

## 재난경보전달을 위한 스마트폰 플랫폼에서의 Push 기술 분석

곽별샘, 전인찬, 최성중  
서울시립대학교  
semix2@gmail.com

### Analysis of Push Technology in Smartphone Platform for Emergency Alert

Kwak, Byul Saim Jeon, Inchan Seong Jong Choi  
University of Seoul

#### 요 약

스마트폰은 항상 사용자가 휴대하고 있으며 필요에 따라 적절한 어플리케이션을 내려 받아 기능을 확장할 수 있다. 이에 많은 서비스 업체들은 사용자가 언제 어디서나 자사의 서비스를 이용할 수 있도록 스마트폰용 어플리케이션을 개발하고 있다. 본 고에서는 현재 스마트폰 플랫폼인 iOS, 안드로이드, MS 에서 제공한 푸시 서비스에 대해 분석한다. 다음으로 국가의 재난경보서비스를 위해 이 기술을 활용할 경우의 문제점 찾아 보았다. 본 연구의 결과 현재의 푸시 기술은 많은 문제점을 가지고 있기 때문에 정부에서 본격적으로 푸시 기술을 활용한 스마트폰 재난경보서비스를 시작하기엔 아직 이른 것으로 판단된다.

#### 1. 서론

스마트폰은 항상 사용자가 휴대하고 있으며 필요에 따라 적절한 어플리케이션을 내려 받아 기능을 확장할 수 있다. 이에 많은 서비스 업체들은 사용자가 언제 어디서나 자사의 서비스를 이용할 수 있도록 스마트폰용 어플리케이션을 개발하고 있다. 이러한 어플리케이션들 중에는 항상 인터넷과 연결하여 수시로 정보를 주고 받는 어플리케이션들이 있다. 예를 들어 CNN[1] 등의 뉴스 어플리케이션들은 항상 최근의 뉴스 정보로 갱신하고 속도 같은 중요한 내용을 즉시 사용자에게 알려준다. 트위터[2]나 페이스북[3]과 같은 소셜 네트워크 서비스들도 스마트폰용 어플리케이션을 제공하며, 자신을 언급한 글이나 자신이 작성한 글에 대한 댓글 등이 작성되었을 때 사용자에게 바로 알린다. 최근에는 메신저나 인터넷 전화 같은 어플리케이션들도 많이 등장하고 있다. 이러한 어플리케이션들은 사용자와 상호작용하지 않는 상태에서도 인터넷 연결을 지속하여 수시로 새로운 정보를 갱신할 수 있어야 한다.

본 고에서는 현재 스마트폰 플랫폼인 iOS, 안드로이드, MS 에서 제공한 푸시 서비스에 대해 분석한다. 결론에서는 현 기술의 문제점에 대해 기술한다.

#### 2. Apple Push Notification Service

애플은 2008 년 애플 세계 개발자 컨퍼런스(Apple Worldwide Developer Conference, WWDC)에서 푸시 알림 서비스(Push Notification Service, PNS)를 소개하고 iOS 3.0 버전부터 모든 개발자가 이 서비스를 사용할 수 있도록 개방했다[4].

푸시 알림 서비스는 어플리케이션이 실행 중이지 않을 때에도 일정 수준의 상호작용이 가능하게 하는 기술로써 운영체제에 의해 기기는 애플의 푸시 알림 서버와 항상 연결되며, 이 연결을 통해 다양한 어플리케이션 메시지가 전달될 수 있게 한다. 이 기술로 인해 어플리케이션 개발자 및 서비스 사업자는 사용자 단말기에서 어플리케이션이 실행 중이 아닐 때에도 중요 정보를 전달할 수 있게 되었다.

기기는 애플의 푸시 알림 서버와의 신뢰성 있는 연결을 위해 일련의 인증 절차를 따른다. 이 과정은 운영체제에 의해 자동으로 일어난다.

푸시 알림 서비스를 이용하기 위해서는 개발자나 서비스 사업자에게 별도의 서버가 요구된다. 이 서버는 애플의 푸시 알림 서버와 어플리케이션 사이에 위치하여 푸시 알림 서버를 통해 사용자 기기로 알림 메시지를 전송한다. 그러기 위해 우선 어플리케이션 지원 서버는 애플의 푸시 알림 서버에 인증을 받아야 한다. 이 과정은 앞서 살펴본 PNS 서버와 기기간의 인증 절차와 유사하다.

