Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 18, No. 3: 754~760 Mar. 2014

스마트폰 기반의 신뢰성 있는 Push Notification 미들웨어 및 모듈 개발

정준석1 · 김태용2*

Development of Reliable Push Notification Middleware and Module based on Smart Phone

Joon-Seok Jung¹ · Tae Yong Kim^{2*}

¹iReal co. Ltd., Busan Cultural Content Complex Bldg. #924, Busan 612-020, Korea

요 약

Push Notification Service 기술은 단위 서비스가 아닌 플랫폼 경쟁을 위한 기반 기술이다. 국내에서는 이러한 메인 플랫폼에서 제공하는 기술을 기반으로 한 서비스 형태만이 연구되고 개발되고 있는 실정이며 기반 기술에 대한 연구는 지속적으로 이루어지지 못하고 있다. 본 연구에서는 국산 기술 확보 차원에서 새로운 형태의 Push Notification System을 제안하고 설계하였다.

ABSTRACT

Push Notification Service technology is to provide platform competition rather than unit-based technology. In Korea, only the type of a service using the technology provided by this main platform has been studied and the study of the based technology has not been developed consistently. In this paper, new types of Push Notification Service capable of domestic technology acquisition was proposed and designed.

키워드: 푸시 서비스, 미들웨어, GCM, APNSc

Key word: Push service, middleware, GCM, APNSc

접수일자: 2014. 01. 10 심사완료일자: 2014. 01. 30 게재확정일자: 2014. 02. 10

Open Access http://dx.doi.org/10.6109/jkiice.2014.18.3.754

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/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.

^{2*}Div. of Computer and Information Engineering, Dongseo University, Busan 617-716, Korea

^{*} Corresponding Author Tae Yong Kim(E-mail:tykimw2k@gdsu.dongseo.ac.kr, Tel:+82-51-320-1738) Div. of Computer and Information Engineering, Dongseo University, Busan 617-716, Korea

Ⅰ. 서 론

Notification Service를 위해 자바의 NIO(New I/O) 기능을 이용하여 미들웨어 기반으로 개발되기도 하였다[1]. 최근에는 Apple이 2009년 Push Notification Service 개발과 함께 서비스[2]를 시작하여 모바일 시장알람 서비스, 모바일 클라우드 시스템 등 모바일 초기시장을 선점한 이후로 Google, Microsoft 등 메인 모바일 플랫폼 업체 뿐 아니라 Facebook, Twitter 등 모바일 메신저 기업들도 자사만의 Push 서비스를 개발하여 경쟁하고 있다.

현재 기술의 흐름은 스마트워크[3] 시대로 접어들고 있으며 국내 역시 2015년까지 전체 노동자의 30%이상을 스마트워크 방식으로 전환하고자 하는 '스마트워크 확산을 위한 노동관계법령 가이드라인[4]'와 같은 법적 정비와 함께 지속적인 사회적 투자가 진행되고 있다. 또한 스마트 모바일기기로 업무 수행을 지원하기 위한 모바일 클라우드 서비스 플랫폼 개발에 대한 관심도 높아지고 있다[5]. 또한 스마트폰 상에서 프로젝트 관리를 위한 소셜 네트워킹 서비스를 위해 일정 통지 및 이슈추적 시스템 구축을 위해 Push 서비스를 활용하는 예도 살펴볼 수 있다[6].

그러나 스마트워크의 기반이 되는 실시간 데이터 연동 및 알림(Push Notification) 기술은 현재 Google과 Apple사에서 만든 GCM(Google Cloud Message)[7], APNS(Apple Push Notification Service)[8,9] 서비스에 의존하고 있는 실정이다. 이 두서비스의 가장 큰 문제점은 전송에 따른 성공여부 판단이 힘들며 이를 보완하기 위해 여러 차례 메시지를 전송하여야 한다. 또한 알림 전송 순서가 서버 상태에 영향을 크게 받게 되며 GCM, APNS 업데이트에 따라 기능을 업그레이드 해줘야하는 단점을 가지고 있다.

이러한 문제점을 개선하고 자체적인 국산 기술 확보 차원에서 신뢰성이 높은 Push Notification 기술을 제안하고자 한다. 이를 통하여 GCM, APNS 서비스에 대한 대외 의존성을 낮추고 업무용 App과 Push 메시지를 직접 통제 가능한 서버 시스템과 Push Notification System의 기반이 되는 미들웨어와 프로토콜을 설계하였다.

