

효율적인 메시지 전송 신뢰성 향상 솔루션

진성빈⁰, 이화섭^{*}

^{0*} 송실대학교 컴퓨터학부

e-mail: jakajaka51@naver.com⁰, hs6959@gmail.com^{*}

Efficient Message Transfer Reliability Improvement Solution

Sung-Bin Jin⁰, Hwa-Seob Lee^{*}

^{0*}Dept. of Computer Science, Soong-Sil University

● 요약 ●

본 논문에서는 서버와 메시지를 주고받을 시 네트워크 환경상의 문제로 발생할 수 있는 메시지 전송 실패 이슈에 대한 신뢰성을 향상시킬 수 있는 3단계 메시지 실패 해결 솔루션을 제안한다. 서버에 업데이트 이슈가 발생시 push 서버를 통해 클라이언트에게 해당 이슈를 전송하게 되는데, 이 때 네트워크 환경상의 문제 등으로 실패 케이스가 발생하는 경우가 있다. 이러한 메시지 전송 실패 이슈에 대한 솔루션으로 재전송 요청, 해당 가능 수행시 서버 DB와 클라이언트 DB 동기화, 학습된 주기를 이용한 폴링(polling) 방식으로 서버로부터 누락된 메시지를 확인 및 실패 처리를 하여 신뢰성을 향상 시키도록 한다.

키워드: 실패 처리(failure treatment), 폴링(polling), 동기화(synchronization), 기계학습(machine learning)

I. Introduction

본 연구는 서버에서의 메시지 전송 시 네트워크 환경상의 문제 등으로 발생하는 실패 이슈에 대한 신뢰성 향상을 그 목적으로 한다. 최근 스마트폰의 보급으로 많은 사용자들이 네트워크를 통해 메시지를 송수신 할 수 있게 되었고, 그에 따라 서버는 스마트폰과 같은 여러 디바이스로 메시지를 실시간으로 푸시해주는 경우가 잦아 졌다. 이러한 상황에서 푸시 서버의 메시지 전송 실패는 금전적 손실뿐만 아니라, 사용자의 신뢰성을 저하시킬 수 있다. 이에 본 논문에서는 3 stage failure treatment solution을 통해 메시지 전송의 신뢰성을 높일 수 있는 방안을 제시한다.

```
{ "multicast_id": 216,
  "success": 3,
  "failure": 3,
  "canonical_ids": 1,
  "results": [
    { "message_id": "1:0408" },
    { "error": "Unavailable" },
    { "error": "InvalidRegistration" },
    { "message_id": "1:1516" },
    { "message_id": "1:2342", "registration_id": "32" },
    { "error": "NotRegistered" }
  ]
}
```

Fig. 1. GCM error code

II. Preliminaries

스마트 디바이스의 대중화로 디바이스 간의 메시지 전송 실패 이슈가 많아졌다. 현재 디바이스 간 메시지 전송을 위한 많은 서비스가 제공되고 있지만, 많은 서비스가 잠정적인 메시지 전송 실패 위험을 가지고 있다.

대표적인 예로 구글 GCM 푸시 서비스의 경우에는 메시지가 실패할 시 Error code를 반환해 주는데 Fig. 1과 같이 다수의 사용자에게 푸시를 하는 경우 Unavailable이라는 에러 메시지를 출력하고 푸시 재시도를 권고하고 있다. Error 발생 시 재전송을 권고하고 있는 것은 GCM 서비스의 메시지 전송 성공률이 100% 보장되는 것이 아니라는 것을 보여주고 있다.

이와 같이 기하급수적으로 증가하는 데이터 통신량으로 인해 전송 지연 문제와 트래픽 오버플로우, QOS(Quality of Service)감소문제가 발생할 수 있다. 모바일 네트워크 환경 특성상 무선망을 사용하기 때문에 링크 에러와 채널 간섭이 존재하는 문제가 있다. 이에 대한 해결책으로 폴링 방식을 통해 서버DB와 클라이언트DB와의 동기화 처리를 한다.

