

논문 2017-12-41

협업가능 표준기반 IoT 시스템을 위한 자가적응 IoT 소프트웨어 플랫폼 개발

(Self-adaptive IoT Software Platform for Interoperable Standard-based IoT Systems)

성낙명, 윤재석*

(Nak-Myoung Sung, Jaeseok Yun)

Abstract : In this paper, we present a self-adaptive software platform that enables an IoT gateway to perform autonomous operation considering IoT devices connected each other in resource-constrained environments. Based on the oneM2M device software platform publicly available, we have designed an additional part, called SAS (self-adaptive software) consisting of MAM (memory-aware module), NAM (network-aware module), BAM (battery-aware module), DAM (data-aware module), and DH (decision handler). A prototype system is implemented to show the feasibility of the proposed self-adaptive software architecture. Our proposed system demonstrates that it can adaptively adjust the operation of gateway and connected devices to their resource conditions under the desired service scenarios.

Keywords : IoT, Software platform, Self-adaptive, Resource-constraint, IoT standards

1. 서론

임베디드 시스템은 우리가 일상생활에서 사용하는 가전제품, 제조현장에서 활용되는 생산로봇 및 기계, 전기에너지 사용량 측정 및 관리 장치, 농수산업 장비, 물류 및 수송을 위한 차량 등 다양한 분야에서 중추적인 역할을 수행해 오고 있다. 이러한 임베디드 시스템은 유무선 근거리 통신기술의 발달, 이동 통신망 보급, 스마트폰 대중화로 인해 서로 연결되어 정보를 공유하고 협업이 가능하게 되는 하나의 거대한 동적 네트워크를 구성할 수 있게 되었

*Corresponding Author (yun@sch.ac.kr)

Received: July 11 2017, Revised: Aug. 17 2017,

Accepted: Nov. 17 2017.

N.M. Sung: Korea Electronics Technology Institute

J. Yun: Soonchunhyang University

※ 본 연구는 순천향대학교 연구비 지원으로 수행하였음 (No. 20170684).

※ 본 논문은 2017년도 정부 (교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 임 (NRF-2017R1D1A3B03032676).

다. 사물인터넷 (IoT, The Internet of Things)으로 불리우는 이러한 현상은 1999년 처음 소개된 이래 정보통신기술의 발전 방향을 이끌어 왔으며, 앞으로 다가올 생활과 산업에 가장 큰 변화와 영향을 줄 것으로 예상되는 기술 분야로 분류되고 있다 [1].

사물인터넷 시장의 성공을 위해서 가장 중요한 것은 시스템 간의 상호호환성 (interoperability)이다. 사물인터넷 특성상 다양한 종류의 하드웨어로 구성되고, 서로 다른 소프트웨어를 탑재한 임베디드 시스템들이 상호연결 되기 위해서는 공통된 데이터 구조와 약속된 인터페이스 방법을 규정하는 표준을 만들고 관련 미들웨어를 사용하는 것이 하나의 해법이라고 볼 수 있다 [2].

이러한 노력의 일환으로 2012년 세계 7개 표준화 기구가 모여 oneM2M 표준 협의체를 구성하였으며 2015년 첫 번째 표준인 릴리즈 1을 발표하였다. oneM2M 표준은 '세계 어디에서나 접근이 가능하고 서로 협업이 가능한 사물인터넷 서비스 개발을 위한 규약'을 목표로 개발되어 왔으며 이를 기반으로 대한민국 에스케이텔레콤의 ThingPlug [3], 미국 인터디지탈의 oneMPOWER [4] 등 많은 회사 및 연구소에서 사물인터넷 서비스 플랫폼을 개발하

여 공개하고 있다.

oneM2M과 같은 서비스 계층의 표준들은 사물인터넷 서비스를 위해 공통으로 필요한 기능, 예를 들어 디바이스 등록, 데이터 검색 및 관리, 구독 및 통지, 애플리케이션 관리, 보안, 과금 등을 중점적으로 다루고 있다. 그러나 디바이스 리소스를 접근하거나 펌웨어 업데이트 등 관리를 위한 동작 제어는 표준간의 중복개발을 피하기 위해 oneM2M 표준화에 참여하고 있는 산업포럼인 OMA (open mobile alliance, [5])와 BBF (broadband forum, [6])에서 규정한 방식을 가져와 oneM2M에서 활용하고 있다. 또한 제한된 환경에서 디바이스 통신을 위한 프로토콜인 CoAP (constrained application protocol, [7])이나 대규모 디바이스 통신을 위한 프로토콜인 MQTT (message queuing telemetry transport, [8])는 프로토콜 바인딩 기법을 통해 시스템 연동을 지원하고 있다.

