

Asynchronous Message Pushing Framework between Android Devices using Remote Intent

Jihun Baek[†] · Yongwoo Nam^{**} · Sangwon Park^{***}

ABSTRACT

When developing an android mobile application the androids intent is used as a mechanism to send messages between local equipment of androids application inner part and other applications. But the androids intent does not support sending messages via each android products intent. If there is a way to support each androids equipments to send messages, it will be easier to make non-stopping services. Non-stopping service is used when the user is using the android to do word or searching services and suddenly changes to a different android product but still maintains the progress what was currently being done without waiting the programs to be loaded. It is possible to send messages to each android products by using the socket, but the connection must be maintained stably which is the weak point. In this paper, I am suggesting a BRIF(Broadcasting Remote Intent Framework) framework to send messages to different android products. BRIF is a framework that uses the Googles C2DM service which services asynchronous transmissions to different android products. This is organized with the C2DM server, RemoteContext Api, web server and RISP(Remote Intent Service Provider) which is will be easy to be used for the developers since there are no big changes for coding compared to the intent code.

Keywords : Application, FrameWork, Intent, Android, Asynchronous

Remote Intent를 이용한 안드로이드 장치 간 비동기식 메시지 푸싱 프레임워크

백 지 훈[†] · 남 용 우^{**} · 박 상 원^{***}

요 약

안드로이드 장치용으로 모바일 애플리케이션을 개발할 때 안드로이드의 인텐트(intent)는 안드로이드 장치에서 애플리케이션 내부와 애플리케이션 간의 메시지 전달 메커니즘으로 사용된다. 그러나 안드로이드의 인텐트는 서로 다른 안드로이드 장치 간의 인텐트를 이용한 메시지 전송은 지원되지 않고 있다. 만약 서로 다른 안드로이드 장치 간에 인텐트를 전송하는 기능이 지원된다면 좀 더 다양한 애플리케이션 구현을 쉽게 할 수 있다. 서로 다른 안드로이드 장치 간에 메시지 전송을 할 때 Socket을 이용하여 메시지를 전송 할 수는 있지만, 항상 연결을 유지하여야 한다는 단점이 있다. 따라서 본 논문에서는 서로 다른 안드로이드 장치 간의 메시지 전송을 위한 BRIF(Broadcasting Remote Intent FrameWork) 프레임워크를 제안한다. BRIF 프레임워크는 구글의 C2DM 서비스를 이용하여 서로 다른 안드로이드 장치 간의 메시지 전송을 비동기적으로 푸싱하는 기능을 서비스하는 프레임워크다. 이것은 기존의 로컬 기기내에서 인텐트를 사용하는 것과 동일한 방법으로 원격지에 사용자 정의 인텐트를 보내는 코드를 쉽게 작성할 수 있다.

키워드 : 애플리케이션, 프레임워크, 인텐트, 안드로이드, 비동기

1. 서 론

최근 스마트폰의 사용자가 크게 증가함에 따라 스마트폰에 대한 연구가 활발히 진행되고 있다. 특히 구글의 안드로이드 플랫폼은 매년 전 세계 스마트폰 시장에서 점유율이 늘어나고 있으며, 안드로이드 플랫폼 기반의 애플리케이션 개수도 급격히 증가하고 있다. 애플리케이션이 급격히 증가함에 따라 사용자들의 요구도 다양해지고 있다. 그러나 사용자들의 다양한 요구를 만족시키기에는 현재 안드로이드의

* 이 논문은 숭실대학교 모바일플랫폼 기반 계획 및 학습인지 모델 프레임워크 기술 개발(4년)에 의하여 연구되었음.
† 정 회 원: 한국의국어대학교 컴퓨터 및 정보통신공학과 석사과정
** 준 회 원: 한국의국어대학교 정보통신공학과 학사수료
*** 중 심 회 원: 한국의국어대학교 정보통신공학과 부교수
논문접수: 2013년 2월 26일
수 정 일: 1차 2013년 4월 11일
심사완료: 2013년 4월 11일
* Corresponding Author: Sangwon Park(swpark@hufs.ac.kr)

메시지 전송 방법으로는 한계가 있는 것들이 있다. 예를 들어 안드로이드 장치에서 보던 동영상과 다른 안드로이드 장치에서 끊김 없이 이어볼 수 있는 서비스가 있다. 또 다른 예로 안드로이드 모바일 장치가 자리마다 설치되어 있는 회의장에서 발표자가 각 자리에 있는 안드로이드 장치에 화면이나 데이터를 전달하여 자신의 의도대로 회의를 진행할 수 있는 서비스는 현재 안드로이드의 메시지 전송 방법인 인텐트(intent)만으로는 구현하기가 쉽지 않다.

안드로이드의 인텐트란 콘텐츠 메시지를 가지고 있는 객체이며 애플리케이션과 애플리케이션 컴포넌트간의 메시지 전송을 하는 곳에 이용된다. 안드로이드의 컴포넌트는 액티비티(activity), 서비스, 브로드캐스트 리시버, 콘텐츠 프로바이더(content provider)가 있다. 액티비티, 서비스, 브로드캐스트 리시버는 인텐트를 이용해 메시지를 주고받을 수 있다.

안드로이드의 액티비티는 하나의 사용자 인터페이스를 나타내는 단위이며 하나의 애플리케이션에 한 개 또는 그 이상의 액티비티로 구성된다. 서비스는 사용자 인터페이스를 가지지 않고 백그라운드(background)에서 실행되며 사용자가 다른 작업을 하는 중에도 백그라운드에서 음악을 재생하거나 데이터를 처리하는 등의 작업을 한다. 브로드캐스트 리시버는 안드로이드 장치 내에 브로드캐스팅되는 메시지를 수신하는 작업을 한다.

서로 다른 안드로이드 장치 간에 메시지를 전송하려면 현재 안드로이드 운영체제에서는 소켓(Socket)[3] 통신으로 서로 다른 안드로이드 장치를 연결해야만 장치 간에 메시지 전송이 가능하다. 소켓통신으로 서로 다른 안드로이드 장치 간에 메시지 전송을 하려면 안드로이드 장치의 애플리케이션들끼리 연결이 되어 있어야만 메시지 전송이 가능하다. 또한 연결이 되어 있어야 한다는 것은 연결을 유지하는데 있어서 불필요한 네트워크 자원낭비가 발생하기 때문에 네트워크 측면의 효율성이 떨어진다. 따라서 소켓통신이 가지는 단점을 해결하기 위해서 본 논문에서는 구글의 C2DM[2] 서비스를 활용하여 서로 다른 안드로이드 장치 간에 메시지를 전송하는 프레임워크를 제안한다.

본 논문에서 제안하는 방법은 다음과 같다. 안드로이드 장치에서 다른 안드로이드 장치로 메시지를 보낼 때 소켓통신과 같은 네트워크 프로그래밍을 하는 것은 쉽지 않은 일이다. 하지만 로컬 프로그래밍하듯이 원격지 안드로이드 장치에 메시지를 보낼 수 있다면 개발자들의 노력이 경감될 수 있을 것이다. 그래서 본 논문에서는 C2DM 서비스를 활용하여 쉬운 메시지 전송을 가능하게 하는 BRIF 프레임워크를 제안한다. 마치 로컬장치에 프로그래밍 하듯이 메시지를 쉽게 보낼 수 있다.

