

CoAP 기반 사물인터넷 시스템 성능평가

추영열[†], 하용준^{**}, 손수동^{***}

Performance Evaluation of CoAP-based Internet-of-Things System

Young Yeol Choo[†], Yong Jun Ha^{**}, Soo Dong Son^{***}

ABSTRACT

Web presence is one of the key issues for extensive deployment of Internet-of-Things (IoT). An obstacle to overcome for Web presence is relatively low computing power of IoT devices. In this paper, we present implementation of an IoT platform based on Constrained Application Protocol (CoAP) which is a web transfer protocol proposed by Internet Engineering Task Force (IETF) for the low performance IoT devices such as Wireless Sensor Network (WSN) nodes and micro-controllers. To qualify the performance of CoAP-based IoT system for such an application as smart grid, we designed a test platform consisting of Raspberry Pi2, Kmote WSN node and a desktop PC. Using open source softwares, CoAP was implemented on top of the platform. Leveraging the GET command defined at CoAP specification, performance of the system was measured in terms of round-trip time (RTT) from web application to the Kmote sensor node. To investigate abnormal cases among the test results, hop-by-hop delays were measured to analyze resulting data. The average response time of CoAP-based communication except the abnormal data was reduced by 23% smaller than the previous research result.

Key words: CoAP, Wireless Sensor Network, IoT, Wi-Fi, Real-Time

1. 서 론

2000년대 초 MIT (Massachusetts Institute of Technology)의 Kevin Ashton에 의해 사물인터넷, 즉 “Internet of Things” (IoT) 라는 개념이 제시된 이후 이 용어는 Radio Frequency Identifier (RFID)를 인터넷에 접속한다는 개념에서 나아가 모든 전자적 기능을 갖는 물리적 개체들을 포함하는 것으로 확장되었다. 사물인터넷에 접속되는 장비의 수는

2020년 말까지 약 26억개에 이를 것으로 추정되고 있다[1].

IoT는 널리 사용되는 용어이지만 그 개념은 다양하게 정의되어왔고 다소의 모호성을 내포하고 있다. [2]에서는 IoT에 대해 “world-wide network of interconnected objects uniquely addressable, based on standard communication protocols” 라고 정의하고 있으며 근래에는 사물들을 연결하는, 또는, 사물들에 접속을 허용하는 서비스를 포함하는 개념으로

※ Corresponding Author : Young Yeol Choo, Address: (608-711) 413, Building No.18, Tongmyong Univ., Yongsang-dong, Nam-gu, Busan, Republic of Korea, TEL : +82-51-629-3752, FAX : +82-51-629-3753, E-mail : yy-choo@tu.ac.kr

Receipt date : Nov. 8, 2016, Revision date : Dec. 5, 2016
Approval date : Dec. 6, 2016

[†] Dept. of Computer & Media Eng., Graduate School, Tongmyong University

^{**} Dept. of Computer & Media Eng., Graduate School, Tongmyong University (E-mail : yongjun86@tu.ac.kr)

^{***} Dept. of Computer & Media Eng., Graduate School, Tongmyong University
(E-mail : ssdzzzzz@tu.ac.kr)

※ This work was supported by Tongmyong University Research Grant (2015).

※ This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20162020106920).

확장되고 있다[3]. IoT는 무선 센서 네트워크 (Wireless Sensor Network, WSN), Machine-to-Machine (M2M) 등의 개념을 포괄하고 있다.

인터넷이 현재와 같이 확산될 수 있었던 중요한 원인 중 하나는 World Wide Web의 탑재라 할 수 있다. 사용자의 측면에서는 대부분 웹을 통해 인터넷 서비스와 정보에 접근한다. 이는 IoT의 경우에도 동일하게 적용될 것이다. 즉, IoT 장비들이 웹을 통해 직접 접속되거나 감시 및 제어되는 것이 IoT의 확산에 중요한 요소가 될 것이다[4]. IoT 장비들이 웹 서비스에 편입되기 위해서는 몇 가지 선행하여 해결되어야 할 점들이 있다.

첫 째로 각 장비들이 인터넷에 접속될 수 있도록 장비들에 대한 네이밍(naming) 방법이 필요하다. 이에 대한 방법으로 게이트웨이 장비를 통해 간접적으로 접속되거나 또는 IPv6의 128 비트 주소체계를 이용하여 주소를 할당하는 방법이 제시되어 있다. IPv6의 128 비트 주소체계는 IoT 장비에 충분한 ID 공간을 제공한다.

둘째로 IoT 장비들의 계산능력과 자원, 특히 에너지 자원이 제한되어 있고 IEEE 802.15.4와 같은 저속, 저신뢰성을 갖는 무선 네트워크로 연결됨에 따라 이를 위한 라우팅 프로토콜이 요구된다. 이를 위해 IETF (Internet Engineering Task Force)는 6LoWPAN (IPv6 over Low power Wireless Personal Area Network)[5] 과 RPL (Routing Protocol for Low Power and Lossy Network)[6] 표준안을 제정하였다.

셋째로 웹과 같은 응용 프로그램과의 호환을 위해 수송계층에서 제공되는 통신에서의 다중 연결(multiple connection) 처리와 중단 노드간의 전송 신뢰성 보장, 그리고 HTTP (Hyper Text Transfer Protocol) 구문을 제한된 자원환경에 맞도록 번역하는 메세징 프로토콜(messaging protocol)이 요구된다. 이를 위해 IETF는 UDP(User Datagram Protocol)을 기반으로 하는 CoAP (Constrained Application Protocol)을 제시하였다[3,7]. CoAP외에 MQTT (Message Queue Telemetry Transport)가 있다[8]. 본 연구에서는 확장성과 국제표준임을 고려하여 CoAP를 중심으로 진행하였다.

