

IoT 기반 날씨 측정 서비스

김정주 · 김희진 · 김동균 · 이상정

순천향대학교

Weather Measurement Service Based on IoT

Jeong-Ju Kim · Hee-Jin Kim · Dong-Kyun Kim, Sang-Jeong Lee

Soonchunhyang University

E-mail : rmadydlfdps@sch.ac.kr · huijin@sch.ac.kr · kdk70@sch.ac.kr · sjlee@sch.ac.kr

요약

최근 기후 변화에 여러 이상 현상들이 발생하면서 전 세계적으로 급격히 변화하는 기상 요인들에 이목이 집중되고 있다. 날씨의 변화는 현대인들의 삶에 많은 영향을 끼치고 있으며, 특정 지역의 국소 영역에서의 날씨 정보 측정에 관심이 증대되고 있다.

본 논문에서는 특정 지역의 날씨의 변화로 인한 피해를 줄이기 위해 사물 인터넷 환경에서 실시간으로 급격히 변화하는 다양한 날씨 정보를 측정하는 IoT 기반 날씨 측정 서비스를 설계 구현한다. 개발되는 날씨 측정 서비스는 IoT를 위한 CoAP 프로토콜을 적용하고, 날씨 정보의 효율적인 활용과 관리가 가능하도록 구현함으로써 사용자들이 웹을 이용하여 차트로 표현된 날씨 정보를 실시간 모니터링 할 수 있도록 개발한다.

키워드

날씨 모니터링, IoT, CoAP 프로토콜, 센서 네트워크

1. 서론

최근 기후 변화에 여러 이상 현상들이 발생하면서 전 세계적으로 급격히 변화하는 날씨 요인들에 이목이 집중되고 있다. 날씨의 변화는 현대인들의 삶에 많은 영향을 끼치고 있으며, 좁은 범위에서는 개개인의 컨디션에, 넓은 범위에서는 자연재해로 인한 국가적인 피해까지 넓은 범위에 걸쳐 영향을 미친다. 이에 따라 기상청에서는 전국에 기상 기지국을 설치하고 변화하는 날씨 요인의 변화를 연구하고 있다. 그러나 전국에

설치된 기상 기지국을 유지하는데 많은 비용이 들고, 대부분 광역 지역 단위로 날씨를 측정하기 때문에 특정 국소 지역의 정확한 날씨 정보의 획득이 어려운 실정이다.

또한 컴퓨터뿐만 아니라 다양한 사물, 데이터 및 서비스를 네트워크로 연결하여 상호 간에 정보를 교환하는 IoT(Internet of Things) 기술이 확산되고 있다. IoT를 기반으로 하는 날씨 측정 시스템은 저비용으로 개발 배치가 가능하여 기존의 광역 기상 기지국보다 더 세밀한 지역의 날씨정보까지

정확히 측정할 수 있다.

본 논문에서는 특정 국소 지역의 세밀한 날씨 정보를 측정하고 모니터링하는 IoT 기반 날씨 측정 서비스를 제공하는 시스템을 설계 구현한다. 구현 개발되는 날씨 측정 서비스는 IoT의 제한적 네트워크 환경을 위한 CoAP 프로토콜 상에서 실시간으로 사용자에게 관심지역의 온도, 습도, 풍향/풍속, 미세먼지 등의 날씨 정보를 제공한다.

II. CoAP 프로토콜

IETF CoRE 워킹그룹의 CoAP (Constrained Application Protocol) 프로토콜은 저 전력, 고 손실, 네트워크 및 소용량, 소형 노드에 사용될 수 있는 특징을 가진 웹 전송 프로토콜이다 [1][2]. HTTP에서 사용하던 REST 서비스의 특성을 따르고 있어 쉽게 접근이 가능하고, 재전송 및 타이머 관리 옵션 등을 포함하여 UDP와 같은 트랜스포트 계층 위에서 비동기적으로 전송을 하여 신뢰성 있는 전달한다. CoAP 프로토콜은 저성능의 CPU와 작은 용량의 메모리를 갖는 노드에서의 데이터 전송을 축약된 메시지 헤더와 옵션 헤더를 갖는다. CoAP 메시지는 요청과 응답의 상호작용으로 전달이 되며 그림 1과 같이 간단한 바이너리 포맷으로 인코딩 된다.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------------|---|---|---|---|---|-----|---|---|------|----------------------|---|---|---|---|---|---|---|---|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| V | | | T | | | TKL | | | Code | | | | | | | | | | Message ID | | | | | | | | | | | | | | | | | | | | |
| Token(if any, TKL bytes) ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Options (if any) ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 1 1 1 1 1 1 1 | | | | | | | | | | Payload (if any) ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

