

<http://dx.doi.org/10.7236/IIBC.2015.15.5.105>

IIBC 2015-5-12

멀티미디어 파일 동시 스트리밍 방법

Simultaneously Streaming Method of Multimedia Files

김진영*, 이은상*, 임소희*, 안병구**

Jinyoung Kim*, Eunsang Lee*, Sohee Im*, Beongku An**

요약 본 논문에서는 멀티미디어 파일 동시 스트리밍 방법을 제안한다. 제안된 방법의 주요한 특징 기여도는 다음과 같다. 첫째, 한 동영상을 두 개 이상의 기기에서 동시에 재생함으로써 공유와 재생을 동시에 하여 사용자에게 편의를 제공한다. 둘째, 스트리밍 방식의 시스템을 둘 이상의 기기에서 동시에 하는 기술을 개발함으로써 통신과 멀티미디어 기술에 이바지 한다. 제안된 방법의 구현 및 성능평가 결과 성공적인 UDP 데이터 통신과 수신하는 쪽에서 데이터를 받음과 동시에 동영상을 효과적으로 출력 할 수 있음을 알 수 있다.

Abstract In this paper, we propose a simultaneously streaming method of multimedia files. The main features and contributions of the proposed method are as follows. First, to share and play at the same time one video playback in more than two devices to provide convenience for users. Second, to contribute communications and multimedia technologies by developing a streaming technology that can work on two or more devices. The results of implementation and performance evaluation show successful UDP data communication and the simultaneous transfer of streaming multimedia with data to the receiving sides.

I. 서론

안드로이드^[1-10]는 운영체제와 미들웨어 그리고 핵심 애플리케이션을 포함하고 있는 모바일 디바이스를 위한 소프트웨어 스택이다. 안드로이드 SDK는 JAVA 프로그래밍 언어를 사용하여 안드로이드 플랫폼 상의 애플리케이션을 개발하기 위해 필요한 도구들과 API를 제공한다.

스마트폰이 보급화 되고 유비쿼터스 네트워크 시대가 되면서 많은 사람들이 유튜브, 페이스북, 멜론플레이어를 통해 언제 어디서든 동영상을 보고 음악을 들을 수 있게 되었다. 누구나 같은 음악을 듣고 동영상을 보면서 '공감'하고 같은 것을 느낄 수 있지만 동시에 감상하기는 어렵

다. 본 논문에서는 동영상, 음악과 같은 미디어 파일을 동시에 공유하고 감상함으로써, '공감' 할 수 있는 애플리케이션 개발을 목표로 한다.

본 논문은 다음처럼 구성되어 있다. II장에서는 유사 애플리케이션의 현황을 파악하고, 안드로이드의 기본 개념에 대해서 설명한다, III장에서는 제안된 방법의 기본 개념과 아이디어에 대해서 설명한다. IV장에서는 제안된 방법의 구현 및 성능평가에 대한 자세한 설명이 이루어지고, V장에서 결론 및 고찰로 본 논문을 맺는다.

II. 안드로이드

안드로이드^[1-10]는 기존의 WIPI, BREW, GVM 등

*준회원, 홍익대학교 컴퓨터정보통신공학과

**중신회원, 홍익대학교 컴퓨터정보통신공학과(교신저자)
접수일자 : 2015년 7월 26일, 수정완료 2015년 8월 26일
게재확정일자 : 2015년 10월 9일

Received: 26 July, 2015 / Revised: 26 August, 2015 /

Accepted: 9 October, 2015

**Corresponding Author: beongku@hongik.ac.kr

Dept. of Computer & Information Communications Engineering,
Hongik University

과 같은 모바일 디바이스를 위한 플랫폼이다. 아주 단순하게 생각하면 PC 위에서 돌아가는 윈도우와 같은 운영체제라고 생각할 수 있다. 안드로이드 플랫폼은 운영체제, 미들웨어, 키 애플리케이션을 포함한 모바일 디바이스를 위한 소프트웨어 집합이다. 표 1 은 안드로이드와 윈도우 응용프로그램 구성요소의 비교를 보여주고 있다. 개발자들이 윈도우에서 애플리케이션을 개발하듯이 안드로이드 SDK를 사용하며 안드로이드 폰에서 동작하는 애플리케이션을 만들 수 있다. 애플리케이션들은 Java 프로그램 언어로 작성해야하고 Dalvik 위에서 실행된다. Dalvik은 구글이 만든 가상머신인데, Linux 커널의 최상위 영역에서 동작한다.