IoT는 가정, 사무실, 산업체 등 다양한 분야에 응용되고 있다[9-13]. 본 논문에서는 스마트 그리드 등

산업 분야에서의 IoT 장비들의 활용과 관련하여 CoAP를 기반으로 IoT 통신 플랫폼을 설계하고 구현한 결과를 기술한다. 하위 네트워크로는 WiFi와 IEEE 802.15.4 프로토콜을 구현하였고 CoAP는 공개 소프트웨어를 활용하였다. 구현된 IoT 통신 플랫폼 상에서 말단 IoT 장비를 모니터링 명령을 전송하는 다양한 실험을 통해 통신 성능을 측정하였다. 이를 통해 시간의 지연이 고려요소가 되는 산업 응용과 특히 마이크로 그리드에서의 CoAP 적용 가이드를 도출하는 것을 목표로 한다. 본 논문의 구성은 다음과 같다.

2장에서 관련 연구와 CoAP 사상에 대해 소개한다. 3장에서 CoAP 기반 통신 시스템의 설계 및 성능 측정 방법에 대해 기술한다. 4장에서 통신 지연 측정 결과에 대해 기술하고 5장에서 결론을 맺는다.

2. 관련연구 및 CoAP

2.1 관련연구

WiFi 등 무선통신과 저전력 IoT 장비들에 대해 IETF 통신 표준을 기반으로 하는 네트워크 구조와 CoAP에 대해서는 [3,14-17]에 특징과 고려사항들이 기술되어 있다. 특히, [17]에서는 각 산업 영역의 응용에 대해 IoT 네트워크의 종단간 통신 과정을 센서 영역, 데이터 연결 과정, 클라우드 관리 플랫폼 과정, 관리 포탈 영역 등 4 과정으로 나누어 각 영역 별 관리 기능을 분석하고 서비스를 정의하였다.

저전력 IoT 장비들의 데이터링크 연결을 위해 사용되는 IEEE 802.15.4는 2006년에 표준안이 비준된 후 저속에도 불구하고 저전력 특성으로 해서 널리 사용되었으나 긴 슬롯 타임으로 인해 산업체 응용에는 사용상 제한이 있었다. 이를 개선하여 짧은 타임 슬롯을 갖는 시분할 다중화 방식의 규정이 2010년 IEEE 802.15.4e 표준으로 확정되었다. 이러한 IoT 하위 계층 네트워크에 대해서는 [18,19]에 기술되었다. [20]에서는 CoAP와 HTTP에 대해 클라이언트-서버 사이 트랜잭션 발생시 소모되는 에너지와 응답시간을 비교하였다. Tmote sky 노드를 사용하여 1 홉(one hop)과 2 홉(two hop) 사이의 지연시간 측정 결과 1 홉에 대해 CoAP를 사용한 경우 평균 지연이 184 ms, HTTP와 TCP (Transmission Control Protocol) 조합의 경우 약 1.8초로 CoAP에 비해 HTTP가 10배 정도 시간지연이 큰 것으로 나타났다. [21]에

서는 컴퓨터, BeagleBone, BeagleBone Black, Raspberry Pi 내장형 컴퓨터에 대해 이를 IoT 게이트웨이로 사용할 경우 각각의 메모리, CPU (Central Processing Unit) 사용률과 CoAP 지연시간을 측정하였다. 이 논문에서는 통신지연에서 통신 채널의 선택이 성능에 중대한 영향을 미침을 밝히고 있다.

이 외에 IoT의 실시간 통신에 대한 연구들이 있다. [22]에서는 Java를 이용하여 CoAP를 구현하고 유선망에서 성능을 실험한 결과를 기술하고 있다. CoAP를 실시간 응용에 활용하는 연구로는 CoHART가 있으나 이는 IETF의 표준 프로토콜 스택에서 벗어난다는 단점이 있다[23]. 통신 지연에 영향을 미치는 요소로 과잉밀집(congestion) 상태가 있다. R. Bhalerao 등은 RTT (Round Trip Time)을 바탕으로 2개의 RTO (Re-transmission Time-Out) 타이머를 사용하여 과잉밀집 상태를 제어하는 알고리즘을 제안하였다[24].

2.2 CoAP

CoAP는 IETF CORE (Constrained RESTful Environments) 워킹그룹에서 저전력, 저신뢰도 성능을 갖는 네트워크 (Low power and lossy network, LLN)에 소속된 IoT 장비들이 웹과의 통합을 위해 HTTP와 호환될 수 있도록 2011년 제정한 프로토콜이다[7]. 주요한 기능과 특징은 다음과 같다. CoAP가 동작하는 계층구조는 Fig. 1과 같다. 그림에서 보는 것처럼 수송계층으로 UDP를 사용하므로 중단간의 신뢰성 있는 전송이나 중복된 메시지의 처리를 보장하지 못한다. 이러한 기능은 따라서 응용계층에서 제공하여야 한다. 단, 포트 (port) 번호는 유지하므로

복수의 연결 서비스는 제공한다.