위에서 기술하였듯이 oneM2M과 같은 서비스 계층의 표준과 이를 기반으로 개발된 소프트웨어 플랫폼들은 리소스와 연결성이 충분한 사물인터넷 디바이스들에게 적용하기에는 이상적이나, 리소스가 제한된 환경 (resource-constrained environment)에서 사용되는 경우, 예를 들어 배터리, 메모리, 네트워크 상태에 맞추어 동작하고, 수집하는 데이터 성격 등을 고려하여 외부에 전송해야 하는 응용 시나리오에는 추가적인 개발이 필요한 상황이다.

본 논문에서는 사물인터넷 표준인 oneM2M을 기반으로 공개된 소프트웨어 플랫폼을 활용하여, 시스템 간 상호호환성을 보장하면서도 연결된 디바이스들의 리소스 조건들을 고려하여 설정된 상황과 시나리오에 따라 스스로 동작을 제어할 수 있는 자가적응 소프트웨어 플랫폼을 제안한다. 공통 데이터 구조와 인터페이스로 대표되는 표준의 이점을 그대로 따르면서, 리소스 모니터링 모듈들을 추가하여 연결된 디바이스들의 리소스 상태를 수집하고 이를 기반으로 설정된 상황과 시나리오에 맞추어 동작하는 제어 모듈을 개발하였다. 최종적으로 프로토타입 시스템을 통하여 제안된 플랫폼의 동작을 검증한다.

II. 관련 연구

1. 사물인터넷 미들웨어

Bandyopadhyay는 사물인터넷 미들웨어가 제공해야할 주요 특징을 디바이스 관리, 상호호환성, 플랫폼 이식성, 맥락 인지, 보안 및 프라이버시로 구

정하고 이에 따른 사물인터넷 미들웨어를 조사해 발표하였다 [9]. Chaqfeh는 기존 사물인터넷 미들웨어를 조사하여 상호호환성, 확장성, 추상화, 상호작용, 동적 인프라, 보안 및 프라이버시 등 향후 필요한 연구분야에 대해 정리하였다 [10]. 특히 서로 다른 도메인과 애플리케이션 간의 상호연동을 위해 미들웨어의 표준화 이슈에 대해 지적하며 표준화된 사물인터넷 미들웨어의 필요성을 제시하였다. 최근에는 Razzaque가 50개가 넘는 사물인터넷 미들웨어를 조사하여 기능적 요구사항 (리소스 디스커버리, 데이터 관리 등), 비기능적 요구사항 (확장성, 보안, 실시간성 등), 구조적 요구사항 (상호호환성, 맥락 인지, 자동 적응성)에 따라 분류하였다 [11]. 조사 결과에 따르면 특정 애플리케이션을 위한 미들웨어 중심으로 주변 환경 변화에 따라 내부 동작을 적응시키는 기능을 하드-코드화하여 탑재하고 있으나, 보다 더 유연하고 동적으로 환경 변화에 적응하는 미들웨어의 필요성을 주장하였다. Yun은 사물인터넷 가전기기를 위한 미들웨어로 oneM2M 표준 기반 소프트웨어 플랫폼인 '&Cube'를 제안하였으며 커넥티드 홈 애플리케이션 개발을 통해 제안한 미들웨어의 활용성과 확장성을 보였다 [12].

