

MQTT 프로토콜 기반의 스마트 출석관리 시스템의 설계

안성우

경남정보대학교

Design of the Smart Attendance Management System based on the MQTT Protocol

Sungwoo Ahn

Kyungnam College of Information & Technology

E-mail : ahnsw@kit.ac.kr

요 약

스마트폰이 생활의 필수품으로 자리잡아감에 따라 인간 편의를 위해 스마트폰을 기본수단으로 많은 서비스가 제공되고 있다. 또한, 모바일 기반 서비스의 고도화로 스마트폰에 부착된 센서와 네트워크 통신을 적극적으로 활용하는 애플리케이션의 수가 증가하고 있다. 이로 인해 배터리 사용시간 단축, 통신량 증가 등 하드웨어 리소스의 부하 증대에 대한 사용자의 불만이 꾸준히 제기되고 있다. 본 논문에서는 최근 대학에서 도입하고 있는 비콘 기반 출석관리 환경에서 이러한 문제를 해결하기 위해 MQTT 프로토콜 기반 통신을 적용한 시스템을 제안하고 설계한다. 제안하는 출석관리 시스템에서는 발간/구독 방식의 경량 프로토콜인 MQTT를 사용함으로써 네트워크 부하 및 배터리 낭비를 최소화 하고자 한다.

키워드

MQTT, Beacon, Publish/Subscribe, Push Notification Service

1. 서 론

스마트폰의 대중화로 인해 스마트폰 사용자는 해마다 꾸준히 증가하고 있으며 2015년 8월 현재 국내 무선전화 가입자 중 70퍼센트 이상이 스마트폰 가입자로 대부분의 국민이 스마트폰을 사용하고 있다고 해도 과언이 아니다[1]. 최근의 스마트폰 단말기는 초창기와 달리 하드웨어 성능이 비약적으로 발전하였고 다양한 종류의 센서 탑재로 응용의 범위도 함께 증가하고 있다. 이에 따라 하드웨어의 장점을 최대한 활용하면서 고도화된 서비스가 계속해서 출시되고 있다.

모바일 서비스의 고도화로 사용자의 편의성이 증대되는 이면에 사용자의 불만도 함께 증가하고 있다. 스마트폰 사용자의 단말에는 사용 서비스에 따라 수십 개 이상의 앱이 설치되고 있으며, 많은 수의 앱에서 센서 및 서버와의 통신을 활용한다. 이로 인해 백그라운드로 앱이 동작하여 배터리 사용시간을 단축하며, 푸쉬 메시지가 경쟁적으로 송신되어 통신량 증가로 이어지고 있다. 불만을 해결하기 위해 사용자들은 백그라운드 동작 앱을

통제하기 위한 추가 앱을 설치하고, 상당수의 푸쉬 메시지를 스팸으로 처리하고 있는 실정이다.

최근 출석관리의 편의성을 위해 대학에서 도입하고 있는 비콘 기반의 스마트 출석관리 시스템도 BLE 통신 및 푸쉬 기반 서버 통신을 활용하기 때문에 통신량과 배터리 소비 부담을 증가시키고 있다. 본 논문에서는 이러한 문제를 해결하기 위한 방법으로 발간/구독 기반의 양방향 서버 통신을 적용한 출석관리 시스템을 제안하고자 한다. 제안한 스마트 출석관리 시스템은 모바일 메시징을 위한 경량 프로토콜인 MQTT[2]를 사용함으로써 서버와 교환되는 메시지 양과 배터리 낭비를 최소화한다. 또한, 사용자 단말기 운영체제별로 제공되는 별도의 푸쉬 알림 서비스를 사용하지 않고 단일 메시지 전송 방식으로 통일시킴으로써 구현 및 유지보수의 편의성을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서 스마트 출석관리 시스템의 대상환경 및 문제정의를 기술한다. 3장에서는 MQTT 프로토콜 기반의 스마트 출석관리 시스템을 제안한 후 설계하고, 4장에서 결론 및 향후 연구를 기술한다.

