

# MOST150기반 MEP를 이용한 프리미엄 고속버스용 멀티미디어 스트리밍 시스템 개발

이 재 규\*, 이 상엽\*, 조 현 중°

## Development of a Multimedia Streaming System using MEP Based on MOST150 for Premium Express Buses

Jae-kyu Lee\*, Sang-yub Lee\*, Hyun-joong Cho°

### 요 약

차량용 멀티미디어 시스템은 자동차 산업에 있어서 매우 중요한 요소 중 하나이다. 특히, 프리미엄 고속버스와 같은 고급상용차에서의 멀티미디어 시스템은 더욱 중요하다. 우리는 본 논문에서 MOST150을 기반으로 MEP(MOST Ethernet Packets)을 이용해 프리미엄 고속버스용 멀티미디어 시스템 구조를 제안했고, Prototype을 설계 및 구현했다. 제안한 시스템 구조를 구현하기 위해 멀티미디어 시스템이 운용될 수 있는 i.MX6기반의 Board를 설계했다. 소프트웨어는 Open Source Platform인 Android를 기반으로 프리미엄 고속버스용 멀티미디어 시스템을 설계했다. MOST는 고속의 멀티미디어 네트워크 기술이며, 차량의 멀티미디어시스템 설계에 사용된다. 기본적으로 링 토폴로지(Ring Topology)를 사용하고, 최대 64개의 MOST 장치를 연결할 수 있다. 또한 MOST 네트워크는 광모듈(POF)을 이용해 통신하기 때문에 EMII(Electro-Magnetic Interference) 및 EMC(Electro-Magnetic Compatibility) 요구조건을 충족한다.

**Key Words** : MOST150, MEP(MOST Ethernet Packets), Premium Express Bus, In-Vehicle Multimedia System, Infotainment System

### ABSTRACT

In-vehicle multimedia systems are one of the most important factors in the automotive industry. Especially, multimedia systems are more important in advanced commercial vehicles such as premium express buses. In this paper, we proposed a multimedia streaming system architecture using MEP(MOST Ethernet Packets) for premium express buses based on MOST150. We have designed and implemented the prototype of proposed multimedia streaming system. We have designed a board based on i.MX6 to operate a proposed multimedia streaming system. The software has designed a multimedia system for premium express buses based on Android which is an open source platform. MOST(Media Oriented Systems Transport) is a high-speed multimedia network technology for in-vehicle multimedia system. The MOST network is able to manage up to 64 devices and ring topology is used basically. In addition, the MOST Network meets EMI(Electro-Magnetic Interference)/EMC(Electro-Magnetic Compatibility) requirements because it uses plastic optical fibers(POF).

※ 본 연구는 산업통상자원부와 한국산업기술진흥원이 지원하는 경제협력연구사업으로 수행된 연구결과입니다.(R0004937)

◆ First Author : Korea Electronic Technology Institute, jae4850@keti.re.kr, 정희원

° Corresponding Author : Korea University Department of Computer Science Information, laycho@korea.ac.kr, 정희원

\* Korea Electronic Technology Institute, syublee@keti.re.kr, 정희원

논문번호 : KICS2016-12-409, Received December 29, 2016; Revised April 21, 2017; Accepted May 17, 2017

## I. 서 론

오늘날 고급차에서 전자제품이 차지하는 비중은 최종 생산단가의 23% 이상을 차지하고 있으며, 대부분의 자동차 혁신의 80% 이상이 전자제품에 기인한다고 추정한다. 차량내의 전자제품에는 제어장치에서 멀티미디어기기를 포함해 다양하게 사용되어지고 있으며, 고급자동차일수록 더 많은 전자제품이 사용되는 추세이다.<sup>[1]</sup> 최근 국내에서 상용화를 시작한 프리미엄 고속버스는 개인이 사용할 수 있는 다양한 시스템과 장치를 갖추고 있고, 특히 각 좌석마다 배치되어 있는 승객을 위한 멀티미디어 시스템은 프리미엄 고속버스만이 가지는 장점이다. 현재 출시되어 있는 대부분의 프리미엄 고속버스의 멀티미디어 시스템은 네트워크를 이용한 멀티미디어 시스템 구성이 아니라 좌석별로 별도의 멀티미디어 시스템이 설치되어 있는 구조이다. 이러한 구조의 시스템은 멀티미디어 기기별로 별도의 미디어 콘텐츠 저장 공간이 필요하기 때문에 비효율적이다. 또한 설령 멀티미디어 시스템이 Ethernet기반으로 구성되어 있다하더라도, 멀티미디어 전송에 특화되어 있고 차량의 네트워크를 구성하는데 있어서 전자파간섭이 없는 MOST 프로토콜 기반 멀티미디어 시스템이 Ethernet기반의 멀티미디어 시스템과 비교해서 안정적이며 효율적이라고 할 수 있다.

우리는 본 논문에서 MOST150 네트워크를 기반으로 MEP를 이용해 프리미엄 고속버스에서 사용할 수 있는 멀티미디어 스트리밍 시스템 구조를 제안했고, MOST 네트워크를 안드로이드 플랫폼에 수용하여 프리미엄 고속버스용 멀티미디어 스트리밍 시스템 소프트웨어를 개발했다. MOST 네트워크기반의 멀티미디어 시스템 설계는 통신에 POF(Plastic Optical Fiber) 물리층을 사용하기 때문에 기존 케이블을 이용한 멀티미디어 시스템 구성과 비교해서 매우 가볍고, 전자파간섭(EMI) 또는 전자파 적합성(EMC)으로부터 자유롭다.