2. 리소스 제한 디바이스 미들웨어

사물인터넷이 주요 연구 주제로 등장하기 전부터 스마트폰과 같이 리소스 제한 디바이스를 위한 미들웨어 연구가 활발히 이루어졌다. Perera는 리소스가 제한된 이동형 디바이스들을 위한 사물인터넷 미들웨어 MOSDEN (mobile sensor data processing engine)을 제안하였다 [13]. 기존의 Global Sensor Network (GSN, [14]) 프로젝트에서 개발된 미들웨어 솔루션에 플러그인 구조를 적용하여 사용성과 확장성을 높였으며, 안드로이드 기반의 스마트폰과 태블릿에서 실제 연결된 센서와 쿼리의 개수에 따른 CPU, 메모리, 에너지 사용량의 변화를 측정하는 실험을 통해 제안한 미들웨어의 우수성을 검증하였다. Lane은 모바일 폰에 탑재된 센서 (가속도계, 자이로스코프 등)를 활용하는 센싱 알고리즘, 애플리케이션, 시스템 구조 및 관련 이슈들을 정리하여 발표하였다 [15]. 또한 Brouwers는 모바일폰 기반 센싱을 위한 미들웨어인 'Pogo'를 제안하였고, 안드로이드에서 구현 및 실험을 통해 프로그램 간결성과 전력 소비를 검증하였다 [16]. 그러나 위에서 기술한 리소스 제한 디바이스를 위한 미들웨어는 비표준 시스템을 가정하여 제안되었으며, 사물인터넷 미들웨어의 가장 중요한 특징인

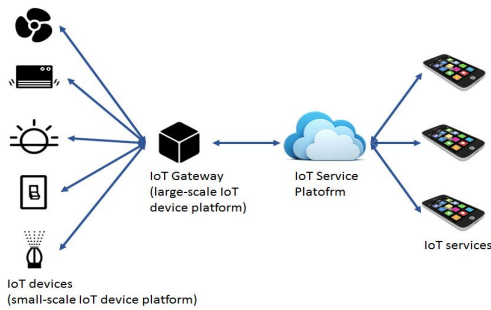


그림 1. 사물인터넷 시스템 구조
Fig. 1 IoT system architecture

확장성과 상호호환성이 필수적인 사물인터넷 연동 서비스 제공 (예, [17])을 위해서는 표준 기반의 사물인터넷 미들웨어에 리소스 제한 환경을 위한 특성을 고려한 기능과 알고리즘 개발이 필요하다.

III. 자가적응 소프트웨어 플랫폼 구조

확장성과 상호연동이 가능한 사물인터넷 시스템을 위해서는 표준 기반 소프트웨어 플랫폼이 필요하다 [17]. 추가로 자가적응에 필요한 기능적 요구 사항과 소프트웨어 구조는 발표된 논문에서 제안하였으며 [18, 19], 다음과 같이 정리할 수 있다.

1. 자가적응 소프트웨어 플랫폼 기능 정의

일반적으로 사물인터넷 시스템은 그림 1에서 보이는 바와 같이 서비스 플랫폼, 게이트웨이, 그리고 연결되는 디바이스들로 구성된다.

본 논문에서 제안하는 자가적응 소프트웨어 플랫폼은 많은 수의 디바이스가 연결되는 하나의 게이트웨이로 연결되는 시나리오를 가정하였다. 이렇게 다수의 디바이스와 다양한 형태의 데이터가 존재하는 서비스 시나리오일 경우, 게이트웨이 플랫폼과 디바이스 플랫폼의 제한된 리소스를 상황에 따라 자율적으로 조절하여 최적의 동작을 보장하기 위한 자가적응 모듈이 필요하다. 예를 들어 그림 2에서 보듯이, 자가적응 모듈안에 존재하는 자가인지 모듈이 '메모리 부족'을 예측할 경우 연결되어진 디바이스 플랫폼에게 데이터 보고 주기를 해당 디바이스의 특성에 맞게 조절함으로써 메모리 부족에 대한 자가적응 기능을 수행할 수 있다.

본 논문에서는 자가적응 소프트웨어 플랫폼에 필요한 기능을 그림 3과 같은 개별 모듈로 정의하였다. 게이트웨이의 메모리 상태를 모니터링하고

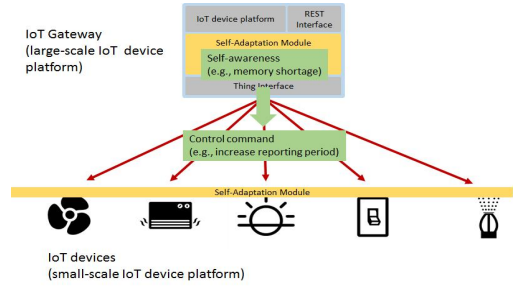


그림 2. 사물인터넷 게이트웨이와 디바이스 자가적응 시나리오
Fig 2 Self-adaptive service scenario for IoT systems

