논문 19-44-11-21 The Journal of Korean Institute of Communications and Information Sciences '19-11 Vol.44 No.11
https://doi.org/10.7840/kics.2019.44.11.2131

소셜관계 기반의 새로운 IoT 플랫폼 구현

유 명 한◆, 김 상 경°

Implementation of New IoT Platform Based on Social Relationship

Myung-han Yu◆, Sangkyung Kim°

요 약

사물인터넷(IoT) 분야에서 다양한 IoT 장치 간의 연결과 데이터 공유의 효율성을 개선하기 위하여 소셜 네트워크를 활용하는 연구가 수행되어 왔다. 상용 SNS와 IoT를 접목한 SWoT(Social Web of Things), IoT 전용의 소셜 네트워크 플랫폼을 가정한 SIoT(Social IoT) 등이 대표적이다. 하지만 기존의 연구는 비 IP 기반 사물의 수용을 충분하게 고려하지 않았으며, 사물과 사물, 사물과 사용자간 소셜관계를 형성하고 효율적으로 사물을 제어하며 데이터를 공유하는 절차에 관한 구체적인 내용이 결여되어 있다. 본 논문에서는 최근 발전하는 IoT 무선통신 분야의 트렌드인 비 IP 기반 저전력 무선통신 운영환경을 반영하고 단순화된 소셜관계에 기반한 새로운 IoT 플랫폼을 제안하였으며 그 구성요소들을 구현하고 시험하였다. 구현한 IoT 플랫폼을 기존 SIoT 플랫폼과 비교하여 성능의 향상을 확인할 수 있었다.

키워드 : 사물인터넷, IoT 플랫폼, 저전력, 소셜관계, SNS
Key Words : IoT, IoT Platform, Low Power, Social Relationship, SNS

ABSTRACT

In the Internet of Things (IoT) field, research has been carried out using social networks to improve the efficiency of connection and data sharing among various IoT devices. For example, Social Web of Things (SWoT), which combines commercial SNS and IoT, and SIoT (Social IoT), which is a social network platform dedicated to IoT, is introduced. However, existing research has not considered the accommodation of non IP-based objects sufficiently and lacks specific details about the process of establishing social relationships between objects and objects, or objects and users. And it also lacks detail about efficiently sharing its data and controlling other objects. In this paper, we proposed a new IoT platform based on simplified social relationships that reflect the non IP-based low power wireless communication environment which is a trend in the recently developed IoT communication field. We compared our new IoT platform with the existing platform, and confirmed the improvement of performance.

※ 본 연구는 2019년도 강릉원주대학교 학술연구조성비 지원에 의하여 수행되었음
◆ First Author : Gangneung Wonju National Univ., Dept. of Computer Science & Engineering, greatymh@gmail.com, 정회원
° Corresponding Author : Gangneung Wonju National Univ., Dept. of Computer Science & Engineering, skkim98@gwnu.ac.kr, 종신회원
논문번호 : 201907-141-0-RN, Received July 30, 2019; Revised September 12, 2019; Accepted September 20, 2019

Ⅰ. 서 론

IoT 플랫폼은 사물로부터 수집된 데이터의 보관 및 제공, 사물 및 소유주의 인증, 각종 제어작업 등의 중추적인 역할을 한다. 현재 IoT 플랫폼 기술 부분에서는 소셜 네트워크 서비스(SNS)를 IoT 환경에 접목하는 방식을 기초로 한 SWoT (Social Web of Things)[1]와 여기서 더 나아가 직접 IoT 단말과 사용자를 위한 SNS를 구성하는 방식의 Social IoT (SIoT, Social Internet of Things)[2] 등 소셜 네트워크 개념과 IoT 플랫폼을 연동하려는 다양한 연구가 진행되어 왔다.

이러한 소셜 네트워킹 관련 기법을 이용하여 전통적인 센서 네트워크상의 권한 설정 기능을 대체하면 기존의 수평 및 수직적인 관계의 구성에서 벗어나 단말의 소유주와 소유물 관계, 또는 협업 관계 등의 다양한 소셜 관계 구현이 가능해지고 이에 따른 다양한 부가기능을 준비할 수 있는 장점이 존재한다.

하지만 SWoT나 SIoT 기술에는 통신기술 분야의 발전에 따라 고려되지 못한 부분들이 있다. 대표적인 예로 LoRa[3]나 Sigfox[4] 등의 LPWAN (Low Power Wide Area Network)[5] 기술, Bluetooth Low Energy 등의 PAN (Personal Area Network) 에 기반한 저전력 사물은 짧은 페이로드 (payload) 크기, 배터리 효율을 위해 수시로 슬립모드 (sleep mode)에 진입하는 등 기존 무선통신에서 볼 수 없었던 특징이 존재한다.

관련 연구[6,7]에서는 이러한 저전력 사물이 SWoT나 SIoT 기술을 이용하려면 직접 소셜관계를 형성하거나 제어할 수 없으며, Atzori et al.[2]는 소셜 에이전트 (social agent)로 불리는 중계기능을 게이트웨이 또는 기지국에 내장하여 운영할 것을 제안하고 있다. SWoT 및 SIoT 플랫폼들은 REST[8] 등 일반적인 IP 기반 환경의 기술에 기반하므로 비 IP 기반 사물에 대해서는 SNS나 SIoT 플랫폼에 간접적으로 접속하도록 중계기능을 제안할 뿐, 저전력 사물들이 상호간에 직접 소셜관계를 구성하고 데이터를 송·수신하며 망을 유지할지에 대해서는 구체적으로 제안하지 않고 있다.[1,2]

이러한 문제를 해결하기 위해 본 논문에서는 IP 기반 사물과 비 IP 기반의 저전력 사물을 포함하여 사물과 사물, 사물과 사용자간의 소셜관계를 정의하고, 그 관계에 기반하여 효과적으로 사물로부터의 데이터를 공유할 수 있는 새로운 IoT 플랫폼을 제안하고 구현하였다.

비 IP 기반 저전력 사물은 게이트웨이를 거쳐 통신하는데, 제안된 IoT 플랫폼은 사물이 여러 게이트웨이들을 거쳐 물리적으로 이동 시 핸드오버 절차를 생략할 수 있도록 구성되었다. 또한, 제안된 IoT 플랫폼에서는 제한된 거리 내에서만 사물을 탐색하고 데이터를 공유하는 기법을 적용해 전파속도를 높이고 사물과 사용자에게 더 가치 있는 근거리의 데이터를 선택적으로 제공한다.

본 논문에서는 사용자 및 그 이용 목적에 따라 유연하게 적용할 수 있음을 검증하기 위해 제안한 IoT 플랫폼 및 IoT 사물 등 구성요소들을 실제로 구현하였다. 구현한 IoT 플랫폼과 기존 SIoT 플랫폼에서 전파 범위 내 사물들에 데이터를 전파하는 상황을 가정한 실험을 통해 실제 성능이 개선되었음을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서 기 연구된 IoT 플랫폼 및 저전력 무선통신 관련기술에 관하여 분석하고 3장에서 새로운 IoT 플랫폼을 제안한다. 4장에서는 구현내용을 상세히 기술하고 성능을 시험하였다. 5장에서 결론을 맺는다.

Ⅱ. 관련연구

IoT의 발전에 따라 다양한 상용 IoT 플랫폼 기술이 제안되었다. IoT 플랫폼 기술은 그 기반에 따라 소셜 네트워크 서비스를 기반으로 하는 Social Internet of Things (SIoT 또는 SoT) 플랫폼과 클라우드 서비스를 기반으로 하는 클라우드 IoT 플랫폼으로 구분할 수 있다.

SIoT 개념은 Atzori et al.[2]이 처음 제시하였다. SIoT는 사물 간에 소셜 네트워크 관계를 형성하고 데이터를 교환하는 방법을 고안하였으며, 기존 상용 SNS를 데이터 저장 및 교환 플랫폼으로 삼는 Social Web of Things(SWoT)[1] 기법과는 달리 IoT 플랫폼 전반에 대한 새로운 설계를 바탕으로 한다. 이는 다른 IoT 플랫폼과 유사하게 여러 서비스 레이어의 구조를 택하고 있으나 핵심은 소셜 네트워크 관계를 구성하는 레이어로 이를 통해 각 사물 및 사용자가 소셜관계를 형성하고 데이터를 주고받도록 하는 것이다.

클라우드 IoT 플랫폼은 Fernando et al.[6]와 Al-Kadhim et al.[7]의 연구에서 보이듯 물리적으로 여러 대의 서버가 하나의 클라우드 형태로 구성되는 전형적인 클라우드 서비스의 형태를 취한다. 이는 스마트시티 등 대규모의 IoT 서비스에 적합한 분산형 구조로 평가된다. SWoT 기법 또한 페이스북, 트위터 등의 상용 소셜 네트워크 서비스를 SaaS (Software as a Service) 및 PaaS (Platform as a Service) 형태로

볼 수 있으므로 클라우드 IoT 플랫폼에 기반한다고 볼 수 있다.

