

논문 2024-3-8 <http://dx.doi.org/10.29056/jsav.2024.09.08>

오픈소스 기반 MQTT 프로토콜을 이용한 스마트 미터 실시간 데이터 수집 및 관리

송재숙*, 배선영**†

Open Source based Smart Meter Real-time Data Acquisition and Management using MQTT Protocol

Jea-Suk Song*, Sun-Young Bae**†

요 약

현재 IoT 기술의 급속한 발전은 산업 전 분야에 혁신적인 변화를 가져왔고, 스마트 그리드 AMI 시스템에도 큰 영향을 줄 수 있는 상황이다. 실제로 AMI 시스템 내 핵심 구성요소인 스마트 미터는 에너지 소비를 실시간으로 모니터링하고 데이터를 수집하는 디지털 장치로 이를 통해 에너지 효율성을 향상시키고, 소비자의 에너지 사용 패턴을 분석하여 효율적인 에너지 관리 방안을 제시할 수 있다. 그러나 향후 통신사업자의 IoT 인프라 기반 하에 수십만 개의 스마트 미터가 대규모로 설치될 경우 방대한 양의 데이터를 신속하고 안정적으로 전송하고 관리하는 것이 필수적인데, 이러한 AMI 시스템에서 효율적으로 데이터를 전송하기 위해 MQTT IoT 프로토콜을 사용할 수 있을 것이다. MQTT는 최소한의 대역폭과 전력 요구 사항으로 장치 간에 메시지를 전송하는 경량의 게시-구독 네트워크 프로토콜로 스마트 미터에서 일반적으로 생성되는 대량의 작은 데이터 패킷에 적합하므로 실시간 검침 데이터 전송에 사용할 수 있다. 따라서 본 연구에서는 향후 통신사업자의 IoT 인프라를 활용한다는 가정 하에 다양한 빅데이터 오픈소스를 기반으로 MQTT 프로토콜을 이용한 스마트 미터 실시간 데이터 수집 및 관리 시스템 프로토타입을 설계하고 구현하였다.

Abstract

Currently, the rapid development of IoT technology has brought innovative changes to all industries and can have a significant impact on the smart grid AMI system. In fact, a smart meter, a key component within the AMI system, is a digital device that monitors energy consumption in real time and collects data. Through this, it improves energy efficiency and provides efficient energy management measures by analyzing consumers' energy use patterns. However, in the future, when hundreds of thousands of smart meters are installed on a large scale based on telecommunications carriers' IoT infrastructure, it is essential to quickly and reliably transmit and manage massive amounts of data. In order to transmit data efficiently in such AMI systems, MQTT IoT protocol may be used. The MQTT IoT protocol can be used to transmit data efficiently in AMI. MQTT is a lightweight publish-subscribe network protocol that transfers messages between devices with minimal bandwidth and power requirements and is suitable for the large amounts of small data packets typically generated by smart meters, so it can be used to transmit real-time meter reading data. Therefore, in this study, we designed and implemented a prototype smart meter real-time data collection and management system using the MQTT protocol based on various big data open sources under the assumption that the IoT infrastructure of telecommunication operators will be utilized in the future.

한글키워드 : MQTT, 스마트 그리드, 첨단계량인프라, 스마트 미터, 사물인터넷, 오픈 소스

keywords : MQTT, Smart Grid, AMI, Smart Meter, IoT, Open Source

* 우송대학교 바이오 헬스사업단

접수일자: 2024.08.28. 심사완료: 2024.09.04.

** 배재대학교 전기전자공학과

게재확정: 2024.09.20.

† 교신저자: 배선영(email: sypae@pcu.ac.kr)

1. 서론

스마트 그리드는 기존 전력 산업에 최신 ICT (Information and Communication Technology)를 융합하고 전력회사와 소비자(일반 고객, 시스템 운영자 등) 간 양방향 통신을 가능하게 함으로써 기존 전력 서비스의 효율성 및 신뢰성을 제고하여 기존 전력망을 강화하고 다양한 새로운 전력 서비스가 창출될 수 있도록 한다[1][2]. AMI (Advanced Metering Infrastructure)의 경우 수용가에 설치된 스마트 미터로부터 정보를 얻고 다양한 시간 간격별로 전력 소비량을 측정하며, 스마트 미터는 모니터링과 과금을 위해 전력 회사와 계통 운영자에게 추가적인 정보를 제공하며 전력회사와 수용가 간 양방향 통신을 제공하고 에너지 사용량에 대한 상세한 모니터링, 제어 및 분석을 용이하게 한다[3].

AMI 시스템은 일반적으로 전력 소비 관련 실시간 데이터를 제공하기 위해 스마트 미터, 통신 네트워크 및 데이터 관리 시스템으로 구성된다. 데이터 집중 장치에서 고속/저속 PLC(Power Line Communication), WiSUN 등 다양한 유무선 통신기술을 이용해 구축된 NAN (Neighborhood Area Network)을 통해 많은 스마트 미터로부터 정보를 수집하며 유무선 장거리 통신 기술을 이용해 이러한 데이터 집중 장치가 수집한 정보를 특정 지역을 담당하는 HES(Head End System)에 전달하고 지역 담당 HES가 중앙의 전력회사 미터 데이터 관리 시스템(Meter Data Management System)에 전달하는 형태이다[4]. 그러나 NAN 내 일간 적시검침율(2시간 이내 검침할 수 있는 확률)과 놓여준 지역의 경제성 측면에서 AMI 시스템 구축에 어려움을 겪고 있다[5][6].