## II. 연구배경 및 관련연구

### 2.1 MOST

MOST(Media Oriented Systems Transport)는 고속의 멀티미디어 네트워크 기술이자 표준이며, 차량용 멀티미디어 통신에 최적화되어 있다. 오디오 및 비디오를 포함하여 차량 내에 적용될 수 있는 다양한 멀티미디어 시스템의 네트워크로 사용된다.

MOST 네트워크는 2001년 등장한 이래 자동차 인

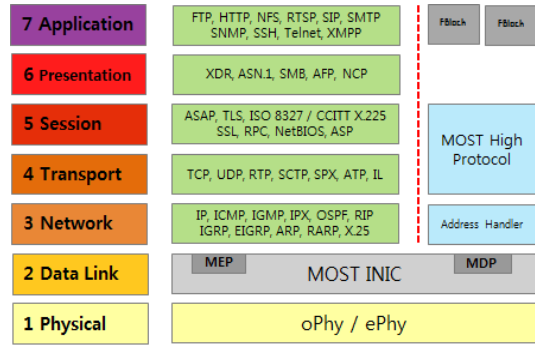


그림 1. MOST 레이어 모델  
Fig. 1. MOST Layer Model

포테인먼트 시스템에서 널리 사용되고 있고, 초기 MOST25인 25Mbps를 시작으로MOST50을 거쳐 현재는 150Mbps대역폭의 MOST150까지 지원하고 있다. 현재 고도화되는 차량용 멀티미디어 시스템에서 기존의 통신방식으로는 대역폭을 충족하지 못하므로 승용차 및 상용차량 내의 멀티미디어 통신에서 MOST 네트워크의 사용은 더욱 더 증가될 전망이다. MOST 사양(Specification)은 그림 1의 ISO/OSI 7 계층(Layer)과 마찬가지로 물리계층(Physical Layer) 및 데이터 링크 계층(Data Link Layer)을 정의한다. 그림 1의 MOST Layer Model에서 Network Service는 Data Link Layer와 Function Block 사이의 미들웨어(Middleware)이며, OSI 7계층의 3 Layer (Network)에서 7 Layer(Application)까지 포함한다. MOST 네트워크는 기본적으로 링 토폴로지(Ring Topology)를 사용하며, Star Topology 및 Double Ring Topology 구성도 가능하다. 1개의 타이밍 마스터(Timing Master)와 63개의 Slave Device로 최대 64개의 장치로 네트워크를 구성할 수 있고, Plug and Play기능을 지원하여 멀티미디어 장치를 네트워크에 쉽게 연결 및 제거할 수 있다.<sup>[2]</sup>

MOST150은 MOST네트워크 중 가장 최근에 나온 프로토콜로 150Mbps 대역폭을 제공하며, 기존 차량용 통신 프로토콜과 완벽히 호환된다. MOST150은 광범위한 비디오 및 오디오 전송을 지원할 수 있는 등시성(Isochronous) 채널과 IP기반 패킷 데이터 전송을



그림 2. MOST 프레임 및 데이터 타입  
Fig. 2. MOST Frame & Data Types

위한 Ethernet채널을 가지고 있는 것이 특징이다. MOST150의 프레임 구조와 데이터 타입은 그림 2와 같다. MOST프레임은 제어(Control)영역과 동기영역(Synchronous), 비동기(Asynchronous)영역으로 나눌 수 있다. 주로 동기영역을 통해서는 QoS(Quality of Service)가 보장되어야하는 오디오 및 비디오 데이터 전송이 이루어지고, 비동기영역과 제어영역은 MOST 메시지 전송이나 Ethernet 네트워크 기반의 통신에 사용된다. 그림 2의 화살표가 나타내는 것과 같이 동기영역(Synchronous)과 비동기(Asynchronous)영역은 고정되어 있는 것이 아니라, 시스템을 구성할 때 시스템 구조에 맞게 할당해서 사용할 수 있다.

