
**에너지 IoT 기반 집합 건물 환경에서
효율적인 수요반응을 위한
경량 에너지관리 에이전트 프로토콜 설계 및 구현**

■ 논문 요약

석사 학위 논문(2018.08.17)

에너지 IoT 기반 집합 건물 환경에서 효율적인 수요반응을 위한
경량 에너지관리 에이전트(EMA) 프로토콜 설계 및 구현

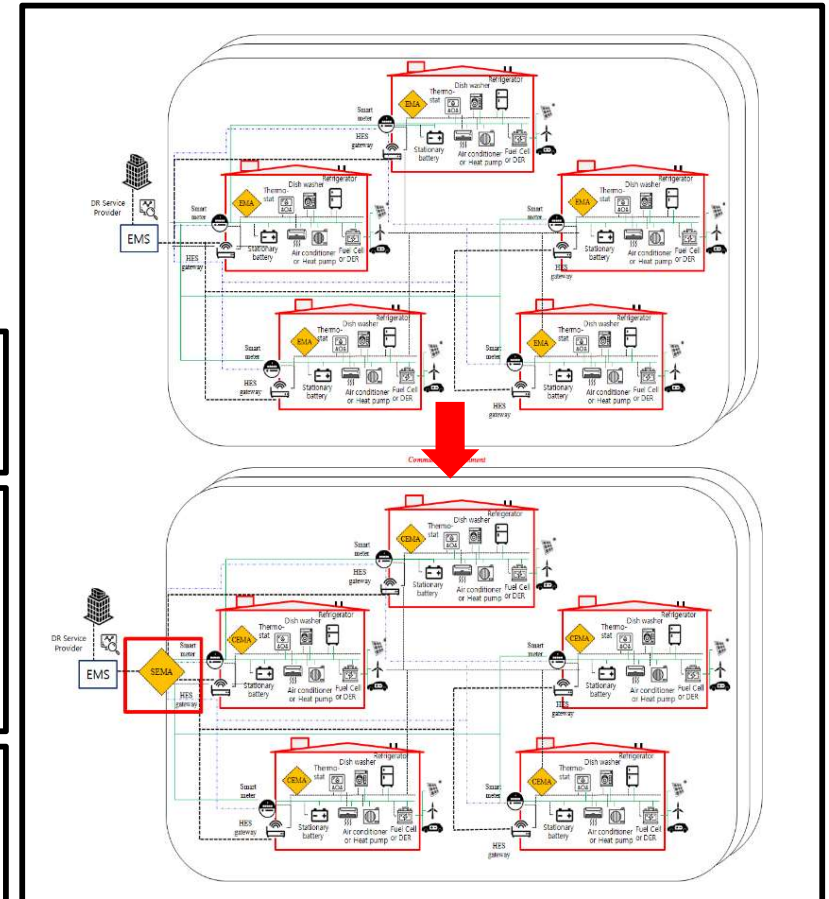
목적: (1)집합 건물 환경으로 수요반응을 하는 목적 (수요반응 영역 확대, 계층 분산화)
(2)에너지 관리 에이전트 프로토콜(EMAP)의 목적

구현: (1)경량 수요반응 프로토콜 기반 OpenADR 2.0b와
에너지관리 에이전트 프로토콜의 4가지 수요반응 서비스 구현
(2)서버 에너지관리 에이전트(Aggregator)의 기능적 중계 기능 구현
(3)에너지관리 에이전트의 Explicit 모니터링 문제점
-Implicit/Explicit 모니터링 통신 기능 구현

실험: (1)경량 수요반응 프로토콜의 데이터 트래픽 및 지연 시간 분석 : 경량 프로토콜 우수
(2)집합 건물 내 이벤트 수요반응 전송 메커니즘 비교 실험 : CoAP 우수
(3)에너지관리 에이전트의 Implicit/Explicit 통신 비교 실험 : 병목현상 ↓, 보안성 ↑