이러한 클라우드 IoT 플랫폼의 경우 클라우드 연결 방법에 IP 환경을 상정한 서비스를 제공하므로 IP가 없는 장치는 클라우드 서비스에 직접 접속할 수 없으며[6,7] SIoT의 경우에도 소셜 에이전트(social agent)로 불리는 중계기능을 게이트웨이 또는 기지국에 내장하여 운영할 것을 제안한다.[2] 기존 SIoT 플랫폼들은 일반적인 IP 기반 환경에서의 운용 개념을 따르므로 IP 기반이 아닌 단말에 대해서는 SNS에 간접적으로 접속하는 중계기능을 제시할 뿐 배터리 효율을 위해 수시로 슬립모드 (sleep mode) 에 진입하는 저전력 단말들이 실제로 어떻게 탐색하여 상호간 관계를 구성하고 데이터를 송·수신하며 망을 유지할지에 대해서는 제안하지 않았다.[2]

LPWAN 무선 통신은 IoT용 무선통신 기술 중 가장 진보된 기술 중 하나로, 소물인터넷이라는 별칭을 가진다. 이는 WSN (Wireless Sensor Network) 분야에서 센서와 게이트웨이 사이의 무선 통신 방법으로 사용된다. LPWAN의 일종인 LoRa 기술의 경우 개방된 공간에서 최대 15km의 통신거리를 가질 수 있으므로[3] 기존 Zigbee 등 저전력 단거리 무선통신 기술에 비해 큰 폭의 커버리지 증가와 4G LTE 등의 셀룰러 망에 비해 높은 경제성을 확보할 수 있다. 광범위한 영역에 적은 수의 중계기를 설치하여도 서비스를 제공할 수 있게 되었으며 위치 기반 서비스 및 자동차 등 교통 네트워크 구성에도 효율적인 형태를 보인다.

이외에도 PAN (Personal Area Network) 기술로 개발된 블루투스의 하위 규격인 Bluetooth Low Energy,[9] 전통적으로 USN 및 IoT 분야에서 많이 사용되어지고 있는 Zigbee (802.15.4)[10] 기술 등 다양한 IoT용 저전력 무선통신 기술이 새로 개발되거나 시장에 이미 보급되어 WSN용 무선통신 기술로 사용되는 추세이다.

이러한 저전력 무선통신 기술들의 단점은 소셜 네트워크 서비스를 이용하기에 충분한 전송속도와 대역폭을 제공하지 못 한다는 점에 있다. 낮은 데이터 속도와 매우 짧은 페이로드 (payload) 크기를 특징으로 하므로 기존의 무선통신 기술과 달리 IP 스택을 구현할 수 있는 환경을 제공하지 않는다.

특히 일반적인 소셜 네트워크 서비스는 REST API를 기준으로 TCP/IP 및 HTTP (때에 따라 HTTPS)에 대한 구현이 필요하나, 저전력 사물들은 동작시간의 대부분을 슬립 모드 상태에서 높은 에너지 효율성을 추구하므로[11] 이에 대한 구현 및 사용이 사실상 불가능하며, 기존의 클라우드 IoT 플랫폼 및 SIoT 플랫폼을 활용하여도 직접 센서 데이터를 공유하거나 제어받기 힘들 것으로 여겨진다.

Ⅲ. 제안하는 IoT 플랫폼

3.1 IoT 플랫폼과 사물간의 구성

본 논문에서 제안하는 IoT 플랫폼과 사물간의 구성도를 그림 1에 나타내었다.

IoT 사물은 IP 기반 사물과 비 IP 기반 사물로 구분되며 센서나 액추에이터 등을 탑재하여 데이터를 수집하거나 물리적으로 제어할 수 있다. IP 기반 IoT 사물은 인터넷을 통해 IoT 플랫폼과 직접 통신을 수행할 수 있다. 반면, 비 IP 기반 사물은 배터리로 구동되는 소형 단말장치의 형태로서 게이트웨이를 통해 IoT 플랫폼과 통신할 수 있으며, LPWAN이나 PAN 등 저전력 사물용 무선 인터페이스에 기반한 사물이다. 사용자 디바이스는 IoT 플랫폼과의 통신 및 데이터 표시를 담당하고 사물과의 간접적인 데이터 교환 및 제어 역할을 수행하는 장치를 의미한다. 데이터 교환 및 제어 기능을 통해 소셜관계가 설정된 소유한 사물에 대해 데이터를 조회하고 액추에이터를 제어하거나 설정 값을 변경할 수 있다. 소유하지 않은 사물에 대해서는 사물이 공개한 데이터를 검색하고 조회만 할 수 있다.

사용자 디바이스는 HTTP REST 등 웹 기반의 통

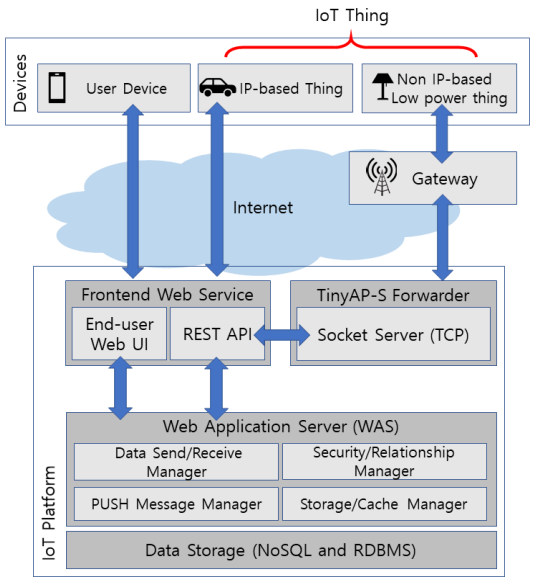


그림 1. IoT 플랫폼과 사물간의 구성도
Fig. 1. Connections between IoT platform and IoT things

신방법을 사용하며 전용 소프트웨어를 통해 새로운 IoT 플랫폼으로 연결한다. 게이트웨이는 비 IP 기반 저전력 사물의 인터넷 접속을 위한 무선 중계기 역할을 수행하는 장치로 프로토콜간의 변환 작업을 담당한다. 사물로부터 데이터를 수신한 후 Payload 부분의 TinyAP[12] 데이터 형식만을 인식하고 이를 소켓 서버로 전달하거나 그 역방향의 변환작업을 수행한다.

TinyAP는 저전력 IoT 단말장치를 위해 고안된 서버-클라이언트 구조의 경량 프로토콜이다. TinyAP에는 별도의 연결협상 절차가 없으므로 단말 및 게이트웨이 간에 교환되는 총 메시지의 수가 적으며 슬립모드 동기화 기법을 바탕으로 송신측에서 미리 동기화한 슬립모드 시간을 예측하고 대기한 후 패킷을 전송할 수 있다.

TinyAP는 IoT 분야에서 널리 이용되는 MQTT-SN[13] 프로토콜과 달리 연결 협상 절차가 없기 때문에 사물로부터 데이터 수신시에 교환되는 제어 메시지의 양이 줄어든다는 장점이 있다. 또한 서버를 통해 특정 사물로 액추에이터 제어 또는 센서 데이터 수집 명령 등의 제어 메시지를 직접 전달 가능하며, 사물과 게이트웨이 간에 슬립모드 시간을 동기화하여 추가적인 제어 메시지의 송수신 없이도 단말이 미리 정한 시간에 게이트웨이로부터 데이터를 수신하여 에너지 효율을 향상시킬 수 있다.[12] 본 논문에서는 TinyAP를 바탕으로 소셜관계 지원을 위해 기능을 확장한 TinyAP-S 응용 프로토콜을 구현하였다.

IoT 플랫폼은 크게 프론트엔드 웹 서비스, TinyAP-S Forwarder, 웹 어플리케이션 서버 (WAS), 데이터 스토리지로 구성된다. 프론트엔드 웹 서비스는 IP 기반 사물 및 사용자 디바이스에 대해 HTTP 연결을 제공하며 REST API의 송·수신자 역할을 수행한다. TinyAP-S Forwarder는 TinyAP-S 프로토콜 지원을 위한 데이터 처리 기능을 가진다.

웹 어플리케이션 서버는 IoT 플랫폼의 핵심적인 기능을 수행하는 여러 개의 컴포넌트를 가진다. 보안/관계 관리자는 사물과 사용자, 사물과 사물간 소셜관계를 설정하고 관리하며, 그 관계에 따른 인증을 수행한다. 데이터 관리자는 디바이스 및 데이터간의 관계를 정의하고 데이터를 추상화하며 송·수신 메시지 형식을 관리한다. PUSH 메시지 관리자는 사용자 디바이스에 PUSH 메시지를 발송하는 작업을 수행한다. 데이터 스토리지/캐시 관리자는 앞서 언급된 모든 기능에서 필요한 데이터를 데이터베이스에 저장하거나 데이터베이스로부터 불러오는 기능을 수행한다.

