

사물인터넷 소프트웨어 플랫폼 기술동향

Internet of Things Software Platforms Technology Trends

홍상기 (S.G. Hong)	재난안전 IoT 연구실 선임연구원
이혜선 (H. Lee)	재난안전 IoT 연구실 연구원
최진철 (J.C. Choi)	재난안전 IoT 연구실 선임연구원
배명남 (M.N. Bae)	재난안전 IoT 연구실 책임연구원
이강복 (K.B. Lee)	재난안전 IoT 연구실 실장

* 이 원고는 2015년도 정부(미래창조과학부)의 재원으로 ETRI 연구개발사업의 지원을 받아 수행된 연구임[15ZC1510, Infraless 보행항법 기반 증강인지 커넥티드 헬멧 시스템 기술개발].

최근 사물인터넷 장치 증가와 관련 서비스의 급속한 확산, 기존의 수직적인 사물인터넷 서비스 구조로부터 수평적인 개방형 생태계로의 변화 추세에 따라 oneM2M, AllSeen, Open Interconnect Consortium(OIC)과 같은 표준협약체 및 산업체 컨소시엄을 중심으로 개방형 사물인터넷 플랫폼 규격 제정 및 오픈 소스 생태계 구축을 위한 노력이 지속되고 있다. 특히 다양한 사물인터넷 장치로부터 데이터를 수집하여 정보를 생성하고, 이종 장치 간의 정보 공유, 분산협업 및 실행제어 기술을 제공하는 사물인터넷 플랫폼은 사물인터넷 서비스 확산을 위한 핵심기술이라 할 수 있다. 본고에서는 오픈 생태계를 지향하는 AllSeen/AllJoyn, OIC/IoTivity, oneM2M 사물인터넷 소프트웨어 플랫폼의 표준화 동향을 살펴보고, 플랫폼의 핵심 기술에 대한 비교 분석을 통해 주요 기술개발 동향을 살펴보고자 한다.

ICT 융합기술 R&D 동향 특집

- I. 머리말
- II. 사물인터넷 소프트웨어 플랫폼 개요
- III. 핵심 공통기술 비교분석
- IV. 플랫폼 상호연동 기술
- V. 맺음말

I. 머리말

사물인터넷 기술은 다양한 장치를 인터넷에 연결하고 장치에 장착된 센서로부터 데이터를 획득, 처리, 전송하여 사용자에게 스마트 홈, 헬스케어, 스마트 카, 스마트 팩토리 등의 서비스를 제공하는 기술이다. 이러한 지능형 서비스 제공을 목표로 하는 글로벌 사물인터넷 시장은 현재 약 200조원 이상의 잠재적 가치를 지니고 있으며, 2020년까지 인터넷에 연결되는 장치의 수는 기하급수적으로 늘어 260억 개에 이를 것으로 예측되고 있다[1].

사물인터넷에 접속되는 장치의 급속한 증가에 따라 라즈베리파이나 인텔 에디슨과 같은 게이트웨이급 하드웨어 플랫폼, 다양한 사물을 구성하여 서비스를 제공하는 경량형 하드웨어 플랫폼인 ARM mBed, Arduino 등의 사용도 점차 증가하고 있는 추세이다[2].

이러한 사물인터넷 하드웨어 플랫폼 기술의 확산 및 성장뿐 아니라, 사물인터넷의 생태계 구축 및 활성화를 위해서는 하드웨어 벤더, 무선통신 업체, 소프트웨어 전문 업체, 이동 통신사 등을 포괄하는 사물인터넷 협의체를 중심으로 사물인터넷 생태계 확장을 위한 단말 플랫폼 기술개발이 필요하다. 특히 수평적 상호운용성을 보장하는 표준 기반의 개방형 사물인터넷 소프트웨어 플랫폼 기술개발이 필수적이라 할 수 있다.

본 문서는 AllSeen, Open Interconnect Consortium (OIC), oneM2M에서 진행되고 있는 주요 사물인터넷 플랫폼에 대한 규격 및 기술을 살펴보고, 사물인터넷 소프트웨어 플랫폼이 공통적으로 제공하는 주요 핵심 공통 기술을 정의하고, 개별 플랫폼이 지원하는 공통 기술을 비교분석한다. 마지막으로 협의체별로 진행되고 있는 사물인터넷 플랫폼 간의 상호연동 기술동향에 대하여 살펴보고자 한다.