## 2.2 관련연구

본 논문에서는 MOST150 MEP를 이용하여 프리미엄 고속버스용 멀티미디어 스트리밍 시스템을 개발했다. MOST관련 연구로는 [3]에서는 안드로이드 인포테인먼트 시스템을 기반으로 MOST 네트워크를 이용한 차량용 오디오 시스템을 구현했고, [4]는 MOST INIC의 Remote Access 기능을 활용한 오디오 스트리밍 서비스에 대해 기술하고 있다. [5]은 MOST High Protocol(MHP) 통신과 MOST Ethernet Packet(MEP)을 이용한 MOST 펌웨어 업데이트 모델을 제안했다. 이처럼 MOST관련 연구는 MOST 네트워크 기반의 오디오 시스템에 관한 연구가 대다수이며, MOST MEP관련 연구는 하드웨어나 소프트웨어 설계 및 구현에 관한 연구보다는 대부분의 연구가 모델이나 구조를 제안한다. MOST150관련 연구로는 MOST150 네트워크를 기반으로 차량주변을 감시하는 Around View 시스템 설계 및 구현에 대해 기술한 연구가 있다. [6] MOST150관련 연구는 현저히 부족한 상태이며, 실제로 MOST150 네트워크를 이용한 차량용 시스템 설계 및 개발에 관한 연구는 더욱 부족한 실정이다. 특히 MOST150 MEP기반의 연구와 시스템 설계 및 구현관련 연구는 거의 없는 상황이고, MOST네트워크를 버스의 멀티미디어 시스템에 적용하기 위해 연구한 사례도 없는 상황이다. 이에 본 논문에서는 프리미엄 고속버스에서 사용될 수 있는 MOST150 MEP 기반의 멀티미디어 시스템을 제안하고 설계했다. MOST네트워크를 기반으로 버스의 멀티미디어 시스템을 구성할 경우 장점으로는 영상과 음원을 쉽게 분리할 수 있어서 버스구조에 적합한 미디어 시스템 설계가 가능하며, MOST150에서 지원하는 MEP는 범용적인 Ethernet 네트워크 유사하여 시스템 설계와 구성을 편리하게 할 수 있도록 한다. 또한 차량의 멀티미

디어 전송에 최적화되어있는 MOST 네트워크는 광케이블로 통신을 수행하기 때문에 다른 네트워크 방식과 비교해서 EMI 및 EMC문제가 없어 안정적인 시스템을 구현할 수 있다.

## III. MOST150기반 멀티미디어 시스템 설계

### 3.1 프리미엄 버스용 멀티미디어 시스템 구조

#### 3.1.1 멀티미디어 시스템 전체구조

본 논문에서 제안하는 MOST150 MEP기반의 프리미엄 고속버스용 멀티미디어 시스템 구조는 그림3과 같다. 그림 3과 같이 Head Unit(서버) 저장소에 미디어 파일들이 있고, 각 좌석의 클라이언트에는 안드로이드 기반의 Media Player 애플리케이션이 운용된다. 클라이언트에는 서버에 존재하는 미디어 파일의 썸네일(Thumbnail) 정보만 존재하며, 사용자가 원하는 영상의 썸네일(Thumbnail)을 선택하면 해당하는 미디어 정보가 MOST150 MEP를 통해 서버에 요청하고 서버에서 미디어 파일을 클라이언트로 스트리밍(Streaming)하는 구조이다. 전체 네트워크는 링(Ring) 토폴로지로 구성된다. 제안하는 구조의 멀티미디어 시스템은 VoD(Video On Demand)와 같은 구조로 클라이언트에 영상을 저장하지 않고 재생한다. 서버에는 미디어 스트리밍을 위해 MPEG2 TS(Transport Stream) 파일(\*.ts)이 저장되어 있다.

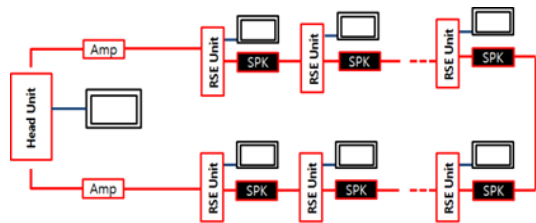


그림 3. 프리미엄 고속버스 내의 멀티미디어 시스템 전체구조  
Fig. 3. Overall structure of multimedia system in premium express bus

#### 3.1.2 하드웨어 사양 및 구성

본 논문에서 멀티미디어 시스템 설계에 사용된 하드웨어 사양은 표 1과 같다. CPU는 Freescale사의 i.MX6 CPU를 사용했으며, DDR3 2GB Memory, Quick Booting을 위한 EMMC Flash Memory, 9.7inch WSVGA Touch Panel 그리고 Debug를 위한 UART 등으로 이루어져 있다.

클라이언트 플랫폼으로 Freescale사의 i.MX6 CPU

표 1. 하드웨어 사양  
Table 1. Hardware Specification

CPU	Freescale i.MX6Q Quadcore Cortex-A9 1.2GHz
Memory	DDR3 64bit 2GB 533MHz
Flash	EMMC 4GB, SPI-NOR Flash 4MB
Power Mngt	Freescale PMIC MMPF0100(12V)
Display	9.7inch WSVGA Touch Panel
Interface	USB, HDMI, UART/SPI, Audio
Network	MOST, Ethernet, CAN, etc

기반의 보드를 사용한 가장 큰 이유로는 i.MX6 CPU는 AEC(Automotive Electronics Council)-Q100인증을 받았으며, 차량용 인포테인먼트(Automotive Infotainment) 시스템 설계를 목적으로 생산되었기 때문이다. AEC-Q100은 자동차에서 사용되는 전자부품에 대한 신뢰성 평가절차를 규정한 문서로 전 세계에서 통용되는 실질적인 업계표준이다. 그러므로 i.MX6 보드를 기반으로 차량용 인포테인먼트 시스템을 설계하는 것은 범용 보드기반의 인포테인먼트 시스템 설계와 비교해서 보다 차량에 적합한 플랫폼에서 시스템을 설계했다고 할 수 있다. 또한, Freescale은 i.MX 보드에 맞는 안드로이드와 Window 그리고 다양한 버전의 리눅스 운영체제를 지원할 뿐만 아니라 하드웨어 및 각 장치를 위한 BSP(Board Support Package)를 지원하기 때문에 개발자의 접근성이 우수하다. 즉, i.MX보드를 이용해 차량용 멀티미디어 시스템을 설계할 때 부트로더(Bootloader)나 디바이스 드라이버나 또는 운영체제에 관한 설계를 최소화한 상태에서 시스템을 설계할 수 있기 때문에 멀티미디어 애플리케이션을 개발하는데 있어서 효과적이다. 마지막으로, i.MX보드는 실제로 다양한 자동차 제조사에서 인포테인먼트 시스템 설계에 사용되어지고 있다.