3.2 소셜관계

사용자가 사물에서 데이터를 얻거나 제어하려면 먼저 적절한 소셜관계를 설정해야 한다. 또한, 사물과 사물간의 데이터 교환 및 자동으로 이루어지는 장치 제어에도 소셜관계 설정이 요구된다. 본 논문에서의 사용자와 사물, 사물과 사물간 소셜관계는 총 3단계로, 그림 2에 소셜관계에 따른 사물 내 접근가능 기능의 제한범위를 나타내었다.

소셜관계는 사물에 대한 모든 제어 권한을 갖는 'Ownership' 관계, 사물로부터 수집된 데이터에 대한 조회 권한만을 갖는 'Guestship' 관계, 그리고 Ownership 관계를 가진 Owner에 의해 사물에 대한 제어 권한 중 일부만을 제공받아 함께 사용하는 'Partial Ownership' 관계로 정의하였다.

Ownership 관계에서 Owner는 사물에 대한 전체 제어 권한을 가진 사용자 또는 사물로, 사물 내의 모든 기능에 접근할 수 있다. Owner는 사물에 대한 설정기능을 통해 포스팅 되는 데이터를 사용할 수 있는 지리적 영역 제한, 보안 설정, 관계 설정 등 사용자의 설정이 필요한 부분의 세부조정이 가능하다. Property는 Owner에 의해 소유되는 사물을 의미한다.

Guestship 관계는 사물탐색 절차를 거쳐 특정 사물의 존재여부를 확인한 사물, 또는 사용자가 해당 특정 사물에 대해 가질 수 있는 가장 기본적인 소셜관계이다. 각 Guest는 탐색절차를 통해 주변에 위치한 사물을 확인하거나 기존에 알고 있는 사물 또는 사용자 아이디를 통해 특정 사물에 접근할 수 있다. Guest는 사물이 가진 데이터를 Owner가 지정한 범위 내에서만 조회할 수 있다. 만약 Owner가 사물의 데이터에 대하여 Guest의 조회 권한을 없애거나 해당 사물의 공개 여부를 Owner만 조회 가능한 비공개 상태로 설정할 경우 Guest는 사물 및 그 데이터에 대해 접근할 수 없다.

Partial Ownership 관계에서 Partial Owner는 Owner가 선택하는 설정에 따라 Owner가 가진 사물 제어기능 중 일부 또는 전체를 제공받을 수 있다. 단,

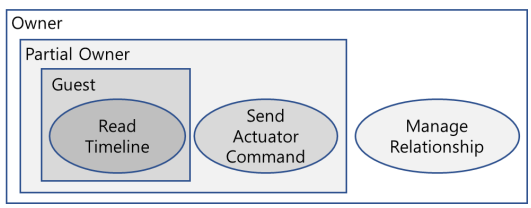


그림 2. 소셜관계에 따른 접근가능 기능의 제한
Fig. 2. Limitations of accessible features based on social relationships

Partial Owner가 사물의 Owner를 바꾸거나 다른 Guest를 Partial Owner로 허용하는 것은 불가능하다. Partial Ownership 관계가 되려면 Guest 사용자의 신청과 Owner 사용자의 수락 과정이 필요하다. Partial Ownership을 신청하고 수락하기까지의 절차를 그림 3에 나타내었다.

Guest가 특정 사물에 대해 Partial Ownership을 희망한다는 요청을 보내면 IoT 플랫폼에서는 PUSH 메시지를 발송하며, 이를 확인한 Owner는 웹 인터페이스를 통해 허용 또는 거절 여부를 결정할 수 있다. Owner가 Partial Ownership을 허용할 경우 IoT 플랫폼은 Partial Ownership 관계를 새로 형성하여 데이터베이스에 저장한다. 24시간 이내에 권한 요청을 수락하지 않으면 요청은 자동으로 거절된다.

[2] 등 다른 연구에서의 소셜관계 대비 더 적은 단계로 구성한 것은 Partial Ownership 관계에서 구체적인 퍼미션을 설정할 수 있는 기능을 도입할 경우 실제 소셜관계의 단계를 줄여 시스템의 복잡도를 낮추면서도 전체적인 기능은 유사하게 구현할 수 있기 때문이다.

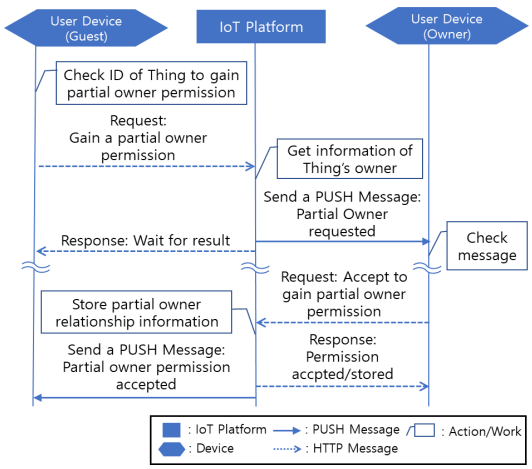


그림 3. Partial Ownership 권한 요청 및 Owner의 수락 절차
Fig. 3. Partial Ownership request and Owner's acceptance procedure

3.3 단말 및 사용자의 가입/인증

IoT 플랫폼이 제공하는 서비스를 이용하기 위해, 사용자나 IoT 사물은 가입이나 등록을 선행하여야 한다. 사용자의 가입에는 상용 SNS와 동일하게 사용자 아이디, 이메일 주소, 비밀번호, 닉네임이 필요하다. 사용자 아이디는 IoT 플랫폼에서 사용자의 고유 식별자로 사용된다. 이메일 주소는 비밀번호 및 사용자 아이디를 분실하였을 경우 개인정보를 찾기 위한 수단으로 사용된다. 닉네임은 사용자끼리 서로를 식별하기 위한 사용자명으로 사용된다.

사용자가 서비스를 사용하기 위해 가입해야 하는 것처럼 사물 역시 사용자 또는 다른 사물과 관계를 형성하고 데이터를 공유하기 위해 먼저 IoT 플랫폼에 등록해야 한다. 사물의 등록 절차는 사용자 가입 절차와 유사하다. 다만, 필요한 정보가 사물 아이디와 사물의 디바이스 타입, 고유키라는 차이가 있다.

고유키는 각 사물의 생산자가 임의로 생성하여 사물마다 부여하는 정보로 사물의 바닥면에 인쇄된 PIN (Personal Information Number) 이나 시리얼 번호 등으로 사물을 물리적으로 소유한 Owner만이 알 수 있는 정보이다. 이는 사물이 Ownership 관계를 형성할 때 Owner 요청을 한 사용자가 사물의 실제 소유주인지를 판단하기 위해 처음 한 차례 입력할 것을 요청한다. 그림 4는 사물의 등록 절차를 나타낸다.

가입이나 등록을 마친 사용자 또는 사물은 IoT 플랫폼에 자신의 신원을 확인하고 접속하기 위한 인증을 수행한다. 사용자의 경우 사용자 아이디와 비밀번호, 사물의 경우 사물 아이디와 인증키가 IoT 플랫폼에 저장된 정보와 일치하는지 확인한 뒤 결과 값을 성공 혹은 실패로 되돌려 받는다. 사용자는 5회 이상, 사물은 3회 이상 연속으로 인증에 실패하면 실패 횟수의 제곱에 해당하는 분 단위 시간 동안 인증이 제한된다. 인증키는 무작위로 생성되는 문자열과 숫자의 조합으로 128-bit 길이를 가진다. 저전력 무선 환경에서는 페이로드의 길이와 통신환경이 제한되어 인증키를 최대한 짧게 하는 것이 유리하므로 생성된 인증키의 앞 32-bit만을 사용한다.

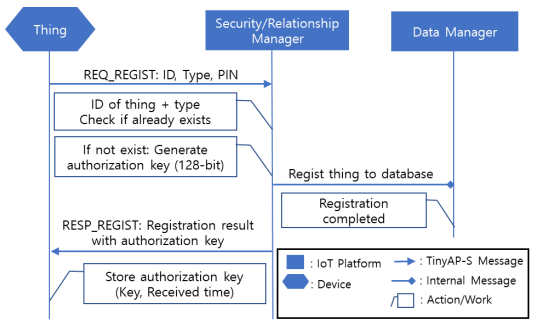


그림 4. 사물의 등록 절차
Fig. 4. Procedure for registering objects

3.4 위치 기반 사물 탐색

IoT 환경에서 대부분의 사물은 자신의 위치를 기반으로 동작한다. 예를 들어, 수백 km 이상 떨어진 사물

간 상대방의 센서 데이터를 직접적으로 필요로 하는 경우는 거의 없다. 반면, 사물이 아닌 사용자는 수백 km 이상 떨어진 자신의 방 안에 위치한 센서의 데이터를 필요로 할 수도 있다.