구글의 C2DM 서비스는 한 안드로이드 장치에서 보내는 메시지를 받아서 다른 안드로이드 장치로 메시지를 비동기적으로 푸시해주는 서비스이다. 하지만 C2DM은 정해진 인텐트에 제한된 문자열 메시지만 보낼 수 있다. 그래서 본 논문에서는 사용자 정의 인텐트를 주고받을 수 있

게 만들었다. 본 논문의 BRIF프레임워크를 통해 사용자 정의 인텐트를 주고받을 수 있다는 것은 자신이 아닌 다른 안드로이드 장치에 인텐트를 보내 브로드캐스팅하거나 서비스를 실행시킬 수 있다는 장점이 있다. 개발자는 마치 로컬에서 인텐트를 쓰듯이 다른 안드로이드 장치에서 인텐트를 원격으로 쉽게 적용할 수가 있다. 이 장점으로 인해 사용자는 자신의 안드로이드 장치에서 다른 안드로이드 장치를 마치 로컬 프로그래밍하듯이 쉽게 원격으로 제어할 수가 있는 것이다.

BRIF프레임워크를 사용하여 N-Screen 등 애플리케이션을 개발할 때 프로그램을 간단하게 작성할 수 있고 안드로이드 장치끼리의 연동을 로컬 프로그램처럼 작성할 수 있다. 또한, C2DM 서비스는 1024byte 이하의 문자열로 구성된 메시지만 푸시해주는 한계가 있다. 그래서 BRIF 프레임워크에서는 메시지 크기에 구애받지 않고 사용자 정의 인텐트를 다른 안드로이드 장치로 보낼 수 있다. 따라서 C2DM의 메시지 크기 제한과 문자열만이 아니라 사용자 정의 인텐트를 보낼 수 있게 구글 C2DM 서비스를 활용하여 비동기적으로 문자열뿐만이 아닌 메시지를 전송하고 받을 수 있는 BRIF(Broadcasting Remote Intent Framework) 프레임워크를 제안한다.

BRIF 프레임워크는 BRIF Server, 구글 C2DM 서비스, RISP 애플리케이션, RemoteContext API로 이루어져 있으며 서로 다른 안드로이드 장치 간 사용자 정의 인텐트를 이용하여 메시지 전송이 가능하다. 또한 기존의 안드로이드의 인텐트 코드 작성 방법과 BRIF 프레임워크의 사용자 정의 인텐트를 작성하는 코드는 크게 다르지 않다. 그래서 기존의 인텐트를 사용하던 개발자들에게는 서로 다른 안드로이드 장치 간 메시지 전송을 쉽게 구현할 수 있는 장점이 있다.

내용을 정리하여 보면 본 논문의 동기로 다음과 같다.

첫째, 원격지 안드로이드 장치에 메시지 전송을 할 때 로컬 프로그램에서처럼 전송할 수 있다면 원격지 안드로이드 장치에 메시지를 쉽게 전송할 수 있다.

둘째, 위에 언급한 것처럼 원격지에 쉽게 메시지 전송을 하려면 C2DM 서비스를 활용한 BRIF 프레임워크를 제안한다.

셋째, BRIF 프레임워크를 사용하면 소켓프로그래밍을 고려하지 않아도 되고 쉽게 프로그램을 만들 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 연구되던 관련연구와 기술들에 대해 기술하고 3장에서는 본 논문에서 제안하는 BRIF 프레임워크를 사용한 메시지 전송 코드를 기존의 인텐트와 비교해보고 예시코드를 보여준다. 4장에서는 본 논문에서 제안하는 BRIF 프레임워크의 전체구조와 모듈별 구조를 살펴본다. 5장에서는 BRIF 프레임워크의 정확성을 검증하기 위해 이 프레임워크에 적용한 애플리케이션의 실험을 하고 6장에서는 성능평가 결과를 기술한다. 마지막으로 7장에서는 결론을 맺고 연구 방향에 대하여 서술한다.

2. 관련 연구

본 장에서는 메시지 전송 시 서로 다른 장치들끼리 통신하는 여러 가지 통신 방법에 대한 연구를 살펴본다.

RPC(Remote Procedure Call)[5, 6, 7]는 한 프로그램이 네트워크상의 다른 컴퓨터에 위치하고 있는 프로그램에 서비스를 요청하는데 사용되는 프로토콜로서, 이때 서비스를 요청하는 프로그램은 네트워크에 대한 상세 내용을 알 필요가 없다. RPC서비스는 크게 서버, 클라이언트, 포트 맵퍼(Port Mapper)로 이루어져 있다. 서로 다른 컴퓨터가 통신을 할 때 소켓을 사용하여 다른 컴퓨터와 통신을 하는 것은 가능하지만 RPC 서비스를 쓰면 다른 컴퓨터와의 통신이 마치 로컬 함수를 호출하는 듯 사용할 수 있다는 것과 복잡한 소켓 프로그래밍을 하지 않아도 되는 것이 장점이다.

RPC는 원격지 함수 호출하기에는 안드로이드 장치에 맞지 않아 안드로이드에서는 IPC로 안드로이드 장치에 적합하게 바뀌었다. 하지만 RPC는 메시지 전송에 적합하지 않은데 안드로이드에서는 서로 다른 프로세스 간에 메시지 전송을 인텐트로 하기 때문이다. 따라서 RPC를 쓰기보다 본 논문의 BRIF프레임워크를 통해서 다른 안드로이드 장치에 메시지 전송을 할 때 인텐트를 확장해서 전송할 수가 있다.

Java RMI(Remote Method Invocation)[8, 9, 10, 11, 12]란 자바 언어로 네트워크상에서 떨어져 있는 객체의 메서드를 투명하게 호출하는 것을 말한다. Java RMI는 RPC의 단순한 원격지 프로그램의 함수 호출에 비해 원격지 객체의 메서드 호출을 가능하게 함으로써 분산객체 시스템(Distributed Object System)에 적합한 메커니즘이다.

Java RMI도 RPC와 마찬가지로 개발자가 소켓통신에 대한 고려가 필요 없고, Java RMI 관련 클래스만으로 프로그램을 만들면, Java RMI 인프라에서 네트워크와 관련된 복잡한 처리를 담당해 준다. Java RMI를 사용하면 원격지 객체의 메서드를 호출할 때 마치 같은 로컬 JVM 내의 객체 메서드를 호출하는 것과 유사한 호출이 가능하다. 호출할 때 메서드의 인자로 사용되는 것은 자바 기본형과 객체 직렬화가 가능한 모든 클래스 객체들이 전달될 수 있다. 호출했을 때의 반환 값으로 자바 기본형과 객체들이 반환될 수 있다. 따라서 개발자가 네트워크 애플리케이션을 구현할 때 Java RMI를 이용하면 소켓과 비교하여 간단하게 프로그래밍을 할 수 있다.