표 1. 안드로이드와 윈도우 응용프로그램 구성요소 비교
Table 1. Comparison of Android with Windows in Application Component

Android	Windows
Activity	Form
Intent Receiver	Event Handler
Service	Linux Daemon
Content Provider	ODBC

아래의 그림 1은 안드로이드의 구조를 보여준다. Applications 에서는 클라이언트, SMS프로그램, 브라우저, 주소록 등을 제공한다. Application Framework은 애플리케이션들이 사용하는 프레임워크를 제공한다. Libraries 에서는 시스템 C Library, Media Library, Surface 관리자, 2D 그래픽 엔진, 3D Library, 경량화된 관계형 데이터베이스 엔진 등이 Library 형태로 제공된다.

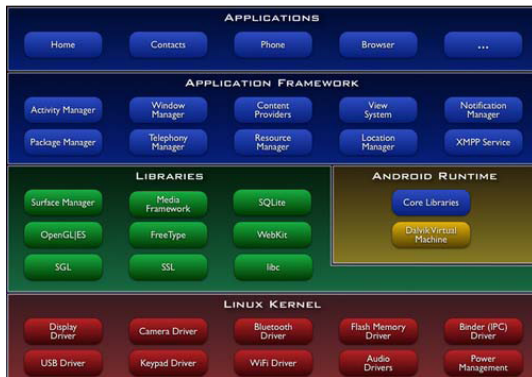


그림 1. 안드로이드 구조
Fig. 1. Android Architecture

갤럭시S4에 기본적으로 탑재된 앱 중에 Group Play가 있다. 내 갤럭시S4에 있는 콘텐츠를 친구에게 공유할 수 있는 아주 손쉬운 방법을 제공한다. Group Play 이용방법은 아주 간단한데 사용자가 그룹을 만들고 다른 사용자가 그룹에 참여하면 된다. 그룹을 만들 때는 비밀번호를 설정할 수가 있어서 공유하고 싶은 친구에게만 콘텐츠를 공유할 수 있다. Group Play에서 공유 가능한 콘텐츠는 음악, 사진, 문서, 게임, 동영상도 공유가 가능하다. 내가 가진 음악을 친구에게 전송할 필요 없이 Group Play를 이용해서 친구들과 같이 들을 수 있다. 그룹에 참여한 갤럭시S4를 왼쪽과 오른쪽 스피커로 지정해서 진정한 스테레오를 즐길 수가 있고 그룹을 만든 기기가 마스터가 되어 다른 갤럭시S4 음량을 조절할 수도 있다.

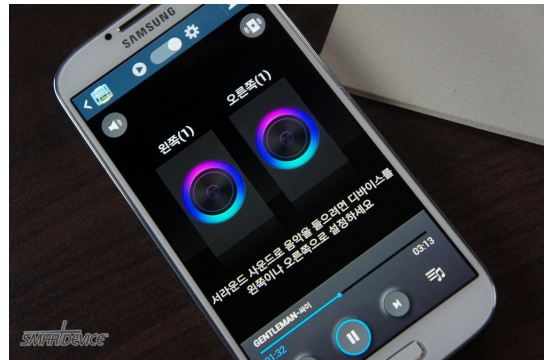


그림 2. 제안된 방법의 기본개념: Group Share 콘텐츠
Fig. 2. Basic concept of proposed method: Group Share Contents

III. 제안된 방법: 안드로이드 기반 스마트폰용 Group Share 콘텐츠

본 논문에서는 그림 2에서 보여주고 있는 안드로이드 플랫폼을 기반으로 한 스마트폰용 애플리케이션 Group Share를 제안한다^[11-15]. 제안된 Group Share 콘텐츠의 기본 개념 및 아이디어는 다음과 같다. 첫째, Group Share는 구글의 안드로이드 플랫폼을 기반으로 한다. 둘째, Group Play의 장점과 단점을 보완하여 강력한 Sharing 기능을 활용한다. 셋째, 기존에 삼성 갤럭시S4 단말기에서만 가능했던 Group Play 기능을 각각의 안드로이드기반 모바일기에서 그룹플레이가 가능하게 하는 애플리케이션 콘텐츠이다.