이러한 점에 착안하여 IoT 플랫폼의 탐색 및 소셜 관계 형성에서 위치 정보를 기반으로 동작하는 기법이 제안되었다.[14] 해당 기법에서 탐색절차는 사물의 위치를 기반으로 수행한다. 사물과 사용자는 GPS 좌표를 기준으로 주변에 위치한 사물에 대한 근거리 탐색을 수행한다. 이를 위해 우선 사물과 사용자는 GPS(사물의 경우)나 무선 이동통신망 신호정보(사용자의 경우)를 사용하여 자신의 위치 정보를 확보하고 이를 IoT 플랫폼과 공유해야 한다.

기존에 제안된 방법에는 사물이 실내에 위치하거나 GPS 모듈을 탑재하지 않는 등, 위치정보를 확보할 수 없는 경우에 대한 구체적인 해결 방안이 제시되지 않았으나, 본 논문에서는 이를 구현 단계에서 사용자의 경우와 동일하게 무선 신호정보를 바탕으로 대략적인 위치를 확인할 수 있는 기능을 탑재할 것을 제안한다.

구체적으로는 저전력 단말의 경우 RSSI 및 SNR, 그리고 3개 이상의 게이트웨이가 가진 IP 주소를 이용한 Geolocation 서비스[15]를 사용하여 거리 계산을 수행하며, IP 기반 사물은 게이트웨이의 경우와 같이 IP 주소를 이용한 Geolocation 서비스를 이용하여 대략적인 위치를 판별한다. 이 방법은 GPS 데이터를 수신할 수 있을 때에도 그 정밀도를 높이는 방법으로 병행하여 사용한다.

3.5 데이터 공유 및 제어

사람이 SNS에서 자신이 가진 정보를 포스팅 하는 것처럼 사물이 자신이 가진 데이터 또는 기타 정보를 다른 사물 및 사용자를 위해 IoT 플랫폼으로 포스팅 하는 것을 데이터 게시라고 한다. 데이터 게시는 Owner가 설정한 데이터 전파대상 제한 조건을 함께 첨부할 수 있다. 전파대상 제한 조건에는 범위 제한 및 디바이스 타입 제한, 권한 제한이 있다. 조건이 설정된 데이터는 해당 조건을 충족하는 사물 및 사용자 디바이스에서만 조회가 가능하다.

특히, 범위 제한을 통해 소셜관계 하에 있는 사물 및 사용자가 멀리 떨어진 상태에서 불필요한 데이터를 받아보지 않도록 유도할 수 있다. 이는 앞서 "4. 위치 기반 사물 탐색"에서 언급한 편의성 측면에서 동일한 이점을 가진다. 소셜관계를 맺은 사물과 사용자간에 전파범위 제한을 포함한 전파대상 제한 방법이 동작하는 과정을 그림 5에 나타내었다.

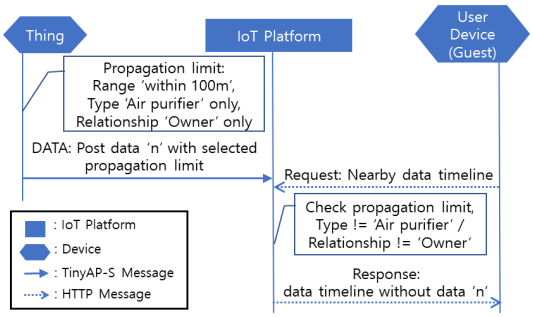


그림 5. 전파대상 제한의 예
Fig. 5. Example of propagation destination restriction

사용자 디바이스가 주변 데이터를 요청하면, IoT 플랫폼은 사용자 디바이스의 현재 위치를 중심으로 데이터가 게시된 위치를 검사한다. 만약 사용자 디바이스 좌표와 사물이 게시한 데이터의 범위정보가 일치할 경우, IoT 플랫폼은 사용자 디바이스에게 해당 데이터를 전달한다. 만약 조건을 만족시키지 못 하면 해당 데이터는 전달되지 않는다.

사물의 활동 데이터를 최근 순서대로 정렬한 것을 "타임라인"으로 정의한다. 활동 데이터는 센서로부터 수집된 데이터, 상태 또는 위치의 변경, 액추에이터 동작 이력 등이 해당된다. 타임라인 조회를 원하는 다른 사물 또는 사용자는 데이터를 최근 순서대로만 정렬할 수 있다.

오래된 데이터는 사물의 Owner에게만 제공하며 사물의 타임라인을 Guest에게도 공개할지에 대한 여부는 사물의 Owner가 설정할 수 있다.

Owner 또는 Owner에 의해 액추에이터 제어가 가능하도록 허가된 Partial Owner는 명령 전송을 통해 사물에 장착된 액추에이터를 제어할 수 있다. 사물이 지원하는 범위 내에서 액추에이터의 종류와 제어방식은 다양할 수 있다.

5.6 비 IP 기반 저전력 사물 지원

본 논문의 IoT 플랫폼은 비 IP 기반 저전력 사물을 IP 기반 사물과 동일하게 지원할 수 있다. 해당 사물들은 슬립모드(sleep mode) 상태에서 오랜 시간을 보내게 되므로 모든 관계 형성 절차는 각 사물과의 통신 없이 IoT 플랫폼 상에서만 이루어지게 된다. 이를 위해 IoT 플랫폼은 사물의 가입절차 및 인증절차에서 고유키를 수신 받아 보관한다.

고유키는 사물의 소유자 인증 단계에서 사용되며 사용자는 고유키 인증을 거친 후 소셜관계 형성에 필요한 모든 절차를 수락하고 사물을 제어할 수 있다.

사물이 오프라인 상태에서도 관계를 형성하고 제어 메시지를 미리 전송해둘 수 있다. 이는 IoT 플랫폼이 버퍼 역할을 하기 때문으로, 플랫폼은 사물이 깨어날 때까지 기다린 후에 데이터를 전송할 수 있다.

소셜관계 형성 및 탐색결과 등은 사물과의 통신 없이 IoT 플랫폼만으로도 처리할 수 있다. 사물이 오프라인 상태일 경우 IoT 플랫폼으로부터 데이터를 전달하는 절차를 그림 6에 나타내었다.

각 사물별로 송신할 제어 메시지는 그 종류에 따라 독립적인 버퍼를 두고 중복되는 명령어는 마지막에 전송된 명령어 한 가지만 전송한다. 이로 인해 사물의 에너지 효율이 증가할 수 있다.

게이트웨이가 자체적으로 보관하고 처리하는 정보가 없으므로 비 IP 기반 저전력 사물들은 물리적으로 이동하여 다른 게이트웨이에 접속하는 경우에도 별도의 핸드오버 절차가 필요하지 않다. ACK (Acknowledge) 신호에 대한 응답, 지연된 제어명령 전송 등의 기능 또한 IoT 플랫폼에서 처리하여 게이트웨이로 전송한다. 이는 사물에서 핸드오버 절차에 대한 구현을 생략함으로써 프로그램의 크기를 줄이고 구현을 단순화하는 효과가 있다.

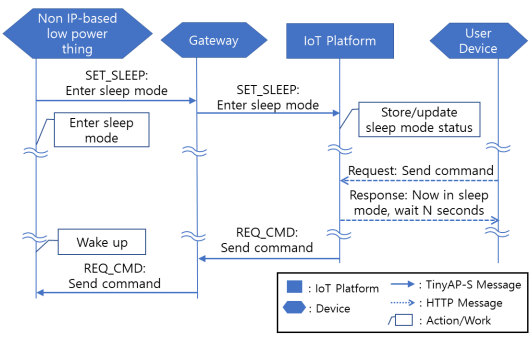


그림 6. 사물이 오프라인 상태(슬립모드)일 경우 IoT 플랫폼으로부터의 데이터 전송절차
Fig. 6. Data transfer procedure from the IoT platform when things are offline (sleep mode)

Ⅳ. 구현 및 시험

각 IoT 사물 및 게이트웨이, IoT 플랫폼을 다음과 같이 구현하고 시험하였다.

4.1 하드웨어 및 펌웨어 구현

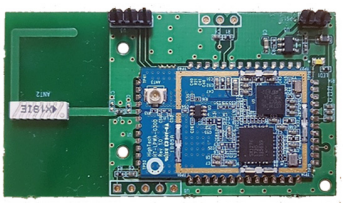
비 IP 기반 저전력 사물은 SEMTech의 SX1276 모뎀과 STM32L0 MCU 기반의 하드웨어를 사용하였다. 이 하드웨어는 UART, SPI 및 I2C 인터페이스를 제공한다. 비 IP 기반 저전력 사물의 펌웨어는 SEMTech사의 SX1276 Standard Library[16]를 사용하였다. 통신은 LoRaWAN 1.0.3 규격에 기반하며 Regional B, Class C를 사용하였다. 개발 환경은 Atollic TrueSTUDIO IDE를 이용하여 C++ 언어로 구현하였다.

