

# 클라우드 컴퓨팅 기반의 스마트폰 블랙박스 시스템 구현

차민수\*, 정순호\*\*

부경대학교 컴퓨터 멀티미디어 공학부\*

부경대학교 컴퓨터공학과\*\*

E-mail : freemagick@naver.com\*

## An Implementation of Black Box System of Smart phone based on Cloud Computing

Min-Soo Cha\*, Soon-Ho Jung\*\*

Dept of Computer Multimedia Science, Pu-Kyong University\*

Dept of Computer Science, Pu-Kyong University\*\*

### 요 약

최근 차량용 블랙박스의 저장된 데이터의 안전성에 대한 관심이 높아지고 있다. 현재의 차량용 블랙박스는 영상녹화 및 저장을 위해 보통 플래시 메모리를 삽입하여 사용하는데 이는 메모리의 분실 또는 데이터 파손 등 데이터 보관성에 문제를 일으킬 수 있다.

본 논문에서는 기존의 차량용 블랙박스의 데이터 안전성을 높이고자 클라우드 컴퓨팅 개념을 도입하여 사고발생 시 중요한 데이터를 서버에 최대한 업로드하고 이를 이용해 제 3자에게 영상 데이터를 제공하거나 사고를 통보 할 수 있는 기능을 가진 블랙박스 시스템을 안드로이드 OS에서 구현하고자 한다.

### 1. 서론

차량용 블랙박스는 교통사고 전후의 사고발생 영상, 발생 위치, 차량운행 기록, 음성 등을 자동으로 기록하는 장치를 의미하며, 차량 운행정보 생성방식, 카메라에 의한 동영상 촬영 등 특성에 따라 '디지털운행기록장치'와 '영상기록장치'로 구분된다. 이들의 차이는 영상녹화 기능의 유무이며, 단순히 정보를 기록하는 것이 '디지털운행기록장치'이고 영상녹화 기능을 가진 것이 '영상기록장치'이다.[1]

현재 국내에서 판매되고 있는 블랙박스의 대부분이 플래시 메모리를 삽입한 '영상기록장치'(이하 블랙박스)이다. 이 같은 블랙박스는 사용자가 직접 구매하여 간단하게 설치할 수 있게 만들어 지기 때문에 전문 블랙박스에 비해 비교적 큰 충격에 약할 수 있다. 이는 대형사고로 블랙박스가 파손될 경우 내장된 플래시 메모리의 파손이나 분실 위험이 있을 수 있고 누군가 고의적으로 블랙박스에 접근하여 메모리의 내용을 모두 지우거나 플래시 메모리의 고장으로 데이터 복구가 불가능 한 경우도 있다.

본 논문에서는 이러한 문제를 해결하기 위해 클라우드 컴퓨팅 개념을 도입하여 영상데이터를 로컬에 저장하는 것뿐만 아니라 별도의 서버에 안전하게 저장하는 방법을 이용하여 충격 감지 시에 중요한 데이터를 최대한 서버에 저장하는 방법을 구현하고자 한다. 서버에 저장된 데이터는 뷰어를 통하여 언제 어느 곳에서든 필요할 때 영상을 열람 할 수 있다. 또한, 사고 발생 시 뷰어를 통해 제 3자

에게 알람을 울리는 기능도 추가 하고자 한다. 이 기능은 교통사고 발생 시 조기에 발견하지 못해서 커지는 인명피해를 줄여 줄 수 있을 것으로 본다.

이러한 기능들을 구현하기 위해서는 영상기록정보 외에도 GPS정보, 충격감지정보 등을 획득하기 위해 이 시스템 구현에서는 GPS, 가속도센서 그리고 네트워크 모듈이 내장되어 있고 개방된 플랫폼으로 인하여 개발이 용이한 안드로이드 OS 기반의 스마트폰을 이용한다.