표 2. 애플리케이션 멀티미디어 공유
 Table 2. Application Multimedia Share

	Group Play	Group Share
사 진	O	O
음 악	O	O
동영상	△	O
문 서	O	O
게 임	O	X

표 2는 본 논문에서 제안한 방법에서 지원 가능한 Group Share 콘텐츠의 기본 기능 및 구현 가능도를 보여 주고 있다. 위에서 설명 하였듯이 Group Share는 같은 곳에 있는 여러 명의 안드로이드 기반 모바일 사용자끼리 와이파이가로 다이렉트 연결하여 같은 음악을 동시에 듣거나 영화도 함께 즐기고 여러 명이 찍은 사진을 전송을 할 필요 없이 공유할 수 있게 해주는 기능들을 가지고 있다^[11-15]. 제안된 애플리케이션의 장점으로서는 **첫째**, 갤럭시 S4 사용자 외에도 Group Play 기능을 사용할 수 있다. **둘째**, 기업에서 회의를 진행할 때 각자 조사해온 영상이나 문서등을 보다 쉽게 공유할 수 있다. **셋째**, 가족모임 또는 친구들과의 모임 등 여러 모임에서 서라운드 시스템의 음악과 영화 등을 즐길 수 있다.

IV. 제안된 방법의 구현

제안된 방법의 효과적인 구현을 위해서 그림 3처럼 자바 실행환경(JRE)을 위하여 OS에 따른 JDK를 다운로드 하고 설치 한 뒤, 삼성 전자에서 제공하는 소프트웨어개발키트(SDK)를 설치해서 보다 수월하게 애플리케이션을 제작할 수 있다. 그림 3은 제안된 방법의 구현을 위한 소프트웨어 개발키트(SDK) 구조를 보여준다. 삼성전자에서 제공하는 모바일 SDK 는 그룹플레이와 같은 애플리케이션을 제작하는 개발자들을 위한 것이다.

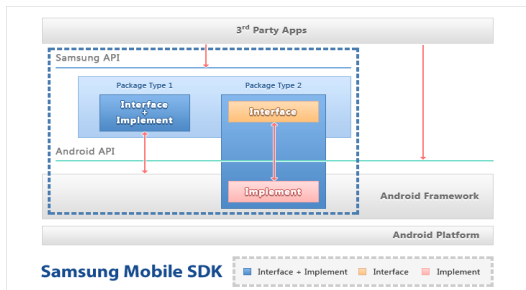


그림 3. 제안된 방법의 구현을 위한 SDK
 Fig. 3. SDK for implementation of the proposed method

1. 구조

본 논문에서 제안된 애플리케이션 개발에 사용된 구조(그림 3, 그림 4)를 좀더 상세하게 설명하면 다음과 같다.

- Group Share는 사진, 비디오, 문서 및 다른 일반적인 애플리케이션 공유 등의 기능을 제공하고 오픈 구조의 목표를 추구하고 있다. 일반 애플리케이션에서는 그룹 내에서 표시되고 플레이 시작 아이콘이 표시되며 장치의 기본 런치에서 직접 시작할 수 있다. 즉, Wi-Fi 또는 모바일 AP를 통해 장치 간에 그룹을 만든다.