II. 대상환경 및 문제정의

본 논문에서는 그림 1과 같이 비콘 기반의 스마트 출석관리 시스템을 대상으로 한다. 시스템에서 사용되는 정보는 출석정보와 비콘 상태정보이며, 출석정보는 강의실에 설치된 비콘 신호를 학생의 스마트폰을 통해서 수신하여 처리된다. 비콘 상태정보는 교수의 스마트폰에서 비콘의 상태정보를 수신하여 서버로 전달된다. 모든 정보는 서버에 저장되며 해당 정보의 요청이 발생할 시 요청자에게 자동으로 전달되도록 한다. 그림 1의 참여자는 학생, 교수, 관리자로 나누어지며 각각 다음의 정보수집 및 요청을 담당한다.

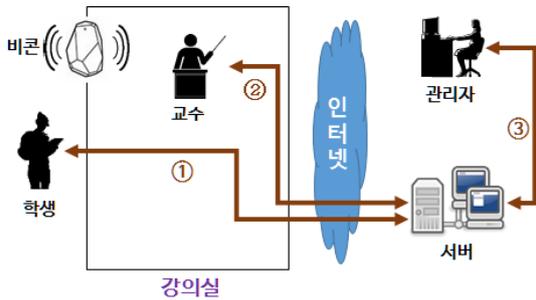


그림 1. 스마트 출석관리 시스템 대상 환경

- ① **학생** : 출석 상태정보의 생산, 출석 확정정보의 수신
 - 비콘 신호의 수신 상태를 파악하여 출석, 지각, 조퇴 등의 정보를 서버로 전송
 - 교수로부터 전달된 자신의 출석 확정정보를 수신
- ② **교수** : 출석 상태정보 수신, 출석 확정정보, 비콘 상태정보의 생산
 - 강의실에 출입한 학생들의 출석 상태정보를 서버로부터 수신
 - 출석을 확인한 후 확정정보를 서버로 전송
 - 비콘의 상태정보(배터리 정보, 신호 강도 등)를 비콘으로부터 전달받아 임계치를 넘어섰을 때 서버로 전송
- ③ **관리자** : 비콘 상태정보의 수신
 - 교수로부터 전달되는 비콘 상태정보를 수신하여 각 강의실에 설치된 비콘의 정보를 모니터링

비콘을 사용하여 학생의 출입정보를 확인하고 출석관리에 활용하는 목적은 자동으로 정보를 갱신하고 확인하는 것에 있다. 따라서 정보를 서버에 전달하고 요청하는 작업에 시스템 참여자의 개입을 최소화하는 것이 필요하다.

서버로부터 자동으로 정보를 전달받기 위해서 기존의 애플리케이션은 푸시 알림 서비스(Push Notification Service)를 사용하고 있다. 그러나 요즘 스마트폰 단말기에는 SNS를 비롯하여 푸시

알림 서비스를 사용하는 애플리케이션이 많이 설치되어 푸시 메시지가 경쟁적으로 수신되고 있다. 이러한 이유로 푸시 메시지를 전달받는 백그라운드 애플리케이션들이 통신량 증가와 배터리 낭비의 원인 중 하나로 지목되고 있다[3]. 따라서 리소스 소비를 최소화하면서 자동으로 원하는 정보를 교환하기 위하여 기존의 푸시 알림 방식이 아닌 새로운 방식으로 사용할 필요가 있다.

III. MQTT 기반의 스마트 출석관리 시스템

본 논문에서는 서버와의 통신에 따른 배터리 소모에 대한 부담을 줄이기 위해 발간/구독을 기반으로 한 경량 프로토콜인 MQTT(Message Queue Telemetry Transport)[2]를 메시지 교환 수단으로 사용한다. MQTT는 제한된 환경에서 낮은 전력 및 대역폭, 높은 지연 등 신뢰할 수 없는 네트워크를 위하여 사용할 수 있도록 설계된 프로토콜로 소형기기의 제어와 센서정보 수집에 적합하다[4]. 실제로 MQTT는 기존의 HTTP 프로토콜보다 배터리를 효율적으로 사용할 수 있다는 것이 검증되었다[5].