Required functions	Self-aware Module	Control Module
SAF 1: Adaptive memory management	Memory Aware Module	Decision Handler
SAF 2: Adaptive delivery handling based on network status	Network Aware Module	
SAF 3: Adaptive operation management based on battery status	Battery Aware Module	
SAF 4: Adaptive behavior management based on data content	Data Aware Module	
		Adaptive reporting period control
		Adaptive delivery control

그림 3. 자가적응형 소프트웨어 기능별 정의
Fig 3 Self-adaptive software functions

DH (Decision handler)로 보고하는 MAM (memory aware module), 디바이스의 네트워크 자원에 대한 상태를 모니터링하고 DH로 보고하는 NAM (network aware module), 디바이스에 연결된 센서들의 배터리 상태를 모니터링하고 DH로 보고하는 BAM (battery aware module), 디바이스에서 수집한 사물 정보를 분석하여 DH로 보고하는 DAM (data aware module)으로 구성되어 있다. 마지막으로 DH는 위 네 개의 모듈로부터 보고된 모니터링 정보를 통해 사물인터넷 서버로 데이터에 대한 송신을 수행할 것인지를 판단하거나 디바이스의 데이터 전송 보고 주기 판단 및 조절하는 역할을 수행한다.

2. 자가적응 기능탑재 게이트웨이 구조

사물인터넷 게이트웨이는 크게 다음과 같은 두 가지 기본 기능을 가진다. 첫째는 센서/액츄에이터 등의 실제 사물과 연결하기 위한 하드웨어 인터페이스를 가지며 이를 통해 사물과 직접적으로 연결을 수행하는 사물 연동 기능이다. 둘째는 사물인터넷

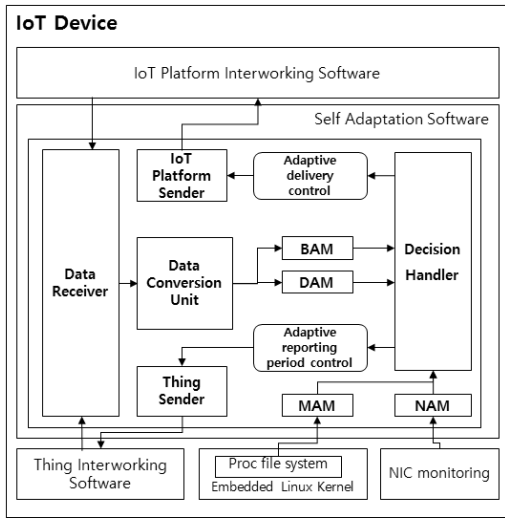


그림 4. 자가적응형 소프트웨어 적용 IoT 게이트웨이

Fig. 4 IoT GW with self-adaptive software

서비스를 위해 서버 플랫폼과 연동을 통한 사물정보 제공 및 사물제어 메시지 수신을 수행하는 사물인터넷 서버플랫폼 연동 기능이다.

본 논문에서 제안하는 사물인터넷 게이트웨이는 그림 4와 같이 사물 연동 기능을 제공하기 위한 사물 연동 소프트웨어 (thing interworking software)와 사물인터넷 서버플랫폼 연동 기능을 제공하기 위한 사물인터넷 서버플랫폼 연동 소프트웨어 (IoT platform interworking software)를 포함하고 추가적으로 실제 디바이스의 메모리, 네트워크, 배터리, 데이터에 대한 모니터링을 통해 자가적응 기능을 수행하는 자가적응 소프트웨어 (self adaptation software)의 세 가지 소프트웨어로 구성된다.

자가적응 소프트웨어 내부의 동작을 살펴보면 MAM 모듈은 사물인터넷 게이트웨이 OS에서 제공하는 메모리 관리 기능을 활용하여 현재 메모리 상태에 대한 모니터링 결과 및 판단결과를 DH로 전송한다. NAM 모듈은 사물인터넷 게이트웨이의 네트워크 인터페이스 카드 (NIC: network interface card)에 할당된 자원을 모니터링 함으로써 네트워크 자원 활용도를 체크하고 판단결과를 DH로 전송한다. 또한 Data Receiver로부터 수신된 사물 정보가 Data Conversion Unit을 통해 자가적응 소프트웨어가 활용할 수 있는 형태로 변환한 후 이를 BAM과 DAM 모듈로 전송하고 각각 BAM은 연결된 디바이스의 배터리 상태를 모니터링 하여 DH로

전송, DAM은 디바이스 데이터에 대한 정보를 분석하여 DH로 전송한다.

