
XMPP 프로토콜기반 안드로이드 휴대폰 정보 푸쉬 시스템

편도길* · 유호** · 정회경***

Android mobile phone information push system based on the XMPP protocol

Do-Kil Pyoun* · Liu Hao** · Hoe-Kyung Jung***

요 약

본 논문에서는 XMPP(eXtensible Message and Presence Protocol) 프로토콜 기반 안드로이드 휴대폰 정보 푸쉬 시스템을 제안한다. 이를 위해 C/S 네트워크 환경에서 자바 언어와 eclipse 개발 도구로 XMPP 기반의 인스턴트 메시징 시스템을 구현하였다. 본 시스템은 PC서버에서 휴대폰 클라이언트까지 메시지를 주고받을 수 있으며, 안드로이드 휴대폰에서 가지고 있는 Smack 패키지로 확장이 가능하게 하였다.

본 논문의 시스템은 기존의 XMPP 프로토콜을 이용한 응용 프로그램들 보다 파싱속도와 전송속도 면에서 성능이 우수함을 확인하였다.

ABSTRACT

In this paper, android phone information push system based on the XMPP protocol is proposed. For this, we implement a XMPP(eXtensible Message and Presence Protocol) based instant message system in C/S network environment by using Java language and eclipse device. This system not only allows sending and receiving messages from the PC server to mobile phone client, but also enables the expansion into the smack package of an android phone.

Thus, this system shown, in this paper has the better performance in transmission and transfer speed than those of the basic XMPP protocol system.

키워드

XMPP, 안드로이드, XML, Smack

Key word

XMPP, android, XML, Smack

* 정회원 : 배재대학교 컴퓨터공학과

접수일자 : 2013. 01. 07

** 준회원 : 배재대학교 컴퓨터공학과

심사완료일자 : 2013. 02. 05

*** 종신회원 : 배재대학교 컴퓨터공학과(교신저자, hkjung@pcu.ac.kr)

Open Access <http://dx.doi.org/10.6109/jkiice.2013.17.3.561>

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

BlackBerry는 2002년 푸쉬 이메일 서비스를 시작으로 정보 푸쉬 서비스를 시작했으며, Apple도 2009년 iPhone 3.0 이후부터 Apple 정보 푸쉬 서버를 지원하고 있다. 또한, 마이크로소프트도 Windows Phone 7에서 마이크로소프트 정보 푸쉬를 지원한다. 안드로이드 기반의 애플리케이션이 급격히 증가하고 있지만, 현재 Google에서 정식적인 안드로이드 기반 푸시 정보 서비스를 지원하고 있지 않은 상황이다.

따라서, 여기서는 안드로이드 기반 푸시 정보 서비스를 지원하기 위한 서버 및 클라이언트 솔루션을 제공하여, 국내외 안드로이드 애플리케이션 개발자 커뮤니티 활성화에 기여하고자한다.

이에, 본 논문에서는 자바 기반 푸시 정보 서버와 XMPP Smack 패키지의 확장기반 안드로이드 푸시 정보 클라이언트 모듈을 설계 및 구현한다.

이를 위해 XMPP 프로토콜 및 안드로이드 휴대폰에 대해 연구하였다. 논문에서 제안한 시스템은 XMPP를 통해 정보를 전송하여 전송 과정이 간단하고, 전송 속도도 빠르다.

II. 관련연구

2.1. XMPP(eXtensible Message and Presence Protocol) [1-5]

XMPP는 인터넷 상의 두 지점 간에 확장 가능한 메시지와 상태정보를 실시간으로 통신하기 위한 XML 기반의 오픈 표준 기술이다. 1999년 Jabber 오픈소스 커뮤니티에 의해 개발되었고, 2000년~2004년에 걸쳐 IETF에 의해 표준화되었다. XMPP Standard Foundation(XSF)의 표준화 작업에 의해 지속적으로 확장되고 있으며, XMPP Extension Protocols(XEP)로 별도 확장 스펙을 관리하고 있다.