사용자가 푸시 알림 서비스를 이용하는 어플리케이션을 내려 받아 실행하면 푸시 알림에 대해 사용자 동의를 묻는 메시지가 나타난다. 사용자가 이에 동의하면 어플리케이션은 푸시 알림 서버에 연결하여 사용자 단말기의 고유 주소를 생성한다. 이렇게 생성된 고유 주소를 어플리케이션 지원 서버로 전달하여 추후 사용자 기기로 알림 메시지를 전송할 때 사용한다.

어플리케이션 지원 서버는 사용자에게 전달할 새로운 정보가 있을 때 단말기 고유 주소를 이용하여 푸시 알림 서버를 이용해 대상 단말기로 알림 메시지를 전송한다.

### 3. Microsoft Push Notifications

마이크로소프트의 윈도 7 폰 등의 모바일 기기들은 애플과 마찬가지로 둘 이상의 어플리케이션을 동시에 수행하는 멀티태스킹을 지원하지 않는다. 따라서 윈도 7 폰에서 동작하는 어플리케이션은 사용자가 종료하거나 다른 어플리케이션을 실행하면 실행을 멈추고 인터넷 연결 등이 모두 끊어진다.

마이크로소프트도 이러한 문제를 극복하기 위해 마이크로소프트 푸시 알림(Microsoft Push Notifications, MPN)이라는 서비스를 제공한다[5]. 이것의 동작 방식 역시 애플의 푸시 알림 서비스, 구글의 C2DM 과 매우 유사하다.

푸시 알림 서비스를 이용하는 어플리케이션은 먼저 MPN 서버에 연결하여 단말기를 인증하여 단말기 고유의 URI 를 받고 이를 어플리케이션 지원 서버에 전달한다. 이 URI 는 추후 어플리케이션 지원 서버에서 특정 단말기를 대상으로 메시지를 전달할 때 단말기를 지정하는 용도로 사용된다. 어플리케이션 지원 서버에서 단말기로 푸시 알림 메시지를 전달하기 위해서는 메시지와 대상 단말기 URI 를 MPN 서버로 전달한다. 그러면 MPN 서버는 대상 단말기로 푸시 알림 메시지를 전달한다.

### 4. Google Cloud to Device Messaging Framework

구글의 모바일 기기용 운영체제인 안드로이드(Android)는 애플이나 마이크로소프트와는 달리 멀티태스킹을 지원하여 하나 이상의 프로세스가 동시에 실행될 수 있다. 그렇기 때문에 안드로이드 기기 위에서 동작하는 어플리케이션은 사용자가 명시적으로 완전히 종료시키기 전까지는 다른 어플리케이션이 실행되고 있는 상태에서도 인터넷 연결을 지속할 수 있다.

그러나 이러한 방식은 앞서 살펴본 것처럼 전체적인 실행 성능을 떨어뜨리고, 배터리 소모를 가속화하여 사용자 경험을 악화시킬 수 있다. 또한 사용자가 명시적으로 어플리케이션을 완전히 종료하는 경우 인터넷 연결이 끊어지기 때문에 이 때 어플리케이션 지원 서버는 사용자 단말기로 중요한 정보를 제공할 수 없게 된다.

이러한 문제를 해결하기 위해 구글은 C2DM(Cloud to Device Messaging Framework)을 제공한다 이 기술은 애플의 푸시 알림 서비스와 매우 유사하게 동작한다[6].

C2DM 을 사용하는 어플리케이션은 우선 C2DM 서버에 등록하여 등록(registration) ID 를 발급 받아야 한다. 이렇게 발급 받은 등록 ID 는 어플리케이션 지원 서버로 전달하여 추후 사용자 단말기로 메시지를 전송할 때 특정 단말기를 지정하기 위해 사용된다. 어플리케이션으로 새로운 정보를 알릴 때에는 어플리케이션 지원 서버에서 전송하고자 하는 정보와 대상 기기의 등록 ID 를 C2DM 서버에 전달한다. 그러면 C2DM 서버는 대상 단말기로 해당 메시지를 전송한다. 애플과 마찬가지로 이 때 사용자 단말기에는 해당 어플리케이션이 실행되고 있을 필요는 없다.

