

## IoT 환경센서 장치를 위한 효율적인 비동기 서비스 업데이트 연구

신슬비, 권동우, 지영민\*

한국전자기술연구원

{seulbi0108, dwkwon, ym.ji}@keti.re.kr

## A Study on Efficient Asynchronous Service Updates for Environmental Sensors

Seulbi Shin, Youngmin Ji, Dongwoo Kwon\*

Korea Electronics Technology Institute (KETI)

## 요약

스마트 인프라 환경에서는 쾌적한 환경 유지를 위해 다양한 센서가 실시간으로 상태를 측정하고 제어한다. IoT 환경센서 장치는 측정 기능과 직접 관련 없는 서비스라 하더라도, 업데이트가 펌웨어 수준에서 이루어지면 데이터 수집이 일시적으로 중단될 수 있다. 제안한 구조는 네트워크 과부하를 고려한 업데이트 분배, 주기적 서버 질의를 통해 컨테이너 기반 서비스를 갱신하는 방식을 제안한다. 제안한 구조의 실효성과 안정성을 검토하기 위해 실제 사무 공간에 적용해 테스트를 진행한다.

## I. 서론

스마트 인프라 기술의 발전에 따라, IoT 장치는 산업 시설, 공공건물 등 다양한 현장에서 환경을 파악하기 위한 핵심 기술로 사용되고 있다. [1] 특히 환경센서는 스마트 인프라 환경 구축에 필수적으로 요구된다. 실내 이용자들의 쾌적한 환경 제공을 위해서는 환경센서가 측정한 공기질, 온도 습도 등의 정보가 요구된다. 또한 기술이 발전하면서 환경센서는 설비 연동 등 다양한 서비스를 수행하는 복합적인 장치로도 사용된다. 이에 따라 실시간 데이터 측정에 영향을 주지 않으면서 서비스를 유지보수할 필요가 있다.

그러나 기존 장치들은 서비스의 단순한 변경임에도 불구하고 전체 펌웨어를 교체하는 방식으로 업데이트가 이뤄지는 경우가 대부분이다. 혹은 원격 업데이트를 지원하지 않기에 수동으로 반복 변경 작업을 수행한다. 이는 불필요한 리소스 낭비와 시스템 안정성 저하를 유발한다. 특히 펌웨어 단위의 업데이트는 전원 종료 또는 네트워크 단절이 발생한다면 치명적인 문제로 이어질 수 있다. 더불어 다수의 장치에서 동시에 업데이트가 진행되면 네트워크 과부하로 인한 통신 지연 등의 문제가 발생할 수 있다. 따라서 장치 수와 네트워크 환경을 고려한 서비스 수준으로 경량화된 업데이트 방법이 필요하다.

본 논문에서는 환경센서 장치를 위한 효율적인 비동기 서비스 업데이트 구조를 제안한다. 효율적인 관리를 위해 웹 환경을 적용하여 상태를 시각적으로 파악한다. 해당 구조는 장치가 주기적으로 서버에 요청을 보내고, 응답으로 필요한 정보를 해석해 사용한다. 이러한 구조를 통해 일시적으로 네트워크가 불안정한 환경에서도 안정적으로 정보를 주고받을 수 있도록 한다. 여러 업데이트 요청끼리는 비동기 실행 구조로 병렬 수행되지만, 하나의 요청 내에 속한 장치끼리는 차례대로 처리된다. 주로 유지보수는 설치 장소 단위로 진행하기에 동일 네트워크의 과부하를 방지할 수 있다.

제안하는 구조는 다수의 스마트 인프라 환경에서 사용할 수 있도록 설계되었으나, 모든 환경에서 테스트를 진행하기엔 현실적인 제약이 존재한다. 따라서 연구는 전자장치의 밀집도와 유동 인구가 높은 사무 공간을 대상으로 진행한다. 추가로 서비스 관리에 적합한 컨테이너 사용을 전제한다. 컨테이너는 IoT 장치 환경에서 효율적인 서비스 관리를 지원한다. [2]

본 논문의 구성은 다음과 같다. 2장에서는 제안한 시스템 구조와 동작 흐름, 연구 환경을 설명하고, 3장에서는 결론 및 향후 연구 과제에 대해 기술한다.

## II. 본론

## 1. 시스템 구조

그림 1은 해당 시스템의 전체 흐름을 도식화하고 있다. 주로 원격 관리 서버는 외부망에 존재하는 경우가 대부분이다. 이와 같은 구조에서는 서버에서 센서로 연결을 시도할 수 없거나, 방화벽 정책에 따라 인바운드 트래픽이 차단된다. 따라서 서버에서 장치에서 서버로 요청을 보내는 방식을 사용한다.