위와 같은 동작을 통해 DH로 각각의 모니터링 정보 및 판단정보 (메모리, 네트워크, 배터리, 데이터)가 통합되면 DH는 현재 게이트웨이에 연결된 사물인터넷 디바이스의 전체 상황에 대해 판단하고 사물인터넷 서버플랫폼으로 데이터에 대한 송신을 수행할 것인지 사물 (즉 디바이스)의 데이터 전송 보고 주기를 조절할 것인지 판단한다.

DH가 사물인터넷 서버플랫폼으로 디바이스에서 수집된 데이터 송신을 수행할 경우 Adaptive Delivery Control 모듈로 데이터를 전달하고 이는 IoT Platform Sender를 통해 사물인터넷 서버플랫폼 연동 소프트웨어로 전달된다. 또한 DH가 사물의 데이터 전송 보고 주기를 조절할 경우 Adaptive Reporting Period Control 모듈로 보고주기 조절 명령을 전송하고 이는 Thing Sender를 통해 사물 연동 소프트웨어로 전달된다.

이러한 자가적응 소프트웨어 동작을 통해 처음 기술한 기본 기능 (사물 디바이스 연동, 서버플랫폼 연동)만 가지는 일반적인 사물인터넷 게이트웨이와는 다르게 사물 연동에 사용되는 자원과 사물인터넷 서버플랫폼 연동에 사용되는 자원을 효율적으로 조절하여 최적의 동작을 수행할 수 있다.

IV. 자가적응 소프트웨어 플랫폼 개발

위에서 기술한 자가적응 소프트웨어를 개발하기 위해서 사물인터넷 표준 기반 오픈소스 연합체인 OCEAN (Open allianCE for iot sTANdard, [20])에서 공개한 소프트웨어 플랫폼인 ‘&Cube’와 서비스 플랫폼인 ‘Mobius’를 활용하였다.

사물인터넷 디바이스에 탑재되는 플랫폼은 데이터 발생 및 전송에 대한 자가적응을 수행하기 위해 게이트웨이로부터 수신되는 자가적응 명령에 따라 센싱 및 보고주기를 조절하는 기능을 구현하였다. 게이트웨이에 탑재되는 플랫폼은 게이트웨이가 활용중인 자원 (메모리, 네트워크, 배터리 등)에 대해 모니터링을 수행하며 &Cube 플랫폼이 제공하는 Mobius 연동 기능을 활용하여 인터넷을 통해 게이트웨이의 정보를 확인하고 제어를 수행할 수 있도록 구성하였다. 또한 현재 게이트웨이의 자원 상황에 따라 연결된 사물인터넷 디바이스의 보고주기를 자율적으로 조절하거나, 사물인터넷 서비스 플랫폼의 요청에 따라 수동으로 조절하여 전체적인 게이트

표 1. 사물인터넷 디바이스 하드웨어

Table 1. IoT device hardware

Class.	Name	Memory	RAM	External Sensor
SoC	TI CC2530	64KB In-System-Programmable Flash	8KB	Temperature, Humidity, Illumination, Watt-hour meter

표 2. 사물인터넷 게이트웨이 하드웨어

Table 2. IoT gateway hardware

Class.	Name	Processor	RAM	OS
Ev. Board	Innopia Tech IOT2	Amlogic S805 (ARM Cortex A5-based)	2GB of LPDDR3 RAM	Linux (Android 4.3)

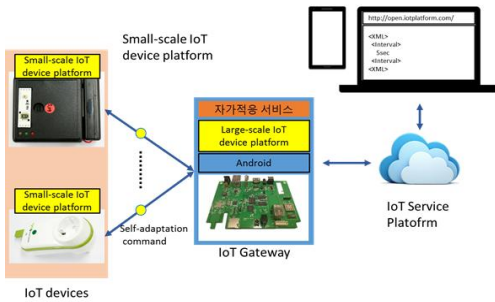


그림 5. 오픈소스 활용 자가적응형 소프트웨어 개발
Fig. 5 Self-adaptive software development

웨어의 자원을 효율적으로 활용할 수 있도록 자가 적응 기능을 개발하였다.