XMPP 프로토콜은 푸쉬 기술의 통신방식으로 클라이언트가 서버로부터의 정보를 갱신하고자 할 때, HTTP 기반의 폴링(Polling) 방식에 비해 훨씬 클라이언트의 자원 소모가 적고 효율적이다.

2.2. 안드로이드(android)[6-8]

안드로이드가 기존의 휴대폰 운영체제인 마이크로소프트의 '윈도 모바일'이나 노키아의 '심비안'과 차별화되는 것은 완전 개방형 플랫폼이라는 점이다. 종전에는 휴대폰 제조업체와 서비스업체마다 운영체제가 달라 개별적으로 응용프로그램을 만들어야 한다.

이에 비하여 안드로이드는 기반 기술인 '소스 코드'를 모두 공개함으로써 누구나도 이를 이용하여 소프트웨어와 기기를 만들어 판매할 수 있도록 하였다. 개발자들은 이를 확장, 대체 또는 재사용하여 사용자들에게 풍부하고 통합된 모바일 서비스를 제공할 수 있게 된 것이다.

III. 정보 푸시 시스템 설계

3.1. 시스템 구성

서버와 클라이언트 모듈은 모두 이클립스 기반의 자바 시스템이다. 서버는 자바기반 소켓 서버로 XMPP 프로토콜을 지원하며, 관리자용 웹 콘솔을 이용하여 클라이언트 접속 정보 모니터링 및 푸시 메시지를 전송할 수 있다.

클라이언트는 안드로이드 플랫폼용 푸시 정보 클라이언트 라이브러리 배포 모듈이다. 안드로이드 기반 애플리케이션으로 서버로부터 푸시된 메시지를 수신하여 사용자에게 통보한다.

안드로이드 푸시 정보 시스템은 XMPP 프로토콜을 지원하는 소켓 서버 시스템과 안드로이드 기반 클라이언트 애플리케이션으로 구성된다.

3.2. 푸시 정보 과정

안드로이드 푸시 정보 시스템의 기본적인 푸시 정보 과정은 그림 1과 같다.

- ① 클라이언트는 푸시 정보 서버에 접속하여, 단말 기본 정보를 서버에 등록하고 연결 상태를 유지
- ② 서버는 메시징 이벤트가 발생한 경우, 클라이언트에 푸시 방식으로 메시지를 전송
- ③ 클라이언트는 서버부터 전송된 메시지를 통보



그림 1. 푸쉬 정보 과정
Fig. 1 Push information procedure

3.3. 서버(androidpn-sever) 설계

푸쉬 정보 서버는 구동을 위해 **Java SE 6** 런타임이 필요하며, 스프링(Spring) 프레임워크를 이용해 클래스간의 의존성을 가지고 **MINA** 프레임워크를 이용하여 구성된 소켓 서버이다. 데이터 통신 방식은 **XML** 스트리밍 방식의 **XMPP** 프로토콜을 사용하고 있으며, **XMPP** 서버 스펙을 구현하였다. 또한, **Jetty** 서블릿 컨테이너를 내장하여 관리자 웹 콘솔 애플리케이션을 포함하고 있다. 서버 구조는 그림 2와 같다.

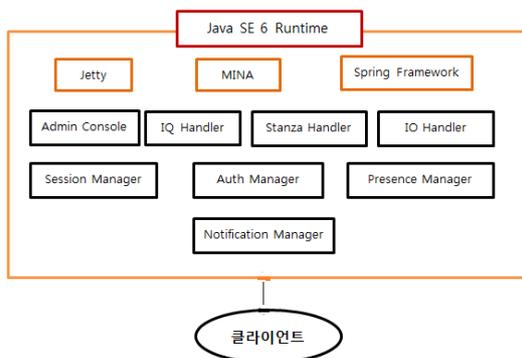


그림 2. 서버 구조
Fig. 2 Server architecture

- **IO Handler** : XML포맷의 스트리밍 바이너리 데이터를 핸들링
- **Stanza Handler** : XMPP 기본 통신 단위인 Stanza 및 클

