

SDV 기반 소형 단말 전용 OTA 시스템의 실용적 구현 및 성능 분석

장하람, 조혜원, 김가연, 이연우, 김준영
성신여자대학교

haramjangr@gmail.com, 20221434@sungshin.ac.kr, 20221326@sungshin.ac.kr,
20221052@sungshin.ac.kr. *jkim@sungshin.ac.kr

Practical Implementation and Performance Analysis of an OTA System for SDV-based Embedded Devices

Jang Ha Ram, Cho Hye Won, Kim Ka Yeon, Lee Yeon Woo, Kim Joon Young*
Sungshin Women Univ.

요 약

본 연구는 소형 단말기기 환경에서의 OTA(Over-The-Air) 성능의 정량적 분석을 통한 소프트웨어 정의 차량(Software-Defined Vehicle, SDV) 적용 가능성을 고찰하고자 한다. 실 OTA 시스템 구현과 더불어 싱글 보드 컴퓨팅 장치 기반의 실질적인 단말 레벨 실험 수행 및 결과에 대한 분석을 진행한다. 추가로 다양한 사례 분석을 위해 클라우드 플랫폼 및 로컬 서버 레벨의 OTA 케이스도 병행해서 분석한다.

I. 서 론

하드웨어 설계가 중심인 기존 자동차 경우 해당 하드웨어를 제어하기 위한 소프트웨어가 사후적으로 탑재되는 구조이다. 실제 소프트웨어 업데이트를 위해서는 수동정비가 필수로 요구되는 구조이다. 한편, 인포테인먼트 및 자율주행 기술의 급격한 발전은 차량이 사용자 요구에 따라 지속적으로 업데이트를 할 수 있는 소프트웨어 중심의 아키텍처를 요구하고 있다. 이에 따라 기존 하드웨어 기반의 차량에서 벗어나, 기능 업데이트와 서비스를 유연하게 하는 SDV 개념이 대두되고 있다[1]. SDV의 구현을 위한 핵심 기술 중 하나는 무선 소프트웨어 업데이트 방식인 OTA 기술이다. OTA 경우 차량 기능을 원격으로 개선, 적용 및 갱신할 수 있으며 SDV의 확장성과 경제성 측면에서 매우 중요한 시스템 중 하나이다[2].

현재 OTA 기술의 적용에 대한 연구는 지속되고 있으며 양산 레벨로써 활용되고 있으나 실제 차량 시스템에서의 실시간 혹은 대용량 적용 경우 다양한 환경 검증 등 필요한 상황이다. 특히 다양한 CPU 구조 및 서버 환경에서 OTA 수행 시 성능 특성과 지연 시간의 변화에 대한 실험적 검증이 필요한바 본 연구에서는 실제 구현 통한 분석을 진행하고자 하며 실험 결과를 바탕으로 추후 SDV 분야에서의 확장 가능성도 같이 고찰한다.

II. OTA 시스템 개념 및 구성

본 연구는 서버 환경, 단말기기별 성능 및 전송파일 크기에 따른 지연시간 차이를 정량적으로 분석하는 것을 목적으로 한다. 이를 위해 그림 1 기반 하에 실험용 OTA 시스템 구현을 진행 하였으며 클라이언트-서버 구조 기반 동작으로 설계했다. 실험 내 소형 단말 경우 싱글 보드 컴퓨터를 사용했으며 대표적인 라즈베리파이 시리즈의 세 가지 모델들을 개별적으로 사용하였다. 단말기기별 로컬 및 클라우드 서버 중 하나에 연결하여 OTA를 수행하였다. 클라우드 서버 경우 AWS 기반 Python Flask 서버를 적용하였다.

네트워크 연결 방식은 유선 LAN과 무선 WiFi 두가지 방식으로 구분하였다. 무선 실험의 경우 라즈베리파이 3B는 하드웨어 특성상 2.4GHz WiFi 대역에서만 연결이 가능하며 해당 대역에서 실험을 진행하였고, 라즈베리파이 4B 및 5B는 5GHz WiFi 대역에서 실험을 수행하였다. 모든 조건에 대해 30 회씩 반복 실험을 진행하여 평균 및 분산을 측정하였다.