그림 1. CoAP 메시지 포맷

CoAP 프로토콜 헤더에는 버전을 나타내는 V, 메시지 타입을 나타내는 T, 토큰필드

의 길이를 나타내는 TKL, 정확한 메시지가 왔는지 중복확인을 위한 Message ID, Code가 있다. 메시지 타입에는 확인형 (Confirmable), 비확인형 (Non-confirmable), 승인 (Acknowledgement), 리셋 (Reset) 총 4가지를 정의하고 있다. Code는 클래스와 상세정보로 이루어져 있으며, c.dd와 같은 형태로 표현이 된다. 클래스는 요청(0), 성공적인 응답(2), 클라이언트 에러 응답(4), 서버 에러 응답(5)을 의미하고, 요청의 경우 0.01은 GET, 0.02는 POST, 0.03은 PUT, 0.04는 DELETE를 나타내며 0.00은 빈 메시지를 나타낸다 [1].

III. IoT 기반 날씨 측정 시스템

3.1 시스템 구성

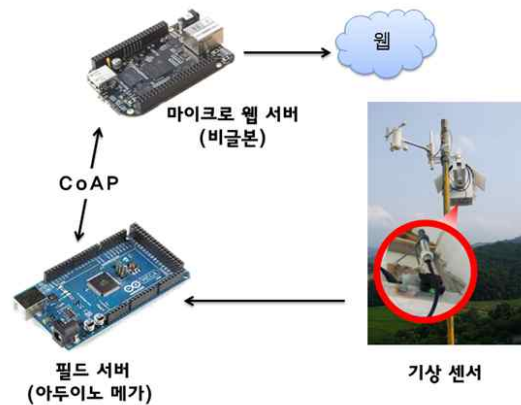


그림 2. 날씨 측정 시스템 구성도

본 논문의 날씨 측정 시스템은 그림 2와 같이 날씨 센서와 필드 서버로 구성되고, 날씨 요인들을 측정하기 위해 온·습도, 풍향·풍속, 미세먼지 농도 센서를 사용하였다. 각 날씨 센서는 옥외에 백엽상을 설치하여 백엽상 내부에 부착하였다. 날씨 센서의 데

IoT 기반 기상 측정 서비스

이터를 수집하는 필드 서버는 아두이노 메가 보드를 사용하였으며, 필드서버에서 날씨 센서 데이터를 웹으로 전송하기 위해 비클본 블랙을 웹 서버로 사용하였다. 각 임베디드 보드의 사양은 표 1과 같다. 웹 서버에서는 전달받은 날씨 센서 데이터의 변화를 웹 페이지에서 차트로 표시 하였다.

표 1. 임베디드 보드 주요사양

| | | |
|---|---------------------|--|
| ① | Arduino Mega | - Microcontroller : ATmega2560 - Input Voltage : 7-12V - Rev C |
| ② | Beaglebone Black | - CPU : AM335x 1GHz ARM® Cortex-A8 - ARM : 512 MB DDR3L |

3.2 CoAP 프로토콜 통신

각 날씨 센서와 필드서버 사이의 통신은 아두이노용 RS-485모듈을 이용하였으며, 웹 서버와 필드 서버 사이의 통신은 CoAP 프로토콜을 사용하였다. 필드서버와 웹 서버 사이의 통신에서 필드서버는 microCoAP [4]를, 웹 서버는 libCoAP [5]를 사용하여 CoAP 프로토콜을 구현하였다. 필드 서버는 CoAP 서버가 되며, 웹 서버는 CoAP 클라이언트가 된다.

그림 3은 웹 서버로부터의 온도 센서 자료의 요청과 응답을 표시하는 메시지의 흐름이다. 사용자 브라우저의 요청에 따라 웹 서버는 GET 요청으로 필드 서버에 저장된 온도 자료의 데이터를 요청하게 되고, 필드 서버는 날씨 센서의 데이터를 수집하여 요청 메시지에 따른 응답 메시지를 작성하여 전송한다. 웹 서버는 수신된 날씨 정보 데이터를 클라이언트의 사용자 브라우저에 차트로 표시한다.