특징은 비동기적 메시지 교환, 낮은 오버헤드, 특히 헤더 부분에서의 오버헤드 경감, best-effort 형태의 멀티캐스트 지원, URI (Universal Resource Indicator) 및 콘텐츠의 형식의 지원 등으로 요약할 수 있다. CoAP의 메시지 계층과 요청/응답 (request/response) 계층의 기능은 다음과 같다.

메시지 계층의 기능은 UDP를 이용하여 중단 노드 간 메시지 교환을 제어하는 것이다. 메시지는 ID에 의해 구분되며 사용자의 요청 메시지와 이에 대한 응답 메시지는 같은 형식을 갖는다. 이 계층에서는 다음과 같은 4가지 형식의 메시지를 정의하고 있다

- Confirmable : 이 메시지의 수신자는 Ack로 응답해야 함.
- Non-confirmable : Ack나 재전송이 불필요한 메시지
- Acknowledgement : Confirmable 메시지에 대한 응답
- Reset : 정상 처리되지 못한 Confirmable 메시지에 대한 응답

요청/응답 계층의 기능은 응용 계층에서 요구되는 서비스의 의미(semantics)를 그에 맞는 하위 계층의 서비스로 연결해주는 것이다. 또한, CoAP는 UDP (User Datagram Protocol)를 사용함에 따라 중복 메시지, 도착순서의 뒤바뀜, 메시지 손실 등이 발생할 수 있으므로 다음과 같은 기능을 정의하고 있다.

- Confirmable 메시지에 대해 exponential back-off 방식을 이용하여 stop-and-wait 재전송 알고리즘 수행
- Confirmable/non-confirmable 메시지에 대해 중복 메시지 검출
- 멀티캐스트 지원

Confirmable 메시지에 대해 Ack 또는 Reset과 같은 응답이 없으면 정해진 재전송횟수 내에서 전송 간격을 지수승 만큼 증가시키며 재전송한다. Non-confirmable 메시지는 응답이 요구되지 않으며 처리되지 못한 메시지는 무시된다.

요청 메시지는 HTTP와 유사하게 같이 4 개가 있다. GET은 request URI에 의해 지정된 정보를 송신해 줄 것을 요청한다. PUT은 request URI로 명시된 자원을 갱신하거나 새로 생성해 줄 것을 요청하는 method이다. POST는 자원의 정보를 제시하는 값

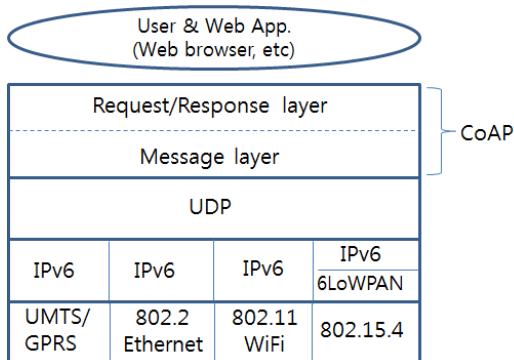


Fig. 1. Layered architecture that CoAP operates.

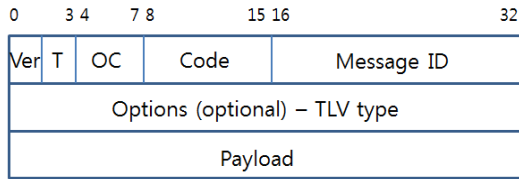


Fig. 2. CoAP frame format.

로 처리해 줄 것을 요청하며 DELETE는 해당 자원의 삭제 요청이다. 이에 대한 응답으로는 Success, Client Error, Internal Server Error 등 다양한 종류가 있으며 이들 각각에 대한 코드값은 [25]에 정의되어 있다. 메시지의 형식은 Fig. 2와 같다. 그림에서 Ver은 2 비트로써 버전 정보를 표시한다. T는 2 비트로 숫자에 따라 메시지의 형식을 표시한다. 0은 Confirmable, 1은 Non-confirmable, 2는 Ack, 3은 Reset을 각각 나타낸다. Code는 GET, PUT 등 각 요청 메시지 또는 응답 메시지의 코드 값을 나타낸다. Option 은 그 값이 (Type, Length, Value)의 형식을 취한다.

3. 성능 측정 시스템 설계

성능측정을 위해 웹페이지에서 CoAP를 통해 무선 센서 노드의 센싱 값을 읽어 출력하는 통신 시스템을 구축하였다. 시스템의 구성은 Fig. 3과 같다.

라즈베리파이2 시스템을 서버로 사용하고 데스크탑 PC가 클라이언트 역할을 수행한다. PC와 서버 사이는 IEEE 802.11n USB 무선 랜카드를 통해 통신한다. 무선 센서노드는 Kmote를 사용하였고 IEEE 802.15.4 통신 사양은 Texas Instrument사의 CC2420 칩에 의해 구현되어 있으며 라즈베리파이2에서는 Xbee S1 보드를 통해 Kmote와 통신한다. Kmote는 TinyOS-2.x를 기반으로 구현되었으며 IEEE 802.15.4 표준과 다른 부분이 있다. Kmote의 Xbee S1과 통신을 위해서는 Kmote에서 구현된 헤더의 AM type이

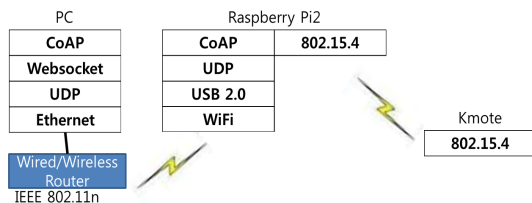


Fig. 3. Configuration of the communication system.