현재 IoT(Internet of Things) 기술의 급속한 발전은 산업 전 분야에 혁신적인 변화를 가져왔

고 NB-IoT/Cat.M1 모듈 가격의 하락으로 많은 통신사업자가 IoT 적용 범위를 확대함에 기존 스마트 그리드 AMI 시스템에도 큰 영향을 줄 수 있는 상황이다. 실제로 통신사업자들의 IoT 통신 인프라를 활용하여 급속도로 무수히 많은 IoT 장치들이 우리 주변에 연결되고 있으며 다양한 통신 프로토콜을 이용하여 유용한 서비스를 제공하고 있다. 또한 스마트 그리드 관련해서도 통신사업자의 경우 통신 기술 관련해서 AMI NAN의 대표적 통신기술인 PLC 대비 향후 LTE 기술의 상대적 가격경쟁력에 의해 LTE 기술이 보다 강화될 것으로 전망하고 있다[7]. 따라서 언젠가 AMI 시스템도 통신사업자가 제공하는 IoT 통신 인프라를 활용할 수 있을 것이며, 향후 IoT 통신 인프라를 이용한 AMI 시스템이 구축되는 경우 스마트 미터와 같은 임베디드 디바이스에 대해 비동기 통신을 가능하게 해주는 경량 메시징 프로토콜인 MQTT(Message Queuing Telemetry Transport) 프로토콜을 이용할 수 있을 것이다. 물론 이외에 주요 IoT 통신 프로토콜인 CoAP(Constrained Application Protocol) 등을 사용할 수 있지만 CoAP의 경우 UDP 기반으로 AMI의 경우 전력사용량 데이터가 과금에 사용된다는 신뢰성 요건 측면에서 배제하였으며, MQTT 프로토콜의 경우 Eclipse 재단에서 2018년 IoT 개발자 설문조사에서 IoT 개발자들이 가장 선호하는 메시징 프로토콜로 선정되어 보안, 신뢰성 등을 염두에 개발된 프로토콜이다[8].

따라서 향후 IoT 통신 인프라 기반 하에서 스마트 미터 통신 관련 MQTT 프로토콜을 이용하는 경우 AMI 시스템에서 데이터 집중 장치를 배제하고 HES에 MQTT 브로커를 설치하여 많은 스마트 미터들로부터 실시간 데이터를 수집할 수 있을 것이다[9]. 또한 IoT 기술을 확산하려는 통신사업자의 네트워크를 저렴하게 사용할 수 있으면서 통신 속도, 지연 시간 등의 AMI 통신 요건

을 충족시킬 수 있다[10]. 일반적으로 NB-IoT의 최대 업로드 및 다운로드 속도는 각각 160Kbps 및 120Kbps 속도이고, LTE Cat.M1은 1Mbps의 최대 업로드 및 다운로드 속도와 10~15ms의 낮은 대기 시간을 제공한다[11].

본 논문에서는 스마트 미터 시뮬레이터를 이용하여 스마트 미터에 대한 임의의 검침 데이터 수집과 관리를 위해 HES 역할의 MQTT 브로커, 빅데이터 시스템 간 흐름 관리, 상위 시스템 연동용 비동기 메시징 플랫폼 등을 빅데이터 오픈소스 기반 하에 MQTT 프로토콜을 이용한 스마트 미터 실시간 데이터 수집 및 관리 시스템을 제안하였다. 제안한 시스템을 설계 후 구현하여 실험해본 결과, 스마트 미터 시뮬레이터를 통한 실시간 검침 데이터가 성공적으로 게시, 전달되어 DB에 저장되었음을 확인하였고 최종적으로 아파치 NiFi를 이용한 MySQL DB 데이터의 ActiveMQ 전송도 테스트 및 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문의 기본 이론이 되는 MQTT 프로토콜에 대해 정의한 다음, 3장에서 제안하는 스마트 미터 실시간 데이터 수집 및 관리 시스템을 설명한다. 여기에서는 제안한 시스템의 전체 개요와 개발 환경에 대해 서술한다. 그리고 4장에서는 제안한 시스템을 테스트한 후 연구결과를 서술하고 5장에서는 결론을 맺는다.

2. 관련기술

2.1 MQTT

MQTT 프로토콜은 Message Queuing Telemetry Transport의 약자로 OASIS와 ISO에서 표준화한 게시/구독 메시징 전송 프로토콜로 IoT 메시징을 위한 사실상 표준으로 인터넷을 통해 IoT 장치들을 연결하는 확장 가능하고 안정

적인 방법을 제공한다. MQTT는 가볍고, 개방적이며, 간단하고, 구현하기 쉽게 설계되어 있어 적은 코드 풋프린트가 필요하거나 네트워크 대역폭이 중요한 기계 대 기계(M2M) 및 IoT 컨텍스트에서 통신하는 것과 같은 제약이 있는 환경을 포함하여 많은 상황에서 사용하기에 이상적이다.

실제로 MQTT 프로토콜은 TCP/IP 또는 순서가 있고 손실 없는 양방향 연결을 제공하는 다른 네트워크 프로토콜을 통해 동작하며 다음과 같은 특징을 갖고 있다.

- 일대다 메시지 배포와 애플리케이션 분리를 제공하는 게시/구독 메시지 패턴을 사용
- 페이로드 내용과 무관한 메시지 전송
- 메시지 전달을 위한 세 가지 QoS를 제공:
 - o QoS=0, "최대 한 번", 운영 환경의 최선의 노력에 따라 메시지가 전달.
 - o QoS=1, "적어도 한 번"은 메시지가 도착하는 것이 보장되지만 중복이 발생할 수 있는 경우임
 - o QoS=2, "정확히 한 번", 메시지가 정확히 한 번 도착하도록 보장.
- 네트워크 트래픽을 줄이기 위해 전송 오버헤드를 최소화하고 프로토콜 교환을 최소화
- 비정상적인 연결 끊김 발생 시 이해관계자에게 알리는 메커니즘

AMI 내 스마트 미터 실시간 데이터 수집 측면에서 TLS 기반으로 보안을 활성화하고 QoS 레벨 2를 사용하는 경우 MQTT 프로토콜을 유용하게 이용할 수 있다. 또한 스마트 미터의 전원 공급 중단으로 인해 비정상적으로 연결이 끊기는 경우 MQTT 브로커의 해당 토픽에 Last Will 메시지를 보냄으로써 상위 시스템에 알릴 수도 있다. 그리고 스마트 미터 등의 FW 업데이트, 특정 설정/제어 명령의 경우 상위 시스템에서 일대다 메시지 배포를 통해 동시에 여러 스마트 미터에 전송할 수 있어 양방향 통신의 이점을

활용할 수도 있다[12].