CORBA(Common Object Request Broker Architecture)[13, 14]는 네트워크에서 분산 프로그램 객체를 생성, 배포, 관리하기 위한 구조와 규격이다. CORBA는 800개 이상의 산업체가 참여하여 구현이 아닌 인터페이스를 정의하는 규격으로 주요 특징으로는 객체 위치 투명성, 연결 및 메모리 관리 등 여러 병행성 및 보안 기능을 제공한다. 따라서 CORBA는 언어, 운영체제, 하드웨어 및 네트워크 프로토콜의 이질적인 의존성으로부터 독립적인 응용의 구현할 수 있는 인터페이스를 제공한다.

구글 C2DM(Android Cloud to Device Messaging) 서비스는 안드로이드 버전2.2부터 지원되는 기능이다. 기존의 안드로이드 장치의 애플리케이션들은 주기적으로 데이터를 가져오기 위해 폴링방식이나 소켓통신을 사용하였다. 그러나 구글 C2DM은 안드로이드 마켓 애플리케이션이 사용하는 구글 서비스를 이용하기에 마켓 애플리케이션이 설치되어 있는 모든 안드로이드 장치에 비동기적으로 메시지 전송을 한다. 구글 C2DM 서비스의 단점은 1024Byte 이하의 메시지 전송만 가능하다는 것과 구글이 송신자가 보낼 수 있는 총 메시지 수를 제한한다는 것이다.

3. BRIF 프레임워크를 사용한 메시지 전송

본 장에서는 BRIF 프레임워크를 사용해서 인텐트 전송을 하는 것이 기존의 안드로이드 장치의 인텐트 코드를 작성할 때와 유사해서 개발자가 쉽게 적용할 수 있음을 보여주고, 소켓통신으로 메시지 전송을 할 때보다 BRIF 프레임워크가 효율적이라는 것을 기술한다.

서로 다른 안드로이드 장치 간에 인텐트 전송을 소켓통신으로 하려면 안드로이드의 서버소켓 클래스와 클라이언트소켓 클래스를 구현해야 한다. 소켓통신은 서버-클라이언트 모델이기에 서로 다른 안드로이드 장치 간에 메시지 전송을 하려면 각 안드로이드 장치에 서버소켓을 이용한 서버구현과 클라이언트소켓을 이용한 클라이언트 구현이 필요하다. 서버소켓과 클라이언트소켓을 이용하여 소켓통신을 구현할 시 메시지를 보내는 쪽이 클라이언트를 실행시켜 다른 안드로이드 장치의 서버에 접속하는 방식이다.

소켓통신을 사용하는 방법은 메시지 전송을 할 때 소켓 연결 방법에 대해 고려해야 한다. 안드로이드 장치에서 메시지 전송을 할 때마다 연결을 할 것인지 아니면 연결을 유지하고 있을지의 방식을 선택해야 한다. 또한 연결을 유지할 때 메시지 전송을 위해 연결을 유지하고 있는 방식은 불필요한 네트워크 자원 낭비가 발생한다. 메시지 전송을 할 때마다 연결을 하는 방식은 용량이 작은 메시지를 전송할 때 메시지 크기보다 연결할 때 드는 네트워크 자원이 더 많이 드는 낭비가 생긴다.

```
Intent intent = new Intent();
intent.setAction("dislab.intent.action.PLAY_VIDEO");
intent.putExtra("file", "movie.mp4");
intent.putExtra("currentFrame", currentPosition);
startService(intent);
sendBroadcast(intent);
```

Fig. 1. Use Intent in android

BRIF 프레임워크를 사용하면 이러한 네트워크 낭비를 줄일 수가 있다. BRIF 프레임워크는 구글의 C2DM 서비스를

```

RemoteContextManager mgr =
RemoteContextManager.getManager();
Mgr.setDevice("010-123-4567");
Context context = mgr.getContext();
Intent intent = new Intent();
intent.setAction("dislab.intent.action.PLAY_VIDEO");
intent.putExtra("file", "movie.mp4");
intent.putExtra("currentFrame",
currentPosition);
context.startService(intent);
context.sendBroadcast(intent);
    
```

Fig. 2. Transfer message with BRIF FrameWork

이용하여 서로 다른 안드로이드 장치 간에 비동기적으로 인텐트를 전송하기 때문에 연결이 유지되어 있지 않아도 된다. 또한 서로 다른 안드로이드 장치 간에 인텐트 전송을 위해서 연결 작업이 필요 없다. 구글의 C2DM 서비스가 안드로이드 장치로 연결을 담당하기 때문에 개발자는 연결에 대해 신경 쓸 필요가 없어진다. 개발자가 서로 다른 안드로이드 장치 간에 메시지 전송을 하려면 인텐트를 만들어 RemoteContext 클래스 코드를 작성하면 인텐트 전송을 위한 준비는 끝나게 된다. RemoteContext는 서로 다른 안드로이드 장치 간에 인텐트를 주고받을 수 있게 할 수 있도록 본 논문에서 제안하는 BRIF 프레임워크의 메시지 전송 방식을 구현한 클래스이다.

안드로이드의 인텐트는 수행될 액션과 데이터, 엑스트라, 플래그 등의 내용을 포함하고 있다. 액션은 수행될 액션을 지정하는 문자열 상수이며 안드로이드의 인텐트 클래스는 정의되어 있는 많은 액션 상수를 가지고 있고 개발자의 맞춤형 액션 문자열을 정의할 수도 있다. 데이터는 액션들과 한 쌍이 되는 처리되어야 할 데이터에 대한 URI다. 예를

들어 액션 문자열이 ACTION_CALL 이라는 정의되어 있는 문자열이면 데이터는 통화번호를 가진 tel:URI가 된다. 엑스트라란 추가정보에 대한 키/값 쌍으로 이루어진 데이터이다. putExtras() 메서드와 getExtras() 메서드를 사용하여 인텐트의 추가정보로 넣고 뺄 수 있다. RemoteContext는 안드로이드의 인텐트를 다른 안드로이드 장치 내로 인텐트를 전송할 수 있게 해주며 기존의 인텐트를 사용했던 개발자라면 쉽게 쓸 수가 있다. 안드로이드의 기존 인텐트 작성 코드와 다른 안드로이드 장치에 인텐트를 보내는 RemoteContext 클래스를 이용한 코드 작성은 크게 다르지가 않고 추가되는 코드를 Fig. 1, 2에서 보여주고 있다.

Fig. 1, 2에서 나타난 인텐트와 RemoteContext의 코드는 dislab.intent.action.PLAY_VIDEO 라는 커스텀 액션 문자열 상수를 정의했고 엑스트라 데이터로 file/movie.mp4, currentFrame/currentPosition을 추가하여 인텐트를 만들었다. 기존의 인텐트는 startService 메서드를 호출할 때 로컬 기기에 있는 서비스를 실행시키고 sendBroadcast 메서드는 로컬 기기내에 인텐트를 브로드캐스트 하여 액션 문자열 상수를 수신하고 있는 브로드캐스트 리시버에 전달하는 것이다. BRIF 프레임워크의 RemoteContext는 기존의 인텐트를 만드는 과정은 같지만 startService 메서드를 호출하면 하나의 다른 안드로이드 장치에 인텐트를 전송하여 다른 안드로이드 장치 내에 서비스를 실행하는 것이고 sendBroadcast 메서드를 호출하면 다른 안드로이드 장치에 인텐트를 전송하여 다른 안드로이드 장치 내에 인텐트를 브로드캐스트하는 것이다.