라이언트와의 연결 특징을 핸들링

- **IQ Handler** : IQ Stanza 데이터를 해석하고 타입별 데이터를 핸들링
- **Session Manager** : 클라이언트 연결 세션 들을 전반적으로 관리
- **Auth Manager** : 클라이언트 사용자 등록 및 로그인 등 인증 관리
- **상태 매니저(Presence Manager)** : 클라이언트 연결 상태(Presence) 정보를 관리
- **Notification Manager** : 서버 이벤트에 따라 클라이언트로 Push 메시지 전송을 담당
- **Presence Manager** : 클라이언트 연결 상태 (Presence) 정보를 관리
- **Admin Console** : 클라이언트의 등록 정보, 접속 상태 모니터링 및 Push 메시지 전송 UI를 가진 관리자용 웹 애플리케이션 모듈

3.4. 클라이언트(androidpn-client) 설계

푸쉬 정보 클라이언트 모듈 안드로이드 OS 1.5이상의 플랫폼을 필요로 하며, **XMPP** 클라이언트 형태로서 서버와 **XML** 스트리밍 기반 통신을 한다. 그림 3에 클라이언트 구조를 보인다.

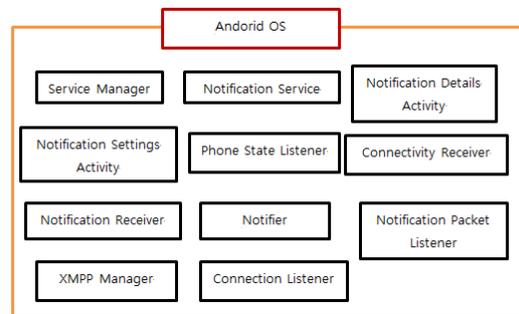


그림 3. 클라이언트 구조
Fig. 3 Client architecture

- **Service Manager** : 클라이언트 환경 설정 로딩 및 통지 서비스
- **Notification Service** : 서버로부터 메시지 수신을 위해 사용자 프로그램이 종료되어도 백그라운드로 수행되는 안드로이드 Service를 상속한 서비스
- **XMPP Manager** : XMPP 클라이언트 모듈로 푸쉬 정보

서버로의 연결, 사용자 등록, 로그인 등을 담당

- **Connection Listener** : 클라이언트와 서버간 연결의 지속성을 유지하기 위해 연결 상태를 모니터링
- **Notification Packet Listener** : 서버로부터의 Notification 패킷 수신을 담당하고, 통지를 위한 브로드캐스트를 수행
- **Notification Receiver** : 서버로부터 수신 후 브로드캐스트된 Notification 메시지 정보를 받아 처리
- **Notifier** : **Notification Manager**를 이용하여 수신된 메시지를 사용자에게 통보
- **Connectivity Receiver** : 클라이언트와 서버 간 연결의 지속성을 위해 네트워크 상태를 모니터링
- **Phone State Listener** : 클라이언트와 서버간 연결의 지속성을 위해 전화기 데이터 연결 상태를 모니터링
- **Notification Details Activity** : 사용자에게 정보 메시지의 상세정보 뷰를 위한 Activity
- **Notification Settings Activity** : 클라이언트 의 메시지 수신, 소리, 진동 여부등의 설정 뷰를 제공하는 Activity

IV. 정보 푸시 시스템 구현

4.1. 구현 환경

시스템 기본 개발 환경은 서버와 클라이언트 모두 Java 언어를 사용하고 있으며 JDK 5.0이상과 Eclipse를 필요로 한다. 상세 개발 환경은 표 1, 2와 같다.