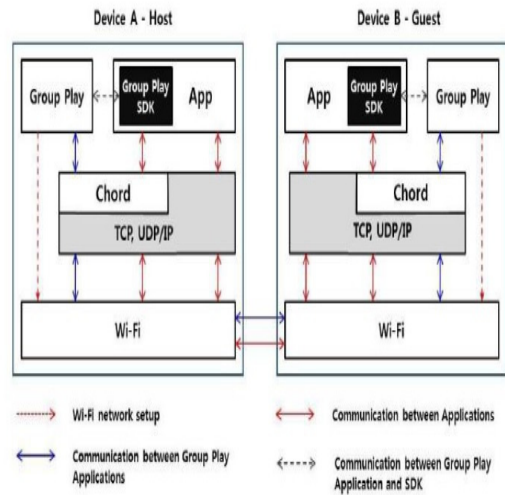


그림 4. Group Share SDK 동작구조
 Fig. 4. Operation Architecture of Group Share SDK

- Group Share SDK는 네트워크를 만들고 장치 간의 통신을 수행하며 다음과 같은 기능을 제공한다[그림 4].
 - ① Group Play를 통해 디바이스 간의 Wi-Fi 네트워크를 만든다.
 - ② Group Play용 다운로드 가능한 응용 프로그램 목록을 표시한다.
 - ③ Group Play를 위해 다운로드 한 응용 프로그램 목록을 생성한다.
 - ④ 이러한 참가자와 응용 프로그램 이름의 수와 그룹 재생 상태를 표시한다.
 - ⑤ Group Share SDK는 그룹 플레이를 통해 Wi-Fi 연결을 설정하고 그룹 플레이 상태를 관리하지만, 응용 프로그램 간 통신을 제공 하지 않는다. 응용 프로그램이 이 기능을 지원 할지도 모르지만, Chord SDK로 대체 하여 사용하는 것이 좋다.

- Chord SDK 는 네트워크 기능으로, 동일한 네트워크에 연결된 장치를 제공 한다. 동일한 Wi-Fi 네트워크에 연결된 피어 장치를 검색하고 메시지를 교환 할 수 있다.
- Wi-Fi는 장치 간 무선네트워크를 만든다.

2. UDP 통신

그림 5는 멀티미디어 파일 동시 스트리밍을 지원하기 위한 UDP 통신 동작을 설명한다. 그림 6은 자료 수신 부분을, 그림 7은 자료 송신 부분을 각각 설명한다.

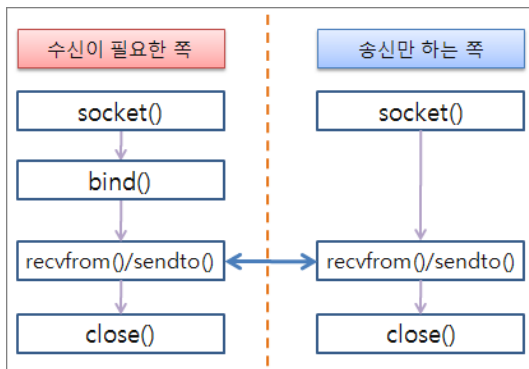


그림 5. UDP 통신
Fig. 5. UDP Communication

가. 자료 수신

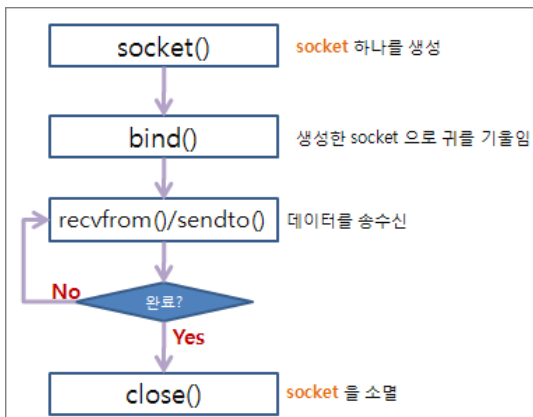


그림 6. 자료 수신
Fig. 6. Data receive

그림 6에서 설명하는 자료 수신부분을 자세하게 각 부분 요소별로 나누어서 설명하면 다음과 같다.