결론

(1)에너지관리 에이전트를 통해 집합 건물 환경에서도 지능화된 에너지관리인 수요반응을 함
(2)에너지 관리 에이전트 프로토콜을 통해 수많은 집합건물의 수요관리 고객에 대한 실시간 수요반응을 할 수 있음(Push 실험을 통해 증명)
(3)집합 건물 환경에서 에너지관리 에이전트 간 통신을 통해 수요반응 및 통합적 스케줄링을 할 수 있음
(4)집합 건물 환경에서 EMA 프로토콜을 경량 IoT 프로토콜인 CoAP, MQTT 이용
-Heavyweight인 HTTP/XML 방식의 수요반응 프로토콜은 초소형 에너지 기기에 제한이 있음(OpenADR2.0b, SEP2.0)
-디바이스나 IoT기기, Gateway에는 제한된 리소스 및 배터리 용량이 제한, IoT(CoAP, MQTT) 프로토콜이 lightweight, 저전력 기반
(5)제한적인 IoT환경에서 데이터 트래픽, 수요반응 전송 속도를 줄이기 위해서 IoT 프로토콜의 메커니즘 비교를 통해
적합한 프로토콜인 CoAP/JSON 이용- (이유 : 집합건물의 수많은 개인지역으로 확대(수많은 노드), 효율적 수요반응 관리가 필요)
-1. CoAP, MQTT 기반 lightweight, 2. CoAP Observe, 3. MQTT Broker 기반 Push 메커니즘, 4. MQTT 기반 멀티캐스트
(6)기존연구의 Explicit모니터링과 에너지관리 에이전트 프로토콜의 Implicit/Explicit 모니터링 비교를 통해 데이터 추상화기능의 필요성을 증명



■ 논문 상세 내용(연구 배경 및 필요성(목적))

(1) 집합건물 환경의 에너지 관리의 필요성 : 블랙아웃(정전), 전력 수요 증가, 에너지 수요관리 대책 강구

집합 건물 환경의 지능화된 에너지 관리(수요반응)가 필요(주거공간의 에너지관리에이전트 집중화 필요)

-여름철이나 겨울철 냉난방기기 판매량과 사용량 급증에 따른 블랙아웃(대규모 정전)은 심각한 문제를 발생

-집합 건물과 일반 가정 사용량이 많게는 피크 시간대 전력수요의 60%를 차지함

-수요반응, OpenADR(Open Automated Demand Response)프로토콜 등장

(2) 국민 DR를 통한 수요반응 영역 확대 및 에너지 IoT, 경량수요반응 프로토콜의 등장

(경량 OpenADR2.0b CoAP/JSON, MQTT/JSON)

-제한적인 IoT환경, 디바이스나 IoT기기, 게이트웨이 ->배터리와 제한된 리소스

-에너지관리에이전트 기술 ->에너지 IoT 기술 발전(에너지 분야와 IoT 분야와 융합)

-Utility 회사 중심의 송배전 산업설비에서 에너지 관리 플랫폼인 에너지관리시스템(EMS)은 효율적 에너지 관리하기 위해 사용

(3) 에너지관리 에이전트 표준화 및 EMA Protocol 표준화가 진행(ISO/IEC 15067-3, ISO/IEC 15067-3-3)

-주거지역, 집합건물, 대학캠퍼스, 빌딩 등에 Interacting EMA를 통해 에너지관리와 수요반응에 대한 표준화가 등장(다양한 구조, 계층화, 분산화)

(ISO/IEC 15067-3-3의 Model of a system of interacting Energy Management Agents)

-상호 작용하는 EMA는 주거지역, 집합건물, 대학캠퍼스, 빌딩 의 지역 사회의 에너지관리와 수요반응을 관리함.

(4) 집합건물 내에 수 많은 수요관리 고객을 수용할 수 있는 수요반응 프로토콜(EMAP)이 필요함

-수요관리사업자가 모두 수요 관리해주기 위해서는 복잡성 증가(병목현상이나 서비스 지연으로 인한 블랙아웃 방지, 서버의 과부하 방지)