하드웨어 Block Diagram은 그림 4와 같다. i.MX6 CPU를 기반으로 eMMC 메모리와 Audio 입출력 및 Touch Panel 등이 있는 것을 볼 수 있다. 특이사항으로는 MOST 네트워크 기반의 통신을 위한 광기반의 물리층 모듈(Optical Receiver/Transmitter)이 있으며, 신호를 처리하기 위한 MOST INIC(Intelligence Network Interface Card)가 있는 것을 볼 수 있다. 실제로 설계된 개발보드는 그림 5와 같고 MOST INIC 연결을 위한 인터페이스가 탑재된 것을 볼 수 있다. INIC보드는 Microchip사의 OS81118 INIC 컨트롤러를 기반으로 설계되었다.

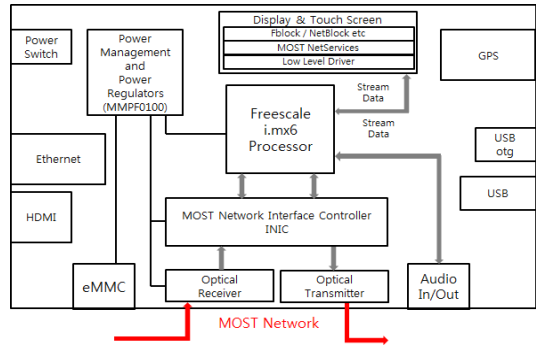


그림 4. 하드웨어 블록 다이어그램  
Fig. 4. Hardware Block Diagram

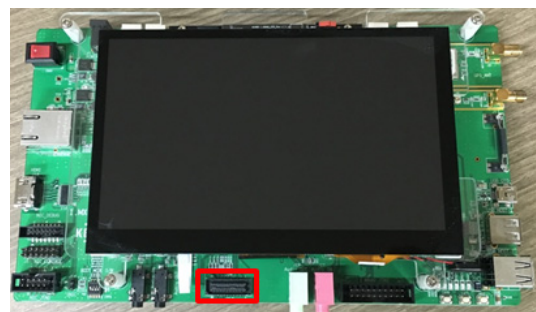


그림 5. MOST INIC 인터페이스 기반의 개발보드  
Fig. 5. Development board based on MOST INIC interface

### 3.2 멀티미디어 시스템 Software구조

이번 절에서는 MOST150 MEP기반의 멀티미디어 시스템 소프트웨어 설계방법 및 절차에 대해 기술한다. 소프트웨어 개발환경과 구조에 대해 기술하고 명령어의 흐름을 포함한 통신구조에 관해 기술한다.

#### 3.2.1 소프트웨어 개발환경

MOST150 및 MOST MEP기반 개발환경은 표 2와 같다. 본 멀티미디어 시스템은 MOST INIC 인터페이스 제어가 필요하기 때문에 INIC인터페이스를 위한 디바이스 드라이버(Device Driver)를 설계했다. 디바이스 드라이버 개발 및 컴파일 환경은 우분투14.04.1, 커널(Kernel) 3.13.0.-32-generic버전과 arm-2014.05 및 gcc-4.6.2 버전의 크로스 컴파일러를 사용했으며, Oracle JAVA-6 버전의 환경에서 작업을 수행했다. 실제로 멀티미디어가 재생되는 클라이언트(Target)는 Android5.0(Lollipop)을 기반으로 하며, 커널은 3.10.32 버전이고 MOST INIC제어를 위한 드라이버의 기본 구조는 MOST Linux Driver3.0.2버전을 사용했다.



표 2. 소프트웨어 개발환경

Table 2. Software Development Environment

개발 운영체제	Ubuntu14.04.1 LTS/Name:Trusty
Kernel Ver	3.13.0.-32-generic
Cross Compiler	arm-2014.05(gcc-4.6.2 etc)
Target 운영체제	Android 5.0(Lollipop)
개발 Tool	Android Studio 2.2.3
Java version	java-6-oracle
Target Kernel	3.10.32
MOST Driver	MOSTLinuxDriver3.0.2

3.2.2 소프트웨어 구조 및 흐름

MOST150 MEP기반의 멀티미디어 시스템 클라이언트의 소프트웨어 구조는 그림 6과 같다. 클라이언트는 안드로이드를 기반으로 설계되었기 때문에 전체 구성은 안드로이드를 기반으로 한다. 클라이언트의 S/W구조는 커널영역에 INIC 하드웨어를 제어하기 위한 MOST 디바이스 드라이버가 존재하며, 애플리케이션과 MOST 드라이버를 연결하기 위한 MOST라이브러리가 존재한다. 또한 미디어를 재생하는 안드로이드 애플리케이션은 MOST Function Block(FBlock)을 기반으로 설계되었고, 서버는 리눅스를 기반으로 설계되었다. 서버의 MOST 디바이스 드라이버 및 제어구조는 클라이언트 시스템과 동일하다. 이는 안드로이드가 리눅스 커널에 기초하기 때문이다. 단, MOST INIC Explorer를 이용해 MOST 통신이 이루어지도록 서버와 클라이언트에 연결된 INIC관련 설정을 해주어야 한다.