MQTT는 다음과 같은 특징을 가지고 있는 모바일에 최적화된 양방향 통신 규약이다. 첫째, 보편성과 유연성을 가지고 있다. 현재 OASIS 그룹에서 표준으로 채택되어 다양한 환경의 클라이언트 라이브러리 및 오픈소스를 지원하고 있다. 둘째, 경량성과 신속성을 가지고 있다. MQTT 프로토콜에서 교환하는 메시지는 프로토콜 오버헤드를 줄이기 위해서 최적화된 단순구조를 가지고 있으며 가장 작은 메시지 사이즈는 2바이트까지 가능하다. 셋째, 메시지 전달의 신뢰성을 제공한다. 즉, 메시지 전달 보장 및 중복 허용 여부에 대해서 3가지의 QoS 레벨을 제공하여 사용 환경에 따라 적합하게 선택할 수 있도록 하였다. 마지막으로 메시지 전달의 안정성을 제공한다. 모바일 네트워크는 특유의 끊김과 불안정에 대비한 보정 기능이 프로토콜에 필히 포함되어야 한다. MQTT는 이를 위해 네트워크 불안정으로 연결이 끊긴 후에 재연결 시의 보정 기능이 프로토콜 수준에 구현되어 있다.

기존의 애플리케이션은 서버에서 원하는 정보를 자동으로 수신하기 위해 푸시 알림 서비스(Push Notification Service)를 주로 이용하고 있다. 푸시 알림 서비스는 구글의 GCM, 애플의 APNS와 같이 운영체제 벤더들이 제공하는 공공 푸시 알림 서비스를 사용하거나 커스텀 푸시 알림 서버를 구축하여 사용할 수 있다. 그러나 기존 푸시 알림 서비스는 대부분 서버로부터의 단방향 푸시만을 지원하는 경우가 많다. 또한 공공 푸시 알림 서비스를 사용하는 경우 운영체제별로 다른 푸시 알림 서비스 구현이 필요하며 메시징 서비스 퀄리티에 대한 품질 보증을 제공하지 않아 고품질의 서비스가 어렵다. 커스텀 푸시 알림 서버

를 사용하는 경우도 기능구현 및 유지보수를 위한 노력이 추가되며 최적화가 되지 않은 경우 서버와의 연결 유지에 프로토콜 오버헤드가 발생하여 배터리 낭비를 가져올 수 있다. 표 1은 MQTT와 대표적인 공공 푸쉬 알림 서비스인 GCM, APNS의 기능을 비교하고 있다[3].

표 1. 공공 푸쉬 알림 서비스와 MQTT의 비교

특징	GCM	APNS	MQTT
통신 방식	단방향 (Push)	단방향 (Push)	양방향 (Push/Sub)
지원 콘텐츠	텍스트	텍스트	모든 데이터 유형(최대 256MB)
QoS 지원	미지원	미지원	지원
동시 송신 능력	Push 수신 자들에 대해 개별적으로 송신해야 함	Push 수신 자들에 대해 개별적으로 송신해야 함	많은 수의 동시 사용자에게 대한 메시지 발행
지원 플랫폼	Android	iOS, Mac OS X	대부분의 모바일 플랫폼, 서버 플랫폼 및 개발 환경 지원

그림 2는 2장에서 설명한 스마트 출석관리 시스템을 구축하기 위한 시스템 구성을 보여주고 있다. 주요 구성 요소는 역할에 따라 센서, 클라이언트, 네트워크(LAN, WAN), 서버의 4부분으로 나누어진다.

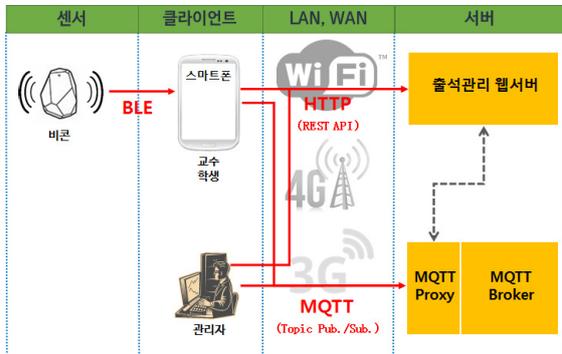


그림 2. 스마트 출석관리 시스템 구성

- **센서** : 스마트 출석관리 시스템은 강의실에 설치된 비콘(iBeacon) 신호를 수신하여 학생들의 강의실 출입여부를 판별한다. iBeacon은 애플에서 2013년 처음 소개한 기술이며 애플은 iBeacon과의 통신에서 발생하는 전력소모를 줄이기 위해 BLE(Bluetooth Low Energy) 통신 방식 사용을 제안하였다[6].
- **클라이언트** : 참여하는 클라이언트의 역할에 따라 교수, 학생, 관리자가 있다. 교수와 학생은 공통적으로 자신의 스마트폰을 통하여 비콘의 신호를 받아 메시지를 서버로 전달한다. 2장에서 살펴봤듯이 학생은 출석

상태정보를 생산하고 출석 확정정보를 수신한다. 교수는 출석 상태정보를 수신, 출석 확정정보와 비콘 상태정보를 생산한다. 관리자는 비콘 상태정보를 수신한다.

