

스마트 에너지 IoT 환경에서 OpenADR 2.0b 수요반응의 Push 메카니즘 필요성 연구

박현진, 박현일, 강성철, 김세영, 이성환, 최진식*
한양대학교, *(교신저자)

phj3372@hanyang.ac.kr, heonil8@nate.com, seongck115@gmail.com, heehouse1@gmail.com,

zmagician@naver.com, *jinseek@hanyang.ac.kr

Research on The Necessity of Push Mechanism on Smart Energy IoT Environment of OpenADR 2.0b Demand Response

Park Hyun Jin, Park Heon Il, Kang Seong Cheol, Kim Sae Young, Lee Sung hwan, Choi Jin Seek*

Hanyang Univ.

요 약

본 논문의 목적은 스마트 에너지 홈 환경에서 수요 반응 표준 프로토콜로 제안되어 사용되는 OpenADR 2.0 b (Open Automated Demand Response)의 Pull Mechanism에 대해 Event Response, Data Traffic량을 Polling Interval 간격 별로 성능을 분석하고 Smart IoT 환경에서 빠른 수요반응 서비스를 제공할 수 있도록 Push Mechanism의 필요성을 제안 한다.

I. 서 론

전기가 부족해 모든 전력시스템이 정지되는 현상이 대규모 정전상태로 블랙아웃(Black-out)이라한다. 2011년 9월 15일 전국적인 이상기후로 무더위가 지속되면서 전력수요가 급증하자 서울을 비롯해 전국 곳곳이 기습적으로 정전되는 사상 초유의 사태가 발생했다.

예고되지 않은 블랙아웃을 방지하기 위해서 예비전력에 대한 대책이 필요하다. 전력거래소에 따르면 수요반응으로 전력계통의 수급위기에 신속히 대응 한다[1]. 수요예측 오차, 자연재해 및 발전기 고장 등 단기 수급 불균형에 순시 적으로 대비해야한다. 그러나 현재의 수요반응은 피크 억제로 전기요금 인상을 억제하며 신규 발전기 건설을 억제하면서 발전기 건설비용이 절감되도록 전력수급의 계획에만 적용되었다. 따라서 공급위주의 안정적인 전력 수급에서 수요관리를 통한 효율적인 전력수급이 필요하기 때문에 수요반응(Demand Response) 개념이 나왔고 전력을 체계적으로 관리하기 위해서 OpenADR 프로토콜이 등장하게 되었다.

본 논문의 목적은 스마트 에너지 홈 환경에서 수요 반응 표준 프로토콜로 제안되어 사용되는 OpenADR 2.0 b (Open Automated Demand Response)의 Pull Mechanism을 분석하고 Smart IoT 환경에서는 Utility 환경과 다르게 Push Mechanism의 필요성을 제안하고자 한다.

II. 본론

1. 수요반응(Demand Response; DR)

수요반응은 수요관리의 하위개념이며, 전력수요 피크 등의 이유로 인해 수급위기 발생 시 전기 요금의 조정이나 부하감축 지시에 의한 전력절감 등을 통해 수급의 균형을 유지하기 위한 활동이다. 이 활동의 목적은 크게 최대 수요의 억제, 최대부하의 이전, 기저부하의 증대로 구분될 수 있다. [2]

2. OpenADR 2.0b

OpenADR은 Open Alliance에서 개발한 지능형 수요 반응(Demand Response, DR)에 적용되는 HTTP 1.1/XML 기반의 표준 프로토콜이다. 2.0b 버전은 다수의 벤더 수용, 인증 테스트 도구, Organization for the Advancement of Structured Information Standards (OASIS) 기반 국제 표준, 대다수 수요 반응 프로그램에 적용되는 장점을 가지고 있다. OpenADR에서 DR 통신 참여자는 Virtual Top Node(VTN)과 Virtual End Node(VEN)으로 구분된다. VTN은 DR의 Event를 제공하는 역할을 하고 End Device 혹은 중간 서버에게 OpenADR Signal을 제공한다[3].

3. Pull and Push Mechanism

	Pull	Push
features	Delays will occur between message publication and delivery.	Delivery is immediate (except when delivery is being rate-limited to avoid overwhelming the endpoint). There is no added latency from pull requests.
	Achieves high throughput at low CPU and bandwidth by allowing batched delivery and acknowledgments as well as massively parallel consumption. May be inefficient if aggressive polling is used to minimize message delivery time.	Delivers one message per request and limits maximum number of outstanding messages.
Recommended for	Large numbers of subscribers which are created dynamically and cannot be manually configured	Subscribers with low traffic
	Requiring configuration with SSL certificates and a web server for push subscription	Subscribers that need closer to real-time performance

table 1. Pull and Push features.[4]

table1은 Push와 Pull 방식에 대한 비교이다. Pull 방식은 메시지 송수신에 지연이 있으나 수동적으로 설정 할 수 없는 많은 subscriber가 있을 때 적절한 방식이다. 반면 Request 메시지를 받았을 경우에만 메시지를 전송할 수 있기 때문에 실시간 통신 방식에는 적합하지 않다. Push 방식은 subscriber가 네트워크 방화벽 뒤에 있는 경우 기술적인 어려움이 있으나 실시간 통신 방식에 적합하고 Data traffic량을 줄이며 pull request의 latency를 줄일 수 있는 장점이 있다.

4. OpenADR 2.0b Communication Process

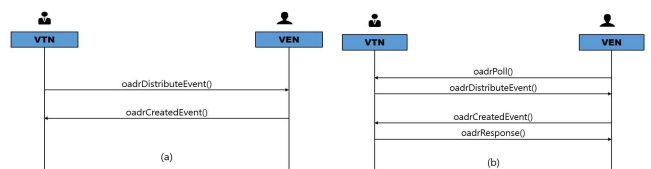


figure1. OpenADR 2.0b Push & Pull communication

figure1은 OpenADR2.0b HTTP/XML 기반에서의 Push와 Pull 통신방식이다. Pull 방식에서는 VEN이 VTN을 향하여 oadrPoll 메시지를 주기적으로 보낸다. Pull 방식은 VEN 상에 있는 HTTP 서버에 대한 요건을 제거하며, VEN의 네트워크 방화벽 문제 가능성을 회피할 수 있다. 하지만 polling 주기에 기인한 지연시간과 Traffic 증가라는 자체적인 한계를