사용 예시에서 보듯이 인텐트와 RemoteContext는 코드 작성 방법에는 크게 차이가 없다. 그러므로 인텐트를 사용하는데 익숙한 애플리케이션 개발자들은 BRIF 프레임워크를 이용하여 쉽게 서로 다른 안드로이드 장치 간의 인텐트 전달을 할 수 있게 된다.

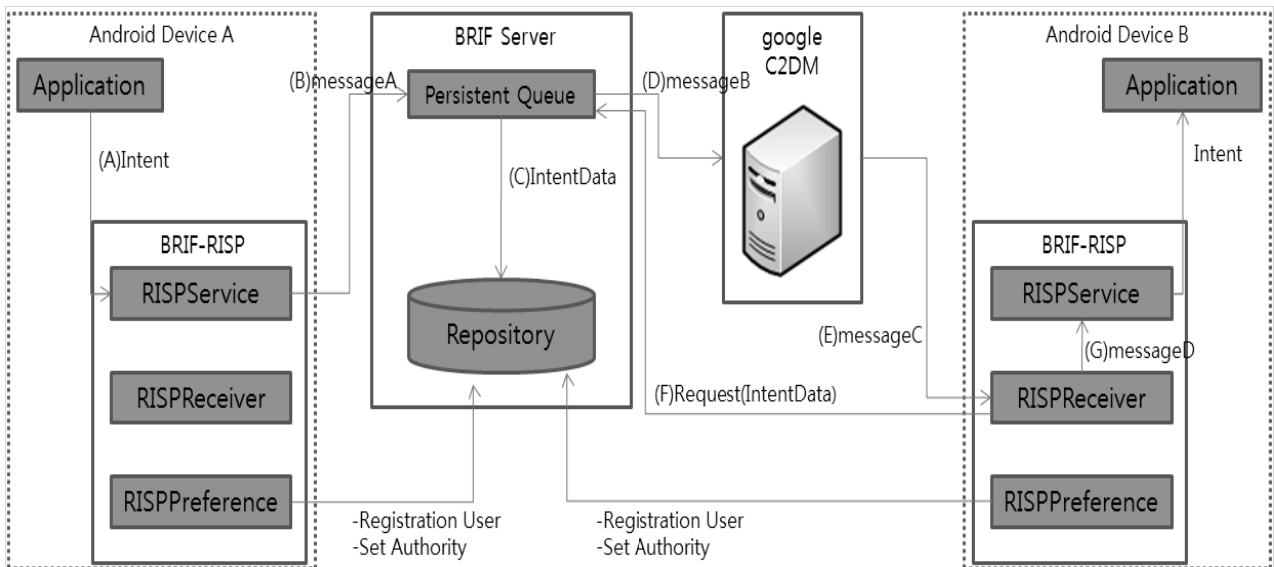


Fig. 3. BRIF FrameWork structure

4. BRIF 프레임워크 구조

본 장에서는 BRIF 프레임워크의 구조와 각 모듈들에 대해서 설명한다. BRIF 프레임워크는 RemoteContext API와 안드로이드 장치 내에 설치되는 RISP 애플리케이션과 BRIF Server, 구글 C2DM 서비스이다. BRIF 프레임워크의 간단한 동작과정을 살펴보고자 하겠다. 먼저 안드로이드 장치 A에서 BRIF 프레임워크를 사용하는 애플리케이션이 RemoteContext API를 통하여 만든 마샬링된 인텐트(A)를 RISP 애플리케이션의 RISPService에게 보낸다. RISPService는 안드로이드의 서비스를 상속받은 클래스로 애플리케이션에서 넘어온 인텐트를 메시지 A(B)에 담아 BRIF Server에게 보낸다. 메시지 A에는 마샬링된 인텐트와 마샬링된 인텐트로 만든 UUID, 자신의 안드로이드 장치 ID, C2DM 등록 ID, 그리고 스마트폰이라든 전화번호가 담겨 있다. C2DM 등록 ID는 RemotePreference에서 안드로이드 장치를 C2DM 서비스를 이용할 수 있게 등록하여 구글로부터 발급받는 고유한 ID이다.

BRIF Server의 Persistent Queue는 받은 메시지 A를 해석하여 이 메시지 A의 인텐트를 받을 안드로이드 장치에게 메시지를 보내도록 구글 C2DM 서버에 메시지 B를 보낸다. 메시지 B(C)에는 메시지를 받을 안드로이드 장치의 등록ID와 인텐트중에 UUID만 담겨 있다.

구글 C2DM 서버는 받은 메시지 B의 등록ID를 보고 해당하는 안드로이드 장치에 메시지 C(E)를 보낸다. 안드로이드 장치B의 RISPReceiver는 안드로이드 장치 A에서 보낸 마샬링된 인텐트의 UUID를 구글 C2DM 서버에게서 받는다. 받은 UUID를 이용하여 BRIF Server에게 마샬링된 인텐트 데이터를 요청한다(F). BRIF Server에게 마샬링된 인텐트 데이터를 받은 후 RISPService에게 인텐트 데이터를 전달한다(G). 마지막으로 RISPService는 마샬링된 인텐트 데이터를 언마샬링하여 원래의 인텐트로 만들어 해당 애플리케이션이나 애플리케이션 내부에 데이터를 전달한다. 다

음 장에서 BRIF 프레임워크의 각 모듈별로 구분하여 설명 하겠다.

4.1 BRIF 프레임워크의 RemoteContext API 구조

Fig. 4의 RemoteContext API는 안드로이드의 Context와 안드로이드의 인텐트를 상속받는 RemoteContext 클래스와 RemoteContextManager 클래스로 구성되어 있다.

RemoteContextManager에서 remoteDeviceUserId와 remoteDevicePhoneNum은 원격지 안드로이드 장치의 userId와 phoneNum을 각각 setPhoneNumberOfRemoteDevice함수와 setRemoteDeviceUserId함수에서 처리된다. getContext함수는 현재 실행되고 있는 애플리케이션 액티비티의 context를 받아와 RemoteContext클래스에 전달한다.

RemoteContext는 안드로이드의 인텐트를 상속받으며 startService함수를 통해 원격지 안드로이드 장치에 메시지 전송을 시작한다. startService의 인자로 들어온 인텐트를 마샬링하여 byte[]로 바꾼 인텐트 데이터를 원격지 안드로이드 장치에 보낸다. userRegister는 애플리케이션 사용자가 처음 자신의 id를 생성하는 함수이다. 애플리케이션 사용자의 전화번호, userId, 비밀번호, 상태 말, 그리고 웹 서버의 역할을 하는 URL을 인자로 넣어 웹 서버로 정보를 보낸다. 웹 서버는 정보를 저장하고 존재하는 id는 리턴 값 -1을 보내고 성공적으로 저장되었다면 리턴 값 1을 보내게 된다.

