앱 프로파일 기반 센서 레지스트리 시스템에 대한 실시간 환경에서의 실험 및 평가

최호진, 정동원†
군산대학교 통계컴퓨터과학과
e-mail:{shfwkd234, djeong}@kunsan.ac.kr

Experiment and Evaluation of the App Profile-based Sensor Registry System in a Real-time Environment

Hojin Choi, Dongwon Jeong
Dept. of Statistics and Computer Science, Kunsan National University

요 약

이 논문에서는 모바일 앱 프로파일 기반의 센서 레지스트리 시스템을 실시간 환경에서 실험하고 평가한다. 이기종 사물인터넷 환경에서 센서 레지스트리 시스템은 모바일 기기에 센서 메타데이터를 제공하여 센서 데이터 처리의 즉시성을 제공한다. 하지만 불필요한 센서 데이터까지 처리하여 성능 측면에서 개선사항이 요구된다. 이를 개선하기 위해 앱 프로파일을 기반으로 한 센서 필터링 기법을 제안한다. 유효한 센서 식별을 위한 센서의 정보를 프로파일 형태로 모바일 기기에 저장 후 사용하여 불필요한 센서 데이터를 필터링함으로써 전체적인 처리 속도가 개선된다. 기존 연구는 시뮬레이션 평가를 통하여 처리 속도의 향상성을 확인한다. 하지만 실시간 환경을 통한 다양한 상황의 실험이 부족하여, 이 논문에서는 기존 센서 레지스트리 시스템과 앱 프로파일 기반의 센서 레지스트리 시스템을 실시간 환경에 맞춰 구축하고 실험 및 평가한다.

1. 서론

최근 사물인터넷(IoT, Internet of Things) 환경이 확산되어 다양한 제품이 출시되고 있다. 원활한 IoT 환경을 구축하기 위해 다양한 연구가 진행됐으나, 기존 연구는 정해진 센서의 정보와 해당 시스템 내에서 처리하는 환경만 고려하는 문제점이 있다. 이는 다양한 IoT 환경에서 제한된 센서를 사용하므로, 이상적인 IoT 환경을 구축하기 위해서 충분히 고려해야 할 사항이다.

IoT 환경에서 모바일 기기의 다양한 센서 활용성과 센서와 모바일 기기 간의 즉시적인 데이터 처리는 매우 중요한 요소이다. 이러한 요소를 충족시키기 위해 센서 레지스트리 시스템(SRS, Sensor Registry System)이 제안되었다. SRS는 센서의 다양한 정보를 등록 및 관리하여 사용자에게 제공한다. 사용자는 SRS를 통하여 센서에 대한 데이터 즉, 센서 메타데이터를 제공 받아, 이기종 간의 센서 데이터 처리를 즉시적으로 가능하게 한다[1].

하지만 SRS는 불필요한 센서 데이터를 필터링하지 못하며, 이를 위해 계속 처리하는 등 성능 측면에서 개선사항이 요구되어 문제점 해결하기 위해 다양한 연구가 진행되었다. [2]는 앱 프로파일 정보를 활용하여 기존의 문제점을 해결한다. 모바일 기기에 유효한 센서 정보를 프로파일 형태로 저장 후, 해당 정보를 활용하여 불필요한 센서 데이터를 필터링한다. 기존 SRS의 문제점을 해결하여 성능 측면에서 향상됨을 확인한다. [3]은 [2]의 제안 방법을 시뮬레이션 환경을 구축하여 실험하고 평가 한다. 그러나 여전히 실제 센서를 이용한 실험이 이루어지지 않아 실제 IoT 환경에서 향상된 SRS의 이점을 표현하기 부족하다.

이 논문에서는 실제 센서를 통해 실험하기 위해 저전력 블루투스(BLE, Bluetooth low energy)를 사용한다. BLE를 활용하여 실제 IoT 환경을 구축 후 기존 센서 레지스트리 방법과 앱 프로파일 기반 센서 레지스트리 방법을 실험 및 평가 한다.