IP 기반 사물은 Raspberry Pi 3를 이용하였으며 WiFi 무선 인터페이스에 할당된 공인 IP 환경을 통해 인터넷에 연결하였다. 센서 데이터 수집은 Python 3으로 구현하였으며, IoT 플랫폼과 연동되는 소셜 네트워크 기능과 데이터 송/수신 기능은 PHP 5 스크립트로 구현하였다.

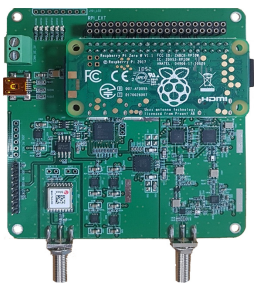
게이트웨이는 Raspberry Pi를 바탕으로 SEMTech SX1301 AP2 Gateway를 포팅하였다. 사물과 게이트웨이는 14~27dBm의 무선 출력으로 통신한다. 100Mbps 이더넷에 할당된 공인 IP 환경으로 인터넷에 연결된다.

사용자 디바이스는 안드로이드 스마트폰으로 구현하였다. 사용자 디바이스 또한 IP 기반 사물이나 게이트웨이처럼 WiFi 무선 인터페이스를 사용, 공인 IP 환경을 통해 인터넷에 연결하였다. 구현과정에서 사용된 디바이스를 그림 7에 나타내었다.

비 IP 기반 저전력 사물에 사용되는 TinyAP-S의 메시지 형식은 크게 기본 헤더와 메시지 부분으로 구성된다.[12] TinyAP의 기본 헤더는 5바이트 길이의 고정 길이를 가지나, LoRaWAN 프로토콜 헤더에는 사



(1) Non-IP based low power thing



(2) Gateway

(3) IP-based thing

그림 7. 실험에 사용된 디바이스 사진
Fig. 7. Devices used for testing

물의 네트워크 주소 및 순번(SEQ) 데이터, CRC 점검 기능이 이미 포함되어 있으므로 TinyAP-S 프로토콜에서는 2바이트로 헤더의 길이를 변경하였다. 그림 8은 변경된 기본 헤더 구조를 나타낸다.

TYPE 헤더의 종류는 TinyAP의 기본 메시지 타입 외에 소셜 네트워크 기능 지원을 위해 표 1의 메시지 타입들을 추가하였다. 확장된 메시지 타입의 이름을 좌측에, 설명을 우측에 기재하였다.

이후 구현내용 상의 모든 TinyAP-S 메시지 형식은 길이(LEN)를 제외한 TYPE 헤더의 종류 및 메시지 부분만을 콤마(",")로 구분하여 나타낸다.

표 1. 확장된 TYPE 종류
Table 1. Extended TYPE for header

Message Type	Description
REQ_OWNERSHIP	Ownership Request
RESP_OWNERSHIP	Ownership Response
REQ_ PARTOWNERSHIP	Partial Ownership Request
RESP_ PARTOWNERSHIP	Partial Ownership Response
REQ_REGIST	Registration Request
RESP_REGIST	Registration Response
REQ_AUTH	Auth Request
RESP_AUTH	Auth Response
REQ_DEAUTH	Logout Request
REQ_TIMELINE	Get Timeline(data) Request
RESP_TIMELINE	Timeline Data Response
REQ_POSUPDATE	Location Update Request
RESP_POSUPDATE	Location Update Response

4.2 IoT 플랫폼 소프트웨어 구현

IoT 플랫폼의 데이터베이스는 MariaDB를 사용하였으며 웹 어플리케이션 서버의 플랫폼은 Eclipse Jetty를 사용하였다. 프론트엔드 웹 서비스로는 Nginx를 다중 구성하였다. 자세한 시스템 사양은 표 2와 같다.

SEMTech packet_forwarder로부터 오는 UDP 데

← 1 byte →	← 1 byte →
TYPE	LENGTH

그림 8. 기본 헤더 구조 (2바이트)
Fig. 8. Basic header structure (2 bytes)

표 2. IoT 플랫폼 주요 소프트웨어 라이브러리 구성
Table 2. Major software configuration of IoT platform

Type	Software	Version
O/S	CentOS	7
Java VM (Server)	OpenJDK	8 (1.8)
Relational Database (RDBMS)	MariaDB	5.6
Web Application Server	Eclipse Jetty	9.4

이터를 수신하고 MAC Command를 송신하기 위해 loraserver[17] 소프트웨어의 HTTP Integration 기능[18]을 활용하였다. 이를 바탕으로 소셜 네트워크 형성 및 유지에 필요한 송, 수신 기능을 개발, 사용하였다.

데이터는 loraserver 시스템에서 버퍼링되고 노드가 슬립모드에서 깨어날 때만 업링크 데이터를 발신하므로 설계된 플랫폼 구조에 부합하다. 모든 사물의 관리 및 유지기능은 LoRaWAN 1.0.3 규격의 Class C에 맞도록 개발하였다. 해당 연결구조를 그림 9에 나타내었다.

TinyAP-S 프로토콜 메시지는 LoRaWAN 메시지의 페이로드에 포함되며 최대 Payload 크기인 51바이트를 넘지 않도록 하였다. RESP_TIMELINE 등 긴 길이의 메시지는 패킷의 맨 마지막에 연속 메시지임을 나타내기 위해 CR/LF (0x0D, 0x0A) 구문을 추가하여 구현하였다. 이는 TinyAP-S Forwarder에서 변환 처리되어 게이트웨이를 거쳐 비 IP 기반 사물로 전달되도록 구성하였다.

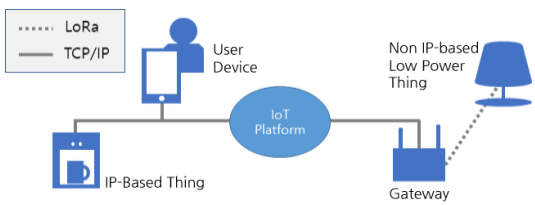


그림 9. IoT 플랫폼과 디바이스간의 연결구조
Fig 9. Connection structure between IoT platform and device

4.2.1 단말 및 사용자의 가입/인증

사물 및 사용자로부터 가입요청이 수신되면 등록절차를 수행한다. 등록 후 로그인(인증)이 완료된 사물 및 사용자에게는 128비트 길이의 인증키를 생성하여 전송한다. 인증키는 생성 시간 정보와 다수의 무작위 값, 사용자 아이디/비밀번호 또는 사물의 고유키 등의 seed를 바탕으로 하여 MD5 해시로 생성된다. 그림 10에 사물 및 사용자의 인증키 해시 값을 생성하는 코

```
// Generate hash for User
function GenerateUserSessionHash(USER_ID,
 USER_PASSWORD) {
  key_val = md5(USER_ID + USER_PASSWORD +
              CURRENT_DATE(Y-m-d H));
  return key_val;
}

// Generate hash for Things
function GenerateDeviceSessionHash(DEVICE_KEY) {
  key_val = md5(DEVICE_KEY +
              CURRENT_DATE(Y-m-d H));
  return key_val;
}
```

그림 10. 인증키 해시 값 생성 코드
Fig. 10. Authentication key hash value generation code

드를 나타내었다.

사물의 경우 별도로 플랫폼의 요구가 있을 때만 해시 값의 비교가 수행되며, 사용자는 매 HTTP 요청마다 인증정보를 Header에 포함하여 전송하고 비교 받는다. 만약 해시 값이 다르다면 IoT 플랫폼에 다시 인증할 것을 요청받는다. 만약 요청한 리소스 대신에 인증 요청 패킷이 돌아오거나 인증 요청 페이지가 나타나면 각 디바이스들은 다시 인증요청을 전송해야 한다.

그림 11에 사용자 및 사물의 인증키 해시 값을 검사하는 코드를 나타내었다.

```
// Hash verification for Things
function CheckDeviceSessionHash(DEVICE_KEY) {
  key_val = md5(DEVICE_ORIGINAL_KEY +
               CURRENT_DATE(Y-m-d H));
  if(key_val == DEVICE_KEY) { return SUCCESS; }
  else { return FAIL; }
}

// Hash verification for Users
function CheckUserSessionHash(HTTP_COOKIE) {
  USER_KEY = HTTP_COOKIE['key_val'];
  key_val = md5(USER_ID + USER_PASSWORD +
               CURRENT_DATE(Y-m-d H));
  if(key_val == STORED_USER_KEY) return SUCCESS;
  else return FAIL;
}
```

그림 11. 인증키 해시 값 검사 코드
Fig. 11. Authentication key hash value check code

4.2.2 소셜관계 형성

최초의 Ownership 및 Partial Ownership 형성은 Guestship 관계에 있는 사용자나 사물의 요청에 의해 진행된다. IoT 플랫폼으로 전송되는 Ownership 요청 메시지의 예를 그림 12에, Partial Ownership 요청 메시지의 예를 그림 13에 나타내었다.