Ⅱ. 본 론

2.1. Push Notification System 시장

Push Notification은 모바일 App에서만 가능하기 때문에 MIS와 SNS에 주로 사용되고 있으며, 국내의 경우는 카카오톡(80%), 라인(6%) 등에서 사용되고 있다. 해외의 경우는 Facebook, Twitter 등에서 알림서비스 목적으로 이용되고 있으며, 다수의 포털사이트 App 서비스에서도 Push Notification기능을 이용하여 사용자에게 정보를 제공하고 있다. 그림 1에서 알 수 있듯이 Push Notification 사용유무에 따라 사용자의 모바일 접속 빈도에서 차이가 발생하고 있다는 것을 알 수 있다[10]. 따라서 제안하고자 하는 기술 개발을 통하여 시장진입이 충분히 가능할 것으로 생각된다.

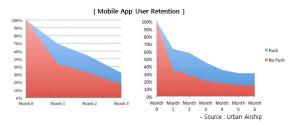


그림 1. 모바일 앱 유저 변동량 Fig. 1 User Retention on Mobile App

2.2. Push Notification System Proxy 서버

Apple의 APNS 및 Google의 GCM Push 서버의 경우 단독으로 동작하는 모바일 운영체제 기반에서 Push 메시지를 전송할 수 있는 서비스이지만 하루에 보낼 수 있는 Push 메시지 수와 길이가 제한적이다. 또한 스마트폰이 휴면 상태로 들어갔을 경우 스마트폰으로 Push 메시지를 전송할 수 있는 방법은 두 가지를 생각할 수 있다. 한 가지 방법은 APNS 및 GCM Push 서버를 이용하여 메시지를 전송하는 방법이다. 또 다른 방법으로서 안드로이드 운영체제에서 서비스를 지속적으로 Push 서버와 연동을 시키는 방법을 생각할 수 있다. 그러나 후자의 경우는 배터리 소모의 주요 원인으로 지적되고 있어 연구 개발의 범위에서 제외시켰다.

Push Notification System Proxy 서버를 운영할 때 스마트폰이 휴면상태에 돌입한 경우에는 Apple의 APNS 및 Google의 GCM Push 서버를 제한적으로 이용하여 Push Notification System App을 휴면상태에서 활동 상

태로 변경하도록 설계하였다. 그리고 Push Notification System App으로 Push 알림 메시지에 해당하는 특정 코드만 전송하도록 하여 빠른 응답 속도를 확보하고 서버부하를 최소한으로 경감시켰다. 또한 실제 전송되는 데이터는 높은 압축률로 처리 가능하도록 하였으며 안전한 데이터 전송이 가능하도록 프로토콜을 구축하여 Push Notification System Proxy 서버에서 가져올 수 있도록 설계 하였다. Push Notification System Proxy 서버 의 구성은 그림 2에 나타내었다.

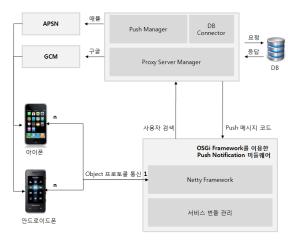


그림 2. Push Notification System 프록시 서버 구성도 Fig. 2 Push Notification System Proxy sever

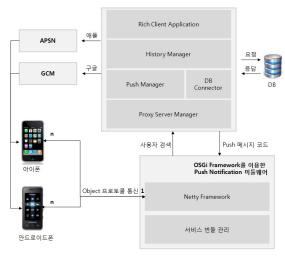


그림 3. Push Notification System 이력 매니저 Fig. 3 Push Notification System History Manager

2.3. Push Notification System History 관리

Push Notification System History 관리를 위해서 Push 메시지 이용 현황 및 사용빈도를 관리할 수 있는 서비스 기능을 추가 하였다. 관리자가 한눈에 확인할 수 있는 Rich Client Application 서비스와 Push Notification System을 이용하여 사용자들의 습관 및 자주 사용하는 서비스를 판단하여 큐레이팅 서비스를 할수 있는 기반을 마련하였다(그림 3 참조).

2.4. Push Notification System 미들웨어

현재 사용되고 있는 네트워크 미들웨어는 상용화된 제품을 찾아보기 힘들며, 특히 사용자가 원하는 기능을 제공하는 네트워크 미들웨어는 찾아보기 힘들다. 본 연구에서는 Push Notification System에 최적화된 미들웨어를 설계하고 프로토콜 규약을 지정하여 서버 구축 및 서비스에 바로 적용시킬 수 있는 미들웨어를 개발하였다.