2. 기존 연구

이기종 사물인터넷 환경에서 끊임없는 센서 데이터의 처리를 위해 다양한 연구가 진행되었다. [1]은 다양한 센서 정보를 등록 및 관리하여 모바일 기기에 센서 정보의 데이터 즉, 센서 메타데이터를 제공한다. 모바일 기기는 센서 메타데이터를 통하여 센서 데이터의 정보를 식별 후, 데이터를 사용한다. 하지만 모바일 기기는 유효하지 않은 센서의 메타데이터까지 요청하여 처리속도 측면에서 개선사항이 필요하다. 그뿐만 아니라 센서 데이터를 수신 받을 때마다 서버로 센서 메타데이터를 요청해야 하므로, 네트워크에 종속적인 문제점이 있다.

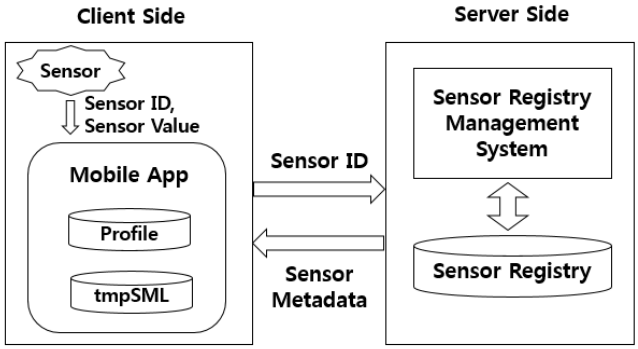
*이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2014R1A1A2058992)
† 책임저자 : 군산대학교 정동원

[2]는 기존 SRS의 단점을 보완하기 위해 앱 프로파일을 기반으로 한 필터링 기법을 제안한다. 모바일 기기에 유효한 센서 정보를 프로파일 형태로 저장한다. 모바일 기기가 센서 데이터를 제공 받으면, 센서 데이터의 유효성을 식별하기 위해, 식별 가능한 센서 정보를 통해 해당하는 센서의 메타데이터를 SRS에 요청한다. SRS를 통해 센서 메타데이터를 받은 모바일 기기는 앱 프로파일 정보를 통하여 센서의 유효성을 식별한다. 식별 후 센서 메타데이터와 유효속성 값을 함께 임시 저장하여, 해당 정보를 활용한다. 그 후에 같은 센서 정보를 수신 받게 되면 서버로 센서 데이터를 요청하지 않고, 임시 저장된 정보를 활용하여 센서 데이터를 식별한다. 유효한 센서와 유효하지 않은 센서를 모바일 기기에서 필터링하여 센서 정보를 사용하기 때문에 서버로 요청하고 수신 받는 과정을 제거하여 처리 속도 측면에서 향상을 보인다.

3. 시스템 구조 및 주요 프로세스

3.1 시스템 구조

그림 1은 앱 프로파일을 활용한 SRS의 시스템 구조를 보여준다. 크게 사용자 영역(Client Side)과 서버 영역(Server Side)으로 나뉜다. 사용자 영역에서 모바일 기기는 BLE 센서를 통하여 센서 아이디와 센서 값을 전달받는다. 해당 센서를 이용하는 모바일 앱은 유효한 센서 유형을 프로파일 형태로 기록하고 있다. 유효한 센서 데이터를 활용하기 위한 임시 저장 리스트(tmpSML)는 센서 데이터 활용을 위해 다양한 센서 메타데이터를 포함한다. 서버 영역은 센서 구성 정보를 데이터베이스 형태로 관리및 제공한다. 센서 구성 정보는 센서를 식별하기 위한 센서 유형과 단위뿐만 아니라, 센서 제조 정보 및 센서 위치 등 전반적인 센서 정보를 포함한다.



(그림 1) 전체적인 시스템 구조

모바일 앱의 Profile 클래스는 해당 모바일 앱의 유효한 센서 정보를 다양한 형태로 저장한다. 이 논문에서 구현하기 위해 모바일 앱 상의 DataBbase 스키마는 다음과 같다.