### 2. 관련연구

#### 2.1. 안드로이드 기반의 블랙박스

안드로이드 OS는 모바일 디바이스를 위한 운영체제와 미들웨어 그리고 핵심 어플리케이션을 포함하고 있는 소프트웨어 플랫폼이다. 안드로이드 SDK는 스마트폰에 맞게 최적화된 다양한 API를 제공하고 개방형 플랫폼을 지원해서 라이선스에서 자유로워 개발이 용이하고 검증된 많은 라이브러리들을 포함하고 있다. 또 한 가지 중요한 점은 구글 맵 API를 사용 할 수 있다는 것인데, 이는 블랙박스 시스템을 위한 최적의 환경이다.

안드로이드 OS용의 블랙박스 어플리케이션은 기본적으로 영상을 보여주는 SurfaceView와 센서를 관리하는 SensorManager, GPS를 관리하는 LocationManager, 영상을 녹화하는 MediaRecorder의 4가지로 구성된다. 앞의 이러한 클래스를 이용하여 Activity를 디자인 하여 제작하는

것이다.[2]

### 2.2. 클라우드 컴퓨팅

클라우드 컴퓨팅은 그리드 컴퓨팅, 분산 컴퓨팅, 병렬 컴퓨팅과 결합한 가상화를 기반으로 한 기술이다. 이 가상화 기술은 운영체제, 미들웨어 또는 응용프로그램들이 물리적인 장치나 서버에서 대신 실행되거나 할당된 것에 대한 형상을 말한다.[6]

클라우드 컴퓨팅의 이러한 특징은 다양한 기기를 서비스 단말로 이용가능하며 사용자의 데이터를 신뢰성 높은 서버에 보관함으로써 안전하게 저장 가능하게 한다. 또, 데이터가 여러 기기에 분산 저장되어지는 것을 방지하므로 저장 장치의 낭비를 초래하지 않고 데이터의 보안성도 높아지게 된다.

## 3. 시스템 설계

### 3.1. 시스템 구성

본 논문의 블랙박스 시스템은 (그림 1)과 같이 서버와 블랙박스 어플리케이션 그리고 뷰어 어플리케이션 3가지로 구성된다.



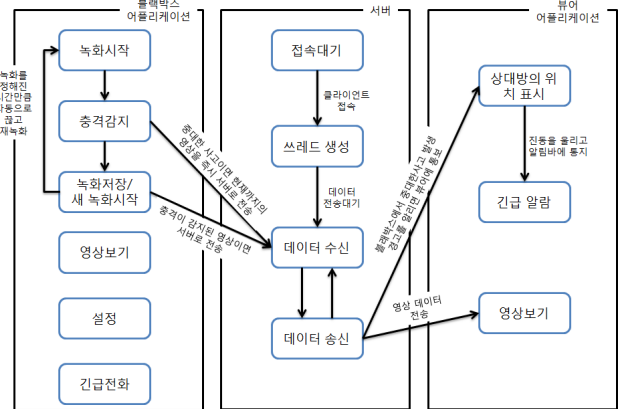
(그림 1) 블랙박스 시스템 구성

서버의 역할은 블랙박스로부터 영상을 전송받거나 뷰어에 저장된 영상과 블랙박스의 위치를 제공한다.

블랙박스 어플리케이션은 주행 중 영상을 항상 녹화하다가 충격이 감지되면 그 시점을 체크해 놓고 정해진 시간이 되면 현재까지의 녹화를 저장한다. 이 때, 충격이 감지된 영상은 바로 서버로 업로드를 한다. 만약, 녹화 중에 아주 큰 충격이 감지되면 대형사고로 판단하고 촬영을 즉시 중단하여 서버로 업로드 한다. 이후 촬영 되는 영상은 바로 서버로 실시간 업로드 되도록 한다.

뷰어 어플리케이션은 서버에 저장된 영상을 다운받아 보거나 블랙박스 사용자의 위치를 실시간으로 표시하여 주고 사고발생 시 알람을 울려준다.

### 3.2. 시스템 동작



(그림 2) 시스템 동작구조

시스템의 주요 동작구조는 (그림 2)와 같다. 블랙박스 어플리케이션에서 녹화가 시작되면 녹화시간을 체크한다. 정해진 시간이 되면 현재까지의 녹화를 저장하고 새로운 녹화를 시작한다. 만약, 녹화 중에 충격을 감지하여 충격정도가 높으면 촬영을 중단한 뒤 현재까지의 녹화화면과 사고이벤트를 즉시 서버로 전송하고, 충격정도가 높지 않다면 중요영상으로 체크하고 녹화가 끝나면 업로드 큐에 추가한다.