II. 사물인터넷 소프트웨어 플랫폼 개요

다양한 사물인터넷 장치에 포함된 이중 센서와 디바

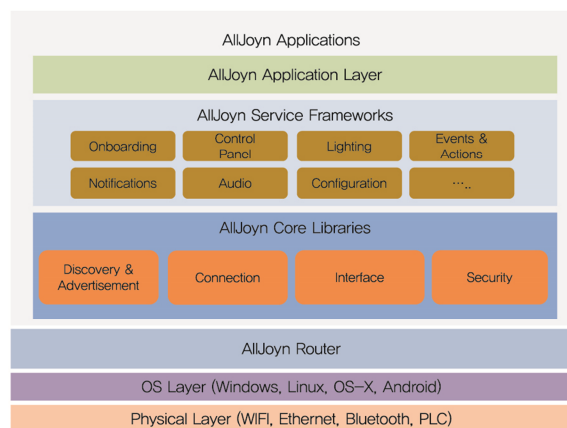
이으로부터 데이터를 수집·처리하여 정보를 생성하고, 이중 장치 상호 간에 전달된 정보를 공유하여 융합 처리하는 분산협업 및 실행제어 기술은 지능형 사물인터넷 서비스 개발에 있어서 필수적이라 할 수 있다.

여기서는 이러한 핵심 기능의 제공을 목표로 하여 현재 사물인터넷 단말 관점에서 고려되고 있는 대표적인 공개 플랫폼인 AllJoyn[3]과 OIC/IoTivity[4], oneM2M[5] 기술에 대해 살펴보고자 한다.

1. AllJoyn

AllJoyn은 Qualcomm 산하 Qualcomm Innovation Center에 의해 시작되었으며, AllSeen 얼라이언스를 중심으로 오픈 소스화하여 개발자 누구나 이 기술을 이용할 수 있도록 하고 있다. AllJoyn은 Ad-hoc 기반 Peer-to-Peer(P2P)기술로 별도의 중계서버 없이도 피어 장치들 간의 자율적인 상호연결을 통해 네트워크를 구성하고 장치들 사이에 분산된 정보를 공유할 수 있다. 또한, AllJoyn은 RTOS, Linux, Android, iOS, Windows 등과 같은 다양한 OS에 적용이 가능하며, CPU, 메모리, 대역폭 등의 리소스가 제한적인 장치도 수용하고 있다.

AllJoyn은 Bluetooth나 Wi-Fi 장치를 제어 및 작동시킬 수 있는 OS계층 위에 소프트웨어 프레임워크 형태로



(그림 1) AllJoyn 소프트웨어 프레임워크

존재한다. 따라서 개발자가 운영체제의 커널이나 부품의 제어에 관한 부분을 신경 쓰지 않아도 되는 장점을 가지며 Bluetooth나 Wi-Fi 기반 위의 기술이어서 모바일 기기에 적합하다. 현재 C, C++, object C, Java 등과 같이 다양한 언어를 지원하는 API와 SDK의 소스코드를 공개하여 Android, iOS 기반의 스마트 단말과 더불어 다양한 임베디드 장치에 탑재될 수 있다.

AllJoyn 소프트웨어 프레임워크는 (그림 1)과 같은 구조를 지니며 주요 특징은 다음과 같다.

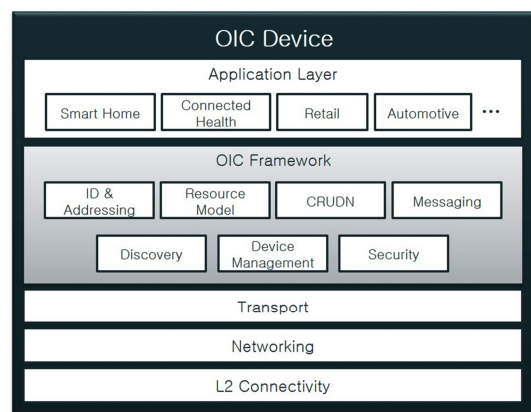
- D-Bus 기능을 확장하였으며, 장치(단말)의 이동성/근접성 관리와 응용 간에 자유로운 분산 연결성을 제공한다.
- AllJoyn은 근거리 기반의 기기 간 P2P기술로 중계서버 없이 장치 간 직접 통신을 제공하는 프레임워크
- Windows, Linux, Android, iOS 등 다양한 환경에서 AllJoyn에 기반을 둔 응용을 개발할 수 있음.
- AllJoyn은 응용서비스 개발자가 네트워크 계층에서 어떤 통신 기술을 사용하는지 고려하지 않아도 됨. 개념적으로는 AllJoyn 프레임워크가 Wi-Fi 혹은 Bluetooth를 상황에 맞게 선택하나 현실적으로는 동일 Wi-Fi 내에 접속 중인 장치 간에 통신을 지원함.
- AllJoyn 장치들은 세션 연결을 통해 Remote Method Invocation(RMI)방식으로 데이터를 교환할 수 있으며, annotation을 추가한 인터페이스 파일을 서로 공유하고, 프록시를 통해 해당 인터페이스의 메소드를 호출하는 방식을 사용함. 소켓 프로그래밍처럼 주고받을 데이터의 순서와 타입에 대해 약속하고 예외처리를 할 필요 없이 메소드를 정의하고 호출하는 방식으로 동작
- AllJoyn 세션 연결 없이도, session-less signal을 통해 통신이 가능한 장치에게 메시지를 브로드캐스트 할 수 있음.

