

XMPP 프로토콜기반 Android 휴대폰 정보 푸쉬 시스템

유호* · 유결* · 최광일* · 박상노* · 정희경*

배재대학교

Android mobile phone information push system based on the XMPP protocol

Liu-Hao* · Liu-Jie* · Kwang-Il Choi* · Sang-No Park* · Hoe-Kyung Jung**

Paichai University

E-mail : {xjue, liujietym}@pcu.ac.kr, blue32@nate.com, psn@hankisul.com, hkjung@pcu.ac.kr

요 약

본 논문에서는 XMPP 프로토콜 기반 Android 휴대폰 정보 푸쉬 시스템을 제안한다 이를 위해 C/S 네트워크 환경에서 JAVA 언어와 eclipse 개발 도구로 XMPP 기반의 인스턴트 메시징 시스템을 구현하였다. 본 시스템은 PC서버에서 휴대폰 클라이언트까지 메시지를 주고받을 수 있다 Android 휴대폰에서 가지고 있는 smack 패키지로 확장이 가능하게 하였다.

본 시스템은 기존의 XMPP 프로토콜을 이용한 응용 프로그램들 보다 파싱속도와 전송속도 면에서 성능이 우수하다.

ABSTRACT

In this paper, android phone information push system based on the XMPP protocol is proposed. It uses Java and eclipse development tools to create an information pushing system in android phone based on the XMPP protocol, and runs on the C/S network situation. Not only, the information can be send and receive from android phone in the PC server, but also the smack packet of the android phone is used fully by this system.

Therefore, the speed of the transmission and the transfer are better than other sending sportocol in the basic XMPP protocol of the system.

키워드

XMPP, android, XML, sever

I. 서 론

안드로이드 기반의 애플리케이션이 급격히 증가하고 있지만, 현재 Google에서 정식적인 안드로이드 기반 푸시 정보 서비스를 지원을 하고 있지 않은 상황이다. 본 논문에서는 이러한 문제를 해결하였다.

이에 반해 BlackBerry는 2002년 Push Email 서비스를 시작으로 정보 푸쉬 서비스를 시작했으며, Apple도 2009년 iPhone 3.0 이후부터 Apple 정보 푸쉬 서버를 지원하고 있다. 또한 Microsoft도 출시 예정인 Windows Phone 7에서 Microsoft 정보 푸쉬 지원과 함께 출시 예정이다

본 논문에서는 XMPP 프로토콜 및 Android 휴대폰의 관련연구에 대하여 기술하고 시스템의 서버와 클라이언트에 대한 설계 내용을 기술한다

II. 관련연구

2.1 XMPP(eXtensible Message and Presence Protocol) [1,2]

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

XMPP 프로토콜은 푸쉬 기술의 통신방식으로 클라이언트가 서버로부터의 정보를 업데이트하고자 할 때, HTTP 기반의 Polling 방식에 비해 훨씬 클라이언트의 자원 소모가 적고 효율적이다

2.2 안드로이드(android)[3]

안드로이드는 운영체제 미들웨어 및 핵심 응용 프로그램을 포함하는 휴대기기용 소프트웨어 스펙이다. Google에 의해 개발되고 오픈 소스(2008년)이다. 이는 Linux 커널 기반의 운영체제이다. Dalvik 가상머신과 Java 언어 기반의 SDK 지원한다. 그리고, Android Market을 통해 애플리케이션을 배포할 수 있다.

Ⅲ. 시스템 설계

안드로이드 푸시 정보 시스템은XMPP 프로토콜을 지원하는 소켓 서버 시스템과 안드로이드 기반 클라이언트 애플리케이션으로 구성된다

| | |
|-------|------------------------------------------------------------------------------------|
| 서버 | Java기반 소켓 서버로 XMPP 프로토콜을 지원하며, 관리자용 웹 콘솔을 이용하여 클라이언트 접속 정보 모니터링 및 푸시 메시지를 전송할 수 있음 |
| 클라이언트 | 안드로이드 기반 애플리케이션으로 서버로부터 푸시된 메시지를 수신하여 사용자에게 정보함 |

3.1 푸시 정보 과정[4]

안드로이드 푸시 정보 시스템의 기본적인 푸시 정보 과정은 아래 그림과 같다.

- ① 클라이언트는 푸시 정보 서버에 접속하여 단말 기본 정보를 서버에 등록하고 연결 상태를 유지한다.
- ② 서버는 메시징 이벤트가 발생한 경우, 클라이언트에 푸시 방식으로 메시지를 전송한다.
- ③ 클라이언트는 서버부터 전송된 메시지를 정보를 한다.