표 1. 서버 구현 환경
Table. 1 Server build environment

구분	설명
시스템 플랫폼	Linux또는windows
프로그래밍 언어	Java (JDK1.5이상)
IDE	Eclipse
Database	HSQldb 또는 MySQL
Servlet Container	Jetty 6.1
빌드 도구	Ant 1.7.1, Maven 2.0
J2EE 스펙	Servlet 2.6,JSP 2.2
핵심 프레임워크 및 라이브러리	Spring2.6, MINA2.0, Hibernate 3.3, SiteMesh2.4

표 2. 클라이언트 구현 환경
Table. 2 Client build environments

구분	설명
디바이스 단말기	LG LU6200
모바일 플랫폼	Android SDK r20
프로그래밍 언어	Java (JDK1.5이상)
빌드 도구	Ant 1.7.1
IDE	Eclipse
핵심 프레임워크 및 라이브러리	Smack 3.1

4.2. 구현 화면

4.2.1. 서버 정보 푸시 시스템

Username을 지정하여 특정 클라이언트에만 푸시 메시지를 전송할 수 있다. Message에 내용을, Title에 제목을, URI에 홈페이지를 입력한다.

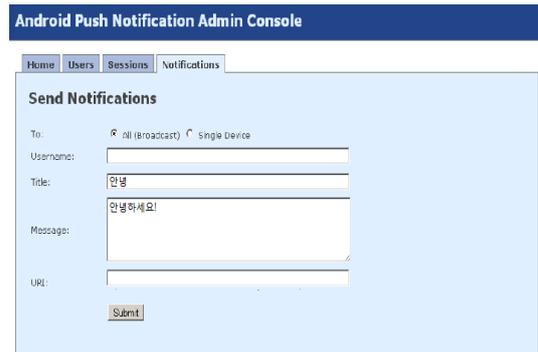


그림 4. 서버 푸시 메시지 전송
Fig. 4 Server push message transfer

4.2.2. 클라이언트 수신 시스템

그림 5는 푸시 메시지 수신 화면이다.

메시지 수신 후에 수신 목록을 클릭했을 경우 전체 메시지 상세 내용을 볼 수 있다. 푸시 정보 설정화면에서 메시지 수신여부 및 사운드, 진동 여부를 설정할 수 있다.



그림 5. Android 2.2 에뮬레이터에서 푸쉬 메시지를 수신
Fig. 5 Push messages receive on the Android 2.2 emulator

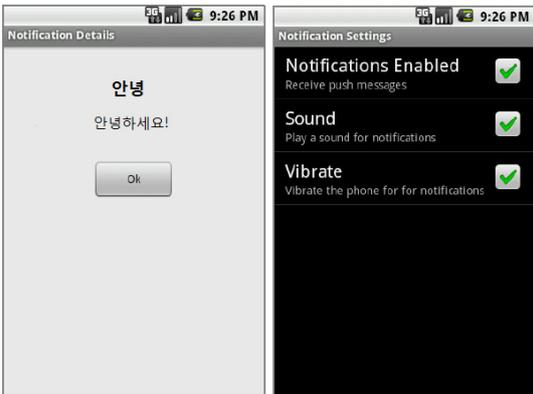


그림 6. 정보 세부 사항과 알림 설정
Fig. 6. The information details and the notification settings

4.3. 시스템 고찰

일반적으로 SMS 메시지를 보낼 때 작은 패킷 형태로 메시지는 먼저 SMSC(서비스 센터)를 통해 처리한다. 다음 휴대폰 송신탑을 통해서 메시지를 휴대폰에 보낸다. 만약 바쁜 서비스 센터는 지연 될 것이다. SMS는 또한 단어 개수 제한이 있다. 메시지의 단어 개수 제한을 초과 하면 MMS로 SMS 메시지를 분할하고 휴대폰에서 조합한다. 따라서, 전송속도가 매우 느리다. MMS메시지 전송방식은 인터넷을 통해서 서비스센터의 서버로 처리한다. 다음에 휴대폰으로 보낸다.

XMPP 프로토콜기반 정보 푸쉬 시스템은 스스로 XMPP 서버를 통해서 정보를 전송한다. 따라서, 서버가

비지 상태가 되지 않는다. 전송 과정이 간단하고 전송 속도도 더 빠르다.