라는 필드를 삭제하여야 한다. CoAP 서버 기능은 라즈베리파이2 보드에 구현하였고 PC(클라이언트)와 CoAP 메시지를 송수신한다. 웹 서버 기능은 PC에 구축하여 라즈베리파이2로 부터의 CoAP 응답을 웹페이지에 출력하였다.

성능인자로는 RTT (Round Trip Time)을 사용하였고 아래와 같은 순서로 설계된 시스템에서 성능을 측정하였다.

- 1) 클라이언트에서 무선 센서 노드의 조도센서 값 요청
- 2) 서버에서 센서 노드로 조도 값 요청
- 3) 센서 노드에서 서버로 조도 값 전송
- 4) 서버에서 클라이언트로 수신된 조도 값 전송
- 5) PC에서 2초간 휴지 기간을 가짐
- 6) 1)~5)의 과정을 100회 반복

RTT의 측정은 4가지 방법으로 진행되었다. 첫 번째와 두 번째 테스트는 웹 통신을 제외하고 실행했다. 첫 번째 테스트에서는 IPv4를 사용하고 두 번째 테스트는 IPv6를 사용했다. 메시지 송수신 과정은 아래의 Fig. 4와 같다.

전체 RTT는 다음 식 (1)로 표현할 수 있다.

$$T_{RTT} = 2D_{PR} + 2D_{RK} + D_{sensing} \quad (1)$$

여기서 T_{RTT} 는 전체 통신지연 시간, D_{PR} 은 데스크탑 PC에서 프레임을 전송하여 라즈베리파이2에서 수신할 때까지의 통신 지연 시간, D_{RK} 은 라즈베리파이2에서 송신하여 Kmote에서 수신할 때까지의 통신 지연 시간, $D_{sensing}$ 은 조도센서 값을 읽어 처리하는 시간을 각각 나타낸다. PC에서 라즈베리파이2로의 전송지연과 역방향으로의 전송지연은 같은 것으로 하였다. 라즈베리파이2와 Kmote 사이의 송수신 시간도 동일하게 가정하였다. 세 번째 테스트에서는 웹 통신을 포함하여 RTT를 측정했다. 웹 페이지에서 센서 노드의 조도 값을 요청하면 PC에 구축되어 있는 CoAP 클라이언트 프로그램을 통해 라즈베리파이2로 전송한다. 세 번째 테스트 과정은 Fig. 5와 같다.

세 번째 실험의 웹 통신을 포함한 전송지연은 식 (2)와 같이 주어진다.

$$T_{RTT_w} = 2D_w + 2D_{PR} + 2D_{RK} + D_{sensing} \quad (2)$$

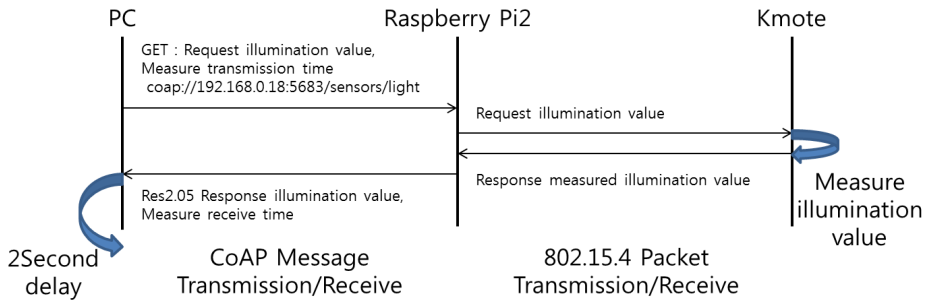


Fig. 4. The test procedure of message communication (without Web communication).

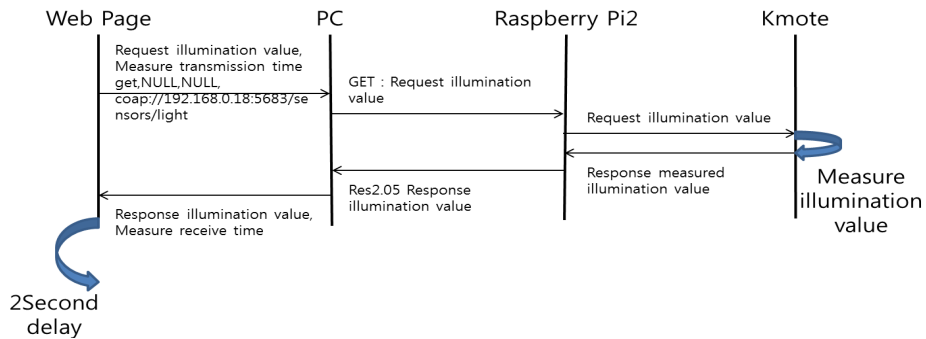


Fig. 5. The test procedure of message communication (with Web communication).

식 (2)에서 D_w 는 웹 통신에 소요되는 시간지연이다. 식 (1)과 (2)의 각 항 중에서 어느 부분이 전체 통신 지연에서 주요한 요소인가를 파악하기 위해 링크 단위별 통신 지연을 측정하였다.

마지막으로 기존의 웹에서 사용하고 있는 HTTP와 TCP 조합의 통신규약 하에서 응답시간을 측정하였다. 이때는 조도 센서를 구동하지 않고 Kmote에서 바로 응답하도록 하였다.