■ socket 생성

```

socket 생성
int sock;

sock = socket( PF_INET, SOCK_DGRAM, 0);
if (-1 == sock)
{
    printf( "socket 생성 실패");
    exit( 1);
}
    
```

■ bind()메소드를 이용하여 커널에 등록

- ① 만들어진 sock은 단지 socket 디스크립터일 뿐이다.
- ② 이 socket에 주소를 할당하고 port 번호를 할당해서 커널에 등록해야 한다.
- ③ 커널에 등록하면 클라이언트로부터 자료를 수신할 수 있는 귀를 달아 놓는 상태가 된다.
- ④ socket에 주소와 port 를 할당하기 위해 sockaddr_in 구조체를 이용한다.
- ⑤ htonl(INADDR_ANY)는 주소를 지정해 주는 것으로 inet_addr("내 시스템의 IP ")로도 지정할 수 있다. 그러나 프로그램이 실행되는 시스템 마다 IP가 다를 것이므로 주소 지정을 고정 IP로 하지 않고 htonl(INADDR_ANY) 를 사용하는 것이 편리하다.

```

bind() 메소드
struct sockaddr_in server_addr;

memset( &server_addr, 0, sizeof( server_addr);
server_addr.sin_family = PF_INET; // IPv4 인터넷 프로토콜
server_addr.sin_port = htons( 4000); // 사용할 port 번호는 4000
server_addr.sin_addr.s_addr = htonl( INADDR_ANY);// 32bit IPv4 주소

if( -1 == bind( sock, (struct sockaddr*)&server_addr, sizeof( server_addr) ) )
{
    printf( "bind() 실행 에러\n");
    exit( 1);
}
    
```

■ recvfrom() 메소드

socket을 커널에 등록했으므로 recvfrom() 또는 sendto()를 이용하여 자료를 송수신할 수 있다. recvfrom() 메소드를 이용하여 클라이언트로부터 전송되어 오는 자료를 읽어 들인다.

```
recvfrom() 메소드
//1번
struct sockaddr_in client_addr;
int client_addr_size;

//2번
client_addr_size = sizeof( client_addr);
recvfrom( sock, buff_rcv, BUFF_SIZE, 0 ,
          ( struct sockaddr*)&client_addr,
          &client_addr_size);
```

- ① recvfrom()을 이용하여 송신 측의 주소 정보를 받아 오기 위해 변수를 선언한다.
- ② recvfrom()으로 자료를 수신한다.
- ③ 수신한 자료가 있다면 자료는 buff 에 저장되며, client_addr에 송신측의 주소 정보가 담기게 된다.
- ④ client_addr를 이용하여 송신한 측으로 자료를 전송할 수 있다.

■ sendto() 메소드를 이용하여 데이터를 전송한다.

```
sendto() 메소드
//1번
sprintf( buff_snd, "%s%s", buff_rcv, buff_rcv);

//2번
sendto( sock, buff_snd, strlen( buff_snd)+1, 0,
        ( struct sockaddr*)&client_addr, sizeof( client_addr));
// +1: NULL까지 포함해서 전송
```

- ① 전송 데이터를 준비한다.
- ② sendto()를 이용하여 클라이언트로 자료를 송신한다.
- ③ 위의 작업을 반복한다.

나. 자료 송신

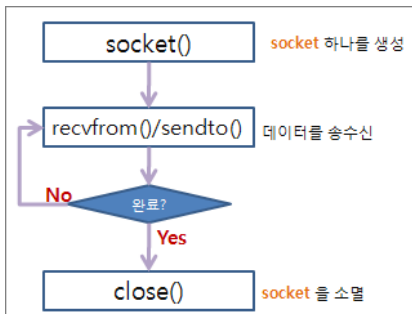


그림 7. 자료 송신
 Fig. 7. Data transmission

자료를 전송하는 쪽은 bind() 함수를 사용할 필요가 없으며, bind()함수를 사용하지 않았기 때문에 다른 시스템에서 전송하여도 전송 사실 자체를 알 수가 없다. 그림 7에서 설명하는 자료 송신부분을 각 부분요소별로 자세히 설명하면 다음과 같다.

■ socket()을 이용하여 소켓을 먼저 생성한다.

```
socket()
int sock;

sock= socket( PF_INET, SOCK_DGRAM, 0);
if( -1 ==sock)
{
    printf( "socket 생성 실패n");
    exit( 1);
}
```

■ 커널에 socket 을 등록할 필요 없이 sendto()로 자료를 송신할 수 있다.

```
sendto()
struct sockaddr_in server_addr;

memset( &server_addr, 0, sizeof( server_addr));
server_addr.sin_family = AF_INET;
server_addr.sin_port = htons( 4000);
server_addr.sin_addr.s_addr= inet_addr( "127.0.0.1");
```

- ① 상대 시스템이 실행 중인지는 모른다. 그러나 전송할 수 있다.
- ② 상대 시스템의 주소 정보를 준비한다.
- ③ sendto() 함수로 데이터를 전송한다.