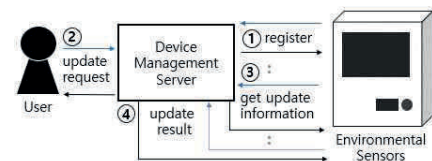


그림 1 시스템 구조

업데이트하는 서비스의 이미지는 레지스트리 서버에 저장되어 있다. 이미지 다운로드 중에는 서비스가 종료되지 않는다. 새로운 컨테이너를 생성하는 단계에서만 서비스가 일시적으로 중단되기에, 서비스 중단 시간을 최소화할 수 있다.

환경센서는 단순한 데이터 측정을 넘어, 기능이 고도화되며 기기 제어 및 데이터 처리 역할까지 수행하는 방향으로 발전하고 있다. 이러한 기능 수행을 위해, 많은 장치가 RTOS나 Linux 기반 운영체제를 탑재하고 있다. [3] 따라서 운영체제 기반 장치를 대상으로 효율적인 서비스 업데이트 구조를 설계했다. 테스트에 사용되는 환경센서는 Linux 기반 운영체제를 사용한다.

실험은 사무 공간을 대상으로 구성한다. 데이터 송수신을 위해서는 무선 네트워크를 사용한다. 벽면에 고정된 복수의 환경센서가 온도, 습도, 이산화탄소 등 데이터를 실시간으로 수집한다. 무선 통신 환경에서는 공간 내

사람과 장비의 이동에 따라 네트워크 품질의 변화가 발생한다. 이러한 환경에서 동시다발적인 이미지 다운로드를 무선 네트워크 환경에서 과부하를 발생시킬 가능성이 크다. 따라서 관리 서버에서 업데이트 명령을 환경 센서로 보내는 과정에서 부하를 고려한 분산 방식이나 시간차 처리 전략이 필요하다.

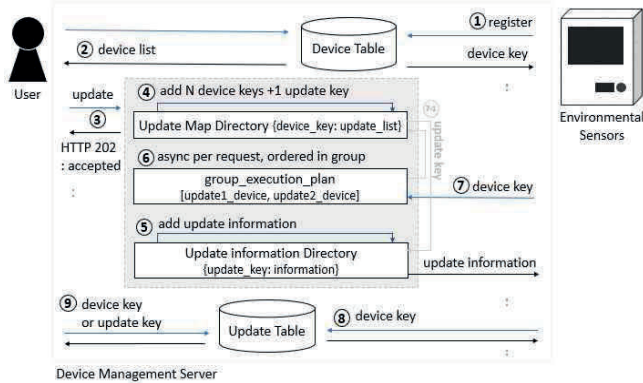


그림 2 관리 서버 동작 흐름

환경센서가 서버로 자신의 MAC 주소와 함께 등록 요청을 전송하고, 고유한 키를 받는다. 키는 앞으로 이루어질 모든 요청에서 환경센서의 식별자가 되어 사용된다. MAC은 고유하지 않아도 되며, 서버에서 사용자가 환경센서를 쉽게 식별하기 위한 이름으로서만 사용된다.

사용자는 원하는 환경센서들의 업데이트 명령을 보낸다. 정보는 장치별 업데이트 예정 목록, 업데이트 상세 정보로 나뉘어 저장된다. 사용자가 보내는 장치 목록은 네트워크 과부하 방지를 위해 설치 장소와 네트워크 환경이 고려되어야 한다. 하나의 요청 내에 포함된 장치들은 순차적으로 처리된다. 관리 서버는 요청 별로 비동기 스레드를 생성하고, HTTP 202 accepted 상태 코드를 응답한다. 현재 업데이트해야 하는 장치를 리스트에 넣어두고, 환경센서에서 요청이 들어오면 정보를 반환한다. 만약 서로 다른 요청에서 동일한 장치에 대한 업데이트가 요청될 경우, 장치에 적용될 정보는 가장 최근에 도착한 요청의 데이터로 갱신된다. 업데이트 완료 이후에는 이전 요청과 이후 요청 둘 다 완료한 것으로 간주한다. 업데이트가 완료되면 환경센서는 결과를 관리 서버로 전송한다. 관리 서버는 비동기 스레드가 결과를 저장하는 테이블을 주기적으로 확인하고, 결과가 확인되면 다음 업데이트 장치를 리스트에 넣는다. 만약 환경센서와의 연결이 중간에 끊기더라도, 스레드에서 시간제한 이후에 다시 업데이트를 수행하도록 처리한다. 웹 클라이언트는 업데이트 결과를 주기적으로 조회하며, 데이터가 확인되면 사용자 화면에 출력한다.