각 사물은 아직 Owner 정보가 설정되지 않았을 경우에 한해 Ownership 요청을 받아들일 수 있다. 그림 12의 예에서 'user_test1' 이라는 사용자가 사물의 Owner가 되기를 희망함을 요청한다. 메시지에는 PIN 정보가 함께 포함되므로 각 사물은 이 값이 자신이 보유한 고유키 정보와 일치하는지를 확인하고 성공 또는 실패를 IoT 플랫폼으로 전송한다. Partial Ownership 요청을 받은 IoT 플랫폼은 우선 사물의 Owner를 확인하고 해당 사용자 디바이스로 Partial Ownership 수락 또는 거절을 확인할 것을 요청한다. Partial Ownership 관계에 관한 정보는 관계를 요청한 사물 및 IoT 플랫폼에서만 관리되고 관계를 요청받은 사물 쪽에는 저장되지 않는다. 따라서 관계가 형성되거나 해제되어도 사물에는 별도의 메시지를 전송하지 않는다.

```
REQ_OWNERSHIP, [USER='user_test1'], [PIN='38533']
```

그림 12. Ownership 요청 데이터의 예
Fig. 12. Example of Ownership request data

```
REQ_PARTOWNERSHIP, [DEVICE_ID='00c747ff']
```

그림 13. Partial Ownership 요청 데이터의 예
Fig. 13. Example of Partial Ownership request data

4.2.3 위치 기반 탐색

소셜 네트워크 기반 IoT 플랫폼상의 위치 정보를 이용한 성능향상을 제안한 관련연구[14]가 있으나 이는 각 사물이 위치 정보를 어떻게 수집할지에 대한 구체적인 정의를 하지 않고 있다. 새로이 제안하는 IoT 플랫폼의 구현에는 사물별 위치정보 수집을 위해 자동화된 서비스를 사용하였다.

비 IP 기반 저전력 사물은 LoRaWAN의 Geolocation 서비스[19]를 활용하여 위치정보 기능을 구현하였다. 이는 게이트웨이 GPS 좌표를 기반으로 동작하며 RSSI 및 SNR, Data Rate[3] 등의 무선 전파 정보를 바탕으로 상세 위치 정보를 수집한다.

IP 기반 사물은 네이버 클라우드 Geolocation 서비스[15]를 이용하여 IP 주소 기반 위치정보를 수집하도

```
function GetGPSByAddress(mapKey, mapQuery) {
  QUERY = "key=" + mapKey + "&query=" + mapQuery
  fp =
  https_open("naveropenapi.apigw.ntruss.com", 443);
  https_url_set(fp, "/map-geocode/v2/geocode");
  https_header_add(fp, "Data", QUERY);
  content =
    xml_to_array(https_get_content(fp));
  mapPoint = array();
  mapPoint['x'] = content['x'];
  mapPoint['y'] = content['y'];
  return mapPoint;
}
```

그림 14. 법정주소를 GPS 좌표로 변환하는 오픈API 코드 (네이버 클라우드)
Fig. 14. OpenAPI code to convert legal address to GPS coordinates (Naver Cloud)

록 구현하였다. 각 사물은 미터 단위의 범위 조건으로 위치 기반 탐색을 수행할 수 있다. 이를 위해 사용자 인터페이스 화면의 지도는 네이버 지도 API의 주소 문자열을 GPS 좌표로 변환하는 서비스[20]를 사용하여 구현하였다. 그림 14는 법정주소를 GPS 좌표로 변환하는 코드이다.

IoT 플랫폼에서 상용 지도 서비스로 HTTP 소켓을 연결하고 지정된 절차에 따라 법정주소를 GPS 좌표로 변환한 위·경도 값을 가져올 수 있다. 이 과정에서 통신 대기시간이 발생할 수 있으므로 IoT 플랫폼상의 모든 요청은 블로킹 콜 (blocking call) 로 처리된다.

4.2.4 데이터 공유 및 제어

데이터의 게시는 LoRa의 일반적인 DATA 메시지 형식으로 전송된다. 그림 15에 센서 데이터를 게시하는 메시지의 예를 나타내었다.

예시의 사물은 타입으로 S_WTEMPHUMI (온·습도 센서), 첫 번째 데이터로 온도에 해당하는 16.4, 두 번째 데이터로 습도에 해당하는 59를 데이터로 전송하였다. 이는 가장 기본적인 데이터 전송 형식에 해당한다.

사물의 현재 위치를 중심으로 IoT 플랫폼에서 데이터를 조회할 수 있는 사용자 또는 사물의 위치 범위를 제한할 수 있다. 여기서 중심이 될 위치는 사물의 이동 여부와 관계없이 각 데이터가 게시된 위치를 기준

```
DATA, [SENSOR=S_WTEMPHUMI], 16.4, 59
```

그림 15. 센서 데이터의 예
Fig. 15. Example of sensor data

```
DATA, [SENSOR=S_WTEMPHUMI], 16.4, 59,
      [SPREAD_POS='0.01']
```

그림 16. 센서 데이터의 예 (전파범위 제한)
Fig. 16. Example of sensor data (limited propagation range)

으로 한다. 전파대상 제한조건을 범위 기반으로 설정한 메시지의 예를 그림 16에 나타내었다. 예시의 SPREAD_POS는 위치 기반 제한을 나타낸다. 데이터는 0.01로 현재 게시된 위치로부터 100m 범위를 나타낸다. 매 데이터 게시마다 자신의 위치를 새로 전송할 필요는 없으며 사물의 마지막 위치는 IoT 플랫폼이 판단하고 데이터베이스에 입력한다.

이의 역방향으로 IoT 플랫폼으로부터 사물로 수신되는 액추에이터 제어 메시지의 예를 그림 17에 나타냈다. 예시에 사용된 제어 메시지는 LED 제어 명령으로, PWM 제어를 통해 조명의 밝기를 제어한다. "100"은 최대 밝기를 의미하며, 0은 반대로 최소 밝기를 의미한다. KEY는 인증키의 해시 값을, ADDR은 제어 대상 장치 (예시에서는 LED)의 주소를 나타낸다. 사물은 보안 유지를 위해 인증키와 수신시간을 합하여 자신이 가진 인증키의 해시 값과 비교한 후 정보가 일치할 경우에만 액추에이터 제어 명령을 수행한다. 센싱 주기, 슬립모드 시간 등의 사물 내부 설정 값을 변경하는 설정 변경 메시지 또한 그림 17과 동일한 패킷 형식으로 수신할 수 있다.

```
REQ_CMD, [KEY='0xcbff...'], [ADDR=0xFF], [PWM='100']
```

그림 17. 액추에이터 제어 메시지 예
Fig. 17. Example of actuator control message

4.3 시험

구현한 플랫폼 및 디바이스를 바탕으로 아래와 같은 시험 시나리오를 구성하였다.

(1) 사용자 디바이스 A에서 주변 탐색을 통해 비 IP기반 사물 A가 존재함을 확인 (Guestship 관계 형성)
(2) 사용자 디바이스 A와 사물 A 관계 형성 (Ownership)
(3) 사용자 디바이스 B가 사물 A와 Partial Ownership 관계 형성
(4) 사용자 디바이스 A가 관계 수락, 제어 권한 할당
(5) 사용자 디바이스 B가 사물 A를 제어
(6) 사물 A가 데이터를 게시하고, 사용자 디바이스 B

가 사물 A가 게시한 데이터를 공유 받음

소셜관계를 설정하기 위해 먼저 사용자 디바이스 A는 인증절차를 거친 후, 주변 탐색을 통해 사물 A의 정보를 확인하였다. 그 결과를 그림 18에 나타내었다.

이후 사용자 디바이스 A와 사물 A간에 Ownership 소셜관계를 맺었다. 탐색된 사물 A에 대해 Owner 요청을 전송하였다. 요청을 수신 받은 IoT 플랫폼은 고유키 입력을 요구하였으며 이를 입력하고 정상적으로 Ownership 관계가 맺어짐을 확인하였다. 그 결과를 그림 19에 나타내었다.

Owner가 설정된 사물 A에 대해 Guestship 관계의 사용자 디바이스로 B부터 Partial Ownership 요청을 전송하였다. 이후 사물 A의 Owner인 사용자 디바이스 A에 PUSH 메시지가 도착함을 확인하였다. 이를 그림 20에 나타내었다.