서버는 블랙박스 어플리케이션에서 보내주는 데이터를 전송받아 해당 파일을 저장해놓고 데이터베이스에 이 파일의 정보를 기록해놓는다. 만약, 블랙박스 어플리케이션에서 사고이벤트를 보내 왔다면 뷰어 어플리케이션에 이를 다시 재전송하여 알려준다.

뷰어 어플리케이션은 서버에 저장된 블랙박스 어플리케이션의 위치정보를 실시간으로 제공받는다. 또, 필요한 영상 데이터를 서버에 요청하면 이를 목록으로 보여주고 선택한 파일을 재생하는 동작을 한다.

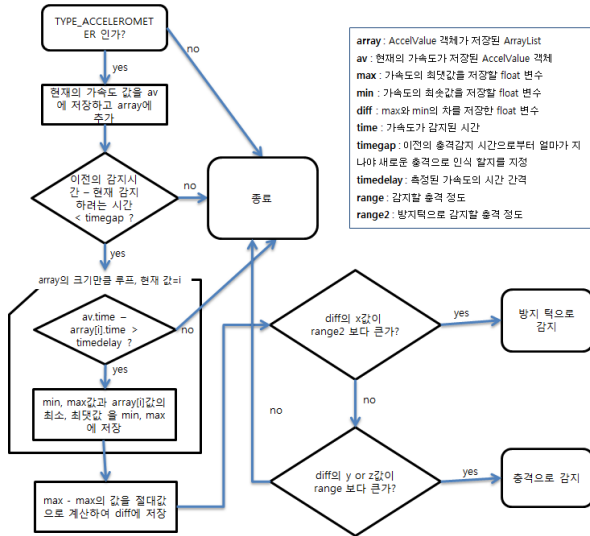
#### 3.2.1. 서버

서버는 어플리케이션의 접속을 항시 대기하다가 접속요청이오면 새로운 Thread를 띄운다. 이후 이 Thread에서 접속이 종료될 때까지 통신한다. 서버와 어플리케이션간의 통신은 DataInputStream과 DataOutputStream을 이용하여 4096kb 크기의 패킷을 정하여 데이터를 주고받는다. 패킷의 처음에는 1바이트의 헤더가 오고 뒤에는 그 헤더의 목적에 맞는 데이터가 위치한다.[7]

#### 3.2.2. 블랙박스 어플리케이션

블랙박스 어플리케이션에서의 구현요점은 가속도 센서와 GPS 센서를 이용하여 충격을 감지하는 알고리즘과 충격

감지 시에 데이터를 서버로 전송하는 부분이다.



(그림 3) 가속도 센서를 이용한 충격감지

가속도 센서는 기기의 가속도 정도를 x,y,z 축으로 나타내어 주는데 이를 이용하면 충격을 감지할 수 있다. 먼저 가속도의 변화가 지속되는 시간이 일정시간 이상인지 판단하고 그 시간동안 진행된 가속도변화의 최댓값에서 최솟값을 뺀 값을 이용하여 충격이 얼마나 큰지를 판단한다. 여기서 방지턱의 경우 x축(기기를 옆으로 누워 후면 카메라를 전방으로 했을 때)으로 충격이 가해지기 때문에 이 값을 제외하고 y축과 z축의 값만 이용한다.[4]

영상의 녹화는 MediaRecorder를 이용하며 영상을 자동으로 끊어 저장하고 재녹화하기 위하여 녹화관리를 위한 클래스를 만들어 이용한다. 현재는 1분으로 설정되어 있고 이 값은 변경이 가능하다. 녹화 중에는 GPS정보와 충격감지정보가 같이 기록되는데 각각의 확장자명은 .acc와 .gpx이다.[6]

영상의 녹화 중에 일반적인 충격이 감지되면 그 충격 시간을 기록해 놓았다가 녹화관리자에 의하여 녹화가 끝나면 영상을 중요영상으로 기록하고 서버로 업로드 한다. 만약, 큰 충격이 감지되면 이를 중대한 사고로 판단하여 즉시 촬영을 중지하고 현재까지의 영상을 저장하여 서버로 업로드 한다.