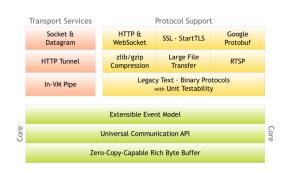


그림 4. Netty 프레임워크 구성도

Fig. 4 Netty framework

Push Notification System의 목표 성능을 확보하기 위하여 다음 사항을 고려하였다.

- 개방형 프레임워크 사용: 기존의 고성능/고확장성 네트워크 미들웨어인 Netty 프레임워크[11]의 사용 (그림 4 참조). Netty Framework를 사용하는 이유는 Netty Framework가 고성능/고확장성 네트워크 미들 웨어를 개발하기 위한 다양한 기반 기술을 제공하고 있기 때문이다.
- 프로토콜 확장성: 특정 프로토콜에만 동작하는 것이 아니라 다수의 프로토콜을 수용 가능하도록 하여 외 부 인터페이스 확장이 가능.

- **암호화 통신**: 기본 데이터 통신에서 암호화 기능을 지원하여 보안성을 확보.
- 동적 바인딩: 새로운 기능이나 프로토콜을 신규로 추가하고 싶은 경우를 고려하여 기존 미들웨어에 새로운 기능을 추가하여 새롭게 빌드를 하는 것이 아니라 새로운 기능만 만들어 동적으로 바인딩 가능.

또한 미들웨어 구축을 위해서 OSGi Framework[12]를 기반으로 개발한 이유는 새로운 서비스 및 프로토콜을 동적으로 추가하기 위해서는 모듈형식으로 동작하는 프레임워크가 필요하기 때문이다. OSGi Framework에서는 이러한 모듈형식의 서비스를 이미 지원하고 있어 미들웨어 구축을 위한 기반 프레임워크로 사용하였다. 따라서 제안하는 Push Notification System을 사용하면 요구사항에 따라 효율적인 개발 프로세스를 진행할 수 있으며 개발 기간의 단축이 기대된다.

Push Notification System 미들웨어 구성은 그림 5에 나타내었다.

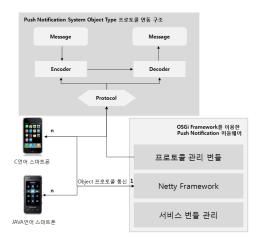


그림 5. Push Notification System 미들웨어 구성도 Fig. 5 Push Notification System Middleware

2.5. 고압축 및 안전한 데이터 전송을 위한 Push Notification System 프로토콜

현재 사용되는 통신 데이터들은 HTTP 또는 XML 프로토콜을 주로 사용하고 있으며 그 외의 시스템에서는 Object 형식의 프로토콜을 사용하고 있다. HTTP 또는 XML 프로토콜을 사용하면 프로토콜 해석이 쉽고 명확하여 외부 인터페이스 확장성을 높일 수 있다는 장점을

가지고 있다. 그러나 HTTP 또는 XML 프로토콜 사용시 암호화를 하지 않고 통신을 할 경우 누구나 쉽게 데이터를 분석할 수 있게 되며 처리해야 할 데이터 량이 많아져 느린 응답성을 가지게 되며 서버의 부하를 야기시키는 단점을 가지고 있다.

Object 형식의 프로토콜은 HTTP 또는 XML 프로토 콜보다 가볍고 프로토콜 해석 자체가 힘들어 보안상 안 정하지만 사용이 불편하며 타 언어와의 연동이 힘들다 는 단점이 있다.

개발된 Push Notification System 프로토콜에서는 HTTP, XML 그리고 Object 기반 프로토콜과 같은 데이터 모두 처리 가능하며, 고압축 기술을 적용하여 적은데이터 양으로 동일 기능을 처리할 수 있다. 또한 프로토콜 해석 자체가 힘들며 타 언어와의 연동을 보다 효율적으로 처리 할 수 있다. Push Notification System 프로토콜 개념도는 그림 6에 나타내었다.

2.6. Push Notification System의 사용자 인터페이스

그림 7에 나타낸 바와 같이 Push Notification System 의 사용자 인터페이스(User Interface)는 사용자 경험에 중점을 두고 꼭 필요한 기능으로 설계하고 환경설정이란 어려운 조작을 요하는 기능은 최대한 배제하고 원터치 실행에 중점을 두었다. 비주얼 디자인에 비중을 두어 한눈에 봐도 누구나 손쉽게 사용할 수 있으며 프로세스 흐름에 맞게 진행될 수 있도록 시나리오를 구성하였다.