4.2 BRIF 프레임워크의 RISP(Remote Intent Service Provider)

BRIF-RISP 애플리케이션은 인텐트를 다른 안드로이드 장치에 전송하거나 받을 수 있는 애플리케이션이다. RISP 애플리케이션의 구조는 개발자가 만든 인텐트를 받아서 BRIF-Server에 보내는 RISPService 클래스와 구글 C2DM 서버에서 보내는 메시지를 받는 RISPReceiver 클래스가 있고 RISP 애플리케이션의 사용자 관리와 설정을 담당하는 RISPPreference 클래스로 구성된다.

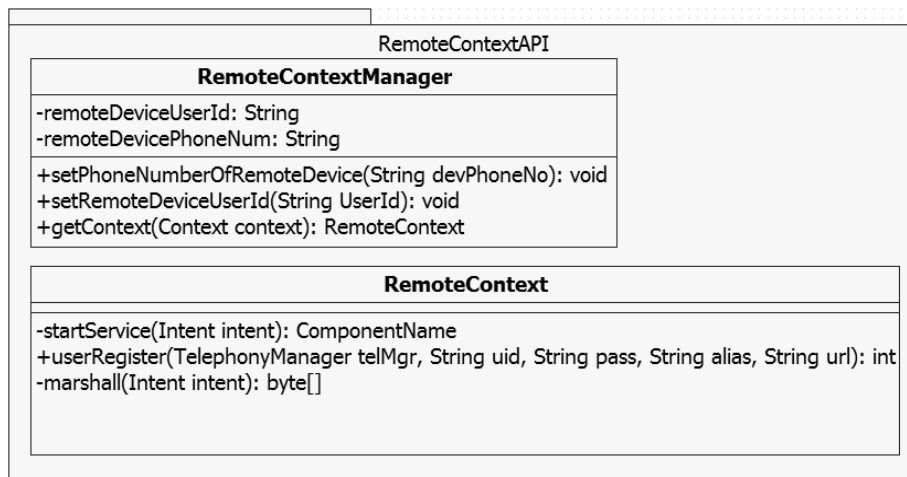


Fig. 4. RemoteContextAPI Package

RISPService 클래스는 안드로이드 Service 클래스를 상속받으며 안드로이드 장치에서 메시지를 송신할 때와 수신할 때로 역할이 나뉜다. 안드로이드 장치에서 메시지를 송신할 때는 애플리케이션 개발자가 만든 마샬링된 인텐트 데이터가 RISPService에게 넘어오면 HTTP POST 방식으로 BRIF Server에 전송한다.

메시지를 수신할 때의 안드로이드 장치에서는 RISP Receiver에서 보내온 마샬링된 인텐트 데이터를 언마샬링하여 안드로이드의 인텐트로 만든다. RISPService에 의해 만들어진 인텐트는 개발자가 만든 인텐트의 액션과 엑스트라 데이터 등을 모두 담고 있는 개발자가 만든 보내기 전의 인텐트와 똑같다. RISPService는 만들어진 인텐트를 실행하게 되고 안드로이드 장치의 애플리케이션이나 애플리케이션 내부를 활성화 시키거나 데이터가 전달되며 서로 다른 안드로이드 장치 간에 메시지 전송이 완료된다.

RISPReceiver는 안드로이드 서비스 중 하나인 Broadcast Receiver 클래스를 상속받으며 구글 C2DM 서버에서 보내오는 메시지에 수신대기하고 있다. 구글 C2DM 서버에서 메시지 C를 보내면 RISPReceiver로 들어오게 되고 메시지 C의 UUID를 이용하여 BRIF Server에 마샬링된 인텐트 데이터를 요청하여 받은 후 RISPService로 마샬링된 인텐트 데이터를 전달한다.

RISPPreference 클래스는 BRIF 프레임워크를 사용하는 애플리케이션 간에 메시지를 전송할 수 있는 권한을 설정한다. 안드로이드 장치 사용자의 장치에 BRIF 프레임워크를 사용하는 애플리케이션들이 설치되어 있을 때 애플리케이션별로 메시지 전송을 수락 또는 거절할 수 있게 등록 또는 해제를 하는 것이다. 등록 또는 해제를 하는 방법은 설치되어 있는 애플리케이션별로 메시지 수신 여부를 결정하고 또한 특정 사용자의 메시지 수신 여부를 결정하도록 한다. 사용자가 수신 여부를 결정하면 RISPPreference는 BRIF Server에 메시지 수신 여부를 등록하게 된다. 수신 여부를 등록할 때는 애플리케이션별로 또는 사용자별로 등록할 수 있다. 먼저 메시지를 수신할 사용자를 등록하고 그 사용자들에게서 메시지를 수신할 애플리케이션을 결정하여 BRIF Server에 사용자 전화번호나 등록ID를 자신의 메시지 권한을 BRIF Server의 Repository에 등록한다. 이렇게 BRIF Server에 등록된 수신 여부 목록은 웹 서버에 BRIF 메시지가 도착했을 때 등록된 수신 목록을 보고 특정 사용자에게 메시지를 보낼지 안 보낼지 결정할 수가 있다.

BRIF 프레임워크의 RISP 애플리케이션은 이 세 가지 RISPService, RISPReceiver, RISPPreference 가 메시지 수신에 따른 반응을 보이며 동작한다. BRIF 프레임워크를 사용하기 위해서는 사용자의 안드로이드 장치에 반드시 RISP 애플리케이션이 설치되어 있어야 한다. 그래야만 BRIF 프레임워크를 사용하는 애플리케이션을 이용할 수 있다.

4.3 BRIF Server

RISP 애플리케이션이 인텐트를 다른 안드로이드 장치에 보내기 위해서는 구글의 C2DM서버와 또 하나의 웹 서버가

필요하다. 이 웹 서버는 안드로이드 장치 내의 RISP애플리케이션이 보내는 인텐트를 받아서 구글 C2DM 서버로 다시 송신하는 역할을 한다. 본 논문에서 BRIF 프레임워크에 사용될 웹 서버는 플랫폼 구축을 쉽게 하기 위해 구글 앱 엔진[15]을 사용하였고 이름을 BRIF Server라 부른다. 구글 앱 엔진을 사용한 이유는 쉽게 서버를 만들 수 있으며 서버 관리를 웹 페이지로 쉽게 할 수 있다는 장점이 있고 유지보수를 구글에서 하여 편하기 때문이다. 따라서 본 논문에서 사용될 BRIF-Server는 구글 앱 엔진을 사용해 만들었다.