- AllJoyn router라는 데몬을 통하여 다른 장치와 통신이 가능하고, router가 설치된 장치가 같은 네트워크 안에 연결되면, router들이 서로의 존재를 인지하고 정보를 교환함. 교환하는 정보는 자신에게 연결된 서비스의 종류, 자신에게 연결된 클라이언트와 해당 클라이언트가 요청하는 서비스 등이 있음. 개념적으로는 같은 네트워크 안에 연결된 장치가 하나로 연결되어 가상의 AllJoyn 버스가 만들어지는 형태
- 보안을 위해 AES128 암호화 알고리즘과 PSK, ECDSA 등의 인증 알고리즘을 지원함.

2. OIC/IoTivity

OIC는 사물인터넷시대를 대비하여 Internet of Things(IoT) 네트워크상에서 이종 OS 및 플랫폼을 사용하는 경량형 장치를 연결하는 데 필요한 요구사항을 정의하고 장치 간 상호운용성을 보장하는 것을 목적으로 기업들을 선도하기 위하여 설립된 조직으로, 2014년 7월 삼성, 인텔 등이 공동으로 설립하였다[6][7].

현재 프레임워크의 기능과 규격을 정의한 core specification과 다양한 시장 요소에 IoT 서비스를 제공하기 위한 프로파일을 포함하는 vertical profiles specification으로 구분하여 OIC 규격을 개발하는 중이다. Core specification은 프로젝트 A와 B 문서로 나누



(그림 2) OIC core 프레임워크

어지는데, 프로젝트 A는 core resource, Constrained Environment Application Protocol(CoAP) 기반 end-point discovery, resource 기반 탐색, CRUDN 메시지, notification 기능을 정의하고 있다. 프로젝트 B는 그룹 관리, IP 카메라 스트리밍, 장치관리, multi-PHY easy-setup, 원격 제어 기능의 규격을 포함한다. 2015년 4월 10일 프로젝트 A의 0.9 버전이 OIC 내부에서 발표되었고 현재 보완 중이며, 곧 1.0 버전을 확정하여 공개 발표할 예정이다.

0.9 버전의 프로젝트 A에 기술된 OIC 아키텍처의 주요 개념을 설명하면 다음과 같다.

- 실제 세계의 개체(entity) 및 개념이 OIC 리소스로 표현됨. 리소스는 고유의 식별 ID를 갖고, Representational State Transfer(RESTful) 기반의 명령을 수행하기 위한 인터페이스를 제공한다. 이러한 정보는 OIC 리소스 모델에 정의됨.
- OIC 아키텍처는 RESTful 구조 모델을 바탕으로 하고 있음. RESTful 구조 모델은 서버-클라이언트 구조로, 서버는 OIC 리소스를 갖고 있으며 클라이언트가 서버에게 리소스의 상태조작 요청을 보내면 서버가 응답으로 리소스의 상태 즉, representation을 응답하게 됨. 클라이언트가 서버로 보내는 리소스 상태조작 요청은 create, retrieve, update, delete, notify을 포함하는 CRUDN 명령으로 전달됨. RESTful 구조 모델에서 서버/클라이언트 간 통신은 stateless 통신
- OIC 아키텍처는 두 가지 추상화 방법을 제공. 첫째, entity handler를 제공하여 비 OIC 장치를 발견하고 이에 대응되는 OIC 리소스를 생성/맵핑 할 수 있도록 함. 둘째, connectivity abstraction primitives를 제공하여 논리적인 RESTful 명령어를 데이터 통신 프로토콜에 맵핑할 수 있도록 함. 현재 CoAP과 HTTP를 지원하고 있음.

〈표 1〉 OIC core 프레임워크 계층별 기능

계층	기능
Application Profiles	- 서비스 도메인에 따라 특화된 데이터 모델 및 공통기능 제공
OIC Framework	- OIC core specification에 명시된 기능 제공
Transport	- QoS 제약에 적합한 end-to-end flow 통신 제공
Networking	- 네트워크에서 OIC 장치 간 데이터 교환 기능 제공
L2 Connectivity	- 물리적 링크계층(예, Wi-Fi, Bluetooth)에 대한 연결 기능 제공

이와 같은 주요 개념을 구현하기 위해 제안된 OIC core framework는 (그림 2)의 계층으로 구성되어 있으며 계층별 기능은 〈표 1〉과 같다.