그러나 폴링 요청주기를 짧게 설정하면 불필요한 요청/응답 트래픽이 많이 발생하게 된다. 이는 서비스의 성능을 저하시키는 문제를 야기하는 또 다른 원인이 된다. 또한 반대로 폴링 요청주기를 너무 길게 설정하게 되면 실시간으로 업데이트 된 데이터를 필요시점에 제대로 동기화 하지 못하는 문제가 발생할 수 있게 된다.

따라서 사용자의 해당 서비스 사용주기를 학습하여 폴링주기를 지속적으로 갱신기능이 필요하다. 이때, training set에 오차가 0인 방식을 취하게 되면 향후 test set들에 대해서 overfitting된 결과를 야기할 수 있기에 가중치를 두어 학습을 하도록 한다.

III. The Proposed Scheme

메시지를 송수신 실패를 처리하기 위해 3단계의 절차를 거치도록 한다. 이는 IoT, 스마트 폰, 스마트 디바이스 등 다양한 장치, 서비스에 대한 메시지를 실시간으로 푸시 전송하여 신뢰성을 향상시킬 수 있는 솔루션이라 할 수 있다. 3 stage failure treatment solution의 각 단계는 다음과 같다.

3.1 재전송 요청

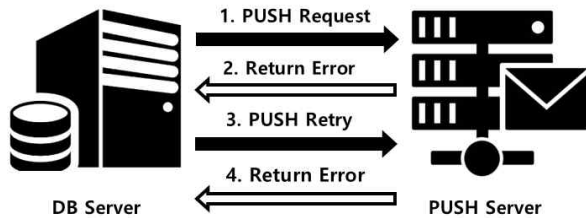


Fig. 2. 메시지 전송 실패 시 재전송 요청

푸시 서버에서 클라이언트의 메시지 전송 시 실패 이슈가 발생할 수 있기 때문에 Fig. 2에서와 같이 실패 시 가장 원초적인 방법으로 리턴 된 푸시 결과 중 실패한 클라이언트들에게 재전송을 요청하여 1차적으로 메시지 전송 실패 처리를 수행한다.

3.2 서비스 요청 시 동기화

사용자가 해당 서버에 서비스를 요청하는 시점에 디바이스는 서버로 데이터 동기화를 요청하도록 한다. 전체적인 동기화 요청 절차는 Fig. 3과 같다.

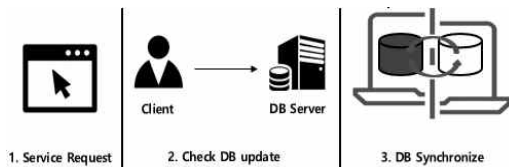


Fig. 3. 서비스 요청 시 확인 후 동기화 처리

앞서 첫 번째 단계의 재전송 요청이 실패했을 경우를 대비해, 서버는 사용자가 서비스를 요청하는 시점에 서버 DB와 클라이언트 DB를 비교하여 최신임을 확인하고 그렇지 않을 경우 동기화를 진행하도록 한다. 현재 안드로이드 어플리케이션으로 메시지를 푸시해주는 구글 GCM의 경우 Fig. 4와 같은 메시지 실패 통계를 보이고 있다.

GCM MESSAGES ON FEB 7, 2016

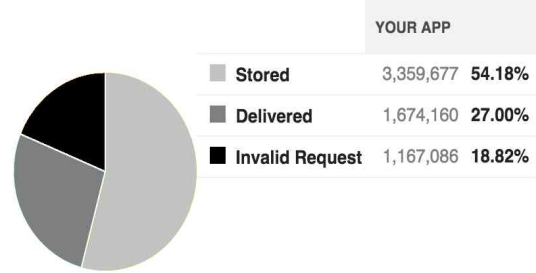


Fig. 4. GCM 메시지 통계

Fig. 4는 실제 안드로이드 어플리케이션에서 GCM을 통해 디바이스들로 푸시를 보낸 통계를 나타내고 있고, 제대로 Delivered된 푸시는 27%정도이다. 54.18%의 stored data는 디바이스들로부터 온 것인데, 이는 푸시를 보냈으나 대상 디바이스들이 전원이 꺼져있거나, 네트워크 연결이 끊기는 등 통신이 불가능한 상황을 나타낸다. 이런 경우 1번은 재전송을 요청하여 푸시를 다시 보낼 수 있지만, 계속해서 디바이스들이 푸시를 받을 수 없는 상황이 발생 할 수 있다. 따라서 서비스를 요청하는 시점에서 동기화 작업을 통해 메시지 전송의 신뢰성을 최대한으로 확보할 수 있다.