- **네트워크(LAN, WAN)** : 통신 인프라는 WiFi, 4G LTE, 3G 통신 등 무선 인터넷과 유선 랜 등 현재 사용되고 있는 LAN, WAN 통신 인프라를 사용한다. 본 시스템에서는 서버와의 통신 시 효율성과 성능을 위해 요청과 응답채널을 별도로 분리한다.
 - **요청 채널** : HTTP를 사용한다. HTTP 기반의 요청을 위해 다양한 REST API 및 라이브러리를 활용한다.
 - **응답 채널** : MQTT를 사용한다. 교수, 학생, 관리자는 서버로 전달할 정보, 서버에서 수신할 정보를 위한 토픽을 생성한다. 정보를 서버에 전달하기 위해서는 토픽을 발간(Publish)하고, 서버에서 수신하기 위해서는 토픽을 구독(Subscribe) 한다.
- **출석관리 웹서버** : HTTP 기반의 요청 및 응답을 처리한다. 클라이언트의 최초 접속 시 인증을 하며 인증 후 클라이언트 및 세션 정보를 생성하여 관리한다. 또한, 토픽 정보를 관리하여 MQTT Proxy가 사용할 수 있도록 한다.
- **MQTT Proxy** : 클라이언트의 발간/구독 요청을 MQTT Broker에게 넘겨주기 전에 권한 확인과 인증을 담당한다. 이를 위해 출석관리 웹서버로부터 세션정보를 넘겨받아 클라이언트를 인증하고 토픽정보를 이용하여 발간, 구독 요청 토픽에 대해 클라이언트가 권한을 가지고 있는지 검증한다.
- **MQTT Broker** : 토픽의 발간/구독을 위한 MQTT 고유의 작업을 담당한다.

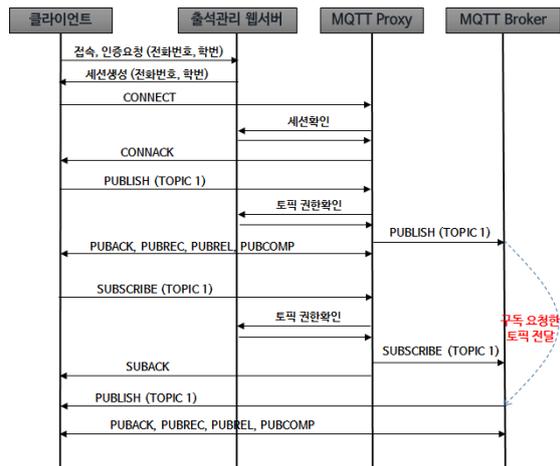


그림 3. MQTT 프로토콜 기반 메시지 교환 시퀀스

그림 3은 MQTT 프로토콜을 기반으로 한 스마트 출석관리 시스템에서 각 구성요소 간 메시지

교환 프로세스를 도식화한 것이다. 각 프로세스에 대한 설명은 다음과 같다.