- Profile(s_type)
- tmpSML(s_id, s_type, s_unit, s_valid)

모바일 앱에 포함된 profile 정보는 유효한 센서 속성 값인 온도와 조도를 포함한다. 또한 tmpSML은 센서 식별을 위한 센서 아이디(s_id), 센서 타입(s_type), 센서 단위(sr_unit), 센서 유효속성 값(s_valid)을 속성으로 포함한다. 유효 속성 값은 유효한 센서 일 경우 1, 아닐 경우 0을 저장한다. 센서 레지스트리(SR)는 센서 메타데이터를 포함한다. 센서 자체의 정보인 센서 식별자, 센서 유형, 센서 단위 등을 기본으로 센서 제조사, 센서 모델 정보, 센서 설치 위치정보 등 센서와 관련된 다양한 정보를 제공한다. 이 논문에서 구현하기 위해 사용한 Sensor 스키마는 다음과 같다.

- Sensor(s_id, s_type, s_unit)

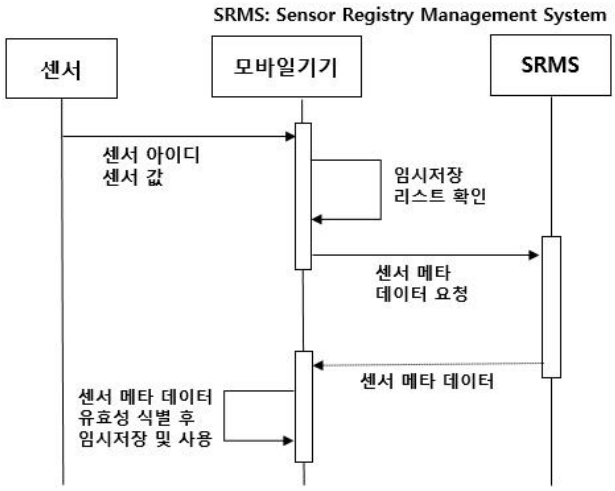
이 논문에서는 구현을 위해 최소 정보인 센서 아이디(s_id), 센서 타입(s_type), 센서 단위(s_unt)을 포함한다. 그림 2는 실제 구현한 Sensor 객체를 보여주고 있다.

	S_ID	S_TYPE	UNIT
1	s001	temp	celsius
2	s002	humi	percent
3	s003	illum	lx
4	s004	temp	fahrenheit

(그림 2) SR의 Sensor 객체

3.2 프로세스

그림 3은 앱 프로파일을 활용한 SRS의 과정을 순차 다이어그램으로 표현한다.



(그림 3) 전체적인 SRS 프로세스

센서는 센서 아이디와 센서 값(해당 센서 측정값)을 출력한다. 모바일 기기는 센서의 정보를 제공 받아 임시 저장 리스트(tmpSML)에 해당 센서 메타데이터의 여부를 확인한다. 센서 메타데이터가 없을 경우에 서버 영역의 SRMS에 센서 메타데이터 정보를 받게 되면 해당 프로파일 정보

를 활용하여 유효성 식별 후, 임시 저장 리스트에 유효 값과 함께 저장한 뒤, 해당 센서 데이터를 사용한다. 센서 메타데이터 및 유효 값을 임시 저장 리스트(tmpSML)에 저장한 후에 같은 센서 정보를 수신 받게 되면 해당 앱은 SRMS에 센서 메타데이터를 요청하지 않고, 임시 저장 리스트의 데이터를 활용하여 센서 값을 처리하게 된다. 임시 저장 리스트를 사용함으로써 서버의 SRMS에 독립적으로 모바일 기기에서 즉시적인 데이터 처리가 가능하다. 기존 SRS는 센서 메타데이터를 SRS에 요청 후 사용하므로, 네트워크 환경에 민감하다. 앱 프로파일을 활용하여 사용하면 초기 임시 저장 리스트를 구성해야 하므로, 기존 SRS보다 처리 속도가 더 소모된다. 하지만 첫 센서 수신시 모바일 기기에 임시 저장 리스트를 구성하고 이 정보를 활용하면, 서버로 센서 메타데이터를 요청하지 않고 바로 처리할 수 있다. 이는 전체적인 속도 개선에 효과를 나타내고, 네트워크 환경에 종속적인 기존 방식의 문제점의 해결이 가능하다.