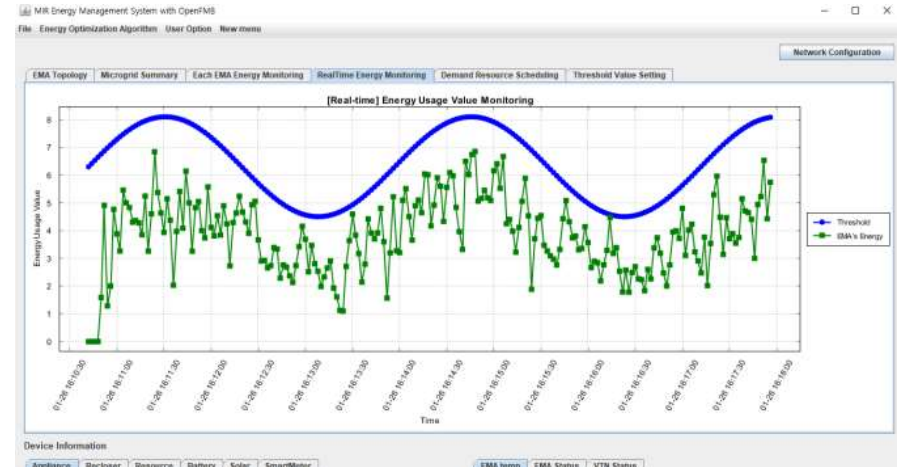
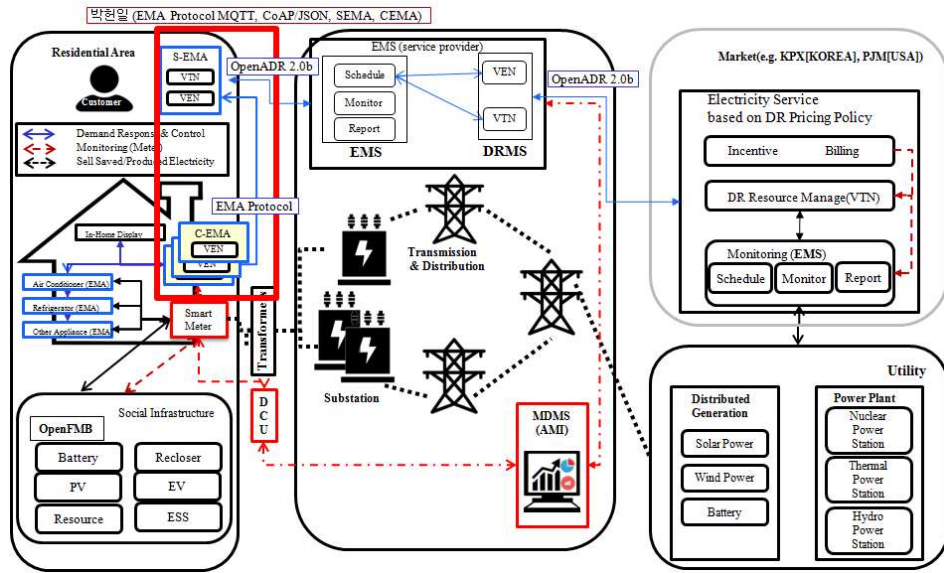
-다단계 구조인 가정과 집합건물들에 대한 Aggregator(중개업자)가 필요(통합적 수요반응(집중화 필요))

-기존 주거공간의 단일 에너지관리에이전트 -> 다단계 서버/클라이언트 에너지관리에이전트의 개념이 필요

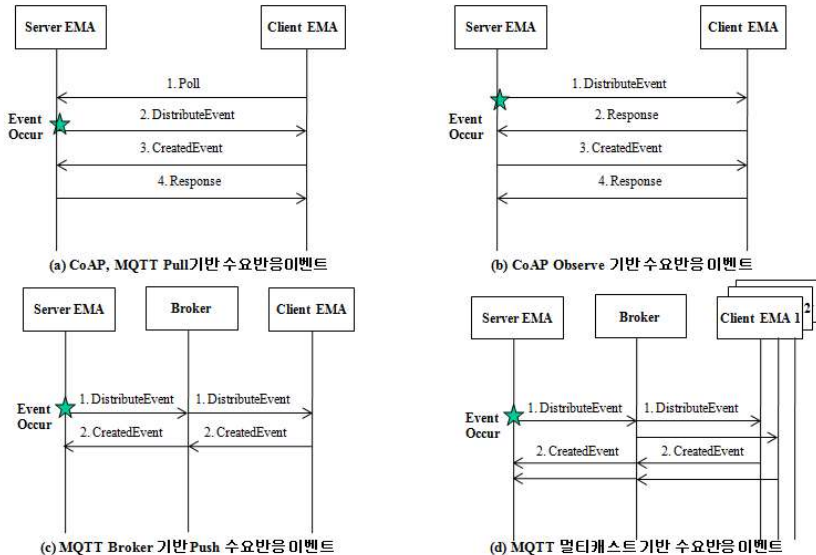
-다양한 수요참여고객의 증가 및 집합건물 환경에서 다양한 서비스 필요 -> 에너지관리에이전트 간 경량 수요반응 프로토콜의 전송 메커니즘이 필요