IoTivity 프로젝트는 OIC specification을 구현하는 오픈 소스 프로젝트로, OIC에서 지원하고 Linux foundation에서 후원하고 있다. IoTivity 프레임워크는 모든 운영체제와 연결 플랫폼의 미들웨어로 동작하며 여러 프로그래밍 언어와 OS를 지원한다. IoTivity는 현재 0.9.1 버전이 공개되어 있다. 0.9.1 버전은 장치 발견 및 연결, 자원 관리, 소프트 센서 매니저, 프로토콜 플러그인 매니저, 객체 매니저(things manager), 네트워크 연결 추상화 등의 기능을 제공하며, 현재 기능이 추가 보완되고 있는 중이다.

3. oneM2M

oneM2M은 2012년 7월, 상호운용성을 보장하는 M2M/IoT 서비스 표준 플랫폼 개발을 위해 설립되었으며, 수직적 서비스 구조를 탈피하여 수평적 서비스 플랫폼을 지향하고 있다. 현재 이동통신사, 벤더, 솔루션 업체 등 200여 업체가 가입하여 표준화를 진행하고 있으며 주요 표준화 활동은 다음과 같다.

- 공통서비스 기능 제공을 위한 사용자 시나리오 및 핵심 요구사항을 정의

〈표 2〉 oneM2M 기술 주요 특징

특징	내용
리소스 기반 아키텍처	- 모든 정보 및 기능 등을 리소스(resource)로 정의하고 RESTful API 로 접근할 수 있는 아키텍처 제공
분산 서비스 플랫폼	- 중앙 서버뿐 아니라 플랫폼이 보유한 개별 장치 및 게이트웨이 등 분산된 시스템에서 서비스 제공
접속망 연동	- 3GPP 네트워크 연동을 통한 장치의 위치 획득 및 트리거링 서비스 등을 제공하며 기타 관련 기능 추가로 정의 중
장치관리 기능	- OMA DM, LWM2M 및 BBF TR-069 와 연동하여 장치관리 기능 제공

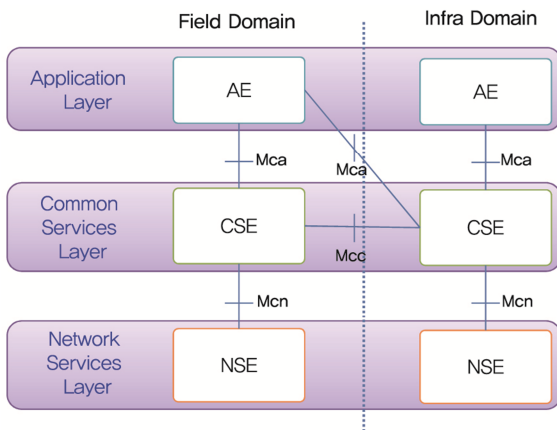
* Lightweight M2M(LWM2M)

* Open Mobile Alliance(OMA)

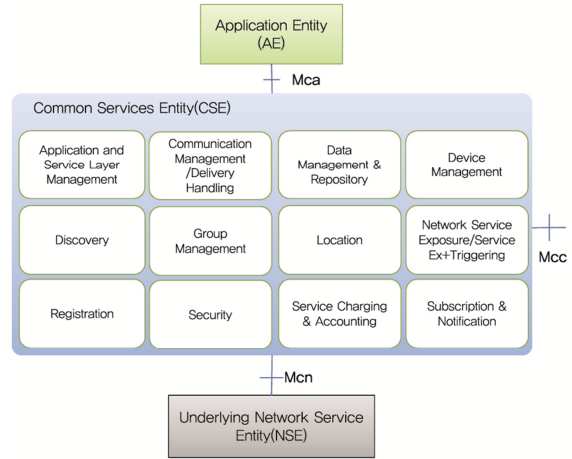
- 도출된 요구사항으로부터 공통 서비스 기능을 도출
- oneM2M entity 및 reference point를 정의
- 다양한 이종 네트워크 연동을 위해 프로토콜 바인딩 규격을 정의(HTTP, CoAP, Message Queue Telemetry Transport: MQTT 등)
- 기기 간 인증, 기밀성, 무결성 및 접근제어 기능 제공을 논의
- M2M장치관리 및 데이터 추상화, 시멘틱 기술개발에 대한 규격화를 포함

oneM2M 기술의 주요 특징은 〈표 2〉와 같다.

oneM2M 기술은 (그림 3)과 같이 end-to-end M2M



(그림 3) oneM2M 계층 모델



(그림 4) oneM2M 공통 서비스

서비스를 위한 3계층 모델, 즉 응용 계층(application layer), 공통 서비스 계층(common services layer), 네트워크 서비스 계층(network services layer)로 이루어진다.

oneM2M이 제공하는 서비스의 핵심이 되는 계층은 공통 서비스 계층[그림 4 참조]으로 M2M 장치와 서비스 응용 사이의 원격 권한 설정 및 활성화, 보안, 네트워크 연결성, 데이터 저장 및 처리, 위치획득, 장치관리 등의 공통 서비스 제공을 목적으로 하는 미들웨어로 정의될 수 있다.