4. 실험 및 평가 결과

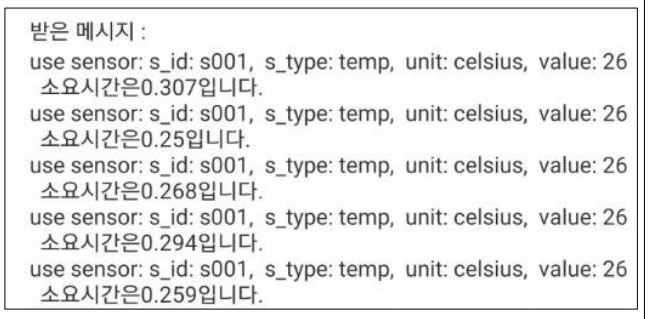
이 논문의 물리적인 구현 환경은 표 1과 같다.

<표 1> 물리적인 구현 환경

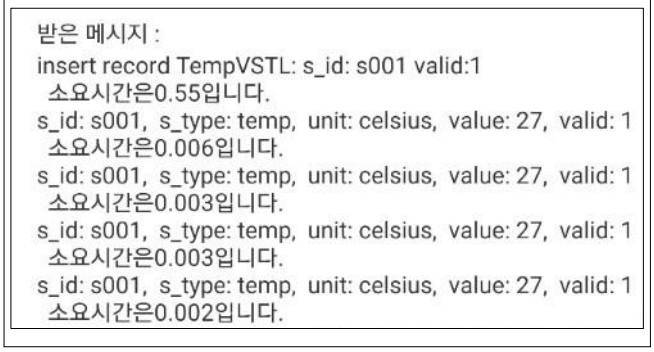
구분	내용
스마트폰 운영체제	Andrioid 5.1.1 (Lollipop)
BLE 모듈	16MHz ARM Cortex-M0, Bluetooth 4.0(BLE)
필터링 저장소를 위한 DBMS	Android SQLite
서버 시스템 운영체제	Microsoft Window 7 Professional K
서버 시스템 사양	Intel(R) Core(TM) i3 CPU 530 @2.93GHz
서버 레지스트리 저장소를 위한 RDBMS	Oracle Data Base 11g Enterprise Edition

그림 4는 실제 센서로부터 전달받은 데이터를 처리하는 모바일 기기 화면 모습을 보여준다. 그림 (4-a)인 기존 방식은 센서 데이터가 들어오는 즉시 센서 메타데이터를 요청한 후 받은 정보를 활용하여 데이터를 처리한다. 그림 (4-a)를 통하여 실제 센서와 관련된 정보들을 확인할 수 있으며, 이 정보를 활용하여 모바일 앱은 다양한 서비스 제공이 가능하다. 그림 (4-b)인 제안 방식은 센서 메타데이터를 프로파일 정보를 활용하여 유효성을 식별 후 임시 저장 리스트(tmpSML)에 저장한 뒤 해당 센서 데이터를 사용한다. 임시 저장 리스트의 저장을 위해 SQLite의 tmpSML 객체에 센서 식별자인 s001과 s001의 센서 메타데이터, 유효속성 값(valid=1)을 저장한다. 실시간 환경에서 실험하는 도중 센서 데이터를 처리하지 못하여 생기는 센서 데이터의 손실이 확인 된다. 이는 실시간 환경에서 생기는 네트워크 지연으로 인한 발생된 센서 데이터 손실이다.