3. 시스템 설계 및 구현

3.1 시스템 개요

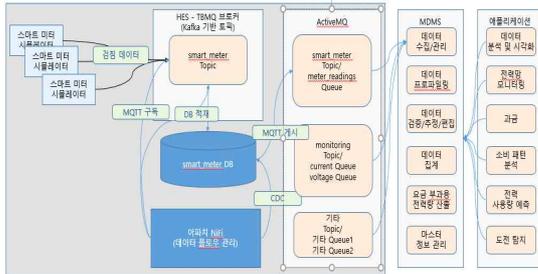


그림 1. 시스템 구조
Fig. 1. System architecture

향후 통신사업자의 IoT 인프라를 활용한다는 가정 하에 다양한 빅데이터 오픈소스 기반 MQTT 프로토콜을 이용한 스마트 미터 실시간 데이터 수집 및 관리 시스템 프로토타입의 아키텍처는 그림 1과 같다. 본 연구에서는 회색 박스에 포함된 부분만 다루며, 또한 시스템 성능적인 측면보다는 실현가능한 많은 아키텍처 중 하나의 아키텍처에 대한 기능성만을 다룬다.

이 시스템은 스마트 미터 실시간 데이터 수집 및 관리를 위해 스마트 미터 시뮬레이터, TBMQ MQTT 브로커[13], MySQL DB, 아파치 NiFi[14]와 ActiveMQ[15]로 구성되어 있으며, 오른쪽의 경우 향후 구현 가능한 MDMS와 잠재적인 애플리케이션을 나타내었다. 개략적인 스마트 미터 데이터 실시간 데이터 수집 및 관리 프로세스는 다음과 같다:

- 1) 스마트 미터 시뮬레이터에서 MQTT 프로토콜을 이용하여 MQTT 브로커의 토픽에 실시간으로 검침 데이터를 게시한다.
- 2) 실시간으로 게시된 검침 데이터는 MQTT

브로커 내 토픽에 저장되며 메시지 구독자가 구독해서 가져갈 때까지 브로커 내에 저장된다.

3) 아파치 NiFi를 이용하여 MQTT 브로커 내 토픽에 저장된 실시간 검침 데이터를 구독하고 DB 저장을 위해 변환 처리한 후 MySQL DB에 저장한다.

4) 아파치 NiFi를 이용하여 MySQL DB 내 테이블에 데이터 적재 시 이를 CDC(Change Data Capture) 기술[16] 기반으로 포착한 후 비동기 메시징 플랫폼 ActiveMQ에 전송할 수 있도록 해당 데이터를 JSON 변환을 한 후 MQTT 프로토콜을 이용해 ActiveMQ의 해당 토픽/큐에 전송한다. 일반적으로 실시간 검침 데이터가 저장되는 MySQL DB의 경우 수십만대의 스마트 미터를 고려한다면 테이블 쓰기 작업으로 인해 성능 차원에서 읽기 전용 레플리카를 구성하는 것이 필수적일 것이다. 따라서 실제 MDMS의 경우 이러한 읽기 전용 레플리카를 사용하여 실시간 검침 데이터를 가져온 후 검침 데이터 검증, 누락 데이터 처리 등의 프로세스를 통해 MDMS 내 별도 DB에 저장하는 것이 일반적이지만 본 연구에서는 이를 생략하였다. 그러나 실제 애플리케이션에서 사용하기 위해서는 이러한 처리가 필수 불가결하며, 따라서 실제로는 동일한 테이블이라기보다는 읽기 전용 레플리카의 테이블에 대해 CDC 기술이 적용되어 MDMS 연동을 위해 ActiveMQ에 전송될 것이다.

5) 오른쪽의 경우 향후 구현 가능한 MDMS와 잠재적인 애플리케이션들로, 예를 들어 전력 사용량 데이터 분석 및 시각화 애플리케이션의 경우 smart_meter 토픽의 meter_reading 큐로부터 비동기적으로 메시지를 받은 후 분석 및 시각화에 활용할 수 있다.

3.2 시스템 개발 환경

3.2.1 스마트 미터 시뮬레이터

스마트 미터는 AMI의 핵심 구성요소로 매 15분 또는 매 1시간 등의 특정 시간 간격으로 많은 고객들의 전력 소비 데이터를 자동적으로 수집하여 다양한 통신 기술을 활용하여 상위로 전송하는데, 여기서는 스마트 미터 내 LEE 모뎀이 탑재되어 스마트 미터가 수집한 검침 데이터를 모뎀을 통해 직접 상위에 전송한다고 가정한다. 실제로 LEE 모뎀 내 다양한 오픈소스 MATT 클라이언트를 사용하여 검침 데이터를 MATT로 실시간 게시하는 전용 프로그램을 작성할 수 있을 것이다. 따라서 스마트 미터 시뮬레이터의 경우 검침 데이터로 15분 간격의 전력 사용량 데이터를 임의로 생성하여 mosquito_pub 클라이언트 [17]를 사용하여 MATT 중개인의 특정 토픽에 게시하는 역할을 한다. 본 연구에서는 이를 위해 임의의 ID값을 갖는 스마트 미터에 대해 임의의 검침 데이터를 생성한 후 mosquito_pub 클라이언트를 이용하여 smart_meter/meter_readings 토픽에 이를 MATT 중개인으로 게시하는 쉘 스크립트를 작성하고 매 1분마다 실행하도록 crenate에 등록하였다. 물론 15분 간격의 수천~수십만 개의 미터가 동시에 MATT 중개인에 검침 데이터를 게시하는 쉘 스크립트도 작성 가능할 것이다. 또한 MATT 중개인의 경우 CA(Certificate Authority) 인증서를 이용하여 TBS 기반 메시지 게시도 가능할 것이다.