애플의 푸시 알림 서비스는 사용자 단말기가 메시지를 수신했을 때 화면에 어떻게 표시하고, 사용자 입력을 받아 어떻게 동작해야 하는지 철저히 제약을 가했지만, 구글의 C2DM 은 이에 대한 제약이 없다. 사용자 단말기가 메시지를 수신하면 브로드캐스트 인텐트를 어플리케이션에 전달한다. (이 때 어플리케이션이 실행 중이지 않으면 자동으로 실행시킨다) 이를 수신한 어플리케이션은 이에 상응하는 적절한 동작을 임의로 결정할

수 있다. 이는 애플의 푸시 알림 서비스에 비해 굉장히 자유도가 높지만 일관된 사용자 경험을 제공하지 못하여 혼란을 야기할 수 있다는 단점이 있다. 또한 C2DM 역시 애플의 푸시 알림 서비스와 마찬가지로 사용자가 알림 메시지를 더 이상 수신하고 싶지 않은 경우 언제든지 이 서비스를 해제할 수 있다.

C2DM 서비스를 이용하기 위해서는 몇 가지 제약이 따른다. 우선 이 기술은 안드로이드 2.2 버전 이상을 요구한다. 2.2 보다 하위 버전에서는 C2DM 을 이용할 수 없는데, 안드로이드 단말기의 업데이트는 구글이 아닌 단말기 제조사에 의해 이루어지기 때문에 모든 안드로이드 단말기가 2.2 버전 이상으로 업데이트 하지 못하고 있는 것이 현실이다. 또한 C2DM 은 구글 서비스를 이용한다. 구글 서비스는 구글의 인증을 받은 단말기에서만 사용 가능하므로, 구글의 인증을 받지 못한 제조사의 단말기는 안드로이드 버전이 2.2 이상이라도 C2DM 을 이용할 수 없다. 그리고 가장 큰 문제는 현재 C2DM 서비스는 모든 개발자가 이용 가능하지 않다. 아직 연구 수준의 서비스이기 때문에 일부 사용을 신청한 개발자에 대해 선택적으로 C2DM 서비스를 사용할 수 있게 되어 있다.

### 5. 기타 유사 방식

앞서 살펴본 바와 같이 C2DM 은 모든 안드로이드 기기에서 동작하지 않고, 모든 개발자가 사용할 수도 없기 때문에 많은 어플리케이션 개발자와 서비스 업체들은 독자적인 대안을 찾고 있다. 그 중 하나가 바로 HTTP Keep Alive 이다[7][8]. 그러나 HTTP 1.1 Keep Alive 는 지속적인 인터넷 연결이 필요한 어플리케이션에 대해 완전한 해법을 제공하지는 않는다. 우선 이 기술을 이용하기 위해서는 구글의 안드로이드처럼 멀티태스킹을 지원해야만 인터넷 연결을 지속할 수 있다. 애플이나 마이크로소프트처럼 멀티태스킹을 지원하지 않는 운영체제에서는 어플리케이션을 종료하면 관련된 자원을 모두 강제로 반환시키거나 사용을 제한하므로 HTTP Keep Alive 연결 또한 더 이상 지속할 수 없게 된다.

가장 큰 문제는 연결 대상이 되는 어플리케이션 지원 서버가 예기치 않게 종료된 경우이다. 이 경우 HTTP 연결이 강제로 끊기는데 이 때 Keep Alive 가 지정된 연결은 최대한 빠르게 연결을 복구하려는 특징이 있다. 어플리케이션 지원 서버가 종료된 경우를 기점으로 여기에 연결되었던 모든 단말기의 HTTP Keep Alive 연결은 동시다발적으로 연결을 복구하기 위한 신호를 짧은 주기로 보낸다. 서버 연결이 완전히 복구될 때까지 주기적으로 이러한 복구 신호를 보내게 되는데 이 때 대량의 네트워크 트래픽이 순간적으로 발생하여 심각한 네트워크 과부하를 야기하게 된다. 이는 대량의 트래픽을 전송하는 분산 서비스 거부 공격으로(Denial-of-service Attack) 간주될 만큼 위협적이다. 특히 모바일 환경에서 3G 네트워크는 통화와 관련된 아주 귀중한 자원이므로 이러한 과부하는 치명적이다.