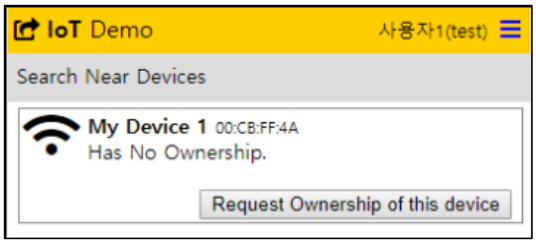
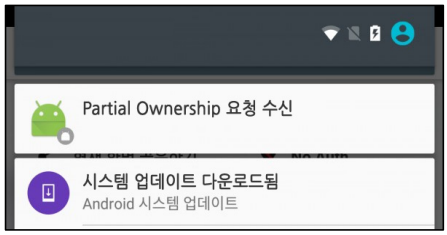


그림 18. 주변탐색으로 확인한 장치 목록
Fig. 18. List of devices identified by 'Search near devices' function

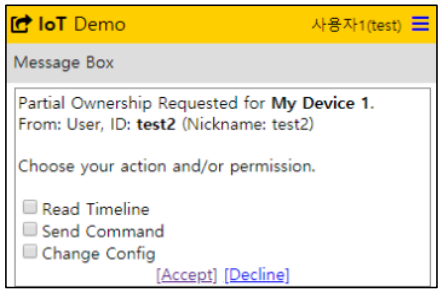
(1) PIN input request

(2) Ownership formation completed

그림 19. 사용자 디바이스 A와 사물 A의 관계형성
Fig. 19. Establishing relationship between user device and thing

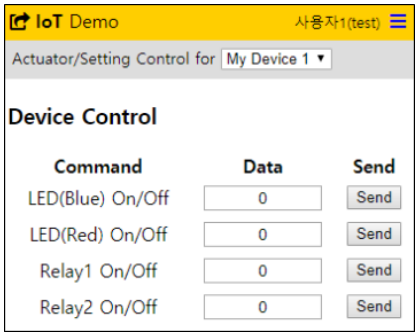


(1) PUSH message received

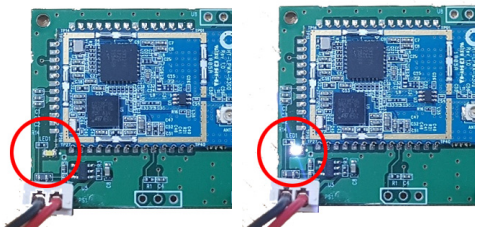


(2) UI for accept or deny request

그림 20. Partial Ownership 요청 메시지 수신 및 수락, 제어 권한 할당
Fig. 20. Partial Ownership request message reception and acceptance, permission assignment interface



(1) User Device B's actuator control message transmission screen

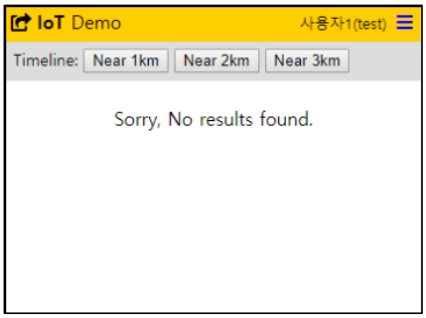


(2) LED control command arrival confirmation of object A

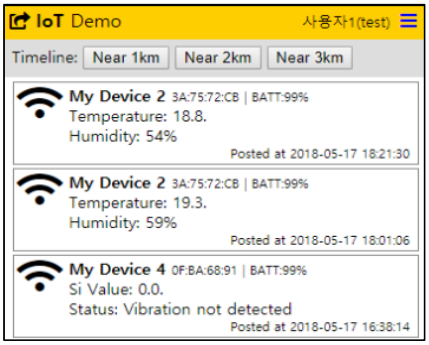
그림 21. 액추에이터 제어 메시지 전송을 통한 LED 켜기
Fig. 21. Turning LEDs on by transmitting actuator control messages

사용자 디바이스 A는 사용자 디바이스 B의 이용자에게 "Send Command" (제어) 권한을 할당하고 수락하였다. 이후 사용자 디바이스 B에서 Partial Ownership 관계를 맺은 사물 A에 액추에이터 제어 메시지를 발신하였다. 제어 메시지를 수신한 사물은 GPIO 포트에 연결된 내장형 LED를 켜도록(ON) 설정하였다. 해당 제어 메시지 수신 동작은 그림 21과 같이 이루어졌다.

사물 A가 범위 조건 1km의 데이터를 게시하고 사용자 디바이스 A의 위치 정보를 해당 위치로부터 2km 이상 떨어진 임의의 위치로 설정한 후, 데이터 타임라인에 접근하도록 하였다. 이를 통해 그림 22의 (1)과 같이 타임라인에 아무 데이터도 표시되지 않음을 확인하였다. 다시 사용자 디바이스 A의 위치 정보를 해당 위치로부터 1km 이내로 변경한 후, 데이터 타임라인에 접근하도록 하였다. 그림 22의 (2)와 같이 타임라인에 데이터가 표시됨을 확인할 수 있었다.



(1) 3km range search results



(2) 1km range search results

그림 22. 사용자 디바이스 위치에 따른 범위제한 조건 탐색 결과
Fig. 22. Search result of range limitation condition based on user device's location

4.4 성능평가

새로운 IoT 플랫폼에 적용된 프로토콜인 TinyAP-S의 성능을 MQTT-SN이 적용되었을 때를 기준으로 시험하였다. 구체적으로는 IoT 플랫폼 성능 중 전체 데이터 송/수신 메시지 크기와 이를 바탕으로 한 에너지 소비량, 사물 이동성 지원에 대한 비교를 수행하였다.

시뮬레이션은 MQTT-SN 및 TinyAP-S를 사용하는 두 개의 비 IP 기반 저전력 사물이 동일한 IoT 플랫폼 연동 기능을 바탕으로 인증 후 센서 데이터를 전송하는 것을 가정하였다. 소셜 네트워크 관련 기능은 사물 인증, 데이터 게시, 인증키 갱신 절차로 간략화 하였다. 슬립모드로부터 웨이크 업 - 사물 인증 및 인증키 갱신 - 데이터 게시 - 다시 슬립모드 진입을 하나의 트랜잭션으로 가정하였다.

4.4.1 동일 기능 하의 바이트 수 비교

각 트랜잭션 당 송·수신되는 메시지별 평균 바이트 수를 바탕으로 무선 송/수신에 필요한 에너지 소비량을 비교하였다. TinyAP-S의 메시지별 평균 바이트 수를 MQTT-SN과 비교한 데이터를 표 3에 나타내었다.

[21]의 연구에 따라 시뮬레이션 파라미터를 결정하였다. 해당 연구의 에너지 소비량 측정결과에 따르면 SF=12 조건 하에서 Payload가 10 비트 증가할 때 마다 무선 데이터 전송시간은 약 80ms 증가한다.[21] 1ms당 1uJ의 에너지를 소비한다고 가정할 경우 MQTT-SN은 TinyAP-S보다 1회 전송당 1,152uJ의 에너지를 더 소비한다.

시뮬레이션은 각 하나의 사물을 하나의 게이트웨이와 1:1로 연동하는 것으로 가정하였다. 이를 바탕으로 앞서 정의한 트랜잭션을 5,000회 반복하였다. 누적된 메시지 및 바이트 수를 그래프로 나타내었다. 그림 23은 데이터 송·수신 과정에서 누적된 총 바이트 수를 나타낸다.

그림 24는 각 사물의 에너지 소비량과 이로 인한 배터리 잔량을 나타낸다. 배터리 총량은 최초 1,000mJ (1,000,000uJ)로 시작하여 1회 전송 시마다 송신되는 바이트의 길이에 따라 감소하였다. 시뮬레이션 결과를 통해 TinyAP-S는 MQTT-SN 대비 동일한 소셜 네트워크 기능을 사용하는 상황에서 송·수신하는 메시지 크기가 더 작음을 확인할 수 있다.

표 3. 평균 바이트 수 비교
Table 3. Comparison of average bytes per transaction

	MQTT-SN	TinyAP-S
Bytes per transaction	33	15

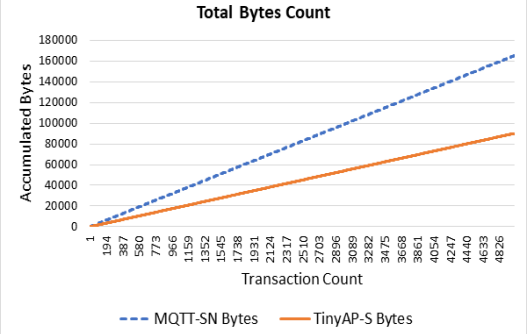


그림 23. 총 바이트 수
Fig. 23. Total Bytes Count

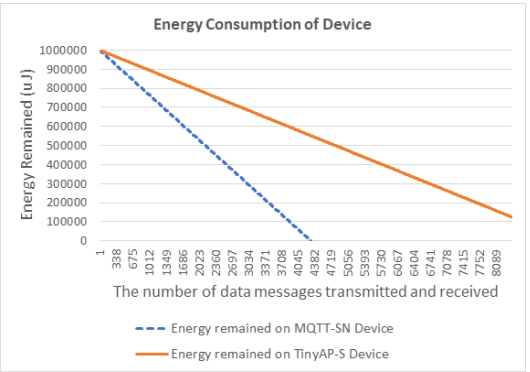


그림 24. 에너지 소비량
Fig. 24. Enegry consumption

TinyAP-S는 슬립모드의 동기화 절차 이후 메시지의 전송이나 슬립모드 재진입 시 별도의 메시지를 교환하지 않는 반면, MQTT-SN은 슬립모드 및 해제과정에서의 연결설정 및 해제 메시지의 교환 작업이 필요하다. 또한, MQTT-SN은 게이트웨이 및 브로커에서 대기 중인 메시지의 존재 여부를 확인하고 이를 별도로 수신하기 위해 PINGREQ 및 PINGRESP 등의 추가적인 메시지를 주고받아야 하기 때문으로 분석된다.[13]