3.2.2 MATT 브로커

MATT 중개인의 경우 스마트 미터와 같은 Riot 장치로부터 MATT 메시지를 수집하여 안정적으로 저장하고 상위 시스템에 전달하는 주요 역할을 하는데, 다양한 유료 또는 무료 SW가 존재[18-21]하는데, 본 연구에서는 적게는 수십만대에서 많게는 수백만 대가 설치되는 스마트 미터 특성상 이를 수용할 수 있는 Things Board[22]에서 개발 및 배포하는 산업용 MATT

중개인인 TBMQ를 선택하였다. TBMQ는 400만 개 이상의 동시 클라이언트 연결을 처리할 수 있는 확장 가능하고 내결함성 있는 브로커로, 단일 클러스터 노드당 초당 최소 300만 개의 메시지 처리량을 지원하고 지연 시간이 짧으며, 클러스터 모드에서는 기능이 더욱 향상되어 100M개 이상의 동시 연결 클라이언트를 지원할 수 있다. TBMQ 설치 방법의 경우 설치요선[23]에서 보듯이 다양한 방법으로 설치할 수 있으며, 본 연구에서는 Ubuntu 22.04에 도커를 이용하여 설치하였다.

3.2.3 MySQL DB

스마트 미터 시뮬레이터로부터 MQTT 브로커에 실시간으로 게시된 검침 데이터는 JSON 포맷으로 변환하여 MySQL DB에 저장되는데, 여기서는 ubuntu/mysql 도커 이미지[24]를 사용하여 컨테이너로 실행하였다. 또한 MySQL DB 내 테이블의 변경 사항을 포착하여 상위 비동기 메시징 플랫폼인 ActiveMQ에 전송하기 위해 binlog[25]를 설정하였다. 도커 실행 후 스마트 미터 시뮬레이터의 데이터를 저장하기 위해 smart_meter DB와 meter_readings 테이블을 생성하였으며, meter_reading 테이블은 meter_id, reading, created_at, received_at과 같이 단순히 구성하였다.

3.2.4 데이터 플로우 관리

MQTT 프로토콜을 이용한 스마트 미터 실시간 데이터 수집 및 관리 시스템 프로토타입 내 데이터 플로우 관리의 경우 빅데이터 환경에서 UI 기능을 통해 데이터를 ETL(Extract, Transformation 및 Load) 할 수 있는 아파치 오픈소스 프로젝트인 NiFi를 사용하였다. NiFi를 이용하는 경우 스마트 미터 시뮬레이터의 실시간 게시 검침 데이터를 실시간 구독 후 NiFi 내 프

로세서를 이용하여 데이터 전처리 후 MySQL DB에 즉시 저장할 수 있으며, 이러한 MySQL DB 내 데이터 적재 이벤트 등을 CDC 기술을 활용한 프로세서를 이용하여 데이터 분석 및/또는 시각화를 위해 MDMS에 전달하기 위해 ActiveMQ 등의 비동기 메시징 플랫폼과 직접 연동할 수 있다. 아파치 NiFi의 경우 다운로드 사이트[26]에서 NiFi 1.27.0 버전을 다운로드하여 설치하였다.

3.2.5 ActiveMQ

Apache ActiveMQ Artemis는 다중 프로토콜로 내장형이며 매우 높은 성능과 클러스터형, 비동기 메시징 시스템을 구축하기 위한 오픈 소스 프로젝트로 메시지 지향 미들웨어(Message Oriented Middleware)의 일종이다. 따라서 메시징 시스템을 사용하면 이기종 시스템을 느슨하게 결합할 수 있으며, 일반적으로 안정성, 거래 및 기타 여러 기능을 제공한다. 본 연구에서는 스마트 미터로부터 수집된 검침 데이터만을 다루었지만, 실제로 스마트 미터의 경우 이러한 에너지 사용량 데이터 외에 과금 데이터뿐만 아니라 전력망 모니터링을 위한 다양한 전압, 전류, 주파수, 역률 등의 데이터도 수집하고 있으며 이러한 데이터들의 수집 및 활용까지 고려한다면 메시지 발신자와 메시지 소비자를 분리해서 완전히 독립적으로 유연하고 느슨하게 결합된 시스템을 고려하는 것이 향후를 위해서도 올바른 것이다. 또한 이 서로에 대한 취약한 종속성이 없기 때문에 향후 필요에 따라 새로운 시스템을 추가하거나 오래된 시스템을 폐기하는 데 더 많은 유연성을 제공할 것이다. ActiveMQ의 경우 다운로드 사이트 [27]에서 ActiveMQ Artemis 2.36.0을 다운로드하여 설치하였다.

4. 연구 결과

4.1 TBMQ MQTT 브로커 실행 및 실시간 검침 데이터 게시 확인

스마트 미터 시뮬레이터를 통한 실시간 검침 데이터가 MQTT 브로커에 게시되는지를 확인하기 위해 우선적으로 TBMQ MQTT 브로커를 실행하였고 http://localhost:8083을 통해 관리자 화면에 접속하였다. 관리자 화면은 그림 2와 같다.

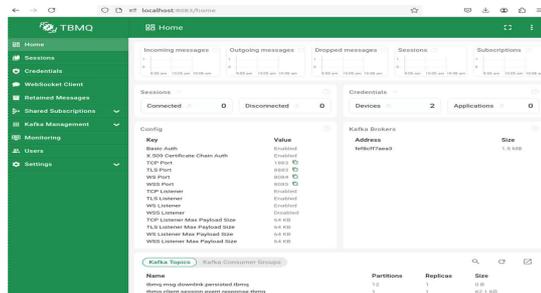


그림 2. TBMQ MQTT 브로커 화면
Fig. 2. TBMQ MQTT broker screen

첫 번째 줄에서 MQTT 브로커로/로부터 들어오는/나가는 메시지 수, 브로커에 연결된 세션 수, 브로커 토픽에 대한 구독 수를 확인할 수 있으며, 두 번째 줄에서는 브로커에 대한 연결/연결해제 수, 브로커 접속 관련 발행된 디바이스 및 애플리케이션 인증 수를 확인할 수 있다. 인증의 경우 기본적으로 사용자이름/패스워드로 설정할 수 있으며, CA 인증서 기반 TLS 인증을 설정하기 위해서는 별도로 인증서를 등록해주면 된다. 세 번째 줄에서는 인증, MQTT 포트와 해당 리스너, 웹소켓 포트와 해당 리스너, 리스너 관련 페이로드 사이즈 등의 브로커 설정 관련 키-값 쌍과 TBMQ MQTT 브로커의 기반 카프카 브로커를 확인할 수 있다. 이외에 MQTT 브로커가 수신 및/또는 전송한 메시지에 대한 카프카 토픽, MQTT 브로커 접속 관련 디바이스와 애플

리케이션의 인증 등을 설정할 수 있다. 최초 화면의 경우 MQTT 브로커에 연결된 디바이스 등이 없기 때문에 들어오는 메시지 수, 세션 수, 구독 수 등이 모두 0일 것이다. 또한 왼쪽의 Sessions, Credentials, WebSocket Client, Retained Messages 등의 메뉴에서 세부 사항을 볼 수 있다.