개발한 사물인터넷 디바이스 플랫폼은 표 1과 같이 컴퓨팅 파워가 낮고 인터넷에 직접 연결이 불가능한 경량의 하드웨어에 탑재하였으며 게이트웨이 플랫폼의 경우 표 2와 같이 충분한 자원을 가지고 하부에 디바이스들이 연결되어 다양한 자원을 수집하고 연산을 처리할 수 있는 하드웨어에 탑재하여 그림 5와 같이 서비스 개발을 위한 환경을 구성하였다.

V. 서비스 개발 및 기능 검증

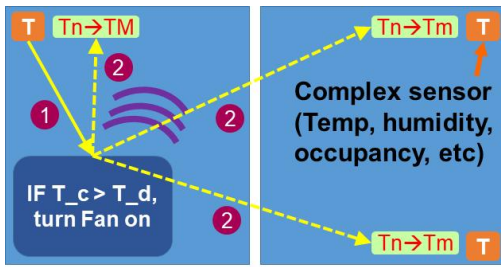
개발한 자가적응 소프트웨어 플랫폼에 대해서 다음과 같이 기능별 검증작업을 수행하였다.

게이트웨이 플랫폼은 안드로이드 운영체제가 제공하는 자원활용 상태 모니터링 기능을 통해 게이트웨이의 메모리 상태를 모니터링하고 (MAM), 네트워크 자원에 대한 상태를 모니터링 (NAM)하여 DH를 통해 연결된 디바이스의 보고주기를 조절하도록 명령을 전송하여 전체적인 게이트웨이 자원 활용성을 유지하는 것을 볼 수 있었다. 또한 디바이스로부터 데이터 수신 시 해당 정보를 분석 (DAM)하여 정보에 대한 중요도를 판단하고 동시에 수집된 배터리 상태 정보를 파악 (BAM)하여 이를 DH에 전달하는 기능을 검증하였다.

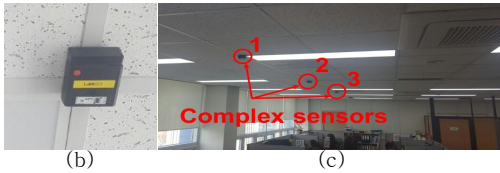
디바이스 플랫폼은 SoC에서 활용되기 때문에 센싱 데이터 발생 시 메모리, 네트워크 자원의 활용 상태와 배터리 레벨을 체크하여 (MAM, NAM, BAM) 보고주기에 따라 게이트웨이로 전송하는 기능의 동작을 확인하고 게이트웨이로부터 보고주기 변경 명령을 수신했을 시 보고주기에 대한 변경이 정상적으로 이루어지는지에 대해 검증하였다.

추가적으로 게이트웨이와 디바이스 플랫폼의 각 단위 기능에 대한 통합 검증을 위해 디바이스-게이트웨이-서버-서비스의 전반적인 사물인터넷 환경 하에서 자가적응 소프트웨어 기능이 필요한 상황을 설정하고 연결된 디바이스들의 보고주기를 조절하는 시나리오를 개발하였다. 아래 시나리오 (그림 6-a)는 보고주기가 T_n 으로 설정된 온도센서 디바이스들이 설치된 방에서 냉방기기의 자동운전 (현재 온도 T_c 가 설정온도 T_d 보다 높으면 동작, ①)에 따라 해당 구역의 디바이스들의 보고주기를 T_n 에서 T_m 으로 조절하고 나머지 디바이스들의 보고주기는 T_n 에서 T_m 으로 조절하는 자가적응 소프트웨어 동작 (②)을 설명하고 있다.