## 2. 실제 환경에서의 적용 및 테스트

그림 3은 환경센서 장치를 위한 효율적인 비동기 서비스 업데이트 구조를 실제 장치 관리 서버에 적용한 모습을 보여준다. 사용자 요청에는 업데이트 조건 버전, 목표 버전, 최대 시도 횟수, 시간제한 정보가 포함된다. 업데이트 조건에 명시된 버전이 장치의 서비스 버전과 맞는 경우에만 업데이트를 수행한다. 만약 모든 버전을 대상으로 업데이트를 진행한다면 all 옵션을 사용한다. 만약 목표 버전과 현재 버전이 같다면 Running 상태를 반환하여 추가 작업 없이 완료된다. 그러나 이미지 변경이 필요한 상황 등 조건과 다른 경우에는 업데이트 과정이 자세히 출력된다.

테스트 환경에서 11개의 장치를 대상으로 업데이트 요청을 수행했다. 유동 인구로 인해 무선 네트워크 품질이 저하되는 오전 시간대에도 안정적인 업데이트 작업을 수행했다. 추가로 제안한 구조는 요청 내 장치를 순차

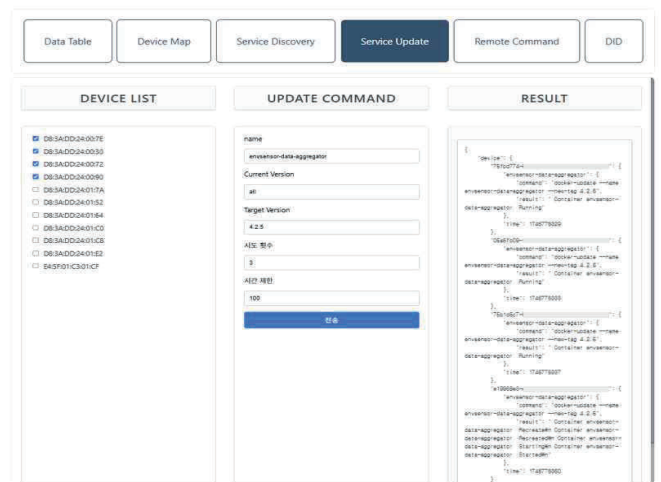


그림 3 장치 관리 웹

적으로 처리하도록 설계되었으며, 시스템 구조 변경 없이 쉽게 적용할 수 있다. 따라서 테스트 환경보다 더 큰 규모의 시스템에서도 안정적으로 적용하고 운영할 수 있다. 다음 장에서는 이러한 구조의 기대 효과와 적용한계, 그리고 향후 개선 방향에 대해 논의한다.

## III. 결론

본 논문에서는 환경센서를 위한 효율적인 서비스 업데이트 구조를 제안한다. 필요한 서비스만 선택적으로 업데이트할 수 있어, 유지보수 효율성과 확장성 측면에서 실용적이다. 이미지 다운로드 중에 오류 상황이 발생하더라도 서비스 동작에는 영향이 없으며, 시도 횟수에 따라 다시 업데이트가 진행되기에 안정적인 동작이 가능하다. 또한 네트워크 혼잡 상황에서도 장치별 순차 처리를 통해 네트워크 부하를 최소화할 수 있다. 결과적으로 제안한 구조는 실제 운영 환경에서의 높은 실효성을 보인다.

다만 아직 환경적 제한이 존재한다. 먼저 장치가 서버로 요청을 보낼 수 있는 환경이 필요하다. 관리 서버와 레지스트리 서버가 외부망에 위치할 경우, 인터넷이 지원되지 않는 폐쇄망 등에서는 적용이 어렵다. 그리고 웹 기반 통신에서 제공되는 기본적인 보안 수준은 유지되지만, IoT 환경에 적용하기 위해서는 강화된 인증 및 데이터 보호 체계가 필요하다.

## ACKNOWLEDGMENT

본 연구는 산업통상자원부(MOTIE)와 한국에너지기술연구원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (RS-2024-00441420)

## 참 고 문 헌

- [1] 이주상, 심창수, 이상한, and 최성준, "건물안전을 위한 스마트센서 IoT 플랫폼", in 한국통신학회 학술대회논문집, 강원, pp 1055-1056, 2020.
- [2] R. Morabito, "A performance evaluation of container technologies on Internet of Things devices," 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 2016, pp. 999-1000, doi: 10.1109/INFOCOMW.2016.7562228.
- [3] M. Farahani, M. A. Rashid and B. Safaei, "From Kernel to Cloud: A Concise Comparative Study of Practical IoT Operating Systems," in IEEE Internet of Things Magazine, vol. 8, no. 1, pp. 92-100, January 2025, doi: 10.1109/IOTM.001.2400090.