■ sendto() 함수를 사용하여 커널에 소켓이 등록되므로, 이번에는 recvfrom() 함수를 이용하여 상대방으로부터 자료를 읽어 들일 수 있다.

```
recvfrom()
sendto( sock, argv[1], strlen( argv[1])+1, 0,
        ( struct sockaddr*)&server_addr, sizeof( server_addr)); // +1: NULL까지 포함해서 전송
//1번
struct sockaddr_in server_addr;
int server_addr_size;
//2번
server_addr_size= sizeof(server_addr);
recvfrom( sock, buff_rcv, BUFF_SIZE, 0 ,
          ( struct sockaddr*)&server_addr,
          &server_addr_size);

printf( "receive: %sn", buff_rcv);
close( sock);
return 0;
```

- ① recvfrom()을 이용하여 송신 측의 주소 정보를 받아 오기 위해 변수를 선언한다.
- ② recvfrom()으로 자료를 수신한다.
- ③ 수신한 자료가 있다면 자료는 buff에 저장된다.
- ④ 수신된 자료를 화면에 출력하고 종료한다.

3. 동영상 재생/정지

동영상 재생 및 정지부분을 각 부분요소별로 자세하게 설명하면 다음과 같다.

```

동영상 재생
/**
 * 동영상 재생 시작
 */
    VideoView        videoViewExample        =
    (VideoView)findViewById(R.id.videoView1);
    Uri videoUri = Uri.parse(
    "android.resource://" +      getPackageName() +
    "/raw/sample_video"
    );
    videoViewExample.setVideoURI(videoUri);
    videoViewExample.start();
    
```

여기서 첫 번째 줄은 main.xml에 배치했던 VideoView 객체를 호출한다. 그 다음 res/raw 폴더에 등록했던 동영상 파일의 경로를 정의한 후 동영상 경로를 Uri 형으로 변형한다. VideoView 객체를 Uri형으로 호출해 페이지를 열었을 시 동영상이 바로 재생되게 한다.

```

동영상 정지
/**
 * 동영상 재생 중지 ( 일시 중지 )
 */
    public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    MediaController mc = new MediaController(context);
    videoViewExample.setMediaController(mc);
    }
    
```

onCreate 메소드안에 MediaController를 정의하는 방식을 사용하였다. MediaController 객체를 생성하고 동영상이 재생될 시 화면에 나타나도록 정의하였다. 화면을 한번 터치하면 재생 컨트롤러가 나타나고, 이 컨트롤러에 있는 재생버튼을 클릭하면 동영상이 재생된다.

4. 사진 뷰어

사진 뷰어 부분을 각 부분요소별로 자세하게 설명하면 다음과 같다.

```

GridView
    GridView gv =
    (GridView)findViewById(R.id.imgGridView);
    final ImageAdapter ia = new ImageAdapter(this);
    gv.setAdapter(ia);
    gv.setOnItemClickListener(new OnItemClickListener(){
    public void onItemClick(AdapterView parent, View v, int
    position, long id){
        ia.callImageViewer(position);
    }
    });
    
```

GridView 의 내용을 표현하기 위해서는 Adapter 클래스가 필요하다. 그래서 BaseAdapter를 상속받아 ImageAdapter 가 구현 되었다. GridView에 들어가는 개별 View는 ImageView 위젯을 사용한다. ImageView 위젯에 Bitmap 정보를 넣어서 넘기면 되는 것이다.

```

Bitmap 정보
    ImageView imageView;
    if (convertView == null){
        imageView = new ImageView(mContext);
        imageView.setLayoutParams(new
        GridView.LayoutParams(95, 95));
        imageView.setAdjustViewBounds(false);

        imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
        imageView.setPadding(2, 2, 2, 2);
    }else{
        imageView = (ImageView) convertView;
    }
    // 이미지 크기를 새로 설정한다.
    BitmapFactory.Options bo = new BitmapFactory.Options();
    bo.inSampleSize = 8;
    Bitmap bmp =
    BitmapFactory.decodeFile(thumbsDataList.get(position), bo);
    Bitmap resized = Bitmap.createScaledBitmap(bmp, 95, 95, true);
    imageView.setImageBitmap(resized);
    
```