MOST150기반 멀티미디어 시스템 서버와 클라이언트의 데이터 및 명령어 전달 구조를 도식화하면 그

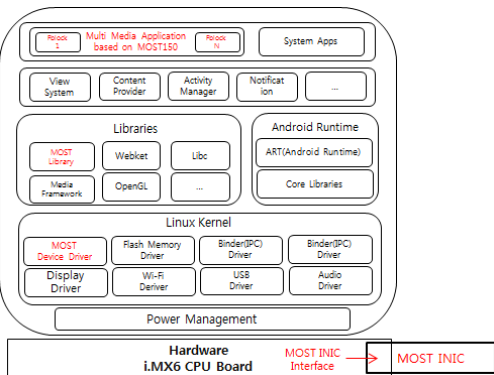


그림 6. 안드로이드기반 클라이언트 플랫폼 소프트웨어 구조 Fig. 6. Software structure of client platform based on Android

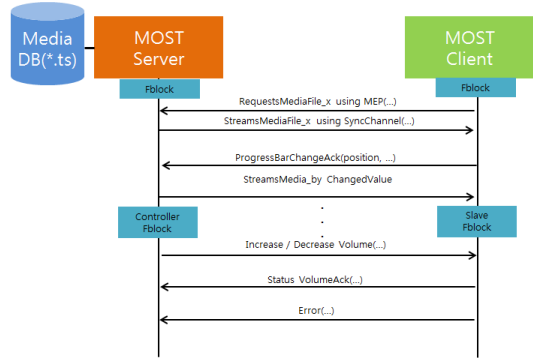


그림 7. 안드로이드기반 클라이언트 플랫폼 소프트웨어 구조 Fig. 7. Software structure of client platform based on Android

그림 7과 같다. 간단히 프리미엄 고속버스를 기준으로 좌측에 있는 클라이언트에서 자신이 원하는 미디어 파일을 요청하면 서버에서 MOST 등시채널로 파일을 나눠서 전송하는 구조이다. 즉, 클라이언트에서는 MOST MEP를 기반으로 선택된 미디어 파일의 정보를 서버에 전달하고, 서버에서는 전송된 정보에 맞는 미디어 파일을 MOST 등시채널(Isochronous)을 기반으로 클라이언트로 전송한다. 클라이언트는 등시채널을 통해 전송되어지는 미디어 정보를 FBlock의 TsReceiver모듈과 안드로이드의 View 및 SurfaceView를 이용해 화면에 출력한다.

MOST 네트워크는 차량용 멀티미디어 시스템을 구축하기 위해 자동차 제조사들이 만든 기술이기 때문에 대부분의 차량용 멀티미디어 시스템의 동작이 Function Block의 메시드로 정의 되어있다. 그러므로 컨트롤러 FBlock과 Slave FBlock을 이용해 음량제어, 미디어 시작 또는 정지 등의 제어가 가능하다. 명령어가 송수신되는 구조는 그림 7 하단부에서와 같이 컨트롤러의 FBlock에서 Slave의 FBlock으로 볼륨을 높이거나 줄이는 명령을 전달하면 Slave는 명령에 따른 동작을 이행하고 동작 수행결과를 다시 컨트롤러로 전송한다. 만약, 컨트롤러에서 전송하는 명령을 정확

표 3. MOSTFunction Block(FBlock) 명령어 구조 Table 3. Instruction structure of MOST Function Block

FBlockID.InstID.FuncID.OpType	User Function
Media.01.Playmode.Set	Set_Playmode(*RX Msg)
Media.01.TimePosImpl.SetGet	Set_TimePos(*RXMsg, *TXMsg)
Media.01.ServerVersionImpl.Set Get	Set_ServerVersion(*RXMsg, *TXMsg)

히 이행하지 못한 경우 에러(Error)메시지를 서버로 전송하는 구조로 동작한다.

표 3은 MOST Fblock(Function Block)을 구성하는 명령어 구조를 나타낸다. 표 3의 예시는 미디어 제어에 있어서 재생모드 설정과 재생위치 변경 그리고 서버정보에 대한 명령어 구조 및 함수 선언과 관련된 예시를 나타낸 것이다. 표 4는 선언된 함수의 내부구조 정의에 대한 예시를 나타낸 것이다.

본 논문의 실제 구현 부분에서 음원을 제어하는 MOST 노드를 독립적으로 두었는데, 각 노드를 제어하기 위해 표 3과 같이 정의한 MOST Function Block을 이용했다.