우선적으로 스마트 미터 시뮬레이터를 통한 실시간 검침 데이터가 MQTT 브로커에 게시되는지를 확인하기 위해 별도의 터미널에서 mosquitto_sub 클라이언트[28]를 활용하여 데이터가 게시될 smart_meter/meterings 토픽을 구독하였다.

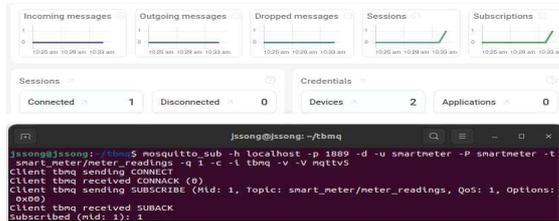


그림 3. 메시지 구독 후 TBMQ MQTT 브로커 및 터미널
Fig. 3 TBMQ MQTT broker and terminal after subscribing to messages

그림 3은 토픽 구독 후의 TBMQ MQTT 브로커 화면으로 구독 수가 1, 연결 세션이 1로 변경되었음을 알 수 있다. 또한 mosquitto_sub 클라이언트가 브로커의 smart_meter/meter_readings 토픽을 성공적으로 구독하였음을 알 수 있다.

다음으로 별도의 터미널에서 mosquitto_pub 클라이언트를 이용한 mqtt_pub.sh 셸 스크립트를 실행하여 smart_meter/meter_readings 토픽에 메시지를 게시하였다.

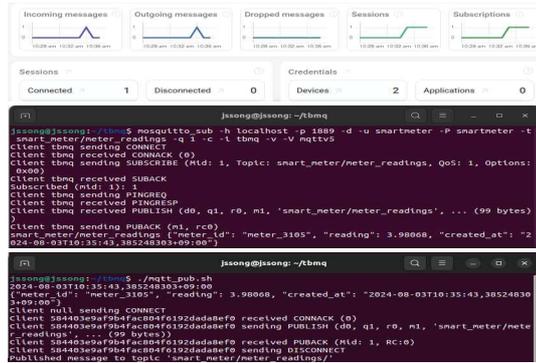


그림 4. 메시지 게시 후 TBMQ MQTT 브로커 및 터미널 화면
Fig. 4. TBMQ MQTT broker and terminal screen after posting message

그림 4의 경우 실시간 검침 데이터를 게시 후 TBMQ MQTT 브로커 화면의 경우 메시지가 1개 들어와서 구독한 mosquitto_sub 클라이언트로 메시지가 나갔음을 보여주고 있다. 왼쪽 아래에서는 mosquitto_sub 클라이언트가 오른쪽 아래의 셸 스크립트를 이용해 게시한 meter_3105에 대한 검침 데이터를 받았음을 보여주고 있다.

전체적으로 스마트 미터 시뮬레이터를 통한 실시간 검침 데이터가 TBMQ MQTT 브로커에 성공적으로 게시되어 이를 구독한 클라이언트에 성공적으로 전달됨을 알 수 있다.

4.2 아파치 NiFi를 이용한 실시간 검침 데이터의 DB 저장

TBMQ MQTT 브로커에 게시된 실시간 검침 데이터를 MySQL DB에 저장하기 위해 아파치 NiFi를 이용하였다. 아파치 NiFi의 경우 https://localhost:8442/nifi를 통해 관리자 화면에 접속할 수 있는데, MQTT 브로커에 게시된 실시간 검침 데이터를 DB에 저장하기 위한 전체적인 데이터 흐름은 그림 5와 같다.

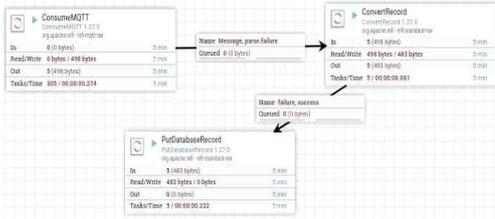


그림 5. 실시간 검침 데이터 DB 저장 NiFi 흐름
Fig. 5. Real-time meter reading data DB saved NiFi flow

우선적으로 TBMQ MQTT 브로커의 smart_meter/meter_readings 토픽에 게시된 실시간 검침 데이터를 갖고 오기 위해 Broker URI, Username, Password, Topic Filter, QoS 등을 TBMQ MQTT 브로커에 맞게 구성하여 ConsumeMQTT 프로세서를 설정하였다. 그림 6과 같이 ConsumeMQTT 프로세서를 선택하여 마우스를 우클릭한 후 View data provenance를 선택하면 프로세서에서 수신한 MQTT 브로커 메시지를 볼 수 있다. 밑 부분의 Provenance Event에서 Content의 View를 선택하면 최종적으로 스마트 미터로부터 검침 데이터를 수신했음을 볼 수 있다.