BRIF Server를 구축하기 전에 먼저 구글 C2DM 서비스를 사용하기 위해서 구글 C2DM 서비스 신청양식을 구글 C2DM 서버에 보내야 한다. 신청양식에는 메일주소, 비밀번호, 네트워크 사용량 등을 작성해야 하며 작성된 신청양식을 구글에 보내 구글C2DM 서버 사용이 허가를 받는다. 허가를 받으면 BRIF 프레임워크의 BRIF-Server에서 구글의 C2DM 서버로 메시지를 보내기 위해 third_party 서버라는 것을 인증받아야 한다. third-party 서버라는 것을 인증받는 것은 C2DM 서버에 HTTP 요청을 보내는 것으로 이루어진다. 인증하는 방법은 미리 신청했던 구글 서비스 신청 양식에 작성했던 본 논문 저자의 메일주소, 메일 비밀번호, C2DM 서비스명, 서버 버전을 C2DM 서버에 전송하게 되면 C2DM 서버가 확인하여 인증키 값을 BRIF Server에 보내준다. 웹 서버는 인증키 값을 받아서 서버 Repository에 저장해서 안드로이드 장치에 메시지를 보낼 때 인증키 값을 불러와 구글 C2DM 서버에 보내는 메시지 헤더에 첨부해서 보낸다. 구글 C2DM 서버는 메시지 헤더에 인증키 값을 보고 인증 여부를 판단하여 인증된 third-party 서버에서 보낸 메시지가면 메시지를 안드로이드 장치에 보내게 된다.

BRIF 프레임워크에서 활용하는 구글 C2DM 서버는 대량의 메시지에 적합하게 설계되지 않았기에(트래픽 초과로 인해 발생하는 문제로 인한 설계) 1024Byte 미만의 메시지만 C2DM 서버에 보낼 수 있다. 따라서 애플리케이션 개발자가 만든 인텐트가 1024Byte를 넘을 경우 문제가 발생한다. 구글C2DM의 메시지 전송 용량 한계에 걸려 전송이 되지 않기 때문이다. 그래서 본 논문에서는 메시지 크기 제한을 해결하기 위해 BRIF Server의 Repository와 Persistent Queue를 이용하여 구글 C2DM 서비스의 메시지 크기 제한을 해결했다. 만약 애플리케이션 개발자가 만드는 인텐트가 1024Byte가 넘을 경우 BRIF Server는 안드로이드 장치에서 받은 메시지의 인텐트 데이터 부분을 Repository에 저장한다.

BRIF Server의 Repository에 저장되면 BRIF Server는 구글C2DM 서버에게 인텐트 데이터가 데이터베이스에 저장되어 있다는 내용을 담은 메시지를 보낸다. 구글C2DM 서버로부터 이 메시지를 받은 안드로이드 장치의 RISPReceiver는 BRIF 서버에게 UUID를 이용하여 마샬링된 인텐트 데이터를 요청한다. 요청을 받은 BRIF Server의 Queue는 요청된 UUID를 가진 마샬링된 인텐트 데이터를 RISPReceiver에게 전송한다.

인텐트를 BRIF 서버에게서 전송받으면 RISPReceiver는 BRIF Server에 인텐트를 받았다는 ACK 메시지를 보내게 되고 도중에 연결이 끊겨 데이터를 받지 못했다면 BRIF Server에게 UUID를 이용하여 마살링된 인텐트 데이터를 다시 요청하게 된다. BRIF Server에서 안드로이드 장치의 RISPService로 마살링된 인텐트 데이터가 완전하게 전송이 됐다면 ACK 메시지를 BRIF Server에게 보낸다. ACK 메시지를 받은 BRIF Server는 Repository에서 인텐트 데이터를 삭제하여 데이터베이스에 불필요한 데이터가 남지 않게 한다.

마지막으로 RISPReceiver는 RISPService에게 마살링된 인텐트를 전달하며 모든 과정이 완료된다.

또한 BRIF 프레임워크의 BRIF Server는 구글 C2DM에 메시지를 보내는 역할을 할뿐만 아니라 사용자들이 RISP 애플리케이션을 통해 수신 여부를 등록하는 안드로이드 장치 간, 애플리케이션 간 메시지 수신 목록을 데이터베이스화하여 가지고 있다. BRIF Server는 사용자 별로 등록해놓은 수신 여부에 따라 안드로이드 장치에게서 받은 인텐트의 송신 여부를 결정한다. 웹 서버의 수신 목록에는 사용자가 BRIF 프레임워크를 사용하는 애플리케이션별로 자신의 안드로이드 장치에 메시지를 수신할 것인가와 특정 사용자의 메시지 수신여부를 등록할 수 있다. BRIF Server에서 사용자의 수신 목록을 저장하는 방법은 특정 사용자를 기기ID와 C2DM 등록ID, 사용자 전화번호로 구분한다. 이렇게 특정 사용자를 구분하면 사용자들은 친구의 전화번호를 등록하여 친구에게만 메시지를 수신하거나 송신할 수 있다. 또한 애플리케이션별로 수신 여부를 지정하여 어떤 애플리케이션은 모든 사용자의 메시지를 수신하게 만들 수 있다.

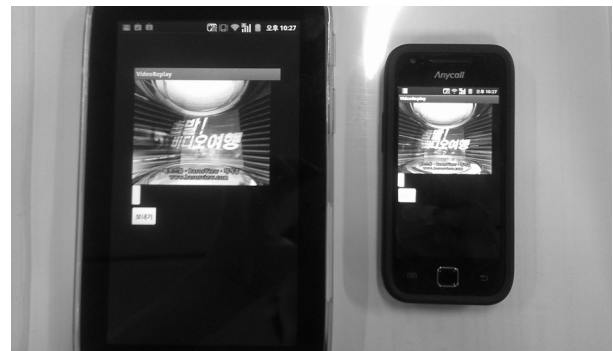
5. BRIF 프레임워크를 사용한 애플리케이션

본 논문에서 제안한 BRIF 프레임워크는 서로 다른 안드로이드 장치 간에 메시지 전송을 위한 프레임워크이며 애플리케이션 설계자는 제안한 BRIF 프레임워크를 기반으로 애플리케이션을 구현한다. 제안한 프레임워크의 검증은 위해 서로 다른 안드로이드 장치 간의 동영상 이어보기 애플리케이션으로 활용 가능성을 검증한다. 동영상 이어보기 애플리케이션은 한 안드로이드 장치에서 보던 동영상 위치를 다른 안드로이드 장치에서 보내 이어 볼 수 있는 애플리케이션으로서 본 논문에서 제안하는 BRIF 프레임워크를 사용한 메시지 전송을 나타내기 위해 애플리케이션을 만들었다.

비디오 이어 보기 애플리케이션의 구현 및 검증 환경은 다음과 같다. RemoteContext API를 애플리케이션 개발자의 프로젝트에 외부 라이브러리로 등록하였고 개발 대상 플랫폼은 Android2.2버전이 탑재된 갤럭시U 스마트폰과 Android 2.2버전이 탑재된 갤럭시 Tab 태블릿 pc로 실험하였다. 또한 두 개의 안드로이드 장치에 테스트 애플리케이션인 RISP애플리케이션을 설치하였다.