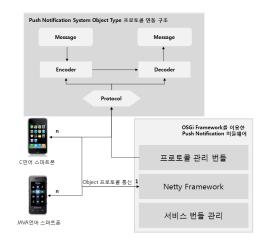


그림 6. Push Notification System 프로토콜 개념도 Fig. 6 Push Notification System Protocol

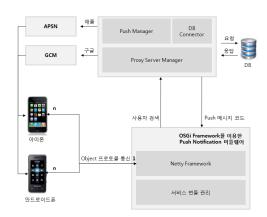


그림 7. Push Notification System 서비스 구성도 Fig. 7 Push Notification System service

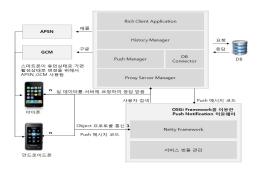


그림 8. Push Notification System 앱 구성도 Fig. 8 Push Notification System Mobile App

2.7. Push Notification System을 위한 Mobile App

그림 8은 Push Notification System을 이용하기 위한 모바일 App의 서비스 과정을 나타내고 있다. 스마트폰 에서 전송한 메시지는 Push Notification System Middle Server에 전달되고 Middle Server는 수신자의 스마트폰 Device Token을 조회 한 후 메시지를 Google의 GCM 또는 Apple의 APNS로 메시지를 전달하여 수신자의 스 마트폰을 깨우고 수신자는 이 메시지를 전달 받게 된다.

Ⅲ. 구현 결과

설계를 바탕으로 아이폰 및 안드로이드 폰용 Push Notification system App과 서버를 구축하여 그림 9와 같이 테스트베드를 구축하였다. 기본적인 테스트는 기능의 구현 여부, 트래픽 스트레스 테스트 및 무작위 대

상을 통한 실사용 테스트로 진행되었다. 기능 테스트를 통해 설계된 기능들이 모두 정상적으로 동작함을 확인 하였다.



그림 9. Push Notification System 테스트 베드 Fig. 9 Push Notification System Test Bed

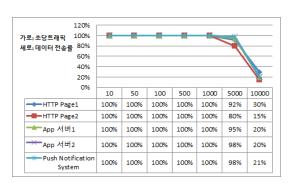


그림 10. 트래픽 스트레스 테스트 결과

Fig. 10 Traffic stress test result

서버에 대한 트래픽 스트레스 테스트는 동일한 환경에서 다양한 웹서비스와 Push Notification System을 각각 구동시키고 가상의 트래픽을 지속적으로 증가시켜 정상적으로 접속 및 데이터가 확인되는 범위를 측정하여 Push Notification System이 서버에 미치는 영향을 테스트 하였으며 서버의 물리적 한계선 이하에서타 서버와 비슷하거나 더 정상적으로 동작함을 확인하였다 (그림 10 참조). 물론 경쟁 서비스와 정확한 비교를 하기 위해서는 테스트환경의 수백 배 이상의 트래픽을 견딜 수 있는 서버 환경에서의 테스트가 요구된다. 그러나 본 테스트 베드 환경의 성능은 일반적 기업 서비스 또는 시, 군 단위의 대시민 서비스를 주된 적용 대상으로 볼 때 다양한 Push 서비스 구축에 적합한 성능을 제공할 수 있을 것으로 보인다.

실사용 테스트는 스마트폰 Push 서비스를 주로 사용하는 2-30대 대학생, 직장인 50인으로 구성되었으며 약 1주일간의 시험 사용 후 설문조사를 통해 기존 서비스와의 성능을 비교하였다(그림 11 참조).



그림 11. 사용자 설문조사 결과 Fig. 11 Question investigation results

대부분의 사용자가 기존 서비스와 비슷함을 지적하였으며 장점으로는 빠른 속도, 개별적 서비스의 가능성 등을 들었으며 단점으로는 다양한 기능의 부족을 지적하였다. 테스트 기간 동안 데이터 유실에 대해 응답한 사용자는 없었으나 데이터 유실 유무에 대해서는 대량의 데이터가 교환되는 환경에서 테스트를 해야 정확한 결과를 얻을 수 있을 것이라 사료된다.