4. 성능 측정 결과

4.1 RTT 측정결과

첫 번째 테스트는 IPv4를 이용하여 100회 실행한 결과 측정값의 평균은 478 ms, 최대치는 1602 ms, 최소치는 373 ms로 측정 되었다. Fig. 6처럼 평균치에 비해 1초 이상 지연된 경우가 있었다. 이러한 데이터는 측정을 반복하여도 나타났는데 이는 네트워크에서의 전파 간섭 또는 충돌로 인한 지연으로 추정된다. 이를 확인하기 위한 실험을 별도로 진행하였고 측정 결과는 뒤에서 기술한다.

두 번째 테스트는 IPv6를 사용하였고 측정구간과

방법은 첫 번째 테스트와 동일하였다. 측정치의 평균은 498 ms, 최대 1,770 ms, 최소 383 ms였다. 첫 번째 테스트에서처럼 평균치와 상당한 차이를 보이는 측정값이 Fig. 7과 같이 나타났다. 이러한 평균에서 1초 이상 벗어난 데이터를 제외하면 첫 번째와 두 번째 테스트의 평균치는 각각 451 ms, 454 ms로 거의 차이가 없음을 보여준다. 즉, IPv4와 IPv6의 프로토콜로 인한 차이는 미미함을 확인할 수 있다.

세 번째 테스트는 앞장에서 기술한 것처럼 PC에 웹서버 기능을 구현하고 이로부터 Kmote까지의

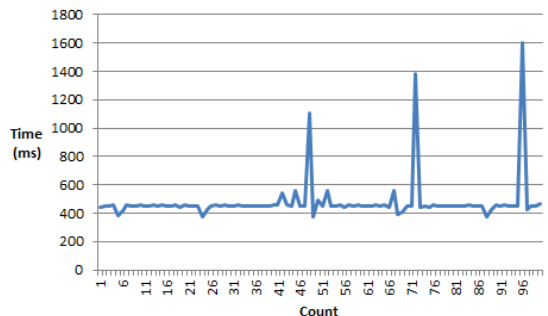


Fig. 6. RTT between PC and Kmote (IPv4).

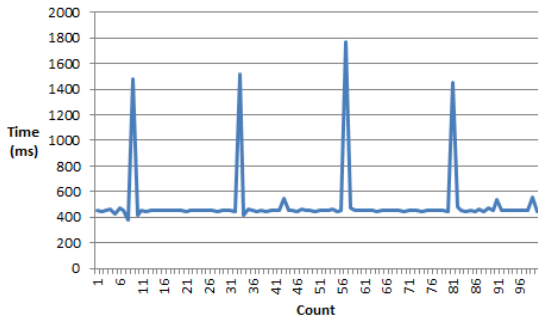


Fig. 7. RTT between PC and Knote (IPv6).

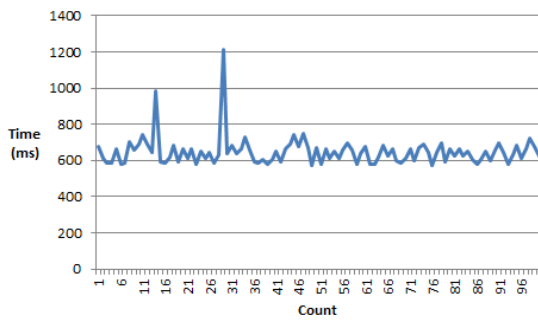


Fig. 8. RTT between PC and Knote (IPv6).

RTT를 측정하였다. IPv4를 사용하였고 평균치는 647 ms, 최대 1,213 ms, 최소 570 ms로 측정되었고 이를 Fig. 8에 나타내었다. IPv6 기반으로 실험한 결과값은 평균이 659 ms, 최대 1,544 ms, 최소 570ms였다. IPv4, IPv6에 상관없이 평균치에서 상당히 벗어나는 RTT 최댓값을 확인할 수 있었다. 이는 어느 정도의 시간제약을 갖는 응용에 사용할 경우 불확정성을 초래하게 된다. 이러한 지연이 식 (1)의 어느 부분에서 발생하는지 확인하기 위해 추가적인 테스트를 실행했다.

마지막으로 HTTP와 TCP 조합의 통상적인 웹 환경에서의 RTT를 측정하였다. 시스템의 구성은 세 번째 실험과 동일하다. 지연시간의 최대치는 80초, 최소 340 ms, 평균치는 4,418.6 ms였다. 이 경우 역시 50초를 상회하는 데이터를 4개를 제외한 경우 평균 지연 시간은 1193.6ms였다.

4.2 구간별 RTT 측정 결과

테스트는 $2D_{PR}$ 의 측정과 $2D_{RK} + D_{sensing}$ 을 측정하는 두 가지 방법으로 진행하였다.

$2D_{PR}$ 의 측정은 PC에서 라즈베리파이2로 CoAP

에서 정의된 GET 명령을 100번 송신하여 PC에 응답이 도착할 때까지의 시간을 측정했다. 메시지의 크기는 1400 바이트 이고 IPv4, IPv6를 사용하여 각각 100번씩 2회 테스트를 수행했다. 통신 프로토콜의 구조는 Fig. 9와 같다.

IPv4로 측정한 결과는 Fig. 10과 같고 평균 65 ms, 최대 1000 ms, 최소 5.8 ms로 측정되었다. 여기서도 일정한 간격으로 높은 값이 측정되었다. 이 값들을 제외하면 평균 지연시간은 45 ms로 T_{RTT} 에서 $2D_{PR}$ 이 차지하는 비중은 대략 10% 정도임을 알 수 있다.