-수요반응 참여고객의 보안성 문제, 서비스 요구 사항 등 발생(기존 연구는 Explicit 사용) -> Implicit/Explicit 모니터링이 필요

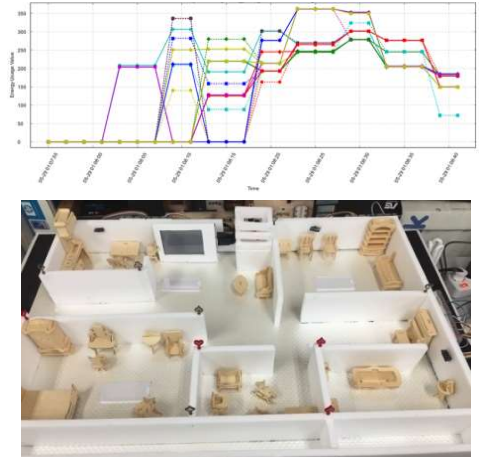
■ 논문 상세 내용(연구 배경 및 실험 환경)



VTN(EMS), Server EMA



Client EMA, OpenWRT

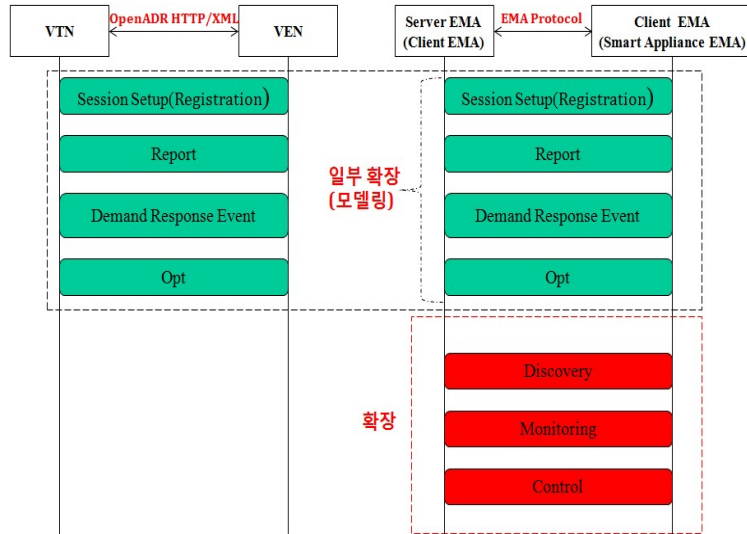


Smart Home(Appliances)

개발하면서 어려웠던 점은 VTN, Server EMA UI를 그려주는 것에 시간을 많이 썼고 OpenWRT(임베디드 시스템 OS)에 C, C++ 프로그램을 크로스 컴파일해서 올리는 과정과 make menuconfig로 라이브러리를 선택하여 OS 이미지를 만들어서 OpenWRT에 포팅 하는 것이었습니다.

■ 논문 상세 내용(집합 건물 환경의 경량 에너지관리 에이전트 프로토콜 구현)