본 논문에서는 LG-LU6200 휴대폰에서 SMS, MMS와 XMPP 프로토콜의 정보 전송 속도를 그림 7과 같은 환경에서 시험하였다. 결과는 표 3과 같다.



그림 7. SMS, MMS와 XMPP 프로토콜의 정보 전송 테스트
Fig. 7 SMS, MMS and XMPP protocol of information transmission test

표 3. Android 모바일에서 프로토콜 정보 전송 속도의 테스트 결과
Table. 3 Information transfer speed test results on the Android phone

	프로토콜		
	XMPP	SMS	MMS
전송 속도	1.2s	2.3s	2.5s

정보 전송 속도의 비교에 따라 XMPP기반 정보 푸쉬 시스템이 성능이 좋은 것을 확인하였다.

V. 결론 및 향후 연구 과제

본 논문에서는 자바기반 푸쉬 정보 서버와 XMPP Smack 패킷의 확장으로 안드로이드 푸쉬 정보 클라이언트 시스템을 구현하였다. 이를 위해 XMPP기반 프로토콜을 사용하여 파싱과 전송속도 면에서 기존 방식보다 우수함을 확인하였다. 이는 서버를 확장하여 여러 개의 안드로이드 애플리케이션들을 동시에 지원하는 플랫폼 서비스에 응용될 수 있다. 클라이언트는 안

드رويد OS 이외의 타 모바일 OS 애플리케이션 지원을 위해 클라이언트 모듈을 확장할 수 있다. 이 시스템은 모든 안드로이드 휴대폰에서 정상적으로 사용할 수 있다.

향후 연구는 안드로이드 휴대폰의 메시지 프로토콜 확장을 통해 이미지, 비디오 등 멀티미디어 메시지 전달하는 등 더욱 생동감 있는 뉴스를 전달하는 애플리케이션에 응용될 수 있을 것이다. 안드로이드 휴대폰의 SMS를 대신하는 가입자가 실시간으로 단문 메시지 전달하는 서비스 등에 응용할 수 있다.

참고문헌

- [1] Jack Moffitt, "Professional XMPP Programming with JavaScript and jQuery", September 2011
- [2] Yun-Myeong Kim, "Jabber Programming", Hanbit Media, 2002
- [3] Sang-Hyeong Kim, "Android Programming conquer", HanbitMedia, 2010
- [4] Android Push Notification System (Project Guide v0.5), Moduad <http://www.slideshare.net/sandeepreddy42/androidpn-guide050ko>, 2010
- [5] RFC2779-2000, Instant messaging and presence[EB], RFC 3921, 2004
- [6] Saint-Andre P. Ed. "Mapping the extensible messaging and presence protocol(XMPP) to common presence and instant messaging (CPIM)[S], RFC3922". 2004
- [7] Bray T.Paoli J. Sperberg-McQueen C.Maler E.Extensible Markup Language (XML) 1.0 (2nd ed), W3C REC-xml, October 2000
- [8] Parviainen, R.Parnes, P. Mobile instant messaging[C]. In: The 10th International Conference on, Telecommunications, pp. 425-430, 2003

저자소개



편도길(Do-kil Pyoun)

2005년 배재대학교 정보통신과 (공학사)
2012년 배재대학교 컴퓨터공학과 (공학석사)

2013년~현재 배재대학교 컴퓨터공학과 박사과정
2002년~현재 침례신학대학교 전산실 근무
※관심분야: 학사관리, 차세대인터넷, UI, 모바일, HTML5



유호(Liu Hao)

2012년 배재대학교 정보통신학과 (공학사)
2012년~현재 배재대학교 컴퓨터공학과 석사과정

※관심분야: C++, HTML5, XMPP



정희경(Hoe-kyung Jung)

1985년 광운대학교 컴퓨터공학과(공학사)
1987년 광운대학교 컴퓨터공학과(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)
1994년~현재 배재대학교 컴퓨터공학과 교수
※관심분야: 멀티미디어 문서정보처리, XML, SVG, Web Services, Semantic Web, MPEG-21, Ubiquitous Computing, USN, 정보기술 아키텍처