4.4.2 사물 이동성 지원에 대한 비교

배터리로 구동되는 저전력 사물은 고정된 환경에서 운영되지 않을 수 있으며 이를 위해 사물의 이동성을 지원하는 것이 중요하다. 사물 이동성 지원의 성능을 측정하기 위해 게이트웨이에 연결되어 있던 비 IP 기반 저전력 사물이 이동하여 다른 게이트웨이로 연결되고 IoT 플랫폼에 접속하여 다시 인증, 데이터를 전송하는데 까지 소요되는 시간을 시뮬레이션 하였다.

TinyAP-S와 MQTT-SN 각 응용프로토콜을 사용하는 노드가 다른 게이트웨이의 통신범위로 이동한 후

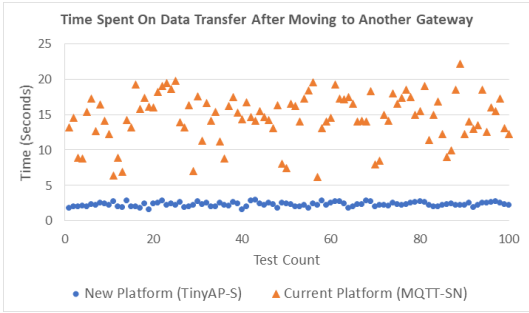


그림 25. 사물 이동 후 데이터 전송까지 걸린 시간
Fig. 25. Time spent on data transfer after moving to another gateway

ACK가 없는 메시지를 1회 전송하여 이를 제안하는 IoT 플랫폼이 저장하기까지 걸린 시간을 100회 측정하여 그림 25에 나타내었다.

사물 이동 후 데이터를 전송하는 데까지 소요되는 시간은 TinyAP-S가 더 짧은 것으로 나타났다. 이는 TinyAP-S 기반 사물은 다른 게이트웨이로 이동하여도 특별히 이동에 관련된 핸드오버 절차를 수행하지 않고 즉시 데이터를 전송할 수 있기 때문이다. 반면에 MQTT-SN 기반 사물은 다른 게이트웨이로 이동하면 게이트웨이가 관리하던 브로커 (broker) 관련 정보를 이용할 수 없으므로 네트워크 가입 절차를 처음부터 다시 수행해야 한다. 이는 게이트웨이 변경 후 데이터를 IoT 플랫폼에 전송하기까지 더 많은 시간이 소요됨을 의미한다. 이러한 성능 개선은 시스템의 실제 효율이 높아지는 것으로 이 시스템의 성능적인 경쟁력이 기존 시스템 대비 높아진다는 점을 의미한다.

V. 결 론

새로운 IoT 플랫폼은 기존 플랫폼 대비 사용자와 사물 또는 사물과 사물간의 탐색 및 관계 형성, 데이터의 공유, 제어 기능면에서 쉽고 유연하게 구성되므로 IoT 플랫폼 기술 발전에 도움을 줄 수 있을 것으로 판단된다. 기존 플랫폼 대비 단순화된 Partial Ownership과 Ownership, Guestship의 세 단계로 구분되는 수직형 관계를 통해 소셜관계를 형성, 사용자가 소유한 사물의 접근 권한과 설정을 쉽게 제어할 수 있으며 소셜관계 형성에 편의를 도모하였다.

향후 IoT 생태계에서는 저전력 무선통신 기술을 바탕으로 한 초소형 IoT 사물이 더욱 활발하게 보급될 것으로 예상된다. 탐색 및 관계형성이 디바이스의 위치를 기반으로 형성되는 IoT 환경의 특성상 실내에

설치된 사물의 위치를 보다 편리하고 정확하게 파악하기 위한 보완이 필요하다. 이는 무선통신 기술의 Geolocation 서비스의 발전 및 본 IoT 플랫폼의 발전이 동시에 이루어져야 할 것으로 판단된다.

소셜 네트워크 기법에 기반을 둔 사람과 사물, 또는 사물과 사물의 관계 형성 및 관리기술은 분명한 장점과 잠재적인 가능성을 가지고 있다. 향후 보다 많은 응용들에의 적용과 운영을 통해 플랫폼의 적정성을 검증하고 분석하여 부족한 부분을 보완해 나가고자 한다.

References

[1] A. Sabri, "A proposed social web of things business framework," *2017 IEEE ICET*, Antalya, Turkey, Aug. 2017.

[2] L. Atzori, A. Lera, G. Morabito, and M. Nitti, "The social internet of things (siot) – when social networks meet the internet of things: Concept, architecture and network characterization," *Computer networks*, vol. 56, no. 16, pp. 3594-3608, 2012.

[3] A. Augustin, et al., "A study of LoRa: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, Oct. 2016.

[4] S. A. Sigfox, "*Sigfox technology overview*," Retrieved July 25, 2019, from https://www.sigfox.com/en/sigfox-iot-technology-overview

[5] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Commun. Surv. & Tuts.*, vol. 19, no. 2, pp. 855-873, 2017.

[6] F. Terroso-Saenz, et al., "An open IoT platform for the management and analysis of energy data," *Future Generation Computer Systems*, vol. 92, pp. 1066-1079, 2019.

[7] H. M. Al-Kadhim and H. S. Al-Raweshidy, "Energy efficient and reliable transport of data in cloud-based IoT," *IEEE Access*, vol. 7 pp. 64641-64650, 2019.

[8] R. Battle and E. Benson, "Bridging the semantic Web and Web 2.0 with representational state transfer (REST)," *Web Semantics: Science, Services and Agents on the World Wide Web 6.1*, pp. 61-69, 2008.

[9] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734-11753, 2012.

[10] L. De Nardis and M.-G. Di Benedetto, "Overview of the IEEE 802.15. 4/4a standards for low data rate Wireless Personal Data Networks," *IEEE 2007 4th Workshop on Positioning, Navig. and Commun.*, Hannover, Germany, Mar. 2007.

[11] D. M. Hernandez, et al., "Energy and coverage study of LPWAN schemes for Industry 4.0," *2017 IEEE Int. Workshop of ECMSM*, Sebastian, Spain, May 2017.

[12] M. Yu and S. Kim. "TinyAP: Lightweight application protocol for LoRa environment," *J. KICS*, vol. 43, no. 1, pp. 27-36, 2018.

[13] A. Stanford-Clark and H. L. Truong, *MQTT for sensor networks (MQTT-SN) protocol specification*, version 1.2 (2013), Retrieved Jul. 30, 2019, from http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

[14] M. Yu and S. Kim, "Improvement of social IoT platform using location information," *J. KICS*, vol. 43, no. 11, pp. 1873-1876, 2018.

[15] Naver Cloud Platform, *GeoLocation API*, Retrieved Jul. 30, 2019, from https://www.ncloud.com/product/applicationService/geoLocation

[16] Semtech Corporation, *SX1276*, Retrieved Jul. 19, 2019, from https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276#download-resources

[17] loraserver.io, *LoRa Server*, Retrieved Jul. 9, 2019, from https://www.loraserver.io/loraserver/overview/

[18] loraserver.io, *HTTP Integration*, Retrieved Jul. 19, 2019, from https://www.loraserver.io/lora-app-server/integrate/sending-receiving/http/

[19] LoRa Alliance, *Geolocation Whitepaper*, Retrieved Jul. 19, 2019, from https://lora-alliance.org/sites/default/files/2018-04/geolocation_whitepaper.pdf

[20] Naver Cloud Platform, *Map API*, Retrieved Jul. 19, 2019, from https://www.ncloud.com/product/applicationService/maps

[21] T. Bouguera, et al., "Energy consumption model for sensor nodes based on LoRa and LoRaWAN," *Sensors*, vol. 18, no. 7, 2018.

유 명 한 (Myung-han Yu)

2011년 : 강원대학교 전자공학과 석사

2018년 : 강릉원주대학교 컴퓨터공학과 박사

<관심분야> IoT, USN, Wireless Networking

[ORCID:0000-0003-3994-7130]

김 상 경 (Sangkyung Kim)

한국통신학회논문지 Vol. 43, No. 1 참조

[ORCID:0000-0002-1410-9918]