- 에너지관리 에이전트 프로토콜의 수요반응 4가지 서비스 (차이) OpenADR2.0과 비교

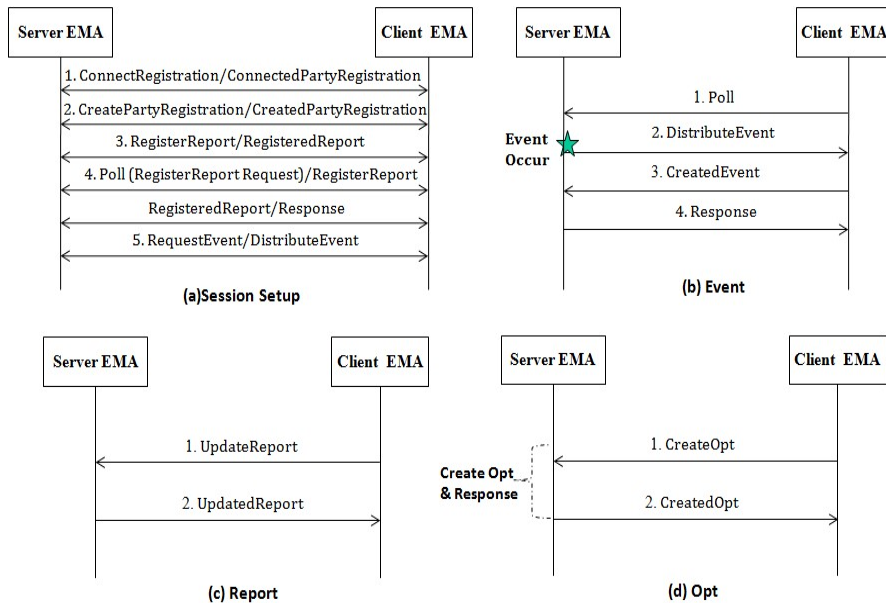


- (a) Session Setup(Registration) 과정
 - 에너지관리 에이전트 간 서로 연결을 수립하고 Report를 교환할 때 **기존의 Report에 대한 에너지 가격 정보나 클라이언트 에너지관리 에이전트의 에너지 관리 정보를 얻음.**

- (b) Event(Demand Response Event)
 - 에너지관리 에이전트 프로토콜은 다양한 이벤트 종류가 있다.** 예로 가격기반 수요반응 이벤트에서 Initial Price, Incentive Price, Negotiation Price가 있다. 서버 에너지관리 에이전트는 실시간적으로 가격에 대한 정보를 이벤트로 발생

- (c) Report(updateReport)
 - 에너지관리 에이전트 간 서로 연결을 수립 할 때 **Report를 교환할 때 실시간 에너지에 대한 가격 정보나 클라이언트 에너지관리 에이전트의 디바이스 정보 등을 얻음.**

- (d) Opt(수요반응 가용상태 및 스케줄링)
 - 에너지관리 에이전트 프로토콜의 Opt는 클라이언트 에너지관리 에이전트가 상위 서버 에너지관리 에이전트에게 수요반응 이벤트의 가용상태 또는 **수요반응 이벤트 프로그램 변경, 수요반응 스케줄링을 요청을 알려주는 서비스**



■ 논문 상세 내용(기존 문제점 및 해결책)

문제점 및 해결책	문제점	해결책
<p>(1)Heavyweight 수요반응 프로토콜의 제한점</p> <p>-집합 건물 환경을 위한 경량에너지관리에이전트 프로토콜</p>	<p>IoT 환경→저전력이고 소형화 된 네트워크 장비 HTTP 수요반응 프로토콜은 Heavyweight 해서 많은 리소스와 데이터 트래픽이 발생. XML은 많은 프로세싱 오버헤드,파싱 과정이 많음 Pull 메커니즘/네트워크 방화벽 문제</p>	<p>-경량 수요반응 프로토콜 및 JSON 사용 (CoAP, MQTT/JSON) -UpdateReport 실험 →CoAP, MQTT/JSON경량프로토콜 데이터 트래픽, 지연시간 감소</p>
<p>(2)Scalability 한계 및 수요 반응의 문제점</p> <p>-통합형 에너지관리에이전트프로토콜</p>	<p>수요반응에 대한 효율적 수요 관리 및 분배 필요, 보안성 문제-트래픽 발생 (예 : UpdateReport) 수요반응사업자로 부터 받은 수요반응 이벤트를 주거공간의 에너지관리에이전트가 Control만 지시 IoT 디바이스들에 Control 지시 (SEP2.0도 다이렉트 Control)</p>	<p>서버 에너지관리에이전트의 Aggregator 역할 다단계 에너지관리에이전트 설계로 인한 데이터 트래픽 감소, 병목 현상이 감소</p>
<p>(3)에너지관리에이전트의 Explicit 모니터링 문제점</p> <p>-Implicit/Explicit 모니터링</p>	<p>모든 디바이스의 정보를 받기 위해서는 응답시간이 오래 걸리고 디바이스를 개수가 증가하게 되면 데이 터 트래픽이 늘어 병목현상이 발생 기존 연구는 디바이스 정보가 모두 보내기 때문에 보안성 문제</p>	<p>에너지관리에이전트 프로토콜의 Implicit/Explicit 기반 모니터링 -데이터 트래픽의 단점을 극복 Implicit 방식 : 보안성(프라이버시) 향상 실험을 통해 증명</p>
<p>(4) 주거 공간 환경에서 수요반응 Event 서비스의 제한점</p> <p>-수요반응 Event 메커니즘을 이용한 에너지관리에이전트 프로토콜</p>	<p>제한적인 IoT환경에서 집합건물 내의 고객의 요구에 맞춘 수요 관리가 필요 (보안성 문제, 네트워크 방화벽문제, 그룹화 정보가 없어 유니캐스트 전송 문제 데이터 트래픽 문제 Topic: Group/#)</p>	<p>경량 프로토콜인 CoAP, MQTT를 사용 (EMAP 프로토콜 설계 및 구현) 경량 수요반응 프로토콜의 다양한 메커니즘 사용(다양한 시나리오 제시) 실험을 통해 증명</p>