III. 핵심 공통기술 비교분석

본 장에서는 사물인터넷 소프트웨어 플랫폼에서 요구되는 주요 핵심 공통 기술을 분류하고 개별 플랫폼이 지향하는 기술별 주요 특징을 비교 분석한다.

1. 서비스 개체 및 장치관리

서비스 개체 및 장치관리는 응용서비스의 설치, 구동, 정지, 해제 등을 위한 응용제어 및 메타데이터 관리, 연결되는 장치에 대한 제어, 진단 등을 위한 장치관리 그리고, 서비스 개체로부터의 데이터 수집, 저장, 검색 등을 서비스 개체 데이터 관리 기능으로 정의될 수 있다.

가. AllJoyn

AllJoyn은 서비스 개체 관리를 위해 D-Bus 타입 시스템과 프로토콜을 활용 또는 확장하여 사용한다. 특히 D-Bus의 object model, interface, system bus를 이용하여 서비스 개체에 대한 관리 및 제어를 수행한다.

나. OIC/IoTivity

OIC/IoTivity는 장치관리 권한설정, 모니터링, 진단 및 유지보수 기능을 제공한다. 객체 매니저는 IoTivity에서 제공하는 서비스 중의 하나로, 네트워크의 객체(things) 그룹을 관리하는 데 유용한 기능을 제공한다. 객체 매니저가 제공하는 API를 사용하여 다수의 객체로 구성되는 그룹을 설정하고 진단할 수 있다.

다. oneM2M

oneM2M 플랫폼은 서비스 개체관리를 위해 응용서비스 계층 관리 데이터 저장관리 기능을 제공한다. 응용서비스 계층관리는 서비스 개체들의 업그레이드, troubleshooting 그리고 설정을 위한 기능과 Application Entity(AE)에서 실행되는 서비스들에 대한 생명주기 관리 및 상태관리 기능을 제공한다. 또한, OMA DM, Broadband Forum(BBF) TR-069, LWM2M 등의 장치 관리 기술들과의 연동을 통하여 장치 설정관리, 장치 진단 및 모니터링, 펌웨어 업데이트 그리고 디바이스 토폴로지 관리 기능 등을 제공할 수 있다.

2. 연결 및 전달제어

연결 및 전달제어 기능은 플랫폼을 기반으로 하는 다양한 사물인터넷 응용서비스 및 디바이스 간의 분산처리 및 서비스의 구독 통지 및 그룹관리 기능을 포함한다.

가. AllJoyn

AllJoyn은 분산버스 시스템을 기반으로 하여 원격 서

비스개체에 대한 연결 및 제어를 수행하게 된다. 이것은 대개 Remote Procedure Call(RPC) 또는 객체지향 개념으로 RMI, 원격 호출을 통해 이루어진다.

나. OIC/IoTivity

OIC/IoTivity는 장치의 정보변경에 대한 구독 및 통지 기능을 제공한다. 클라이언트는 서버에게 retrieve 명령을 사용하여, 서비스 구독이 가능한 리소스의 상태가 변경될 경우 알림을 받을 것을 요청할 수 있다. 연결 및 전달제어를 위한 서버 간 분산, 통신관리 및 전달제어는 향후 확장할 수 있다.

다. oneM2M

oneM2M에서는 다른 Common Services Entity(CSE), AE, Network Services Entity(NSE)들과의 데이터 통신을 관리하기 위해서 Communication Message Delivery Handling(CMDH) 기능을 제공한다. CMDH는 언제, 어떤 방법으로 데이터를 전송할 것인가에 대한 CMDH 정책을 기반으로 데이터 전송을 관리 제어를 수행한다. CMDH 정책에 대한 파라미터는 QoS제어, 데이터 통신 경로 설정, 네트워크 활용 가능상태인지, store-and-forward 처리를 위한 버퍼관리 기능 등을 포함한다. 그리고 서비스 구독 기능은 특정한 리소스의 변화에 대한 이벤트를 사전에 구독하고, 구독한 리소스에 대한 변화가 발생할 때, 이벤트 정보를 보고자 하는 AE 또는 다른 CSE에게 변화된 정보를 알려주는 서비스 구독 통지기능을 지원한다.