(a)



(b)

Fig. 5. Example screen of Video application

애플리케이션 개발자가 개발하던 프로젝트에 Remote Context API를 프로젝트의 외부 라이브러리에 등록하면 BRIF 프레임워크의 인텐트를 사용할 수 있다. Remote Context API에는 RemoteContext 클래스와 RemoteContext Manager 클래스가 있다. 안드로이드 장치의 SdCard 내에 동일한 샘플 동영상 파일을 넣어놓고 비디오 이어 보기 애플리케이션을 실행하였다.

비디오 이어보기 애플리케이션은 한 안드로이드 장치에서 동영상을 보다가 다른 안드로이드 장치에서 이어 볼 수 있게 만든 애플리케이션이다.

비디오 이어 보기 애플리케이션은 보내기 버튼을 누르면 다른 안드로이드 장치에 인텐트가 보내지도록 애플리케이션을 구현하였다. 애플리케이션의 보내기 버튼을 누르면 인텐트를 다른 안드로이드 장치의 RISPReceiver에게 전달되며 일련의 과정을 통해 인텐트를 받은 RISPService가 인텐트를 언마살링하여 원래의 인텐트로 변환하여 실행한다. Start Service로 인텐트를 실행하면 비디오 이어 보기 애플리케이션을 실행하여 엑스트라 데이터에 있는 키/값을 통하여 다른 안드로이드 장치에서 보내기 버튼을 누르기 전까지 보던 위치부터 동영상을 재생하게 된다. Fig. 5의 (a)는 갤럭시U에서 장치에서 재생되는 동영상 위치를 보내기 전 화면이고 (b)는 갤럭시U에서 보낸 동영상의 마지막 위치가 갤럭시 Tab에 전송되어 동영상이 재생되고 있다. 이 실험을 토대로 BRIF 프레임워크가 잘 동작한다는 것을 보여주었고 있다.

6. 성능 평가

본 논문에서 제안하는 BRIF 프레임워크는 기존의 인텐트를 다른 기기에도 쉽게 보낼 수 있게 만든 편의성을 증진시킨 프레임워크이다. 따라서 구글의 C2DM 서비스가 가지는 메시지 크기에 따른 성능과 초당 메시지 개수에 따른 성능을 평가하였다.

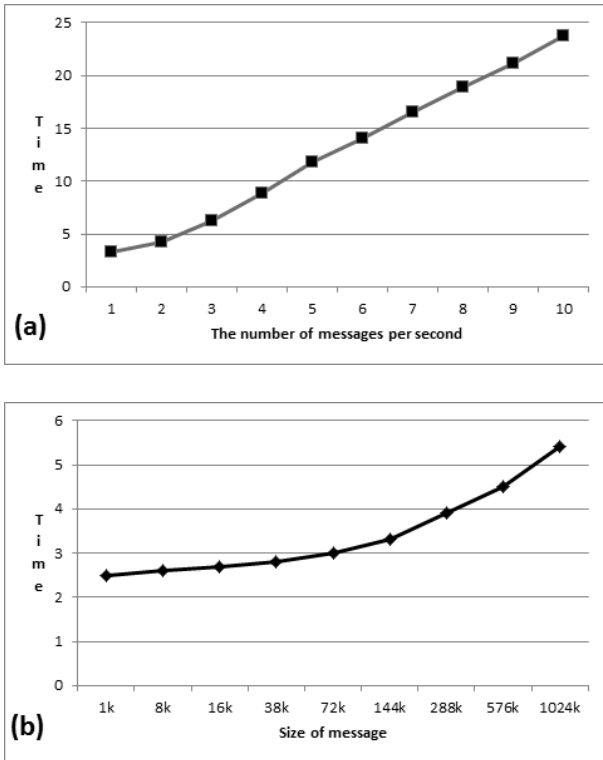


Fig. 6. Message from Android device time, until the arrival of the other Android devices

BRIF 프레임워크를 사용할 때의 성능이 Fig. 6에 나와 있다. Fig. 6은 (a)에서 초당 메시지 개수를 늘려가면서 안드로이드 장치에서 메시지를 보내 다른 안드로이드 장치에도 도착하는 시간을 측정하고, (b)에서는 한 메시지의 크기를 증가시키면서 안드로이드 장치에서 보낸 메시지가 다른 안드로이드 장치에 도착하는 시간을 측정하였다. BRIF 프레임워크의 성능을 측정하는데 쓰인 안드로이드 장치의 하드웨어 스펙은 Table 1에 나타나 있다.

Fig. 6의 (a)에서 보듯이 본 논문에서 제안하는 BRIF 프레임워크를 사용하여 메시지 전송을 할 때 초당 메시지 개수가 증가해도 메시지 개수 당 처리시간이 증가하지 않는 것을 알 수 있다. 또한 (b)에서는 메시지 크기가 증가함에 따라 측정된 처리시간도 크게 차이나지 않는 것을 알 수 있다. 하지만 BRIF 프레임워크는 메시지 크기가 작아도 메시지 하나가 처리되는 시간은 평균 3초가 걸린다. 하나의 메시지를 처리하는 전체의 시간이 평균 3초이고 구간별로 나누어 보면 C2DM 서비스가 안드로이드 장치로 메시지를 푸

Table 1. Specifications of the Android devices that are used to measure performance

Device name	SAMSUNG galaxyU	SAMSUNG galaxyTab
network	WI-FI	WI-FI
OS	Android 2.2	Android 2.2
CPU	S5PC111 1GHz	SP5C110 1GHz
Memory	512MB	512MB

시하는 시간이 평균 2.2초가 걸리고, C2DM 서비스로부터 메시지를 받은 후 BRIF Server에게서 인텐트 데이터를 받아오는 데는 평균 1초의 시간이 걸린다. C2DM 서비스가 메시지를 전달하는 시간이 평균 2초정도 걸리기에 메시지 하나의 처리시간이 2초 이하로 처리되는 않고 있다.

본 논문의 BRIF 프레임워크의 성능을 높이려면 C2DM의 푸시 서비스에 의존하지 않고 메시지를 푸시하는 방법을 찾아야 하지만 현재의 안드로이드 운영체제에서는 구글의 C2DM 서비스의 메시지 처리속도에 의존적일 수밖에 없다. BRIF 프레임워크의 성능은 구글의 C2DM 서비스에 의존적이지만 안드로이드 장치 간 메시지를 주고받는 일반적인 애플리케이션에서 충분히 쓸 수 있으며 작은 크기의 메시지를 주고받는 애플리케이션에 적합하다. 또한 BRIF 프레임워크는 로컬기기에서 인텐트를 주고받듯이 서로 다른 안드로이드 장치 간에 인텐트를 주고받기에 개발자들의 노력이 경감된다.

보통 안드로이드 장치 간 메시지를 주고받을 때 소켓통신을 사용하는데 소켓통신은 안드로이드 장치 간에 연결을 필요로 하며 크기가 큰 메시지를 주고받을 때 적합하다. 그리고 소켓통신은 메시지를 주고받을 때 상시 연결이 되어 있어야 하며 소켓통신을 할 때 연결에 드는 시간을 감안하면 연결시간이 필요 없는 BRIF 프레임워크는 소켓통신보다 효율적이다.