- “클라이언트”는 “출석관리 웹서버”에 접속 하면서 인증요청을 한다. 인증요청 시 스마트폰의 전화번호와 학번을 함께 전달함으로써 “출석관리 웹서버”가 해당 정보로 세션(Session)을 생성하고 세션ID를 발급할 수 있도록 한다.
- “출석관리 웹서버”는 인증이 성공하면 해당 클라이언트에 대한 세션ID를 발급하여 저장하고 “클라이언트”에게 세션을 발행한다.
- “클라이언트”는 토픽을 발간 또는 구독하기 위해 “MQTT Proxy”에게 CONNECT 패킷을 보낸다. CONNECT 패킷의 Payload에는 세션ID를 함께 전달하여 클라이언트 인증에 사용하도록 한다.
- “MQTT Proxy”는 CONNECT 패킷에서 세션ID를 추출하여 “출석관리 웹서버”를 통해 클라이언트를 인증한다.
- 연결이 성공한 후 “클라이언트”는 토픽 발간을 위해 MQTT의 PUBLISH 패킷을 “MQTT Proxy”에게 전달한다. “MQTT Proxy”는 “출석관리 웹서버”를 통해 토픽 권한을 확인한 후 PUBLISH 패킷을 “MQTT Broker”에게 넘긴다. 이후 PUBLISH 패킷 수신에 대한 응답 패킷을 QoS 레벨별 절차에 따라 “클라이언트”에게 전달한다.
- 연결이 성공한 후 “클라이언트”는 토픽 구독을 위해 MQTT의 SUBSCRIBE 패킷을 “MQTT Proxy”에게 전달한다. “MQTT Proxy”는 토픽 권한 확인 후 SUBSCRIBE 패킷을 “MQTT Broker”에게 넘긴다.
- “MQTT Broker”는 SUBSCRIBE 요청한 토픽이 PUBLISH 패킷을 통해 도착하면 해당 “클라이언트”에게 PUBLISH 패킷을 보냄으로써 푸쉬 알림 서비스를 한다.

• 출석 상태정보

/attend_status/room_number/lecture_code/date,time/phone_number,ID

• 출석 확정정보

/attend_result/lecture_code/date/period/ID/result

• 비콘 상태정보

/beacon_status/room_number/battery_present
/beacon_status/room_number/broadcasting_power
...

그림 4. 발간/구독되는 토픽(Topic)

그림 4은 MQTT 프로토콜에서 발간/구독 패킷으로 전달되는 토픽을 보여주고 있다. 사용되는 토픽은 출석 상태정보, 출석 확정정보, 그리고 비콘 상태정보이다. 출석 상태정보는 학생의 스마트폰에서 발간되는 토픽으로 교수 스마트폰에서 구

독을 통해 푸쉬 받을 수 있다. 출석 확정정보는 교수의 스마트폰에서 발간되는 토픽으로 서버에 최종 저장되는 출석정보이며 학생 스마트폰에서 이 토픽을 구독함으로써 자신의 확정 출석정보를 실시간 확인할 수 있다. 비콘 상태정보는 교수의 스마트폰에서 발간되는 토픽으로 비콘의 다양한 상태정보를 포함한다. 메시지 전달의 효율성을 위해 비콘의 상태정보를 주기적으로 보내지 않고 각 상태에 대한 임계치를 넘어섰을 때만 토픽으로 생성해서 발간하도록 한다.

IV. 결론 및 향후연구

본 논문에서는 대학교에서 출석 관리를 효과적으로 할 수 있는 스마트 출석관리 시스템을 제안하고 설계하였다. 시스템은 비콘과의 통신을 기반으로 하고 있으며 서버로부터 푸쉬 알림 서비스를 받기 때문에 배터리 사용시간 단축과 통신량 증가의 문제가 발생할 수 있다. 이러한 리소스 부하를 감소시키기 위해 본 논문에서는 MQTT 프로토콜을 사용하여 서버와의 메시지 교환을 하도록 제안하였다. MQTT를 사용함으로써 출석관리 시스템은 기존의 방식과 비교하여 교환되는 메시지의 사이즈를 줄이는 것이 가능하였다. 동시에 발간/구독 방식을 통하여 서버와의 연결유지로 발생하는 통신 오버헤드 및 배터리 소비를 줄일 수 있었다. 향후 연구에서는 제안된 출석관리 시스템을 구현하여 MQTT 적용에 따른 성능을 검증하는 것이 필요하다.

참고문헌

- [1] 김현희, “[통신정책국]무선통신 가입자 통계 (’15, 8월 기준),” 미래창조과학부, 2015.
- [2] OASIS, “MQTT Version 3.1.1,” <http://mqtt.org>, 2014.
- [3] 지용득, “MQTT 기반 관리형 Push 메시지,” 한국 IBM, 2013.
- [4] 심승현, 김학범, “사물인터넷과 MQTT 기술,” 정보보호학회지, 제24권, 제6호, pp.37-47, 2014.
- [5] S. Nicholas, “Power Profiling: HTTP Long Polling vs. MQTT with SSL, on Android,” <http://stephendnicholas.com/archives/1217>, 2012.
- [6] 김대엽, 김수형, 진승현, “iBeacon 기술 동향 및 문제점 분석,” 한국컴퓨터종합학술대회, pp.390-392, 2014.