3. 서비스 공개 및 검색

서비스 공개 및 검색은 사물인터넷 서비스를 제공하기 위한 서비스 제공자의 서비스 공개(advertisement), 탐색(discovery) 및 검색(search) 등의 기능을 포함한다.

가. AllJoyn

AllJoyn 프레임워크는 서비스 개체에 대한 서비스 공

개와 AllJoyn 서비스를 탐색할 수 있는 기능을 제공한다. AllJoyn 검색 프로토콜은 AllJoyn 네트워크에 들어 오거나 나가는 서비스의 동적 특성을 관리하고, 알림을 고지할 수도 있다. AllJoyn 프레임워크는 서비스 공개 및 탐색을 위해 IP 멀티캐스트를 기반으로 name-based discovery와 announcement-based discovery 두 가지를 지원하며 14.06 버전 이후는 성능 개선을 위하여 Multicast DNS(mDNS)기반의 프로토콜을 지원하는 Next Generation Name Service(NGNS)가 적용되었다.

나. OIC/IoTivity

OIC/IoTivity는 OIC 서비스개체에 대한 검색을 위해 엔드포인트(endpoint) 탐색과 리소스 기반 탐색, 두 가지의 서비스 탐색 기능을 제공한다. 엔드포인트 탐색은 CoAP, HTTP를 기반으로 하여 OIC 엔드포인트가 네트워크에 있는 다른 OIC 엔드포인트를 찾는 방법이고, 리소스 기반 탐색은 클라이언트가 다른 OIC 피어에 공개한 정보(예, 리소스 인스턴스, 리소스 타입)를 찾는 기능이다. 리소스기반 탐색에는 direct 탐색, indirect 탐색, advertisement 탐색이 있다.

다. oneM2M

oneM2M은 CSE의 discovery 기능을 통하여 리소스와 속성이 포함하고 있는 서비스와 응용에 대한 정보를 검색할 수 있다. Discovery 기능은 사전에 정의된 리소스의 필터링 정보(creation time, matching string, upper limit, ordering parameter 등)를 이용하여 다른 CSE에게 서비스 검색결과를 제공한다.

4. 이종네트워크 관리

이종네트워크 관리기능은 다양한 네트워크 프로토콜을 지원하여 장치 간의 데이터 교환을 위한 네트워크 추상화 및 이종네트워크 바인딩과 주변의 근거리 네트워

크의 접근제어를 위한 인터워킹을 포함한다.

가. AllJoyn

AllJoyn은 다양한 네트워크 개체 사이에서 메시지의 실제 이동을 관리하기 위해 TCP 또는 UDP와 같은 전송기법을 사용한다. AllJoyn 전송계층은 데이터 시퀀스의 변화, 캡슐화 및 공개 및 발견 설정 등을 담당하기에 별도의 기본 전송 메커니즘을 사용한다.

AllJoyn 라우터는 서비스 검색과 데이터 전송을 위해 다수의 전송방식(유무선 IP 통신, Bluetooth, local 소켓 등)을 지원하며 각각 지원되는 통신 프로토콜을 위한 추상화를 지원한다.

나. OIC/IoTivity

OIC 프로토콜은 HTTP/CoAP기반의 REST-like 인터페이스이지만 이들보다 높은 추상화 레벨을 갖고 있어서 Bluetooth 클래식, Bluetooth Low Energy(BLE), ZigBee 등의 다중 네트워크 통신이 향후 추가될 수 있다.

OIC는 맵핑 primitives를 제공하여 논리적인 CRUDN 명령을 데이터 통신 프로토콜/기술에 맵핑할 수 있도록 하며, 프로토콜 플러그인 매니저(예, MQTT 플러그인 매니저)를 통해 통신 프로토콜 독립성을 지원한다. 이를 통해 MQTT와 같은 non-OIC 프로토콜을 사용할 수 있다.

다. oneM2M

oneM2M표준에서 이종네트워크 관리 기능은 크게 non-M2M 데이터모델을 oneM2M에 특화된 데이터 모델 즉, 현재의 ZigBee, Z-wave, BACnet, ANSI C12, mBus의 표준을 이용하여 근거리 네트워크의 장치를 oneM2M 표준 데이터 모델로 변환하는 인터워킹 프록시(interworking proxy) 기능과, 타 CSE와의 접근을 위한 통신 프로토콜 바인딩 기술로 나누어질 수 있다.

oneM2M에서는 oneM2M의 primitive를 HTTP, CoAP, MQTT와 같은 전송계층 프로토콜에 맵핑하기

위한 규격을 제시하고 있다

5. 상호등록 및 보안

상호등록 및 보안은 사물인터넷의 중요한 이슈인 보안을 포함하는 기능으로 응용서비스 및 장치 간의 상호등록 및 인증 그리고 연결 및 접근제어 등을 포함한다.