IPv6로 측정한 결과는 평균 59ms, 최대 616ms, 5.8ms였다. 여기서도 무선 통신 중의 데이터 충돌로 추정되는 최댓값은 3회 발생했다. 그러나 측정된 이들 최댓값이 낮아 IPv4보다 평균 지연시간이 낮게 나타났다. 이러한 비정상적 최대 지연시간들을 제외하였을 경우 IPv4와 IPv6를 사용한 실험의 평균 RTT는 큰 차이가 없었다.

$2D_{RK} + D_{sensing}$ 의 측정은 라즈베리파이2에서 Knote로 하여금 조도 센서 값을 읽어 전송하도록 요청하여 응답이 도착할 때까지의 시간을 측정했다. 두 시스템 사이의 통신은 IEEE 802.15.4 통신규약을 따라 수행하였다. 측정 결과는 Fig. 11과 같고 평균 지연시간은 360 ms, 최대 지연 368 ms, 최소 지연 355 ms였다.

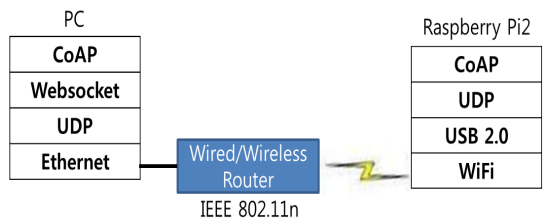


Fig. 9. Testbed configuration to measure $2D_{PR}$.

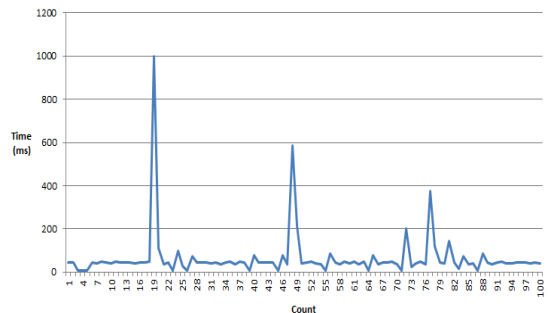


Fig. 10. The results of $2D_{PR}$ measurement (IPv4).

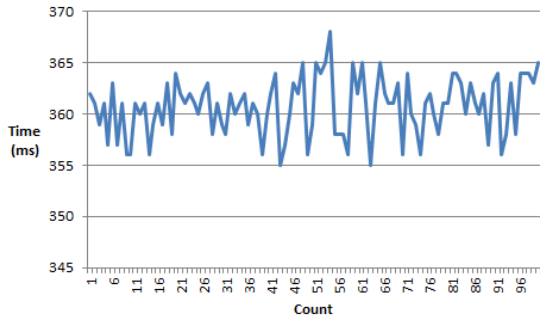


Fig. 11. The results of $2D_{RK} + D_{sensing}$ measurement.

평균 지연시간 대비 최대 지연시간과 최소 지연의 편차가 각각 8 ms, 5 ms로 대단히 낮음을 알 수 있다. 즉, $2D_{RK} + D_{sensing}$ 값은 T_{RTT} 에서 차지하는 비중은 크나 앞의 RTT 측정에서 나타난 전송 지연 비정상 최대치들의 원인은 아님을 알 수 있다.

또한, IEEE 802.15.4 통신 지연을 확인하기 위해 $2D_{RK}$ 를 측정하였다. $2D_{RK}$ 는 라즈베리파이2에서 프레임 전송하고 K mote에서 수신 즉시 에코 기능을 이용해 라즈베리파이2로 응답을 보내 응답이 수신될 때까지 걸린 시간을 측정하였다. 앞의 실험을 포함하여 이 실험의 측정 결과를 Table 1에 요약하였다. 이외에도 Fig. 9의 구성에서 유무선 공유기와 라즈베리파이2 사이의 통신을 유선으로 바꾸어 실험을 진행했다. 측정 결과는 평균 0.73 ms, 최대 0.88 ms, 최소 0.61 ms로 최대 1 ms를 넘지 않는 것으로 나타났다.

4.3 결과 분석

이상의 실험에서 Wi-Fi 무선 통신이 평균치를 상회하는 RTT 값의 주요 원인을 확인하였다. 이때의 이상 지연 데이터 값은 무선 데이터의 충돌이 원인으로 파악된다. 즉, IoT 노드의 주변에 Wi-Fi 네트워크를 사용하는 기기가 많을 경우 심각한 통신지연을 야기할 수 있으므로 사용 무선채널을 트래픽 상황을 감지하여 능동적으로 변경하는 것이 필요하다. 전체 RTT 값을 살펴보면 IEEE 802.15.4 통신과 조도센

서 값을 읽어 처리하는 시간이 대부분을 차지하였다. 이는 센서 네트워크 노드의 컴퓨팅 능력과 자원의 제약성에서 기인하므로 센서 노드 플랫폼에 대한 개선이 있어야 감소될 것이다.