BRIF 프레임워크는 비동기적으로 메시지를 푸시하는 C2DM서비스를 사용하기에 C2DM 서비스로부터 메시지를 받을 때 메시지를 받는 안드로이드 장치가 웹 서버에 접속하여 인텐트 데이터를 받아오면 된다. 그래서 BRIF 프레임워크는 보내는 쪽의 안드로이드 장치가 웹 서버에 메시지를 올리는 것까지 고려해야 하기 때문에 크기가 작지만 많은 개수의 메시지를 다루기에 적합하다. 또한 BRIF 프레임워크는 안드로이드 장치에서 서비스로 동작하기에 사용자가 일일이 신경 쓸 필요 없이 백그라운드에서 동작하는 애플리케이션에 적합하다.

또한 BRIF 프레임워크를 사용한 애플리케이션은 크기가 작은 여러 메시지를 주고받기에 적합하다. 또한 안드로이드 장치 간에 인텐트를 주고받기에 개발자들의 노력이 경감된다. 그리고 소켓통신을 할 때보다 두 장치 간의 연결이 필요 없는 비동기방식으로 메시지를 주고받기에 상시 연결이 필요한 소켓통신보다 효율적이다. 이러한 결과를 볼 때 본

논문에서 제안하는 BRIF 프레임워크는 여러 방향으로 다양하게 사용될 수 있을 것이다.

7. 결 론

본 논문은 서로 다른 안드로이드 장치의 메시지 전송을 위한 BRIF 프레임워크를 제안한다. 제안하는 프레임워크는 서로 다른 안드로이드 장치 간에 메시지 전송 단위를 BRIF 메시지로 정하고 RISP 애플리케이션, BRIF-Server, 구글 C2DM 서비스로 모듈이 구분된다. 애플리케이션 개발자가 만든 BRIF 메시지를 RISP 애플리케이션과 BRIF-Server 그리고 비동기적으로 메시지를 보낼 수 있는 구글 C2DM 서비스를 이용한 메시지 전송 구조이다. BRIF 프레임워크는 구글 C2DM 서비스를 이용하여 비동기적으로 메시지 크기에 제한 없이 메시지를 전송한다. 또한 로컬 기기내에 메시지 전송을 하기 위해 쓰인 인텐트를 다른 안드로이드 장치로 메시지 전송을 할 수 있기에 재사용성을 높였다. 마지막으로 구글 C2DM 서비스를 활용하여 비동기적으로 메시지를 전송하기 때문에 소켓 프로그래밍을 하여 네트워크를 구성할 필요가 없기에 연결성을 고려하지 않아도 되는 애플리케이션 개발을 할 수가 있다.

본 논문에서 제안한 BRIF 프레임워크는 비디오 이어보기 애플리케이션에 적용하여 메시지 전송의 활용성을 검증하였고 서로 다른 안드로이드 장치 간에 메시지 전송이 요구되는 다른 안드로이드 애플리케이션에 응용 가능하다. 향후 연구로는 서로 다른 안드로이드 장치 간에 메시지 전송이 아닌 다른 모바일 플랫폼 장치 간에 메시지 전송이 가능하도록 BRIF 프레임워크를 확장하는 것이다. 인텐트는 안드로이드 서비스이므로 메시지 전송 단위를 바꾸고 모바일 플랫폼 중 안드로이드와 iOS 간에 메시지 전송이 가능하도록 확장해야 할 것이다.

참 고 문 헌

- [1] D. Chaum, "Blind Signatures for Untraceable Payments," *Advances in Cryptology-Proceeding of Crypto'82*, Springer-Verlag, pp.199-204, 1982.
- [2] Google C2DM : <https://developers.google.com/android/c2dm/>
- [3] V. Getov, G. von Laszewski, M. Philippsen, and I. Foster, "Multi paradigm Communications in Java for Grid Computing", *Communications of the ACM*, Vol.44, No.10, pp.118-125, 2001.
- [4] A.D Birrel, B.J. Nelson, "Implementing Remote Procedure Call", *ACM Trans. On Computer System*, Vol.2, NO.1, pp.39-59, Feb., 1984.
- [5] John Bloomer, *Power Programming with RPC*, O'Reilly & Associates, 1992
- [6] R. Shrinivasan, "RPC:Remote Procedure Call Protocol

Specification Version 2", RFC 1831, August, 1995.

- [7] Sun Microsystems. RMI specification, available at <http://java.sun.com/products/jdk/rmi/>
- [8] Insop S., Karray F., Guedea, F., A Distributed realtime system framework design for multi-robot cooperative systems using real-time Corba, *IEEE International Symposium on Intelligent Control*. pp.793-798, 2003.
- [9] Rafal Metkowsk, Piotr Bala, *Parallel Computing in Java: Looking for the Most Effective RMI Implementation for Clustres*, *Lecture Notes in Computer Science 3911*, pp.272-277, Springer-Verlag Berlin, 2006.
- [10] C. Nester, M. Philippsen, and B. Haumacher, "A more efficient RMI for java", In *Proc. of the ACM Java Grande Conference*, pp.152-159, June, 1999.
- [11] Seung-Jun Bang, Jin-Ho Ahn, "Implementation and Performance Evaluation of Socket and RMI based Java Message Passing Systems", *KSI, Vol.8, No.5*, pp.11-20, Oct., 2007.
- [12] Steinke and Bernd, "High definition multimedia display architecture for tiny mobile Smartphones," In *MobiMedia '07: Proceedings of the 3rd international conference on Mobile multimedia communications*, pp.1-4, Aug., 2007.
- [13] Object Management Group, *Realtime CORBA Joint Revised Submission*, *OMG Document orbos/99-02-12 ed.*, March, 1999.
- [14] OMG, *The Common Object Broker: Architecture and Specification, Revision 2.3*, *OMG Document Formal/98-12-01*, June, 1999.
- [15] Google.com Developer's Guide : <http://code.google.com/appengine/docs/whatisgoogleappengine.html>

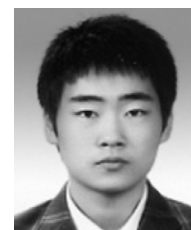


백 지 훈

e-mail : lacidjun@hufs.ac.kr

2004년 한국외국어대학교 정보통신공학부 (학사)

2011년~현 재 한국외국어대학교 컴퓨터 및 정보통신공학과 석사과정
관심분야: 모바일 컴퓨팅, 데이터베이스, 플래시 메모리



남 용 우

e-mail : skadyddn@hufs.ac.kr

2003년 한국외국어대학교 정보통신공학과 (학사)

관심분야: 모바일 컴퓨팅, 데이터베이스, 플래시 메모리



박 상 원

e-mail : swpark@hufs.ac.kr

1994년 서울대학교 컴퓨터공학과(학사)

1997년 서울대학교 컴퓨터공학과(석사)

2002년 서울대학교 컴퓨터공학과(박사)

2002년~2003년 세종사이버대학교

디지털콘텐츠학과 전임강사

2010년~현재 한국외국어대학교 정보통신공학과 부교수

관심분야: 모바일 컴퓨팅, 데이터베이스, 플래시 메모리