그림 1. 푸시 정보 과정

3.2 서버(androidpn-sever) 설계[1,4,5]

푸시 정보 서버는 구동을 위해 java SE 6 런타임이 필요하며, Spring 프레임워크를 이용해 클래스간 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

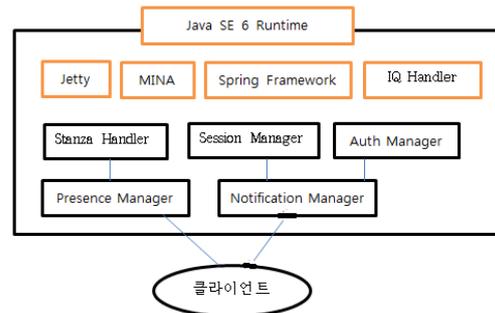


그림 2. 서버 아키텍처

- Stanza Handler: XMPP 기본 통신 단위인 Stanza 및 클라이언트와의 연결 특징을 핸들링한다.
- IQ Handler: IQ Stanza 데이터를 해석하고 타입별 데이터를 핸들링한다.
- Session Manager: 클라이언트 연결 세션들을 전반적으로 관리한다
- Auth Manager: 클라이언트 사용자 등록 및 로그인 등 인증 관리한다
- 상태 매니저(Presence Manager):클라이언트 연결 상태(Presence) 정보를 관리한다.
- Notification Manager: 서버 이벤트에 따라 클라이언트로 Push 메시지 전송을 담당한다.

3.3 클라이언트(androidpn-client) 설계[1,2,4]

푸시 정보 클라이언트 모듈을 안드로이드 OS 1.5 이상의 플랫폼을 필요로 하며 XMPP 클라이언트 형태로서 서버와 XML 스트리밍 기반 통신을 한다.

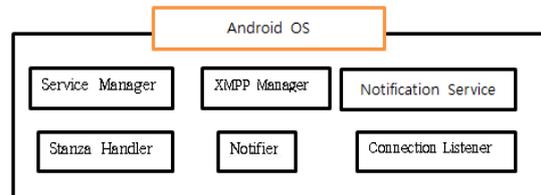


그림 3. 클라이언트 아키텍처

- Service Manager: 클라이언트 환경 설정 로딩 및 통지 서비스이다
- Notification Service: 서버로부터 메시지 수신을 위해 사용자 프로그램이 종료되어도 백그라운드로 수행되는 안드로이드 Service를 상속한 서비스이다
- XMPP Manager: XMPP 클라이언트 모듈로 푸시 정보 서버로의 연결 사용자 등록, 로그인 등을 담당한다
- Connection Listener: 클라이언트와 서버 간 연결의 지속성을 유지하기 위한 연결 상태를 모니터링 한다.
- Notifier: Notification Manager를 이용하여 수신된 메시지를 사용자에게 통보한다.

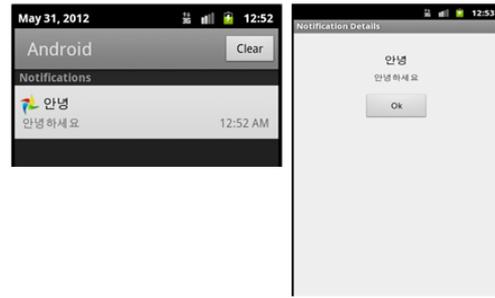


그림 5. 메시지 수신 화면

메시지 수신 목록 화면 및 이를 터치했을 경우 전체 메시지 상세 내용을 볼 수 있다.

IV. 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 다음과 같다.

4.2 구현 화면

(1) 서버 푸쉬 정보 시스템



그림 4. androidpn 서버 푸시 메시지 전송

Username을 지정하여 특정 클라이언트에만 푸쉬 메시지를 전송할 수 있다.

(2) Android 2.1 물레이터에서 푸쉬 메시지를 수신한 화면이다.

V. 결론 및 향후 연구 과제

본 논문에서는 서버 시스템을 확장하여 여러 개의 안드로이드 애플리케이션들을 동시에 지원하는 플랫폼 서비스에 응용될 수 있다 클라이언트는 안드로이드 OS 이외의 타 모바일 OS 애플리케이션 지원을 위해 클라이언트 모듈을 확장할 수 있다. 이 시스템은 모든 Android 휴대폰에서 정상적으로 사용할 수 있다.

향후 연구는 메시지 프로토콜 확장을 통해 이미지, 사운드, 비디오 등 멀티미디어 메시지 전달하는 등 더욱 생동감 있는 뉴스를 전달하는 애플리케이션에 응용될 수 있다 SMS(Short Message Service)를 대신하는 가입자가 실시간으로 단문 메시지 전달하는 서비스 등에 응용될 수 있다.

참고문헌

- [1] Professional XMPP Programming with JavaScript and jQuery, 2011.9
- [2] Jabber Programming, 2002.
- [3] Android Programming Complete Guide 2010.9
- [4] Android Push Notification System(Project Guide v0.5), 2010 Moduad
- [5] RFC2779-2000, Instant messaging and presence[EB], RFC 3921, 2004.