조도 센서값을 읽어 처리하는 시간은 약 312 ms로 가장 큰 지연요소였다. 이 값은 Table 1에서 $D_{sensing} = (2D_{RK} + D_{sensing}) - 2D_{RK} = 312.2$ ms로 구해진다. 이 값을 Table 1의 T_{RTT} 값에서 제하면 CoAP 통신지연은 약 186 ms가 된다. Fig. 11에서 보듯 $2D_{RK} + D_{sensing}$ 와 $D_{sensing}$ 값은 편차가 크지 않아 비교적 정확한 값이라 할 수 있다. $D_{sensing}$ 값은 K mote의 센서를 작동시켜 조도를 측정하여 가져오는 것으로 시스템의 통신과정과는 무관하다. 또한, 4.1절에서 기술한 바와 같이 IPv6의 경우 T_{RTT} 값은 충돌로 인한 이상 데이터 3개를 제외한 평균은 454 ms이다. 이 경우 CoAP 통신지연은 142 ms이다. 이 평균치들을 [20]에서의 평균 지연 184 ms와 비교하면 이상 데이터를 포함한 경우는 비슷한 성능을, 제외한 경우는 약 23%의 성능 개선을 보여준다. 앞에서 4번째로 수행된 HTTP와 TCP 조합의 기존 웹 통신과 비교하면 HTTP의 경우 CoAP보다 통신지연이 8.4배이다. 따라서 사물인터넷 장비에서는 자원의 제약성 외에 통신지연 측면에서도 CoAP의 사용이 불가피함을 확인할 수 있다.

5. 결론

‘Web presence’는 IoT의 확산을 위해서 중요한 요소가 된다. IETF에서는 이를 위해 CoAP 표준을 제시하였다. 본 논문에서는 CoAP 표준에 대해 소개하고 관련 연구들을 요약하였다. 스마트 그리드에의 적용을 위해 PC, 내장형 시스템, 무선 센서 노드를 이용하여 시스템을 구축하고 공개 소프트웨어를 활용하여 CoAP를 구현하였다. CoAP의 GET 명령을 이용하여 웹 서버 기능을 거쳐 무선 센서 노드의 센서값을 읽어오는 과정의 RTT를 측정하여 구현된 시스템의 성능평가를 수행하였다. 이 과정에서 발견된 데이

Table 1. Summary of RTT test results

(unit: ms)

	T_{RTT_W}	T_{RTT}	$2D_{PR}$	$2D_{RK} + D_{sensing}$	$2D_{RK}$
Average time	659	498	59	360	47.8
Max. time	1544	1700	616	368	63.6
Min. time	570	383	5.8	355	42.0

터 충돌현상의 발생처를 파악하기 위해 통신 구간별 RTT를 측정하여 WiFi 구간이 원인임을 확인하였다. 이상 데이터를 제외한 경우 CoAP 통신 지연에서 약 23%의 성능향상을 보였다. CoAP 기반의 IoT 통신은 시간의 단축, 멀티캐스트 지원 등의 장점이 있다. IoT 장비의 적은 자원과 낮은 성능에서 웹 기능의 지원을 위해 CoAP는 필수적이거나 HTTP 기능의 부분집합이라는 점에서 전 기능을 지원하지 못하는 한계가 있다. 또한, TCP의 흐름제어 기능을 갖추지 못함에 따라 통신의 신뢰성은 응용 프로그램에서 구현해야 하는 단점이 있다. 향후, 본 연구결과의 활용 방안은 다음과 같다.

ESS (Energy Storage System)는 마이크로그리드에서 주요한 구성 요소중 하나이다. 즉, 태양열, 풍력 발전 등을 이용하여 생산된 전력을 ESS에 저장, 활용하고 저장된 전력이 일정 수준이하로 내려가면 상용전력으로 전환하여 사용하는 등의 운용과정에서 발전설비와 전력소비 설비, 계통설비 등의 상태에 대한 양방향 통신이 요구된다. 여기서 전환과정 및 각 설비간 통신 시 IoT 통신이 활용된다. ESS 전원과 상용전원 사이의 전환 등의 명령 전달, 상태 모니터링 등에서 IoT 통신의 지연의 최대치를 바탕으로 가정, 사무용 건물, 공장 등의 응용 시 전환 시기, 모니터링 주기 등을 설정해야 한다. 또한 이러한 통신은 유선, WiFi, IEEE 802.15.4 등의 방법을 모두 사용하므로 본 연구 결과는 ESS와 풍력, 태양력 발전 설비가 설치될 마이크로 그리드 환경에서 전환 기준, 감시기준으로써 응용 프로그램에 활용될 수 있다. 마이크로그리드 환경에서 ESS를 설치하여 CoAP 기반 IoT 기능을 구현할 경우 이상의 실험 결과를 고려하여 설계하는 것이 필요하다.