3.3 폴링 주기 학습 및 동기화

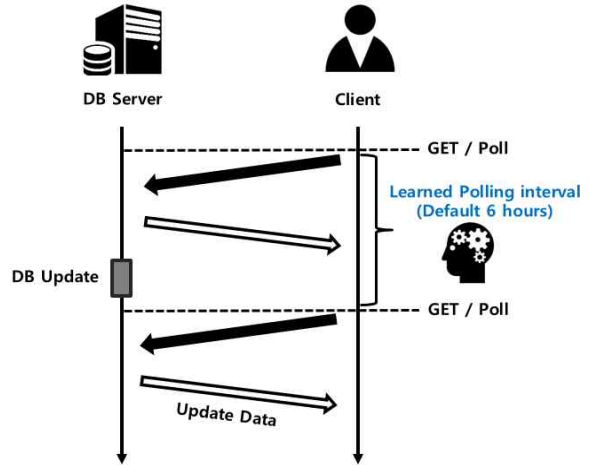


Fig. 5. 폴링 주기 학습 및 동기화

3번째 실패 이슈 해결 솔루션은 100% 메시지 실패 처리를 위해 백그라운드에서 학습된 주기로 폴링하여 예상치 못한 문제에 대한 동기화 처리 방식이다. 이 때, 사용자의 data로부터 학습은 Regularized linear regression을 이용하여 최적의 주기를 찾아내도록 한다. 학습한 training set을 토대로 앞으로의 test set을 적용 시켰을 때의 랑다값으로 가중치를 주어 오차조정을 한다.

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

$$\min_{\theta} J(\theta)$$

사용자가 해당 서비스를 이용하는 주기를 DB에 쌓아가고 이를 학습하며 초기 설정되었던 폴링 주기를 조정하며 동기화처리 작업을 한다.

IV. Conclusions

현재 상용되고 있는 대부분의 서비스는 데이터를 관리하고 처리하고 있다. 메시지 전송 시 데이터 송수신이 제대로 이루어지지 않는다면 서비스의 신뢰성을 저하시키는 문제를 발생시키게 될 것이다.

본 논문에서는 서버에서 클라이언트로 메시지 전달단계에서 발생할 수 있는 신뢰성 문제분석과 3 stage failure treatment solution을 제시하였다.

제시된 솔루션을 통해, 다양한 디바이스에서의 메시지 전송 신뢰성을 높여 최적화된 메시지 전송에 도움이 될 것이다. 최종적으로는 더욱 효율적이고 신뢰성 있는 서비스 개발이 가능하고 소프트웨어의 품질 향상을 기대할 수 있다.

References

- [1] Google Developers, <https://developers.google.com/cloud-messaging/http#example-responses>
- [2] Kanghee Kim, Jinjoo Yeo, JinHyuk Kim, and Sang Bang Choi, "The Design and Implementation of Network Measurement System for Mobile Platforms", Journal of The Institute of Electronics Engineers of Korea Vol. 50, NO. 2, Feb 2013
- [3] Andrew Ng, Machine Learning Lecture, <https://www.coursera.org/learn/machine-learning>
- [4] Cisco, Defining the Polling Intervals, http://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/collaboration_manager/1-1/user/guide/cm_ug/polling.pdf
- [5] Wen Fu, Data Synchronization In a Network-volatile Mobile Ecosystem, Chapter 2.1 Unreliable wireless Network <https://ecommons.usask.ca/xmlui/bitstream/handle/10388/ETD-2014-09-1765/FU-THESIS.pdf?sequence=3&is>