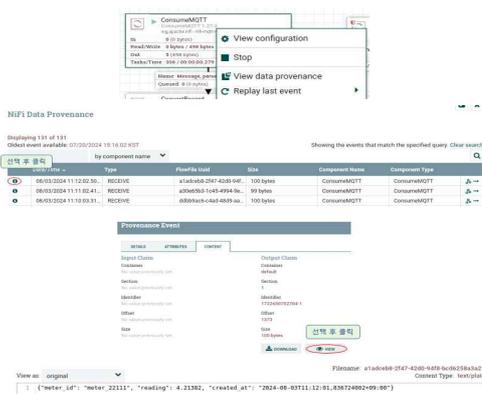


그림 6. ConsumeMQTT 프로세스 수신 메시지
Fig. 6. ConsumeMQTT processor received message

또한 ConsumeMQTT 프로세서에서 수신한 MQTT 메시지를 MySQL DB에 적재하기 위해서는 데이터를 변환해야 하는데 이를 위해 그림 5에서 ConvertRecord 프로세서를 설정하였으며, 마지막으로 변환된 데이터를 MySQL smart_meter DB 내 meter_readings 테이블에 넣기 위해 MySQL DB 구성에 맞게 PutDatabaseRecord 프로세서를 설정하였다. 그림 7에서 DBCPConnectionPool 서비스 구성 부분에 세부적인 MySQL DB 접속 정보와 이를 위한 드라이버 정보를 설정하였다.

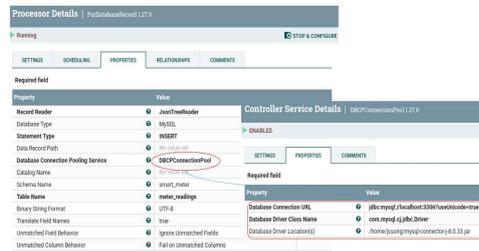


그림 7. PutDatabaseRecord 프로세서 설정 세부사항
Fig. 7. PutDatabaseRecord processor settings details

아파치 NiFi에서 DBCPConnectionPool 서비스의 경우 별도의 프로그래밍을 통해 실시간 검침 데이터의 DB 저장을 구현하기 위한 멀티프로세싱 관련 프로세스, 큐, DB 접속 관련 Pool 관리 등과 같이 빅데이터 처리 관련 작업을 손쉽게 처리할 수 있도록 구현해 놓은 것으로 디버깅, 성능 측면에서 많은 시행착오 등을 줄일 수 있다.

아파치 NiFi를 이용한 실시간 검침 데이터의 DB 저장이 성공적으로 수행되었는지를 확인하였는데, 그림 8은 MySQL DB에서 SELECT 문을 이용하여 meter_readings 테이블의 데이터를 조회한 결과로 성공적으로 수행되었음을 알 수 있다.

```
mysql> select * from meter_readings order by received_at desc limit 10;
```

id	meter_id	reading	created_at	received_at
25	meter_28867	4.46984	2024-08-03T11:29:01.735939837+09:00	2024-08-03 11:29:02
24	meter_17184	4.46437	2024-08-03T11:28:01.318152509+09:00	2024-08-03 11:28:03
23	meter_2598	1.94385	2024-08-03T11:27:01.968472762+09:00	2024-08-03 11:27:02
22	meter_13992	1.93141	2024-08-03T11:26:01.517360693+09:00	2024-08-03 11:26:02
21	meter_2947	3.79151	2024-08-03T11:25:02.044660255+09:00	2024-08-03 11:25:02
20	meter_13450	4.40773	2024-08-03T11:24:01.669350585+09:00	2024-08-03 11:24:02
19	meter_24743	1.89602	2024-08-03T11:23:01.281744242+09:00	2024-08-03 11:23:02
18	meter_4690	1.87662	2024-08-03T11:22:01.928237308+09:00	2024-08-03 11:22:02
17	meter_7766	1.87844	2024-08-03T11:21:01.478571376+09:00	2024-08-03 11:21:02
16	meter_26292	3.64272	2024-08-03T11:20:02.678591267+09:00	2024-08-03 11:20:03

그림 8. MySQL smart_meter DB 내 meter_readings 테이블 조회 결과
Fig. 8. Meter_readings table inquiry result in MySQL smart_meter DB

4.3 아파치 NiFi를 이용한 ActiveMQ 연동

일반적으로 MySQL DB에 저장된 실시간 검침 데이터를 다양한 애플리케이션에서 활용하기 위해서는 MDMS와 같은 별도의 상위 시스템으로 전송해서 별도의 처리를 통해 제공해야 한다. 따라서 MySQL smart_meter DB 내 meter_readings 테이블에 실시간 검침 데이터 적재 시 이를 CDC 기술 기반으로 포착하여 MDMS와 같은 상위 시스템에 비동기적으로 전송할 수 있도록 아파치 NiFi를 이용하여 ActiveMQ 비동기 메시징 플랫폼에 전송하였다. MySQL DB에 적재되는 데이터를 ActiveMQ에 전송하기 위한 전체적인 데이터 흐름은 그림 9와 같다.

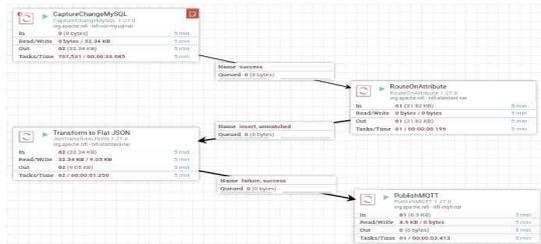


그림 9. ActiveMQ 연동 NiFi 흐름
Fig. 9. ActiveMQ interworking NiFi flow

우선적으로 MySQL DB smart_meter DB meter_readings 테이블에 데이터가 적재 (Insert) 시 이를 CDC 기술 기반으로 포착하기 위해 그림

10과 같이 MySQL DB 구성을 사용하여 CaptureChangeMySQL 프로세서를 설정하였다.



그림 10. CaptureChangeMySQL 프로세서 설정 세부사항
Fig. 10. CaptureChangeMySQL processor settings details

이 프로세서는 3.2.3 MySQL DB에서 설정한 mysql-bin 로그 파일을 이용해 데이터 적재 시 이를 포착할 것이다.



그림 11. CaptureChangeMySQL 프로세서 포착 CDC 메시지
Fig. 11. CaptureChangeMySQL processor capture CDC message

CaptureChangeMySQL 프로세서를 선택하여 마우스를 우클릭한 후 View data provenance를 선택하면 프로세서에서 수신한 MySQL DB의 binlog 파일의 내용을 볼 수 있는데, 그림 11의 경우 meter_28259에 대해 실시간 검침 데이터가 smart_meter DB 내 meter_readings 테이블에 적재되었음을 보여주고 있다.