위에서 기술한 시나리오를 실제로 검증하기 위해 그림 6-b에서 보이듯이 4장에서 개발한 디바이스 플랫폼을 탑재한 온/습도센서가 내장된 복합센서들과 스마트 소켓이 연결된 냉방기기를 실험실 공간에 배치하였다 (그림 6-c). 최종적으로 냉방이 이루어졌을 때 냉방이 이루어지는 지역을 집중적으로 모니터링하기 위해 근처 디바이스의 보고주기가 T_m 으로 변화하고 (그림 6-d의 좌상단/우하단 그래프 참조) 나머지 디바이스들은 보고주기가 T_m 으로 변화하는 모습을 그림 6-d 우상단/좌하단 그래프에서 확인할 수 있다. 그래프에서 X축은 시간, Y축은 보고주기를 나타내며 액추에이터 (냉방기기-스마트 소켓 제어)가 가동이 된 구역 (복합센서 1, 스마트 소켓 위치)에서 냉방기기로 인한 전력소모가 지속적으로 일어나기 때문에 목표 온도 도달 시 냉방기기

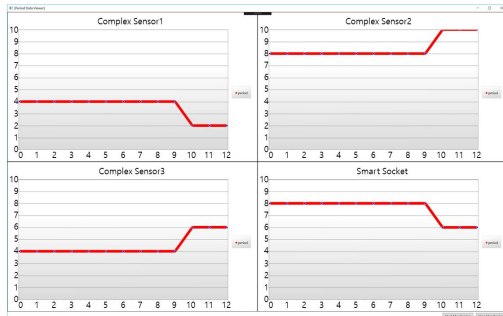


(a)



(b)

(c)



(d)

그림 6. 자가적응 소프트웨어 동작 검증
Fig. 6 Self-adaptive software evaluation

가동을 즉시 중지하기 위한 온도 모니터링이 중요하므로 해당구역을 집중 모니터링 (보고주기 $T_n \rightarrow T_m$)을 하면서 상대적으로 중요도가 떨어지는 구역 (복합센서 2, 3 위치)의 온도 데이터의 보고주기를 길게 조절함으로써 ($T_n \rightarrow T_m$) 단위시간당 연산량과 네트워크 자원 활용도를 일정하게 유지하는 자가적응 통합기능이 정상적으로 동작함을 확인하였다.

또한 IoT 게이트웨이의 메모리/네트워크 자원 활용 측면에서 보았을 때, 1차적으로 개별 디바이스에서 수신되는 온도/전력 사용량 데이터를 IoT 서버 플랫폼으로 전송하기 위한 데이터 가공 작업 (HTTP/XML 포맷 변환) 시 메모리 자원 활용도 증가, 2차적으로 가공된 데이터를 서버로 전송할 때 네트워크 자원 활용도가 증가하게 된다. 즉, 하루에 연결된 모든 디바이스의 매 보고주기마다 올라오는 데이터에 대해 데이터 중요도 및 상황 분석에 따른 보고주기의 조절을 통해 서버 전송을 위한 데이터 가공이 이루어짐에 따라, 게이트웨이의 메모리와 네

트워크 자원 활용도에 대한 조절이 가능하다. 그림 6에서 가정한 시나리오와 게이트웨이의 활용 가능 메모리/네트워크 자원에 제약사항을 두고 기능 검증을 수행한 결과 상황 변화에 따른 디바이스들의 보고주기를 변경하면서도 동일한 자원 활용도를 보이는 것을 확인하였다.

VI. 결론

사물인터넷 시스템의 협업을 위해 반드시 필요한 소프트웨어 플랫폼의 기능은 확장성과 상호연동성이며 이를 위해 국제 사물인터넷 표준을 활용한 소프트웨어 플랫폼을 개발하기 위한 많은 연구가 이루어지고 있다. 본 논문에서는 이에 추가적으로, 기존에 연구된 플랫폼에서 고려하지 못했던 사물인터넷 디바이스의 소형화, 경량화, 저전력, 상황 변화에 따른 대응을 위한 자가적응 소프트웨어 플랫폼의 필요성을 제시하였다. 특히 리소스가 제한된 디바이스의 최적화된 리소스 사용을 위해 사물인터넷 디바이스들의 메모리, 네트워크, 배터리, 데이터를 모니터링하여 게이트웨이로의 보고주기 와 사물인터넷 서비스 서버로의 데이터 송신을 조절하는 자가적응 소프트웨어 모듈을 개발하였다. 단위 기능 검증과 시나리오를 통한 통합 기능 검증을 통해 본 논문에선 제안한 자가적응 소프트웨어 플랫폼의 확장성 및 시스템 간의 상호연동성을 보였다. 또한 사물인터넷 게이트웨이와 디바이스의 리소스와 상황 변화에 따른 자가적응 기능의 유용함을 시나리오 검증을 통해 확인하였다.