REFERENCE

- [1] Cisco, *Cisco Visual Networking Indexcomplete Forecast for 2015 to 2020*, Technical Report, 2016.
- [2] INFSO, *Internet of Things in 2020, Roadmap for the Future*, EU Commission Information Society and Media, V.1.1, 2008.
- [3] M.R. Palattella et. al., "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Communications Surveys and Tutorials*, Vol. 15, No. 3, pp. 1389-1406, 2013.
- [4] R. Want, B.N. Schilit, and S. Jenson, "Enabling the Internet of Things," *IEEE Computer*, Vol. 48, No. 1, pp. 28-35, 2015.
- [5] IETF, *Transmission of IPv6 Packets Over IEEE 802.15.4 Networks*, RFC 4944, 2007.
- [6] IETF, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, IETF, RFC 6550, 2012.
- [7] IETF, *Constrained Application Protocol*, IETF CoRE Working Group, 2011.
- [8] D. Thangavel, X. Ma, A. Valera, H.X. Tan, and C.K.Y. Tan, "Performance Evaluation of MQTT and CoAP via a Common Middleware," *Proceeding of 2014 IEEE Ninth Int'l Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 1-6, 2014.
- [9] J. Kim, S.K. Kang, and J. Park, "Bluetooth-based Tree Topology Network for Wireless Industrial Applications," *Proceeding of International Conference on Control, Automation and Systems*, pp. 1305-1308, 2015.
- [10] S. Jung and H.S. Lee, "Implementation of Internet-based IoT Environment for Multiple Robot Systems," *Proceeding of 2016 31st Institution of Control, Robotics and Systems Annual Conference*, pp. 23-25, 2016.
- [11] A.T. Nguyen, T. Hoang, Q.V. Thai, and T.T. Q. Bui, "Design of an Advanced Wearable Sensor Platform for Multiple Applications," *Proceeding of International Conference on Control, Automation and Systems*, pp. 1703-1708, 2015.
- [12] C. Perera, C.H. Liu, S. Jayawardena, and M. Chen, "A Survey on Internet of Things from Industrial Market Perspective," *IEEE Access*, Vol. 2, pp. 1660-1679, 2015.
- [13] 50 Sensor Applications for Smart World, http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/ (accessed Aug., 20, 2016).

- [14] Z. Sheng, S. Yang, Y Yu, A. Vasilakos, J Mccann, and K. Leung, "A Survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities," *IEEE Wireless Communications*, Vol. 20, No. 6, pp. 91-98, 2013.
- [15] R. Fantacci, T. Pecorella, R. Viti, and C. Carlini, "A Network Architecture Solution for Efficient IoT WSN Backhauling: Challenges and Opportunities," *IEEE Wireless Communications*, Vol. 21, No. 4, pp. 113-119, 2014.
- [16] B.C. Villaverde, D. Pesch, R. D. P. Alberola, S. Fedor, and M. Boubekeur, "Constrained Application Protocol for Low Power Embedded Networks: a Survey," *Proceeding of 2012 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 702-707, 2012.
- [17] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. M. Leung, "Recent Advances in Industrial Wireless Sensor Networks toward Efficient Management in IoT," *IEEE Access*, Vol. 3, pp. 622- 637, 2015.
- [18] A. Rajandekar and B. Sikdar, "A Survey of MAC Layer Issues and Protocols for Machine-to-Machine Communications," *IEEE Internet of Things Journal*, Vol. 2, No. 2, pp. 175-186, 2015.
- [19] B.X. Yen, W. Lee, Y. Kim, and M. Yoo, "Performance Evaluation of Wireless Networked Control System Based on IEEE 802.15.4e with Redundancy," *Journal of Korean Institute of Communications and Information Sciences*, Vol. 38B, No. 7, pp. 572-580, 2013.
- [20] W. Colitti, K. Steenhaut, N. D. Caro, B. Buta, and V. Dobrota, "Evaluation of Constrained Application Protocol for Wireless Sensor Networks," *Proceeding of 18th IEEE Workshop On Local and Metropolitan Area Networks*, pp. 1-6, 2011.
- [21] C.P. Kruger and G.P. Hancke, "Benchmarking Internet of Things Devices," *Proceeding of 12th IEEE International Conference on Industrial Informatics*, pp. 611-616, 2014.
- [22] B. Konieczek, M. Rethfeldt, F. Golatowski, and D. Timmermann, "Real-Time Communication for the Internet of Things Using jCoAP," *Proceeding of 2015 IEEE 18th International Symposium on Real-Time Distributed Computing*, pp. 134-141, 2015.
- [23] J. Skodzick, et al., "CoHaRT: a P2P-based Deterministic Transmission of Large Data Amounts Using CoAP," *Proceeding of 2015 IEEE International Conference on Industrial Technology*, pp. 1851-1856, 2015.
- [24] R. Bhalerao, S.S. Subramanian, and J. Pasquale, "An Analysis and Improvement of Congestion Control in the CoAP Internet-of-Things Protocol," *Proceeding of 2016 13th IEEE Annual Consumer Communications and Networking Conference*, pp. 9-12, 2016.



추 영 열

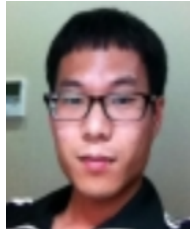
1986년 서울대 제어계측공학과 졸업
1988년 동 대학원 석사
2002년 포항공대 컴퓨터공학과 박사
1988년 1994년 포항산업과학기술연구원 선임연구원

1994년~2002년 포스코 기술연구소 책임연구원
2002년~현재 동명대학교 컴퓨터공학과 교수
2005년 독일 Fraunhofer IESE Visiting Scientist
2006년~2012년 유비쿼터스 항만ITRC 센터장
2015년 2월~2016년 1월 미국 SIUE Visiting Scholar
2010년 12월~2016년 12월 부산정보산업진흥원 이사
관심분야: 컴퓨터 네트워크, WSN, Ambient Intelligence, 공장자동화, 네트워크 보안



하 용 준

2012년 동명대 컴퓨터공학과 졸업
2014년 동 대학원 석사
2014년~현재 동 대학원 박사과정
관심분야: WSN, Routing, IoT.



손 수 동

2013년 동명대 컴퓨터공학과 졸업
2015년~현재 동 대학원 석사과정
관심분야: IoT, CoAP.