### 6. 결론

본 고에서는 현재 스마트폰에 탑재된 운영체제가 제공하는 푸시 기술에 대해 분석하였다. 스마트폰은 휴대용이기 때문에 전원과 컴퓨팅자원의 효율적인 사용이 매우 중요하다. 따라서 재난 시 이를 신속하게 알리는 서비스를 위해 현재의 푸시 기술은 많은 문제점을 가지고 있다.

모든 푸시 패킷은 플랫폼 제작사를 통해 온다. 이 영향으로 서버에서부터의 전달지연시간을 보장할 수 없다. 실제 지진발생을 통보하는 어플리케이션의 경우 30 분에서 1 시간이상의 전달지연 시간을 보여 준다. 또한, 우리나라의 경우 국가재난 시 방송, 통신 사업자의 전달의무는 규정되어 있지만, 플랫폼회사의 의무에 대해서는 어떠한 법적 규정도 없다. 따라서, 푸시 기술을 통해 재난경보를 전달할 경우 이에 대한 법적 근거를 빠른 시일 안에 제정해야 할 것이다.

안드로이드 플랫폼의 경우 제공되는 푸시기술을 사용하지 않고 백그라운드 서버를 사용할 수 있다. 하지만, Keep-Alive 패킷으로 인해 불필요한 트래픽을 유발할 수 있다. 또한 서버 장애 시 Keep-Alive 프로토콜에 의해 더 많은 트래픽을 유발하기 때문에 재난과 같은 급박한 상황에서 통신장애를 이리켜 오히려 재난관리에 악영향을 끼칠 수 있다.

현재의 푸시기술은 아직 매우 제약적이다. 따라서 다른 기능과 연동되는 어플리케이션을 구현하기가 매우 어렵다. 예를 들어, 특정지역의 수신기에만 경보가 울리는 지역기반 서비스를 구현하기가 매우 어렵다.

본 고에서 분석한 푸시기술의 여러 문제점을 볼 경우 아직 정부에서 본격적으로 이 기술을 활용한 재난경보서비스를 시작하기엔 아직 이르다는 판단이다. 본 고에서 언급된 문제점이 해결되어야 국가기관이 공신력 있는 서비스를 시작할 수 있을 것이다.

**감사의 말:** 이 연구는 기상청 기상지진기술개발사업 (CATER 2010-1193)의 지원으로 수행되었습니다.

## 7. 참고문헌

- [1] CNN App for iPhone - <http://itunes.apple.com/kr/app/cnn-app-for-iphone-us/id331786748?mt=8>
- [2] Twitter for iPhone, iPad, iPod Touch - <http://itunes.apple.com/us/app/twitter/id333903271?mt=8>
- [3] Facebook for iPhone - <http://itunes.apple.com/app/facebook/id284882215?mt=8>
- [4] About Local Notifications and Push Notifications, Apple - [http://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Introduction/Introduction.html%23/apple\\_ref/doc/uid/TP40008194](http://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Introduction/Introduction.html%23/apple_ref/doc/uid/TP40008194)
- [5] Understanding Microsoft Push Notifications for Windows Phones, Microsoft - [http://windowsteamblog.com/windows\\_phone/b/wpdev/archive/2010/05/03/understanding-](http://windowsteamblog.com/windows_phone/b/wpdev/archive/2010/05/03/understanding-microsoft-push-notifications-for-windows-phones.aspx)

[microsoft-push-notifications-for-windows-phones.aspx](http://windowsteamblog.com/windows_phone/b/wpdev/archive/2010/05/03/understanding-microsoft-push-notifications-for-windows-phones.aspx)

- [6] Android Cloud to Device Messaging Framework, Google - <http://code.google.com/intl/ko-KR/android/c2dm/index.html>
- [7] HTTP persistent connection - [http://en.wikipedia.org/wiki/HTTP\\_persistent\\_connection](http://en.wikipedia.org/wiki/HTTP_persistent_connection)
- [8] RFC 2616 (HTTP/1.1), 8.1 Persistent Connections - <http://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html>