(a) 기존 SRS방식

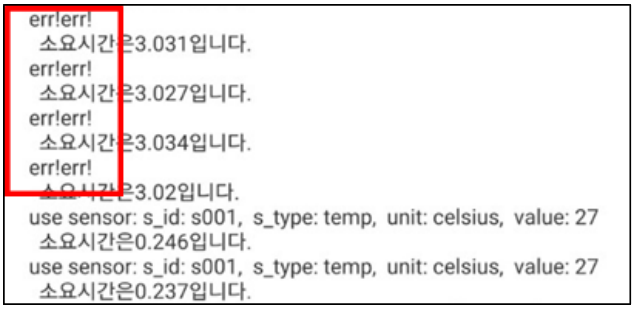


(b) 제안 SRS방식



(그림 4) 실제 SRS 구현을 통한 데이터 처리 결과

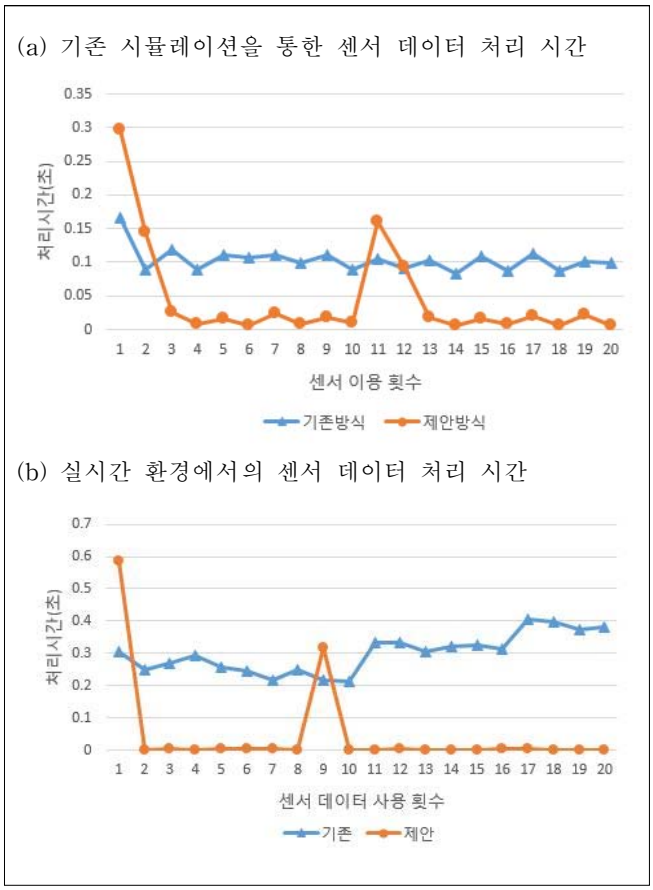
그림 5와 같이 기존 방법은 네트워크 지연으로 인해 센서 메타데이터를 요청하는 사이, BLE 센서의 데이터는 계속 수신되어 이전 센서 메타데이터가 처리 되는 동안 발생하는 새로운 BLE 센서 데이터는 손실되게 된다. 이는 네트워크에 종속적인 기존 SRS의 단점을 확연히 보여준다.



(그림 5) 처리 지연으로 인한 데이터 손실

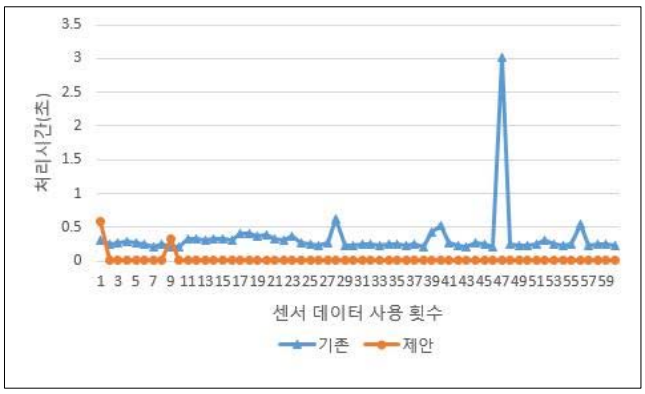
그림 6은 SRS 및 제안된 SRS의 센서 처리 속도를 그래프로 표현한다. 센서 s001, s003을 각 10번, 총 20번 사용하였다. 그림 (6-b)의 그래프를 보면, 기존 SRS는 항상 같은 프로세스를 수행하므로 평균 0.35초에 데이터 처리가 가능하다. 제안 SRS는 초기 센서 데이터 수신 시 앱 프로파일을 이용한 임시 저장 리스트를 처리하기 위해 기존 방식보다 더 많은 시간을 소요하게 된다. 처음 데이터 생성 시 평균 0.6초가 걸려 기존의 방식에 비해 2배 정도 처리 시간이 더 소모되지만, 그 후에는 모바일 기기 내부에서 자체적으로 처리하여 기존 방식보다 처리 속도가 빨라진다. 처음 데이터 구축을 제외한 제안 방식의 평균 처리 속도는 0.002

초 이다.