가. AllJoyn

AllJoyn은 서비스 공개와 발견을 통해 응용 서비스를 알리거나 필요한 응용을 찾는다. 필요시 응용 간에 세션을 맺어 RMI통신을 수행하는 형태로 통신하므로, 장치를 등록하거나 등록정보를 관리하지는 않는다.

AllJoyn 시스템은 디바이스 상호 간의 인증, 암호화된 데이터 교환을 위해 응용레벨에서의 end-to-end 보안 프레임워크를 제공한다. 보안 프레임워크는 D-Bus 시스템에서 정의된 Simple Authentication and Security Layer(SASL) 프로토콜을 기반으로 한다.

나. OIC/IoTivity

OIC/IoTivity는 보안 리소스를 위한 기능과, 접근제어를 위한 보안권한 설정, Datagram Transport Layer Security(DTLS) 기반의 session protection 기능 등을 포함하는 보안기능을 제공한다.

다. oneM2M

oneM2M은 CSE에서 제공하는 등록기능을 통하여 응용 및 장치의 상호등록을 수행하고, 다른 CSE 간의 상호등록 기능도 제공한다. 또한, oneM2M은 다른 서비스 노드와 데이터 송수신을 위해 보안기능을 제공한다. oneM2M이 제공하는 보안기능은 access management, security association establishment, identity protection, security administration, sensitive data handling 등이 있다.

6. 기타

앞서 살펴본 핵심 공통기술 이외에도 사용자와 개발자의 편의성을 고려한 오픈 소스 활용성, 장치 및 OS 독립성, 개발환경의 용이성, 사용자 커뮤니티의 활성화 등 운영환경도 사물인터넷 플랫폼의 활성화를 위한 중요한 기술적 고려사항이다.

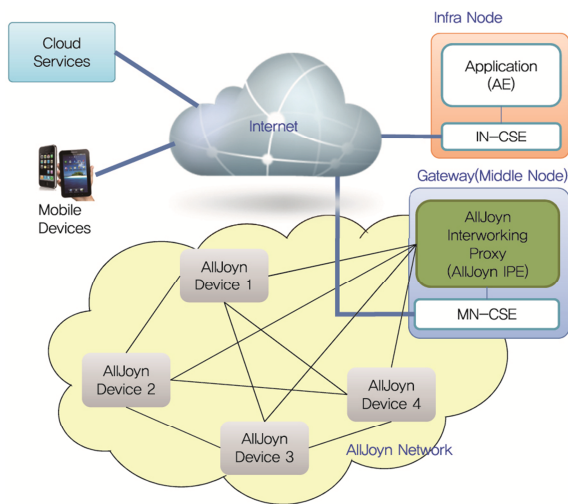
IV. 플랫폼 상호연동 기술

현재의 사물인터넷 플랫폼 기술은 개별 협의체를 중심으로 응용도메인별 공통 플랫폼을 위한 핵심 공통 기능 및 프레임워크정의, 장치관리, 네트워크 관리, 보안 등의 표준기술을 지향하고 있다. 또한, 상호운용성 및 적합성 검증을 목표로 하고 있으며, 다른 산업체 표준 및 네트워크와의 상호연동에 대한 협의를 진행하고 있다.

oneM2M에서는 2014년 9월부터 근접 로컬영역에서의 RPC 기반 사물인터넷플랫폼인 AllJoyn과 광역네트워크에서의 REST 기반 oneM2M 플랫폼 간의 상호연동을 통한 시너지 창출을 목적으로 상호연동 표준을 개발하고 있으며, 논의되는 표준의 주요범위는 다음과 같다 [8].

- oneM2M과 AllJoyn 플랫폼 상호연동 시나리오를 정의하고, 새로운 표준 요구사항의 도출을 목적으로 함.
- 새롭게 도출된 요구사항을 만족시키기 위한 기능구조에 대한 설정, 리소스 맵핑 및 상호연동 절차 등을 포함하는 oneM2M 구조를 제안
- 현재의 oneM2M 기능 규격에 수정 보완이 필요한 부분에 대해 정의하고 새롭게 정의되어야 할 규격에 대한 검토를 포함

oneM2M과 AllJoyn이 middle node 즉 게이트웨이에 올라가는 인터워킹 기술을 이용하여 상호연동을 지향하는 것처럼[그림 5] 참조], oneM2M과 OIC/IoTivity,



(그림 5) oneM2M/AllJoyn 상호연동

AllJoyn과 OIC/IoTivity도 게이트웨이의 인터워킹을 활용하여 프로토콜 변환, 데이터 모델 변환 등의 과정을 통해 상호연동이 가능할 것이다.