### 3.2.3. 뷰어 어플리케이션

뷰어 어플리케이션은 서버에서 위치정보를 실시간으로 제공받고 이를 구글맵을 이용하여 표시한다. 맵 위의 위치 표시를 위하여 MapOverlay를 이용하고 사고가 날 경우 사고지점을 표시하고 알림 바에 통지를 한다.[5][8]

## 4. 구현 및 평가

### 4.1. 구현결과

서버와 블랙박스, 그리고 뷰어의 3가지 부분으로 구현하

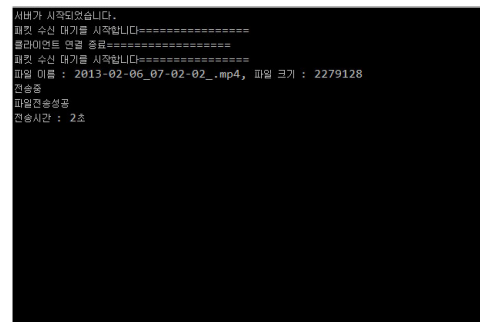
였다. 서버는 윈도우 환경의 Java와 MySQL을 이용하였고 블랙박스와 뷰어 어플리케이션은 안드로이드 OS의 Android SDK를 이용하여 구현하였다. 상세 개발환경은 <표 1>과 같다. 네트워크는 주로 KT의 와이브로 에그를 이용하여 와이브로 망에서 개발을 진행하였고, LTE망 테스트를 위하여 2<sup>nd</sup> Device로 Galaxy Note를 이용하였다.

<표 1> Software & Hardware 개발환경

Software		
항목	서버	블랙박스 & 뷰어
O.S	Windows 7	Android
Language	Java 1.6	Android SDK 10
Library	MySQL 5	Google APIs 10
Hardware		
항목	서버	블랙박스 & 뷰어
Model	-	Galaxy S2
CPU	Intel E6600	Exynos 4210
Memory	2GB	1GB
Sensor	-	GPS, 가속도, 근접
Network	100Mbps Lan	3G, WIFI(WiBro egg)

### 4.1.1. 서버

서버는 GUI를 사용하지 않고 텍스트 기반으로 작성되었으므로 서버의 상태를 알려주는 메시지만 화면에 출력한다. 서버로 업로드 되는 데이터는 업로드 날짜와 파일명, 기기 ID(차후 여러 디바이스를 구분하기 위함) 등을 데이터베이스에 기록하고 저장된다.



(그림 4) 서버 화면

### 4.1.2. 블랙박스

블랙박스 어플리케이션은 크게 메인화면과 영상관리화면, 영상재생화면, 설정화면으로 구성된다.



(그림 5) 블랙박스 화면 구성

메인화면은 주차/주행모드 버튼으로 녹화방식을 변경할 수 있고 듀얼뷰 버튼은 지도를 보여주며 긴급전화 버튼은 전화를 바로 걸 수 있는 버튼들을 표시한다. 영상보기 화면은 기기에 저장된 영상을 관리하는 화면을 띄우고 설정 버튼은 어플리케이션의 설정을 변경 할 수 있는 화면이다.

### 4.1.3. 뷰어

뷰어 어플리케이션은 서버에 접속하여 상대방의 위치를 맵에 표시하거나 서버에 저장된 영상파일을 재생하는 기능을 가진다. 영상목록화면에서 원하는 파일을 터치하면 영상이 다운로드 되고 재생된다.



(그림 6) 뷰어 화면 구성

## 4.2. 평가

본 논문에서 구현한 블랙박스 시스템을 이용하여 70~80여회 야외에서 테스트를 해보았다. 테스트에는 도보, 자전거, 자동차를 이용하여 진행하였는데, 도보와 자동차 테스트 시에는 고의로 스마트 폰에 충격만 주었고 자전거 테스트 시에는 간접적으로 사고 상황을 만들어 충격을 주었다. 세 상황 모두 블랙박스의 기본기능은 잘 작동하였고 네트워크로의 파일 전송도 잘 되었다. 하지만 가속도를 충격값으로 환산하는 경우 내장된 가속도 센서의 값과 충격 정도의 크기에 대한 비교 기준이 없어서 두 값을 잘 조율하기가 어려웠다.