■ 논문 상세 내용(집합 건물 환경의 경량 에너지관리 에이전트 프로토콜 구현)

구현 내용	그림	구현
<p>(1)경량 수요반응 프로토콜 기반 OpenADR 2.0b와 에너지관리 에이전트 프로토콜의 4가지 수요반응 서비스 구현</p>		<p>에너지 관리 에이전트 프로토콜(EMAP)가 다른 수요반응 프로토콜(OpenADR2.0b SEP 2.0의 데이터모델링을 따랐기 때문에 수요반응 프로토콜과 상호운용성이 있음 OpenADR 2.0b의 수요반응 명령어들과 Smart Energy Profile 2.0 (SEP2.0)의 과금, 제어 및 상 태 정보에 대한 데이터모델링을 사용</p>
<p>(2) 서버 에너지관리 에이전트(Aggregator) 의 기능적 중계 기능</p>		<p>1.서버 에너지관리 에이전트는 에너지관리시스템 으로부터 수요반응 이벤트를 중계해서 클라이언 트 에너지관리 에이전트에게 바로 전달 2. 서버 에너지관리 에이전트의 자체적인 수요 반응 스케줄링 통해 클라이언트 에너지관리 에이 전트에게 수요반응 이벤트를 내림 (외부 그리드와 통합적 수요반응 가능)</p>
<p>(3)Implicit/Explicit 모니터링 통신 기능</p>	<p>(a) Implicit</p> <pre>Implicit Object{ "SrcEMA" : String, "DestEMA" : String, "Type" : String, "EMANUM" : Integer, "EMAID" : String, "Power" : double, "Margin" : double, "Generate" : double, "Storage" : double, "Time" : Date }</pre> <p>(b) Explicit</p> <pre>Explicit Object{ "SrcEMA" : String, "DestEMA" : String, "Type" : String, "EMANUM" : Integer, "EMAID" : String, "Power" : double, "Margin" : double, "Generate" : double, "Storage" : double, "Time" : Date, Explicit: { ("DeviceEMAID" : Integer, "Sort" : "LED", "Power" : double, "State" : String, "Dimming" : Integer, "Priority" : Integer), ("DeviceEMAID" : Integer, "Sort" : "PV", "State" : String, "Power" : double, "Priority" : Integer), ("DeviceEMAID" : Integer, "Sort" : "ESS", "Mode" : Integer, "Power" : double, "State" : "On", "Capacity" : double, "Soc" : double, "Vol" : double, "Hz" : double, "ChargedEnergy" : double, "Priority" : Integer), ("DeviceEMAID" : Integer, "Sort" : "Recharger", "State" : String, "Power" : double, "Vol" : double, "Hz" : double, "Priority" : Integer), ("DeviceEMAID" : Integer, "Sort" : "Resource", "State" : String, "Power" : double, "Priority" : Integer) } }</pre>	<p>-Implicit 메시지 방식 (1)모니터링에 있어 사용자가 누군지 모르고 디 바이스 제어 같은 의사 결정이 어려움 (2)광범위한 스마트 홈 네트워크 환경에서 사용 자가 얼마나 많을지 모름 (3)데이터 트래픽 양이 적고 보안성은 높기 때문 에 공격자의 침입을 막을 수 있음</p> <p>-Explicit 메시지 방식 (1)상위 서버관리 에이전트나 에너지관리시스템 에서는 모든 디바이스 데이터가 있음 - 통계적 분석과 빅데이터 분석을 할 수 있는 장점이 있음.</p>

실험 하면서 어려웠던 점은 **OpenADR**이라는 프로토콜에 대한 전반적인 4가지 서비스에 관해서 이해도와 **MQTT, CoAP** 표준문서를 이해하고 구현, 각각 **Wireshark**실험 데이터가 나오기에 1대 3대 9대 20대를 각각 100번 넘게 실험하고 분석하는 것에 오래 걸렸습니다.