Ⅳ. 기대효과

본 연구를 통해 개발된 Push Notification 기술을 이용한 스마트 App은 시간 및 공간의 제한을 받지 않고 업무 진행이 가능한 스마트워크 환경을 제공할 수 있는 성능으로 구현되었음을 확인하였다. 이를 통해 효율적인 기업 업무 처리를 위한 제반 요구사항에 대응하여 프레임워크 기반에서 Push Notification 서비스를 직접구현할 수 있으므로 기업의 정보보안을 유지 할 수 있으며, 메시지에 대한 로그를 조회함으로써 상황에 따라 송수신자를 추적할 수 있는 기능도 활용할 수 있다. 뿐만 아니라 업무 처리에 따른 개인 통신비용 절감을 위하여 기존의 MMS, SMS 등의 유료서비스를 대체하는 효과도 기대된다.

또한 기술우위확보 및 관련 우수 지적재산권 선 확보 를 통한 기술마케팅 및 기술사업화도 가능할 것으로 생 각되며 앞으로 병원, 대학교, 기관 등 업무상 필요한 정보를 상호 공유하고 신속한 전달이 필요한 환경에 적용가능할 것으로 보인다.

Ⅴ. 결 론

본 연구에서 제안하는 Push Notification System의 장점은 GCM 또는 APNS의 의존도를 낮추는 기술로서 해외 네트워크 이용에 대한 트래픽과 전송 속도를 낮추는 효과가 기대된다. 이는 대량의 메시지 전송에 따른 서버 트래픽 관리 기술 확보, 서버와 스마트 폰간의 간결한 소켓 구성에 대한 통신 기술력과 경쟁력 확보, 마지막으로 기술과 기술, 산업과 산업의 융·복합을 통한 시너지 효과를 기대 할 수 있다.

이러한 장점을 극대화하기 위해서는 제안된 기술을 활용하는 다양한 서비스 모델 개발과 동시에 국내 대형 모바일 플랫폼 홀더들의 적극적 기술도입 유도와 홍보 가 동시에 이루어져 할 필요가 있다.

감사의 글

본 연구는 2013년도 교육부의 지원을 받아 수 행된 산학협력 선도대학(LINC) 육성사업에 의하여 이루어진 연구로서, 관계부처에 감사 드립니다.

REFERENCES

- [1] Sung-bak Kim and Ji-hoon Song, *Java I/O & NIO Network Programming*, Hanbit Media, 1999.
- [2] Wikipedia. Apple Push Notification Service [Internet]. Available: http://en.wikipedia.org/wiki/Apple_Push_Notification_Service.
- [3] Yong-woo Choi. Smart Work[Internet]. Available: http://olv.moazine.com/rviewer/index.asp.
- [4] Disability Smart Work Lab.. Smart Work[Internet]. Available: http://smwork.or.kr/?c=5/23&uid=140.
- [5] Dae-Sang You, Kwang-Il Ko, Seung-Ryol Maeng, and Go-Whan Jin, "Mobile Cloud Service Platform for

- Supporting Busniess Tasks," Journal of the Korea Institute of Information and Communication Engineering, vol. 17, no. 9, pp. 2113-2120, Sep. 2013.
- [6] Dae-Ki Kang and Won-Tae Chang, "Project Schedule Notification and Issue Tracking System Based on Social Networking Service on a Smartphone," Journal of the Korea Institute of Information and Communication Engineering, vol. 14, no. 3, pp. 669-677, Mar. 2010.
- [7] Google Developer Official Site. [Internet]. Available: http://developer.android.com/google/gcm/index.html.
- [8] APNS-I-Phone Push Installation Information. [Internet]. Available: http://codingworshipper.tistory.com/327.
- [9] Push Notification for JSP. [Internet]. Available: http://j07051. tistory.com/374.
- [10] User Retention on Mobile App. [Internet]. Available: http://www.mobizen.pe.kr/1355.
- [11] Netty 4.0 New & Noteworthy. [Internet]. Available: http://netty.io/wiki/new-and-noteworthy.html.
- [12] OSGi Framework. [Internet]. Available: http://www.supinan. com/html/newpage.html?code=8.



정준석 (Joon-Seok Jung)

2010년 한국해양대학교 공학사 2012년 한국해양대학교대학교 공학석사 2014년 아이리얼(주) 선임연구원

※관심분야: 신호해석, 레이더, 모바일 기반 응용 소프트웨어



김태용(Tae Yong Kim)

전자통신공학과 공학박사 1993년 부경대학교(공학사) 1997년 오카아마대학(공학석사) 2001년 오카아마대학(공학박사) 2002년 ~ 현재 동서대학교 컴퓨터정보공학부 교수

※관심분야: 위성통신, 마이크로파 회로해석 및 설계, 마이크로 센서 응용, 센서 네트워크