Bitmap 정보를 설정하는 소스 이다. ImageView의 레이아웃 설정은 setLayoutParams으로 하며, setScaleType 을 사용하여 이미지를 View 크기에 어떻게 넣을 건지 설정한다. 그리고 아래에 소스에서는 이미지를 Bitmap 객체에 불러와 크기를 새로 치수에 맞게 새로 설정하였다.

```
managedQuery 정보
String[] proj = {MediaStore.Images.Media_ID,
                MediaStore.Images.Media.DATA,

                MediaStore.Images.Media.DISPLAY_NAME,
                MediaStore.Images.Media.SIZE};
Cursor imageCursor =

managedQuery(MediaStore.Images.Media.EXTERNAL_CON
TENT_URI,
            proj, null, null, null);
```

여기서 중요하게 다루어지는 내용 중 하나는 managed Query를 사용하여 SD 카드에 그림들을 전부 불러오는 부분이다. 첫 번째 들어가는 파라미터에 따라 다양한 정보들을 불러 올 수 있도록 Uri 정보를 넘기는데 미디어 정보인 이미지를 가져오기 위해서는 MediaStore.Images.Media.EXTERNAL_CONTENT_URI 필요하다. 그리고 어떤 데이터를 받을지 String 배열을 파라미터로 설정해 넘기면 된다. 리턴값으로 Cursor 객체를 받게 되는데 그것을 이용해서 화면을 표현하면 된다.

```
managedQuery 정보
Cursor                imageCursor                =
managedQuery(MediaStore.Images.Media.EXTERNAL_CON
TENT_URI,
            proj, "_ID='"+ thumbID +'"", null, null);
if (imageCursor != null && imageCursor.moveToFirst()){
    if (imageCursor.getCount() > 0){
        int                imgData                =
imageCursor.getColumnIndex(MediaStore.Images.Media.DAT
A);
        imageDataPath = imageCursor.getString(imgData);
    }
}
```

managedQuery 의 함수와 파라미터 정보를 사용해 동 영상, 이미지 등 다양한 정보를 Cursor 객체로 받을 수 있다. 그리고 쿼리의 where 절처럼 두 번째 쿼리에 값을 설정하면 특정 데이터를 불러 올 수 있다.

그림 8은 수신측에서 수신된 멀티미디어 파일 동시 스트리밍 결과를 보여주고 있다.