■ 논문 상세 내용(실험 및 구현 결과(대표적 실험만))

실험 내용	그래프
집합 건물 내부 이벤트 수요반응 전송 메커니즘 비교 실험	<p>The graphs illustrate the response times (in ms) for different event management architectures (EMA) under various conditions. The first graph shows Pull-based event response times, the second shows Push-based event response times, and the third shows response times for all EMAs. The protocols compared are CoAP/JSON (blue), MQTT/JSON (red), and HTTP/XML (green). Error bars are included for each data point.</p>

실험 시나리오 및 결과				실험 시나리오	CoAP/JSON	MQTT/JSON	실험 결과	결론																																										
<table><tr><td rowspan="5">EMS/ Server EMA/ VTN</td><td rowspan="5">Host PC</td><td>CPU</td><td>Intel Core i7-5600U</td></tr><tr><td>CPU Speed</td><td>2.6GHz</td></tr><tr><td>RAM</td><td>1GB</td></tr><tr><td>OS</td><td>Ubuntu 14.04 LTS 32bit</td></tr><tr><td>Language</td><td>JAVA, Jruby</td></tr><tr><td rowspan="5">EMA/ Client EMA/ VEN</td><td rowspan="5">Gateway</td><td>Model</td><td>Buffalo WZR-HP-G300NH</td></tr><tr><td>CPU</td><td>Atheros AR9132 rev 2 (0xb9)</td></tr><tr><td>CPU Speed</td><td>400 MHz</td></tr><tr><td>RAM</td><td>64 MB</td></tr><tr><td>OS</td><td>OpenWRT 15.05 Chaos Calmer</td></tr><tr><td>Language</td><td>C, C++</td><td></td><td></td></tr><tr><td rowspan="5">MQTT/ CoAP</td><td rowspan="5">Device</td><td>Model</td><td>RaspberryPi3</td></tr><tr><td>CPU</td><td>ARM Cortex-A53</td></tr><tr><td>CPU Speed</td><td>1.2GHz</td></tr><tr><td>RAM</td><td>1GB</td></tr><tr><td>Language</td><td>C</td><td></td><td></td></tr></table>				EMS/ Server EMA/ VTN	Host PC	CPU	Intel Core i7-5600U	CPU Speed	2.6GHz	RAM	1GB	OS	Ubuntu 14.04 LTS 32bit	Language	JAVA, Jruby	EMA/ Client EMA/ VEN	Gateway	Model	Buffalo WZR-HP-G300NH	CPU	Atheros AR9132 rev 2 (0xb9)	CPU Speed	400 MHz	RAM	64 MB	OS	OpenWRT 15.05 Chaos Calmer	Language	C, C++			MQTT/ CoAP	Device	Model	RaspberryPi3	CPU	ARM Cortex-A53	CPU Speed	1.2GHz	RAM	1GB	Language	C			1. 평소 Pull 기반 수요반응 서비스 트래픽 비교 (UpdateReport 실험)	Pull 기반 CoAP/JSON	Pull 기반 MQTT/JSON	(SEMA에서 트래픽측정) CoAP와 MQTT와 트래픽 양은 비슷했음	리소스가 제한적인 집합건물에 경량 수요반응 프로토콜 MQTT/JSON, CoAP/JSON 이 필요
						EMS/ Server EMA/ VTN	Host PC	CPU	Intel Core i7-5600U																																									
								CPU Speed	2.6GHz																																									
								RAM	1GB																																									
OS	Ubuntu 14.04 LTS 32bit																																																	
Language	JAVA, Jruby																																																	
EMA/ Client EMA/ VEN	Gateway	Model	Buffalo WZR-HP-G300NH																																															
		CPU	Atheros AR9132 rev 2 (0xb9)																																															
		CPU Speed	400 MHz																																															
		RAM	64 MB																																															
		OS	OpenWRT 15.05 Chaos Calmer																																															
Language	C, C++																																																	
MQTT/ CoAP	Device	Model	RaspberryPi3																																															
		CPU	ARM Cortex-A53																																															
		CPU Speed	1.2GHz																																															
		RAM	1GB																																															
		Language	C																																															
2. 수요반응 Event 을 한대에 내릴 때(Pull)	Pull 기반 CoAP/JSON	Pull 기반 CoAP/JSON	Event 응답 시간은 HTTP/XML<MQTT/JSON <CoAP/JSON 순서로 빠름 (CoAP가 MQTT보다 1.5배 빠름)	리소스가 제한적인 집합건물에 경량 수요반응 프로토콜 MQTT/JSON, CoAP/JSON 이 필요																																														
3. 수요반응 Event 을 한대에 내릴 때(Push)	CoAP Observe	Broker 기반 MQTT Push	20대 일때 MQTT Push Un 보다 CoAP Observe(Push) 응답속도가 81배 빠름	집단 건물 내의 많은 EMA가 있고 긴급한 실시간 수요반응 프로토콜은 CoAP/JSON이 적합.																																														
4.수요반응 Event 을 모든 EMA에 내릴때	CoAP Observe 유니캐스트	Broker 기반 MQTT Push 유니캐스트 / Broker 기반 MQTT Push 멀티캐스트 메커니즘	20대 일때 MQTT Push Unicast보다 CoAP Observe(Push) Unicast가 응답속도가 3.4배 빠름 20대 일때 MQTT/JSON Push Muticast보다 CoAP Observe(Push) Unicast가 응답속도가 2.5배 빠름	-집단 건물 내의 많은 CEMA 가 있고 여러 개의 중복 메시지를 전달 할 때 CoAP/JSON이 적합. (하지만, 집합 건물의 SEMA 가 부하가 많을 때 MQTT Broker 기반의 MQTT/JSON Push Multicast을 쓰면 보내는 쪽에서 1/N(CEMA)로 트래픽이 적음)																																														