V. 맺음말

지금까지 표준단체, 산업협의체를 통한 표준 기술개발동향 및 오픈 소스의 형태로 제공되고 있는 주요 사물인터넷 소프트웨어 플랫폼들에 대해 살펴보았다.

AllJoyn은 다수 장치에 분산된 응용 간의 RPC 통신을 제공하는 소프트웨어 프레임워크이다. 경량형이면서 낮은 의존성을 지향하는 D-Bus를 기반으로 확장된 사물인터넷 플랫폼이고, 현재 다양한 언어와 운영체제, 보안 프로토콜을 지원하고 있다.

oneM2M은 Resource Oriented Architecture(ROA)를 기반으로 중앙 서버뿐 아니라 플랫폼이 보유한 개별 장치 및 게이트웨이의 분산 시스템 구성을 통해 서비스를 제공한다. 이를 통해, 분산 구조의 정보 저장 관리, 제약이 많은 사물 및 로컬 네트워크 환경 성능을 고려한 데이터 처리/운용 및 관리 지역화, 데이터 처리 부하 분산 등의 전통적인 자원제약 문제 영역을 해소할 뿐만 아니라 인터넷 연결을 통해 특정 사물에 대한 네트워크적

인 접근을 가능하게 하고 있다.

OIC/IoTivity는 경량형 IoT 도메인에 적합한 플랫폼으로 스마트 홈 도메인 적용을 중심으로 확장 중이다. 현재 HTTP/CoAP 기반 통신프로토콜을 지원하고 있고, OIC/IoTivity의 실질적인 적용을 위해 지속적인 표준 확립이 진행되고 있다.

지금까지 살펴본 주요 사물인터넷 플랫폼 중에서 현재 어떤 플랫폼도 아직 시장에서 독보적인 주도권을 장악하지 못하고 있으며 각 표준이 지향하는 서비스 도메인 및 사용자 시나리오에 특화되어 계속 발전하고 있는 상황이다. 이런 현실을 반영하여, 향후에는 oneM2M과 같은 글로벌 표준 플랫폼과 AllSeen 및 OIC와 같은 산업체 컨소시엄들이 지향하는 플랫폼 사이의 플랫폼 상호연동 노력이 지속될 것으로 보인다. 또한, 상호 협력을 목적으로 국제표준의 상호연동을 지향하는 사물인터넷 플랫폼 규격 및 기술개발이 이뤄질 것이며, 이러한 플랫폼 연동 기술을 기반으로 다양한 사물인터넷 서비스가 확산될 것이다.

약어 정리

AE	Application Entity
BBF	Broadband Forum
BLE	Bluetooth Low Energy
CMDH	Communication Message Delivery Handling
CoAP	Constrained Environment Application Protocol
CSE	Common Service Entity
DTLS	Datagram Transport Layer Security
IoT	Internet of Things
LWM2M	Lightweight M2M
M2M	Machine To Machine
mDNS	Multicast DNS
MQTT	Message Queue Telemetry Transport
NGNS	Next Generation Name Service
NSE	Network Service Entity

OIC	Open Interconnect Consortium
OMA	Open Mobile Alliance
P2P	Peer-to-Peer
REST	Representational State Transfer
RMI	Remote Method Invocation
ROA	Resource Oriented Architecture
RPC	Remote Procedure Call
SASL	Simple Authentication and Security Layer

참고문헌

- [1] 파이낸셜뉴스, “삼성·LG ‘IoT 표준규격’ 동맹 맺고 글로벌 시장 주도한다,” 2014. 11. 13.
- [2] 김선태 외, “IoT 단말 플랫폼동향 및 생태계 구축,” 전자통신동향분석, 제29권 제4호, 2014. 8.
- [3] AllSeen alliance, “Introduction to the AllJoyn Framework,” Dec. 10th, 2013.
- [4] OIC, “OIC Core Specification - PROJECT A, Part 1, V0.9,” Apr. 10th, 2015.
- [5] oneM2M-TS-0001-V-1.6.1, “oneM2M Functional Architecture,” Jan. 30th, 2015.
- [6] OIC, “OIC Security Specification - PROJECT A, V0.96,” Apr. 18th, 2015.
- [7] IoTivity, <https://www.iotivity.org/documentation>
- [8] oneM2M-TR-0014-V-0.1.0, “oneM2M and AllJoyn Interworking,” Dec. 5th, 2014.