다음에는 CaptureChangeMySQL 프로세서에

서 포착한 CDC 메시지를 ActiveMQ에 전송하기 위해 Insert CDC 이벤트에 대해 그림 12와 같이 RouteOnAttribute 프로세서를 설정하였다. 여기서는 MySQL DB의 Insert에 대해서만 구성하였는데, Delete, Update, 스키마 변경 등에 대해서도 설정가능하다.



그림 12. RouteOnAttribute 프로세서 설정 세부사항
Fig. 12. RouteOnAttribute processor settings details



그림 13. JoltTransformJSON 프로세서 설정 세부사항
Fig. 13. JoltTransformJSON processor settings details

또한 그림 13에서 보듯이 포착된 CDC 메시지의 경우 type, timestamp, binlog_filename,...,columns와 같이 중첩된 JSON 메시지로 상위 시스템에서 필요한 데이터는 columns 키에 해당하는 값일 것이다. 따라서 JSON 데이터의 구조를 변환하기 위해 그림 13과 같이 JoltTransformJSON 프로세서를 설정하였다.

오른쪽은 JSON 데이터 구조 변환을 위한 Jolt Spec.으로 이는 그림 13의 포착된 CDC 메시지를 id, meter_id, reading, created_at, received_at의 키와 값을 갖는 flat한 JSON 포맷으로 변환해준다.

마지막으로 변환된 데이터를 ActiveMQ smart_meter 토픽의 meter_readings 큐에 전송하기 위해 ActiveMQ 구성 정보를 이용하여 PublishMQTT 프로세서를 설정하였다. 여기서는 MQTT 프로토콜을 사용하였는데 ActiveMQ에서 지원하는 다른 프로토콜도 사용할 수 있다.

4.4 ActiveMQ 실행 및 확인

ActiveMQ의 경우 http://localhost:8161을 통해 관리자 화면에 접속할 수 있다.

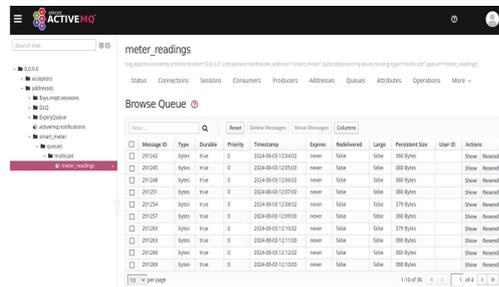


그림 14. ActiveMQ 접속 화면
Fig. 14. ActiveMQ access screen

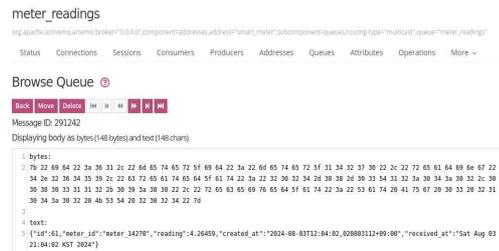


그림 15. smart_meter 토픽 내 meter_readings 큐 내 메시지 내용
Fig. 15. Message content in a meter_readings queue within a smart_meter topic

그림 14에서 보듯이 MySQL smart_meter DB 내 meter_readings 테이블에 실시간 검침 데이터 적재 포착 시 ActiveMQ의 smart_meter 토픽의 meter_readings 큐에 적재되고 있음을 알 수 있다. 그림 15는 큐 내 메시지 내용을 보여주고 있다.

물론 ActiveMQ 토픽의 큐 메시지를 가져가는 실제 MDMS와 같은 상위 시스템과의 연동이 없기 때문에 Expires 부분의 경우 never로 표시되어 있다.

4.1에서 4.4까지 스마트 미터 시뮬레이터의 실시간 검침 데이터의 MQTT 게시, TBMQ MQTT 브로커에서 MQTT 메시지 저장, 아파치 NiFi를 이용한 MQTT 메시지 구독 및 MySQL DB 저장, 아파치 NiFi를 이용한 MySQL DB 데이터의 ActiveMQ 전송을 테스트 및 확인하였다.

5. 결론

본 연구에서는 다양한 빅데이터 오픈소스 기반으로 MQTT 프로토콜을 이용한 스마트 미터 실시간 데이터 수집 및 관리 시스템을 설계하고 구현하였다. 스마트 미터 시뮬레이터를 이용하여 스마트 미터에 대한 임의의 검침 데이터 수집 및 관리만 고려하였으며, HES 역할의 MQTT 브로커, 빅데이터의 시스템 간 흐름 관리, 상위 시스템 연동용 비동기 메시징 플랫폼 등을 빅데이터 오픈소스 기반 하에 설계 및 구현하였다.

그러나 실제 AMI 내 스마트 미터의 경우 단순한 검침 값이라기보다는 국제 표준인 DAMS/COSHES 프로토콜[29]를 통해 많은(수십개의) 매개변수를 측정하고 저장할 수 있는 기능을 갖추고 있다. 예를 들어, 변조, 사기, TH(고조파), IBIS 코드, 주파수, 미터 이벤트, 역률, 무효 전력, 수출/수입 에너지, 부하 분산 등과 같은 매

개변수를 모니터링, 계산 및 등록한다. 또한 Tor(계시 요금제) 등 여러 요금제를 갖고 있으며 달력이 내장되어 있어 요일/월/년(휴일) 등을 구분한다. 따라서 MATT 프로토콜을 이용해 AMI 내 스마트 미터로부터 이러한 데이터들을 전송하거나 또는 설정하기 위해 그리고 전력회사의 보안 정책에 따른 최초 등록 시 활성화 조치를 반영하기 위해 스마트 미터 등록, 활성화, 검침 데이터 전송, 명령/설정 수신 등에 대해 매우 전략적인 토픽 설계가 필요할 것이다. 이외에도 스마트 미터의 경우 인증서 등을 포함하여 전력 회사의 통신 보안 조치를 고려해야 할 것이며, 이 부분의 경우 TBS 기반 보안 활성화를 통해 적절히 대응할 수 있을 것이다. 또한 실제 AMI 배치 시에는 수십만개에서 수백만개의 스마트 미터가 MQTT 브로커에 15분 간격으로 스마트 미터 데이터에 해당하는 다양한 특정 토픽으로 메시지를 전송하며, 필요한 경우 MDMS와 같은 상위 시스템으로부터 설정 명령을 받기 위해 자신의 ID에 해당하는 토픽을 구독해야 할 필요가 있다, 따라서 성능 측면을 고려하여 지속적인 연결 또는 비정기 연결 방식 등을 고려해야 할 것이다.