■ 논문 상세 내용(논문의 결론 및 향후 과제/데모)

- 본 논문의 결론

- (1)EMA를 통해 집합 건물 환경에서도 지능화된 에너지관리인 수요반응을 함
- (2)에너지 관리 에이전트 프로토콜을 통해 수많은 집합건물의 수요관리 고객에 대한 실시간 수요반응을 할 수 있음(Push 실험을 통해 증명)
- (3)집합 건물 한 경에서 에너지관리에이전트 간 통신을 통해 수요반응 및 통합적 스케줄링을 할 수 있음
- (4)집합 건물 환경 에서 EMA 프로토콜을 경량 IoT 프로토콜인 CoAP, MQTT 이용
 - Heavyweight인 HTTP/XML 방식의 수요반응 프로토콜은 제한이 있음(OpenADR2.0b, SEP2.0)
 - 디바이스나 IoT기기, Gateway들에는 제한된 리소스 및 배터리 용량이 제한, IoT(CoAP, MQTT) 프로토콜이 lightweight, 저전력 기반
- (5)제한적인 IoT환경에서 데이터 트래픽, 수요반응 전송 속도를 줄이기 위해서 IoT 프로토콜의 메커니즘 비교를 통해 적합한 프로토콜인 CoAP/JSON 이용- (이유 : 집합건물의 수많은 개인지역으로 확대(수많은 노드) , 효율적 수요반응 관리가 필요)
 - 1. CoAP, MQTT 기반 lightweight, 2. CoAP Observe, 3. MQTT Broker 기반 Push 메커니즘, 4. MQTT 기반 멀티캐스트
 - CoAP 가 실시간 수요반응의 효율성이 있음을 증명
- (6)기존연구의 Explicit모니터링과 EMAP의 Implicit/Explicit 모니터링 비교를 통해 데이터 추상화 기능 의 필요성을 증명

-본 논문의 향후 과제

- 에너지관리에이전트 프로토콜은 스마트 홈과 스마트 공장 등에서 스마트 시티로 확대 적용
(스마트시티 기반의 에너지관리에이전트를 분산전원 및 Advanced Metering Infrastructure(AMI)과 결합 하는 추가적 연구가 필요)
- 다양한 IoT 기술 및 플랫폼과 결합하여 IoT 기기들과 MQTT, CoAP을 이용하여 통합 서비스 프레임워크를 구축
- 4차 산업 혁명이 등장함에 따라 집합 건물 내의 에너지관리에이전트는 홈 게이트웨이나 스마트 IoT 디바이스(인공지능 스피커, 공유기 등)와 새롭게 결합하고 서로 간 서비스 결합이 활발히 발생

-DEMO

https://www.youtube.com/watch?v=6_pg-Md-EKk