표 4. MOSTFunction Block(Fblock) 명령어 예시  
Table 4. Example of MOST Function Block Instruction

```

=====
MOST_Set_Playmode(*RxMsg){
    Playmode = RxMsg->Data[0];
    Set_Playmode(Playmode);
    return(NO_REPORT);}
=====
    
```

#### IV. 멀티미디어 시스템 구현 및 테스트

##### 4.1 차량용 멀티미디어 시스템 구현

제안한 구조와 설계결과를 바탕으로 MOST150 MEP기반의 멀티미디어 시스템을 테스트하기 위한 구조는 그림 8과 같다.

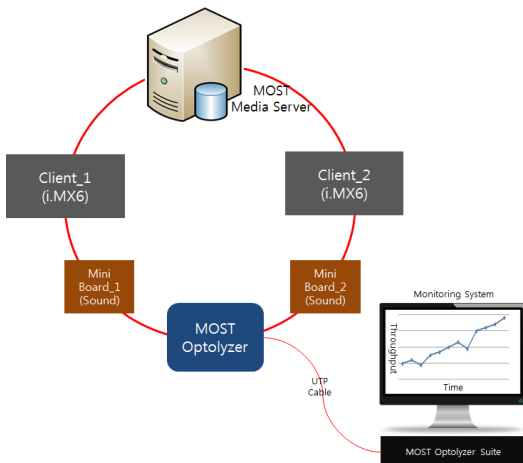


그림 8. MOST150 MEP기반 멀티미디어 시스템 테스트구조  
Fig. 8. Test structure of multimedia system based on MOST150 MEP

테스트실험은 미디어 재생을 위한 i.MX6기반의 클라이언트 보드 2개와 리눅스 기반의 미디어 서버 1대, 음량제어를 위한 MOST INIC기반의 보드(MOST Mini Board) 2개를 포함해 총 5개의 MOST 노드로 테스트 환경을 구현했다. 또한 그림 8과 같이 MOST 패킷과 연결 상태를 확인할 수 있는 Microchip사의 Optolyzer가 연결된 것을 볼 수 있다.

그림 9는 Optolyzer Suite에서 실제로 MOST 네트워크 구성을 확인한 것이다. 그림과 같이 0번 노드에서 4번 노드까지 5개의 노드가 링 토폴로지 형태로 연결되어 있는 것을 확인할 수 있다. 또한 MOST INIC 관련 디바이스 드라이버는 리눅스 커널 상에서 데몬(Demon) 형태로 실행되고 있으며, INIC 장치가 연결되면 MOST 장치(mdev0)로 인식하고 채널을 할당한다. 디바이스 드라이버에 의해 MEP채널이 할당되면 그림 10과 같이 meth0가 생성되며, 이 장치는 일반 네트워크 인터페이스와 같은 구조인 것을 확인할 수 있다. MOST150의 MEP통신은 그림 10의 meth0 IP 및 MAC 주소를 기반으로 이루어진다. meth0 네트워크의 MAC 주소는 그림 10의 초기 값에서 서버가 실행될 때 연결된 클라이언트에 설정 값을 일괄적으로 할당 및 설정한다.

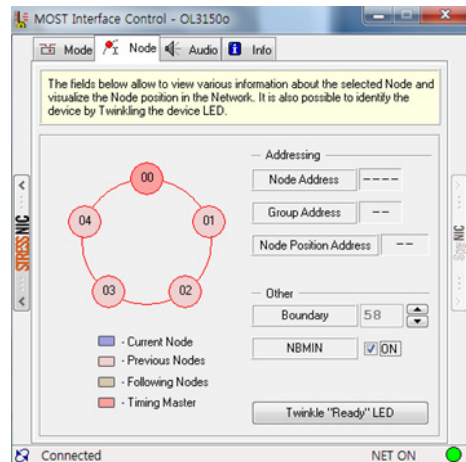


그림 9. MOST 링 네트워크와 노드 구성  
Fig. 9. MOST Ring network and Node configuration

```

root@android:/ # netcfg
netcfg
ip6tn10 DOWN 0.0.0.0/0 0x00000000 00:00:00:00:00:00
meth0 UP 10.0.0.1/8 0x0001043 ff:ff:ff:ff:ff:ff
eth1 UP 0.0.0.0/0 0x0001043 00:00:0f:11:70:00
eth0 DOWN 0.0.0.0/0 0x0001002 00:0e:c6:81:79:01
sit0 DOWN 0.0.0.0/0 0x0000000 00:00:00:00:00:00
lo UP 127.0.0.1/8 0x0000049 00:00:00:00:00:00
root@android:/ #
    
```

그림 10. meth0를 포함한 네트워크 구성  
Fig. 10. Network configuration with meth0

4.2 차량용 멀티미디어 시스템 구현

구현 결과를 테스트하기 전에 그림 11와 같이 핑(Ping)테스트를 수행했다. 핑(Ping)테스트는 네트워크 상에서 각 호스트간의 통신이 수행되고 있는지 확인하는 명령어이다. MOST로 네트워크가 구성되어 있지만 일반적인 Ethernet 프로토콜 통신과 동일한 것을 볼 수 있다.