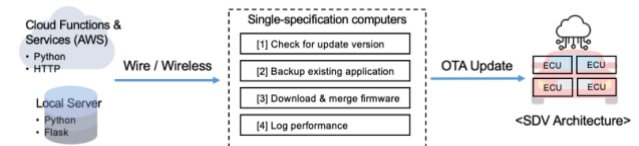


그림 1. OTA 시스템 실험 구조

OTA 업데이트 대상은 실제 펌웨어가 아닌, 전송량을 통제하기 위해 제작된 실험용 zip 파일을 사용하였으며, 파일 용량은 100MB, 200MB, 500MB, 1GB의 네 가지로 구성하였다. 각 용량에 대해 단일 파일을 그대로 전송하는 방식과, 동일한 파일을 세 파트로 분할하여 순차적으로 다운로드하고 병합하는 방식으로 실험을 나누어 수행하였다. 그림 1 내 실험구조를 파이썬으로 OTA 프로토타입을 설계하고 리눅스 환경 안에서 실험을 진행하였다. 업데이트를 실행할 때마다 CPU 및 메모리 사용량, 업데이트 시간을 기록하여 관찰하였다.

	Raspberry Pi 3	Raspberry Pi 4	Raspberry Pi 5
CPU	Broadcom BCM2837 (Cortex-A53)	Broadcom BCM2711 (Cortex-A72)	Broadcom BCM2712 (Cortex-A76)
CPU core	Quad-core 1.4GHz	Quad-core 1.5GHz	Quad-core 2.4GHz
RAM	1GB LPDDR2	8GB	4GB
WiFi	2.4GHz only	2.4GHz / 5GHz	2.4GHz / 5GHz

표 1. 주요 단일 싱글 보드 컴퓨터 보드별 성능[4][5]

III. 실험 결과

본 논문에서의 실험은 OTA 서버 위치, 단말기 사양, 네트워크 연결 방식, 파일 용량에 따른 전송 속도, 메모리, CPU 사용량 차이를 비교분석 하였다. 우선 그림 1 과 같이 클라우드/로컬 서버 간 OTA 성능 차이 송수신 속도 측정 결과 대부분 조건 내 로컬 서버가 클라우드 서버보다 빠른 전송 속도를 보였다. 클라우드/로컬 서버에 따른 OTA 성능 경우 플랫폼 위치에 따른 의존성을 보이며 추가로 장치 성능에 따른 송수신 차이도 확인할 수 있다.

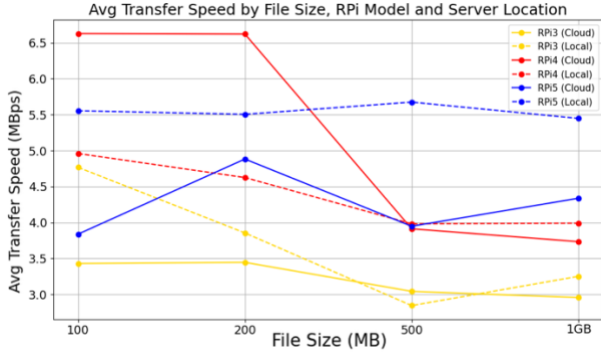
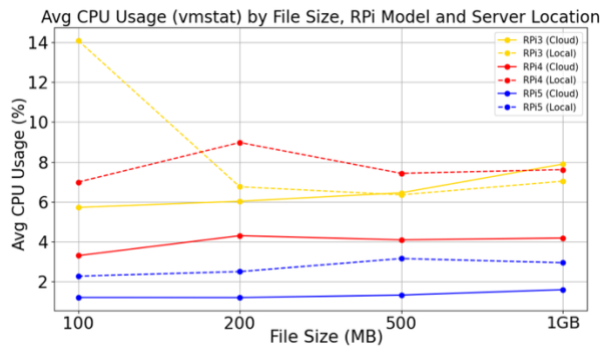
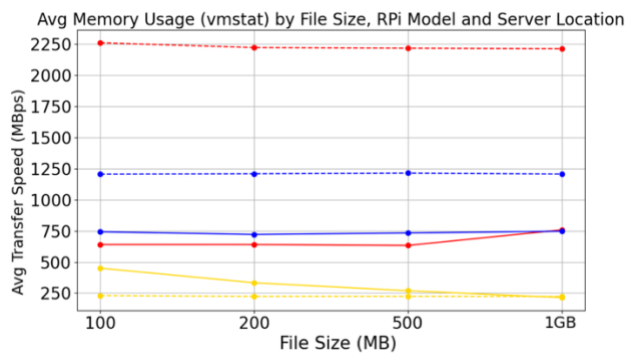


그림 2. 클라우드와 로컬서버 비교

특히 장비 간의 성능 차이는 OTA 처리에 직접적 요소이다. 그림 3(a) 내 RPi 5 경우 RPi 3 대비 최대 40% 이상의 높은 처리 속도를 보였다. OTA 파일 크기 증가에 따라 일정한 메모리 소모 현상도 그림 3(b)에서 볼 수 있다. OTA 파일 크기와 전송 처리 시간 및 메모리 점유의 상관 관계 존재와 더불어 고성능 CPU 단말 경우 대용량 OTA 의 안정적 대응 및 고속 송수신 처리가 가능함을 확인할 수 있다.



