEM) Specification 1.0, May, 2004.
- [7] ISO/IEC 13818-1 : Information technology Generic coding of moving pictures and associated audio information Part 1 : Systems, 1996.
- [8] ATSC Standard A/53C with Amendment No. 1 and Corrigendum No.1 : ATSC Digital Television Standard, Rev. C, 21 May, 2004.
- [9] ATSC Standard A/65B : Program and System Information Protocol for Terrestrial Broadcast and Cable, Rev. B, 18 Mar., 2003.
- [10] ISO/IEC 13818-6 : Information technology Generic coding of moving pictures and associated audio information Part 6 : Extensions for DSM-CC.
- [11] 김세훈, 동기화된 데이터 방송 애플리케이션 제작을 위한 스트림 생성기 개발, 서강대학교 영상대학원 석사학위논문, 2003년.
- [12] ATSC Standard A/90 with Amendment 1 and Corrigendum 1 and 2 : Data Broadcast Standard, 26 July, 2000.
- [13] ATSC Standard A/95 : Transport Stream File System Standard, 25 Feb., 2003.
- [14] Digital Video Broadcasting ; DVB Specification for data broadcasting(ETSI EN 301 192 v1.4.1), Jun., 2004.
- [15] Digital Video Broadcasting ; Implementation guidelines for Data Broadcasting(ETSI TR 101 202 v1.1.1), Feb., 1999.

## 강 승 미



e-mail : chohon@skku.edu

2001년 숙명여자대학교 전산학과(학사)

2001년~현재 삼성전자 정보통신총괄 통신연구소 선임연구원

2004년~현재 성균관대학교 대학원 컴퓨터공학과 석사과정

관심분야 : 분산 컴퓨팅, 데이터 방송, 이동 컴퓨팅 등



### 선 승 상

e-mail : threes@ece.skku.ac.kr

1999년 성균관대학교 정보공학과(학사)

2001년 성균관대학교 대학원 전기전자및  
컴퓨터공학과(공학석사)

2003년~현재 성균관대학교 대학원 컴퓨  
터공학과 박사과정

관심분야: 유비쿼터스 컴퓨팅, 이동 에이전트, MANET, 시스템  
보안 등



### 엄 영 익

e-mail : yieom@ece.skku.ac.kr

1983년 서울대학교 계산통계학과(학사)

1985년 서울대학교 대학원 전산과학과(이  
학석사)

1991년 서울대학교 대학원 전산과학과(이  
학박사)

2000년~2001년 Dept. of Info. and Comp. Science at UCI 방문  
교수

2005년 한국정보처리학회 학회지 편집위원장

현 재 성균관대학교 정보통신공학부 교수

관심분야: 유비쿼터스 컴퓨팅, 분산 컴퓨팅, 이동 에이전트, 시  
스템 소프트웨어, 시스템 보안 등