중요영상으로 판별된 데이터의 경우 서버로 업로드 하는 시간이 일반적인 상황에서는 최소한 녹화속도보다는 빨라야한다. 기본적인 녹화시간이 1분이기 때문에 적어도 59초 안으로는 전송이 완료되어야한다는 뜻이다. 하지만 가장 느린 3G망으로도 테스트 결과 16초안으로는 전송이 완료되기 때문에 문제는 없을 것으로 보고 있다. 다만, 대형사고가 발생할 경우는 스마트 폰이 파손되기 전에 서버로 전송을 완료해야 하므로 전송시간이 더욱 더 빨라야 한다. <표 2>에는 각 망에 대한 테스트 결과가 나타나 있다.[3]

<표 2> 각 모바일 망에 대한 테스트 결과

	LTE	WiBro	3G
측정위치	부산시 남구 대연동 지역		
통신사	SKT	KT	KT
평균 전송속도	약 40Mbps	약 9Mbps	약 3Mbps
평균 전송시간	약 1.5초	약 8초	약 16초

## 5. 결론

블랙박스는 교통사고의 원인을 규명하는데 있어 최후의

보루이다. 하지만 전제조건은 녹화된 영상데이터가 온전하게 보관되어있어야 한다는 것이다. 기존의 블랙박스는 플래시 메모리를 이용하여 데이터를 저장한다. 플래시 메모리는 콤팩트해서 분실의 위험이 존재하고 뜻하지 않게 데이터를 유실할 수도 있다.

본 논문에서 구현한 클라우드 컴퓨팅 기반의 블랙박스 시스템은 이런 문제에 대하여 해결책을 제시하였다. 실시간으로 영상을 서버로 업로드 하는 것이 가장 좋은 솔루션이지만 이는 많은 트래픽을 유발하고 전력소모도 커지게 된다. 따라서 충격을 감지하여 중요영상만을 서버로 업로드하고 대형사고일 경우는 즉시 서버로 업로드 하는 방식을 취하면 데이터 트래픽과 데이터의 안전성이라는 두마리의 토끼를 모두 잡을 수 있다. 또한 뷰어 어플리케이션을 통하여 제 3자에게 본인이 녹화한 영상을 간단하게 보여줄 수 있고 긴급한 사고에서의 알람기능으로 조기대처가 가능하다.

모바일 통신망을 이용하는 만큼 고용량의 데이터 요금제를 이용하는 것이 아니라면 데이터 사용에 거부감이 있을 수 있겠지만 블랙박스 전용데이터 정책제정의 요금제를 통신사와 협력하여 만들어 낸다면 이는 통신사에는 새로운 수익모델이 될 수 있고 사용자에게는 안전한 블랙박스 환경을 제공하는 기회가 될 것이다.

## 참고문헌

- [1] 손해보험협회 정책연구팀(2011), “국내의 차량용 블랙박스 현황과 제도화 방향(上)”, 통권 제510호, pp. 56-64
- [2] 송민섭, 장종욱(2011), “안드로이드 OS 기반의 차량용 블랙박스 시스템 구현”, 한국해양정보통신학회 2011 추계종합학술대회, pp. 483-486
- [3] 한국통신사업자연합회, 이재혁(2011), “Special Theme2 : 4G 시대, LTE VS WiBro 대전 개막”, 통신연합 57호, pp. 14-20
- [4] 김상현(2011), “안드로이드 프로그래밍 정복”, 한빛미디어, pp. 1372-1378
- [5] 후루가와 히데카즈, “안드로이드 프로그래밍”, 정보문화사, pp. 209-304
- [6] Panagiotis Kalagiakos(2011), “Cloud Computing Learning”, Application of Information and Communication Technologies (AICT), 2011 5th International Conference on
- [7] Java API(2013), “Java SE 6 API”, <http://docs.oracle.com/javase/6/docs/api/>
- [8] Google(2013 January 31), “Google Map API”, <https://developers.google.com/maps/?hl=ko>