그림 12는 실제 멀티미디어 시스템을 구현한 것이다. 그림 12 중앙부분에 리눅스 기반의 서버가 있고, 양쪽에 클라이언트 보드에서 멀티미디어가 실행되는 것을 볼 수 있다. 멀티미디어 데이터가 클라이언트에 저장되어 실행되는 것이 아니라 서버로부터 전송받은 데이터가 실시간으로 재생되고 있는 것이다.

그림 13은 실행되고 있는 클라이언트 플랫폼 화면

```

root@android:/ # ping 10.0.0.2
ping 10.0.0.2
PING 10.0.0.2 <10.0.0.2> 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.845 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.841 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.919 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.919 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.794 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.783 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.933 ms
    
```

그림 11. MOST150 네트워크(MEP)기반 Ping 테스트  
Fig. 11. Ping Test based on MOST150 Network(MEP)

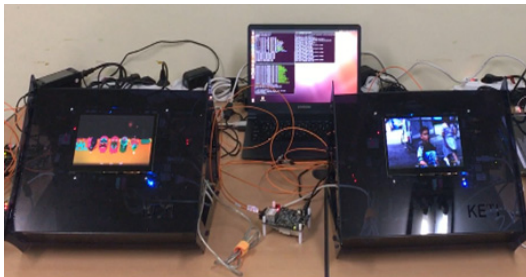


그림 12. MOST150 MEP기반 멀티미디어 시스템 구현  
Fig. 12. Implementation of multimedia system based on MOST150 MEP

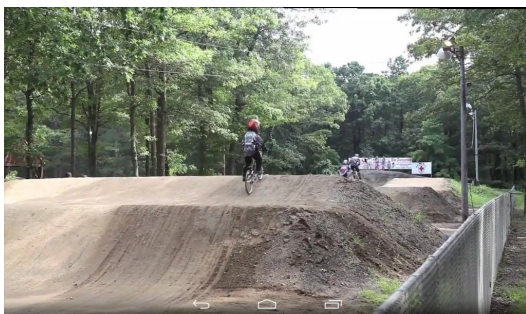


그림 13. 클라이언트 플랫폼의 실행화면  
Fig. 13. Screen of executed client platform

을 자세히 나타낸 것이다. MOST 등시채널을 통해 서버로부터 전송받은 스트리밍 데이터를 실시간으로 화면에 출력하는 것을 볼 수 있다.

본 논문에서는 1대의 서버와 클라이언트 플랫폼 2대를 기반으로 제안한 멀티미디어 시스템 프로토타입을 구현했으며, 구현된 시스템을 기반으로 제안한 멀티미디어 시스템과 설계한 하드웨어 및 소프트웨어 시스템이 정상적으로 구동되는지 테스트를 수행했다. 현재 2개의 클라이언트 플랫폼 기반의 프로토타입 및 테스트 환경에서 실제 프리미엄 고속버스 좌석에 맞게 21개의 클라이언트를 운용하기 위한 네트워크 구성을 정리하면 표 5와 같이 나타낼 수 있다. 표 5의 등시채널(Isochronous)을 통해서서는 영상이 전송되고, 동기영역(Synchronous)채널을 통해서서는 오디오(Audio)가 전송되며, MOST150의 MEP(Asynchronous/Ethernet) 채널을 통해서서는 데이터 통신이 이루어진다. 또한, 표 5의 MOST VoD 서버의 대역폭은 96.768Mbit/s가 할당되어 있다. 이를 21개 좌석으로 분산해보면 각 클라이언트마다 할당되는 대역폭은 4.608Mbit/s가 되고, 이는 현재 2개의 클라이언트로 구성된 시스템에서 클라이언트를 21개로 확장할 경우에도 H.264코덱으로 압축된 HD급 비디오를 전송하는데 무리가 없을 것으로 보인다. 즉, 표 5와 같이 프리미엄 고속버스 내에서 21개 클라이언트 좌석의 영상정보 스트리밍에 필요한 대역폭을 합산하면 142.848Mbit/s이다. MOST150 규격은 150Mbit/s(147.456Mbit/s) 대역폭을 지원하므로 현 시스템을 21개의 클라이언트 (142.848Mbit/s)로 구성해도 VoD스트리밍에는 문제가 없을 것으로 예상된다.

표 5. MOST150기반의 프리미엄 고속버스용 대역폭 할당  
Table 5. Bandwidth Allocation Matrix based on MOST150 for Premium Express Buses

Device Name	Device ID	Device Instance	Channel Usage	Bandwidth (MBit/s)
MOST 150 Server	0x311	0	VoD (Isoc)	96.768
MOST 150 Client	0x324	1	Decode PCM Audio (Sync)	1.536
.	.	.	Decode PCM Audio	1.536
MOST 150 Client	0x324	21	Decode PCM Audio	1.536
Async (Ethernet)				13.824
Total				142.848 (150)

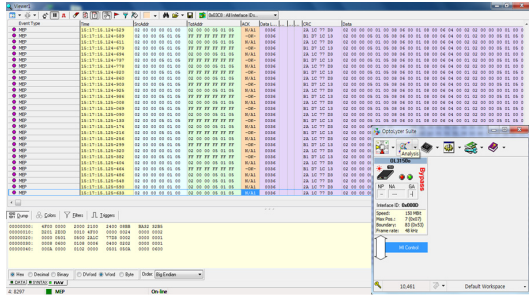


그림 14. MOST150 데이터 프레임 분석  
Fig. 14. MOST150 Data Frame Analysis

마지막으로, 구현된 시스템에 Optolyzer를 연결하여 MOST150 데이터의 패킷을 확인한 결과 그림 14에서와 같이 현재 송수신 되는 데이터가 설계한 멀티미디어 애플리케이션의 Function Block (FBlock)대로 수행되고 있는 것을 확인할 수 있었다.