(그림 6) 처리 시간 그래프

그림 (6-a)와 (6-b)는 시뮬레이션 환경의 실험과 실시간 환경을 바탕으로 한 실험 데이터 그래프를 나타낸다. 두 그래프를 비교하면 시뮬레이션을 통해 실험한 데이터와 실시간 환경에서 한 실험 데이터가 평균적인 처리 속도에서 차이가 발생한다. 그러나 두 그래프의 전체적인 추세를 보면 기존 방식과 제안 방식의 유사함이 확인된다. 그림 (6-b)의 경우 데이터 사용 횟수가 적어 센서 데이터의 손실이 나타나지 않았지만, 데이터 사용 횟수가 증가함에 따라 돌발적인 네트워크 지연으로 센서 데이터의 손실이 확인된다.



(그림 7) 데이터 처리 지연에 따른 처리 시간 그래프

그림 7은 실시간 환경에서 구현한 기존 SRS에서 데이터 손실이 발생한 상황의 처리 시간 그래프 이다. 네트워크 지연에 따른 데이터 처리 시간은 평균 3초 정도가 소모되며, 3초 동안 센서 데이터를 수신 받지 못하게 된다. 지연시간 3초는 서비스 측면에서 사용자가 충분히 느낄 수 있는 시간이다. 이러한 현상은 센서 데이터 사용 횟수를 증가 시키면 시킬수록 더욱 증가한다. 하지만 앱 프로파일 기반 센서 레지스트리 시스템은 사용 횟수를 증가 시켜도 자체적으로 센서 데이터를 처리하여 네트워크에 무관한 안정적인 데이터 처리를 보여준다.

5. 결론 및 향후 연구

이 논문에서는 기존 SRS와 앱 프로파일 기반 SRS를 실제 BLE 센서로 환경을 구축하여 실험하고 평가하였다. 기존 연구는 시뮬레이션을 통해 실험하여, 실시간 환경에 맞춘 실험 및 평가가 요구되었다. 이 논문에서는 실제 BLE 센서를 통하여 실험 및 평가를 하였고, 앱 프로파일 기반 SRS의 이점을 확인하였다. 처음 앱 프로파일을 활용하여 임시 저장 리스트를 구축하는데 기존 SRS보다 더 많은 처리를 요구하지만, 그 후엔 서버로 센서 메타데이터를 요청하지 않고 자체적으로 처리하여 전체적인 처리 속도가 향상됨을 확인하였다. 그뿐만 아니라 실제 실험 과정에서 기존 SRS는 네트워크 지연으로 인한 센서 데이터의 손실을 추가 확인하였다. 이는 앱 프로파일 기반의 SRS가 처리 속도뿐만 아니라 센서 데이터의 손실을 보호할 수 있는 장점을 추가 확인하였다.

향후에는 다양한 장소에서 안정적인 서비스를 위한 실험이 요구되며, 제안된 앱 프로파일 기반 SRS의 초기 센서 데이터 처리 속도를 감소시키기 위한 연구가 요구된다.

참고문헌

[1] D. Jeong and J. Ji, "A Registration and Management System for Consistently Interpreting Semantics of Sensor Information in Heterogeneous Sensor Network Environments," *journal of KIISE* : Databases, Vol. 38, No. 5, pp. 289-302, 2011

[2] D. Jeong, H. Yoo, S. Lee, "Sensor Filtering based on Mobile App Profiles for Enhancing the Processing Performance of Sensor Registry System," Korea Information Processing Society (KIPS), Proceedings of the 2015 Spring Conference of the KIPS, Vol. 22, No. 1, pp. 273-276, 2015

[3] H. Choi, H. Yoo, D. Jeong, K. Jeon, "Implementation and Evaluation of the Sensor Registry System based on Mobile App Profiles," Korea Information Processing Society (KIPS), Proceedings of the 2016 Spring Conference of the KIPS, Vol. 23, No. 1, pp.945-948, 2016