(a) CPU 사용량 비교



(b) Memory 사용량 비교

그림 3. 싱글 보드 장치 모델별 메모리 사용 비교

유선 LAN 과 무선 WiFi 간의 성능 비교에서도 유선 환경이 전송 속도 및 안정성 면에서 우위를 보였다. WiFi 실험내 Pi 4 및 Pi 5 는 5GHz 대역에서 안정적인 성능을

보였으며 특히 클라우드 OTA 경우 지연 및 재시도가 다수 확인되는데 실제 OTA 설계 시 네트워크 조건에 따른 동적 전송 전략이 필요함을 시사할 수 있다.

파일 크기를 변수로 측정한 결과, 전송 지연은 용량에 따라 비선형적으로 증가하였으며, 특히 1GB 파일 전송 시 일부 환경에서는 6 분 이상 소요되는 경우도 확인되었다. 또한, 파일을 세 개의 파트로 분할한 전송 방식은 단일 파일 전송 대비 전송 속도에서 개선되는 경우가 있었으나, 병합 처리 시간이 추가되어 성능 차이가 상쇄된다. 특히 클라우드 OTA 환경에서는 분할 전송이 전체 처리 시간을 줄이는 데 효과적이었으며, 반면 로컬 OTA 에서는 단일 전송이 더 효율적인 경우도 존재하였다. 이러한 결과를 바탕으로 볼 때 OTA 대상 파일의 크기와 네트워크 조건에 따라 분할 전송 여부를 결정하는 것이 중요할 것이다.

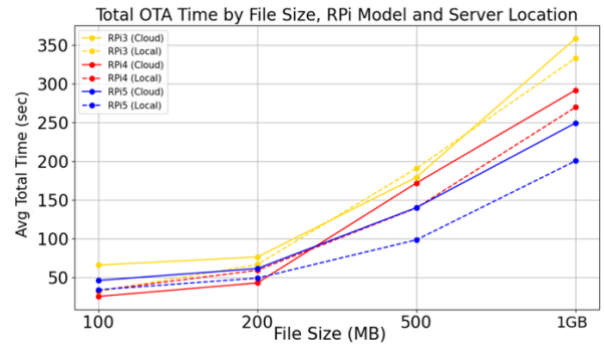


그림 4. 업데이트 파일 용량별 성능 비교

VI. 결론

실험 결과를 통해 실질적인 OTA 시스템 설계의 고려 지점을 알 수 있다. 다양한 환경 상황과 사용되는 HW 레벨의 사양을 고려하여 설계를 진행해야 한다는 근거를 마련하였다. 다만, 본 연구는 표본의 제한과 OTA 구현에 있어 발생할 수 있는 통신중단, 보안인증, 무결성 검증 등의 다양한 시나리오 미반영의 한계를 지닌다. 후속 연구에서는 다양한 장치, 통신 과정에서의 오류 대응 방안과 보안 요소를 포함한 현실적인 실험 설계를 고려함으로써, 실제 OTA 시스템과 유사한 환경 조성에 SDV 최적화되어 활용 가능 OTA 시스템 모델 설계 및 검증이 필요할 것이다.

참 고 문 헌

- [1] Luo, Y., & Lu, S. (2025). *Incremental Over-the-Air Update for Software-Defined Vehicles with C-V2X*. In Proceedings of the IEEE Vehicular Technology Conference (VTC2025-Fall).
- [2] Teixeira, P. V., Raposo, D., Lopes, R. E., & Sargento, S. (2024). *Software Defined Vehicles for Development of Deterministic Services*. arXiv preprint arXiv:2407.17287.
- [3] Eric Gamess, Sergio Hernandez. "Performance Evaluation of Different Raspberry Pi Models for a Broad Spectrum of Interests." *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 13, No. 2, 2022, pp. 819- 829.
- [4] Raspberry Pi Ltd. (2025). *Raspberry Pi 5 Product Brief*. Retrieved from <https://pip.raspberrypi.com>