향후 연구에서는 스마트 미터 데이터의 다양성을 고려한 전략적인 토픽 설계와 AMI 내 설치되는 스마트 미터 수량에 따른 성능 측면을 고려한 스마트 미터와 MQTT 브로커의 연결 방식에 대한 추가 연구가 필요할 것이다. 또한 대다수 전력회사의 AMI 보안 요구사항을 고려하여 기본적으로 인증서를 활용한 TLS 기반 보안 통신을 사용해야 할 것이다.

이 논문은 2024학년도 배재대학교 교내학술연구비 지원에 의하여 수행됨.

참 고 문 헌

- [1] Seungil Moon, "Smart Grid Concept", Information & communications magazine, Vol. 27, No 4, pp.3-9, 2010, <https://www.dbpia.co.kr/pdf/pdfView.do?no deId=NODE01395279>
- [2] Changhoon Lee, "Smart Grid Technology Trends and Future Prospects", Information & communications magazine, Vol. 38, No. 9, pp. 71-77, 2021, <https://www.dbpia.co.kr/pdf/pdfView.do?no deId=NODE10596208>
- [3] Seongho Ju, Yonghoon Lim, Jongmook Baek, Power Electronics Annual Conference, pp. 109-110, 2010, <https://www.dbpia.co.kr/pdf/pdfView.do?no deId=NODE01474176>
- [4] KEPRI, "Development of PLC based AMI System and Smart Meter for Underground", 2015, https://www.kepri.re.kr:20808/keeping_file/n otice_item/20151014134445.pdf
- [5] Moon-Suk Choi, Su-Kyung Kang, Byung-Seok Park and In-Yong Seo, Development and Performance Analysis of Hybrid Communication Technology for Advanced Metering Infrastructure System, The Transactions of the Korean Institute of Electrical Engineers KIEE, Vol. 69, No. 4, pp.610-616, 2020, DOI : 10.5370/kiee.2020.69.4.610.
- [6] Junsu Shim, AMI Effect Analysis of Power-Line Communication System on Underground, Master of Engineering Research Report Seoul National University Graduate School of Engineering, 2018, DOI:10.23170/snu.000000149921.11032.000104 3
- [7] Suk-Gwon Chang, Techno-Economic Analysis and Business Model Evaluation of Smart Grid Communication Technologies, Telecommunications Review, Vol. 25, No. 1, pp. 90-109, 2015, <https://www.kci.go.kr/kciportal/ci/sereArticleSearch/ciSereArtiView.kci?sereArticleSearchBean.artiId=ART001969815>
- [8] Eclipse IoT, "IoT Developer Survey Results", 2018, https://www.globalict.kr/upload_file/s202108/52345177837115076.pdf
- [9] Integration with Smart Electricity Meter, https://developer.tuya.com/en/docs/iot/smart_meter?id=Kbd6gnb08j0a8
- [10] IoT Basic Pricing, <https://b2b.tworld.co.kr/s/basicproduct/basicProductPlanList.bc?mcl Cd=13>
- [11] AMIIoT, Where IoT Meets Smart Metering, <https://www.at-com.com/technical-articles/143-amiot-where-iot-meets-smart-metering.html>
- [12] MQTT Protocol, <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf>
- [13] TBMQ, <https://thingsboard.io/docs/mqtt-broker/>
- [14] Apache NiFi, <https://nifi.apache.org/documentation/v2/>
- [15] ActiveMQ Artemis, <https://activemq.apache.org/components/artemis/>
- [16] Debezium Documentation, <https://debezium.o/documentation/reference/stable/index.html>
- [17] mosquitto_pub man page, https://mosquitto.org/man/mosquitto_pub-1.html
- [18] EMQX, <https://www.emqx.com/en>
- [19] VerneMQ, <https://vernemq.com/>
- [20] HiveMQ, <https://www.hivemq.com/>
- [21] Mosquitto, <https://mosquitto.org/>
- [22] Thingsboard, <https://thingsboard.io/>
- [23] TBMQ Installation Options, <https://thingsboard.io/docs/mqtt-broker/install/installation-options/>
- [24] MySQL SW | Ubuntu, <https://hub.docker.com/r/ubuntu/mysql>
- [25] MySQL binlog,

<https://debezium.io/documentation/reference/stable/connectors/mysql.html#enable-mysql-binlog>
[26] NiFi Download, <https://nifi.apache.org/download/>
[27] Artemis Download, <https://activemq.apache.org/components/artemis/documentation/>
[28] mosquito_sub man page, https://mosquitto.org/man/mosquitto_sub-1.html
[29] Core Specification, <https://www.dlms.com/core-specifications/>

저 자 소 개



송재숙(Jea-Suk Song)

1993.2 한남대학교 전자계산공학과 졸업
1995.2 한남대학교 전자계산공학과 석사
2007.2 한남대학교 컴퓨터공학과 박사
2022.8-2023.12 : 우송대학교 IT융합학부
초빙교수
2024.6.1 - 현재 : 우송대학교 바이오헬스
사업단 연구교수
<주관심분야> OS, IoT, BigData, AI



배선영(Sun-Young Bae)

2000.8 배재대학교 응용수학과 졸업
2004.2 충남대학교 정보통신공학과 석사
2015.3-현재 : 배재대학교 전기전자공학과
조교수
<주관심분야> IoT, BigData, AI, Deep
Learning