### V. 결 론

본 논문에서는 MOST150 네트워크를 기반으로 MEP(MOST Ethernet Packets)를 이용해 프리미엄 고속버스에서 사용할 수 있는 멀티미디어 스트리밍 시스템 구조를 제안했고, i.MX6기반의 하드웨어와 안드로이드를 기반으로 프리미엄 고속버스용 멀티미디어 시스템 프로토타입(Prototype)을 개발했으며, 개발결과를 바탕으로 멀티미디어 시스템을 구현 및 테스트하여 구현된 멀티미디어 스트리밍 시스템이 정상적으로 동작하는 것을 확인했다. 차량용 멀티미디어 네트워크로 MOST를 사용하는 가장 큰 장점은 네트워크 통신에 POF(Plastic Optica Fiber) 물리층을 사용하기 때문에 전자파간섭(EMI) 또는 전자과적합성(EMC)의 영향이 없어 안전한 차량용 네트워크를 구성할 수 있다. 또한 본 논문에서는 MOST150에서 제공하는 MEP를 이용해 멀티미디어 시스템을 설계했다. 이는 제어채널을 이용한 명령어 전달이나 동기영역(Synchronous) 및 등시영역(Isochronous)을 이용한 미디어 전송을 MEP의 IP(Internet Protocol) 또는 MAC(Media Access Control Address)주소를 기반으로 사용자 정의하여 통신할 수 있기 때문에 MOST기반 응용프로그램을 설계하는데 매우 효율적이며 다양하게 활용될 수 있다.

향후 연구방향으로는 현재 5개의 노드(Node)로 구성되어 있는 MOST 링(Ring) 네트워크 노드를 최대 21개까지 확장해서 시스템을 설계할 것이다. 확장된 MOST 링 네트워크 구조에서 스트리밍 데이터 영역

(Streaming Data Area)과 MEP 할당에 따른 QoS(Quality of Service)에 대한 연구를 수행할 것이다. 또한 기존의 미디어파일 공유를 위해 DLNA(Digital Living Network Alliance)나 UPnP 또는 Miracast 공유기법을 기반으로 MOST네트워크 기반의 범용화된 미디어 공유기술 및 공유기술 적용 방법에 관한 연구를 수행할 것이다. 마지막으로, MOST RBD(Ring Break Diagnosis) 오류검출 기능을 이용해 고장진단 및 복구 소프트웨어 설계와 알고리즘에 대한 연구를 수행할 것이다.

### References

- [1] G. Leen and D. Heffernan, "Expanding automotive electronic system," *Computer*, vol. 35, no. 1, pp. 88-93, 2002.
- [2] A. Grzember, *MOST®: The Automotive Multimedia Network from MOST25 to MOST150*, Poing, Franzis, 2011.
- [3] J. K. Lee, S. Y. Lee, and H. S. Choi, "Implementation of MOST audio system for in-vehicle infotainment system based on Android platform," in *Proc. KICS, Winter Conf.*, pp. 449-450, 2013.
- [4] D. K. Park, J. K. Lee, and H. S. Choi, "Design for low-cost MOST streaming system with INIC remote access function," in *Proc. KICS, Winter Conf.*, pp. 451-452, 2013.
- [5] S. H. Yun, S. Y. Lee, and J. J. Ko, "MOST update model based on MOST high protocol and MOST ethernet Packet," in *Proc. KICS, Int. Conf. Commun.*, pp. 663-664, 2013.
- [6] S. Jang, "Design and implementation of an around view monitoring system on MOST150 network," *J. KIICE*, vol. 18, no. 11, pp. 2765-2770, 2014.



이 재 규 (Jae-kyu Lee)



2010년 : 전북대학교 공학사  
2012년 : 전북대학교 공학석사  
2015년~현재 : 고려대학교 컴퓨터정보학 박사과정  
2012년~현재 : 전자부품연구원 임베디드, SW 센터 연구원  
<관심분야> In-vehicle 네트워크 시스템, 차량용 임베디드 시스템

조 현 중 (Hyun-joong Cho)



1996년 : 경북대학교 공학사  
1998년 : 포항공과대학교 공학석사  
2006년 : Computer Engineering Virginia Polytechnic Institute and State University (Ph.D)  
2009년~현재 : 고려대학교 컴퓨터정보학과 교수  
<관심분야> 실시간 임베디드시스템, HCI

이 상 엽 (Sang-yub Lee)



2003년 : 연세대학교 공학사  
2005년 : 연세대학교 공학석사  
2015년~현재 : 고려대학교 컴퓨터정보학 박사과정  
2005년~2009년 : 삼성전기 중앙연구소 선임연구원  
2009년~현재 : 전자부품연구원

임베디드, SW 센터 선임연구원  
<관심분야> In-vehicle 네트워크 시스템, 차량용 임베디드 시스템