References

- [1] L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A survey," Computer Networks, Vol. 54, No. 15, pp. 2787-2805, 2010.
- [2] P.A. Bernstein, "Middleware: A Model for Distributed System Services," Communications of the ACM, Vol. 39, No. 2, pp. 86-98, 1996.
- [3] <https://sandbox.sktiot.com/>.
- [4] <http://www.interdigital.com/solution/onempower-platform>.
- [5] oneM2M. TS-0005-Management-Enablement (OMA)-V-1.0.1. Technical Specification (2015).
- [6] oneM2M. TS-0006-Management-Enablement (BBF)-V-1.0.1. Technical Specification (2015).

- [7] oneM2M. TS-0008-CoAP-Protocol-Binding-V-1.0.1. Technical Specification (2015).
- [8] oneM2M. TS-0010-MQTT-Protocol-Binding-V-1.0.1. Technical Specification (2015).
- [9] S. Bandyopadhyay, M. Sengupta, S. Maiti, S. Dutta, "Role of Middleware for Internet of Things: A Study," International Journal of Computer Science and Engineering Survey, Vol. 2, No. 3, pp. 94-105, 2011.
- [10] M.A. Chaqfeh, N. Mohamed, "Challenges in Middleware Solutions for the Internet of Things," Proceedings of the International Conference on Collaboration Technologies and Systems pp. 21-26, 2012.
- [11] M.A. Razzaque, M. Milojevic-Jevric, A. Palade, S. Clarke, "Middleware for Internet of Things: A Survey," IEEE Internet of Things Journal, Vol. 3, No. 1, pp. 70-95, 2016.
- [12] J. Yun, I.Y. Ahn, N.M. Sung, J. Kim, "A Device Software Platform for Consumer Electronics Based on the Internet of Things," IEEE Transactions on Consumer Electronics, Vol. 51, No. 4, pp. 564-571, 2015.
- [13] C. Perera, P.P. Jayaraman, A. Zaslavsky, D. Georgakopoulos, P. Christen, "MOSDEN: An Internet of Things Middleware for Resource Constrained Mobile Devices," Proceedings of IEEE International Conference on System Sciences, pp. 1053-1062, 2014.
- [14] K. Aberer, M. Hauswirth, A. Salehi, "Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks," Proceedings of the International Conference on Mobile Data Management, pp. 198-205, 2007.
- [15] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, A.T. Campbell, "A Survey of Mobile Phone Sensing," IEEE Communications Magazine, Vol. 48, No. 9, pp. 140-150, 2010.
- [16] N. Brouwers, K. Langendoen, "Pogo, a Middleware for Mobile Phone Sensing," Proceedings of the International Middleware Conference, pp. 21-40, 2012.
- [17] J. Yun, I.Y. Ahn, S.C. Choi, J. Kim, "TTEO (Things Talk to Each Other): Programming Smart Spaces Based on IoT Systems," Sensors, Vol. 16, No. 4 (467), pp. 1-21, 2016.
- [18] N.M. Sung, J. Kim, S.C. Choi, I.Y. Ahn, J. Yun, "IoT Devices With Self-Adaptive Software," Proceedings of IEMEK Fall Conference, pp. 360-362, 2015 (in Korean).
- [19] N.M. Sung, J. Kim, I.Y. Ahn, J. Yun, "Design of Self-adaptive Software Platform Module for Internet of Things Devices," Proceedings of KICS Winter Conference, pp. 827-828, 2015 (in Korean).
- [20] <http://www.iotocean.org>.

Nak-Myoung Sung (성낙명)



He received his B.S. and M.S. degrees in Electronic & Information Engineering from the Hankuk University of Foreign Studies, in 2010 and 2015 respectively.

He is currently a senior researcher with IoT Platform research center at Korea Electronics Technology Institute (KETI).

Email: nmsung@keti.re.kr

Jaeseok Yun (윤재석)



He received his B.S. in Electronics Engineering from Chonnam National University in 1997, and M.S. and Ph.D. degrees in Mechatronics from Gwangju Institute of

Science and Technology (GIST) in 1999 and 2006, respectively. He is now an assistant professor with the Department of Internet of Things at Soonchunhyang University.

Email: yun@sch.ac.kr