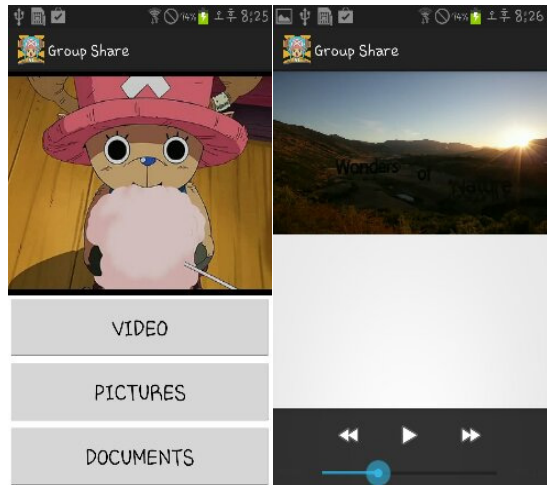


그림 8. 수신된 Group Share 콘텐츠
 Fig. 8. Received Group Share Contents

V. 결론 및 고찰

본 논문에서는 안드로이드 플랫폼을 기반으로 하여 기존의 Group Play의 문제점을 보완하고 개선한 새로운 개념의 멀티미디어 파일 동시 스트리밍을 위한 애플리케이션을 제안 개발하였다. 삼성에서 개발한 기존의 Group Play는 삼성내의 기기끼리도 콘텐츠공유가 제약을 받았다. 예를 들어 갤럭시S4와 갤럭시 노트3의 그룹플레이가 되지 않았다. 하지만 본 논문에서 제안한 방법은 이것을 보완하기 위해 안드로이드 플랫폼을 이해하고 삼성에서 무료로 배포한 SDK를 연구하여 안드로이드 OS 체제의 모바일끼리 콘텐츠를 자유롭게 공유할 수 있도록 애플리케이션을 개발제작 하였다. 제안된 방법의 구현 및 성능 평가 결과 성공적인 UDP 데이터 통신과 수신하는 쪽에서 데이터를 받음과 동시에 동영상을 효과적으로 출력할 수 있음을 알 수 있다.

References

- [1] <http://egg-bread.tistory.com/1267>
- [2] <http://www.wikipedia.org>
- [3] <http://smartdevice.kr/648>
- [4] <http://developer.android.com>
- [5] <http://developer.samsung.com/samsung->

mobile-sdk/start

- [6] <http://developer.android.com/sdk/android-1.5.html>
- [7] <http://developer.android.com/reference/android/webkit/WebViewClient.html>
- [8] <http://developer.android.com/reference/android/view/View.html>.
- [9] <http://www.eclipse.org/downloads>
- [10] <http://java.sun.com/javase/downloads>
- [11] Mohammad Reza Zakerinasab, Mea Wang, "A cloud-assisted energy-efficient video streaming system for smartphones," IEEE Proc. of IWQoS 2013, pp.1-10, June 2013.
- [12] Hsien-Po Shiang and Mihaela van der Schaar, "Multi-user video streaming over multi-hop wireless networks: A distributed, cross-layer approach based on priority queuing," JSAC, vol.25, issue 4, pp.770- 785, May 2007.
- [13] Zhenyu Yang, Ming Li and Wenjing Lou, "CodePlay: Live multimedia streaming in VANETs using symbol-level network coding," IEEE Proc. of ICNP 2010, pp.223-232, October 2010.
- [14] Minseh Kim, Beongku An, "Video Contents Security Streaming," JIWT, vol.12, no.5, pp 67-74, October 2012.
- [15] Takuya Fujihashi, Ziyuan Pan, Takashi Watanabe, "Traffic Reduction for Multiple Users in Multi-view Video Streaming," IEEE Proc. of ICME 2012, pp.182-187, July 2012.

저자 소개

김진영(준회원)



• 2014년 : 홍익대학교 컴퓨터정보통신 공학과 졸업 (BS)
 <주관심분야 : 정보보안, 네트워크보안, 무선네트워크>

이은상(준회원)



• 2014년 : 홍익대학교 컴퓨터정보통신 공학과 졸업 (BS)
 <주관심분야 : 정보보안, 네트워크보안, 무선네트워크>

임소희(준회원)



• 2014년 : 홍익대학교 컴퓨터정보통신 공학과 졸업 (BS)
 <주관심분야 : 정보보안, 네트워크보안, 무선네트워크>

안병구(중신회원)



• 1988년 : 경북대학교 전자공학과 (B.S)
 • 1996년 : New York University (Polytechnic)Dept. of Computer and Electrical Eng., NY, USA (M.S)
 • 2002년 : (미)New Jersey Institute of Technology (NJIT), Dept. of Computer and Electrical Eng., NJ, USA (Ph.D)
 • 1989년 ~ 1994년 : 포항산업과학기술연구원(RIST), 선임연구원
 • 2012년 : 대한전자공학회 컴퓨터소사이터티 회장
 • 2003년 ~ 현재 : 홍익대학교 컴퓨터정보통신공학과 교수
 <주관심분야 : Wireless Networks, Ad-hoc & Sensor Networks, Multicast Routing, QoS Routing, VLC, Cognitive Radio Networks, Energy Harvesting, Physical Layer Security, Mobile Cloud Computing, Cross-Layer Technology, Network Coding, Cooperative Communication, Bioinformatics>

※ This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2012R1A2A2A01046780).