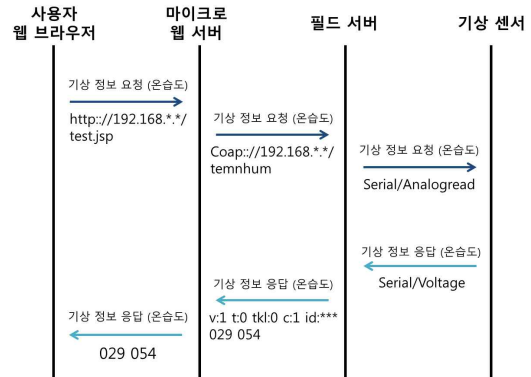


그림 3. 자원 요청에 따른 메시지 흐름도 (온·습도 예)

```
ubuntu@arm:~/libcoap-4.1.1/examples$
./coap-client -m get coap://192.168.100.3/temnhum
v:1 t:0 tk:0 c:1 id:65498
029 054
```

그림 4. 웹 서버의 온·습도 자원 요청 및 결과

그림 4는 실제 웹 서버 상에서 필드 서버에게 온·습도 센서 자료의 데이터를 요청한 결과이며, 029 054는 온·습도 자원이 가지고 있는 실시간 데이터를 출력한 것이다. 029는 현재 외부의 온도를 출력한 것이며, 054는 현재 외부의 습도를 출력한 것이다.

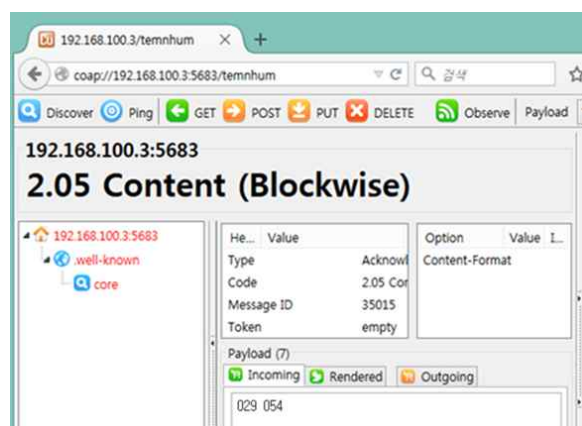


그림 5. Firefox Cooper를 이용한 결과 검증

그림 5는 웹 서버로 수신된 데이터의 값이 필드 서버에서 보낸 데이터와 동일한지 검증하기 위해 Mozilla Firefox의 CoAP 플러그인인 Cooper를 이용해 확인한 결과이다. 확인 결과 payload의 값은 온도 29℃, 습도 54%로 동일한 결과를 보인다.

날씨 측정 서비스를 국소 관심 지역에 배치하면 저비용으로 날씨 정보의 효율적인 활용과 관리가 가능하다.

참고문헌

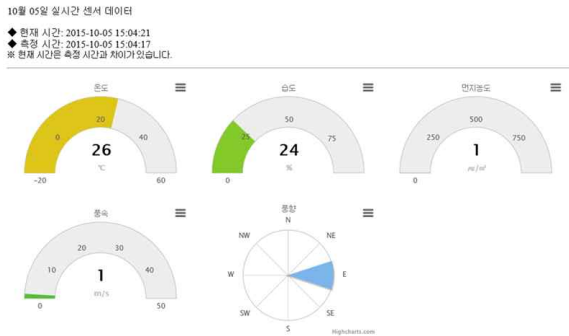


그림 6. 날씨 정보 데이터 차트

웹 서버의 요청에 따라 실시간으로 수집된 날씨 데이터는 그림 6과 같이 JQuery를 기반으로 한 HighCharts [6]를 이용하여 웹에서의 차트 모니터링을 수행한다. 사용자는 웹 브라우저를 통하여 날씨 데이터의 변화를 실시간으로 모니터링 할 수 있다.

IV. 결 론

본 논문에서는 특정 국소 지역의 세밀한 날씨 정보를 측정하고 모니터링하는 IoT 기반 날씨 측정 서비스를 제공하는 시스템을 설계 구현하였다. 개발된 시스템은 날씨 센서와 필드 서버로 구성되고, 날씨 요인들을 측정하기 위해 온·습도, 풍향·풍속, 미세먼지 농도 센서를 사용하였다. 필드서버와 웹서버사이의 통신은 웹 기반 IoT 서비스를 위해 제한적 네트워크 환경을 위한 CoAP 프로토콜을 적용하여 구현하였다.

본 논문에서 제안된 IoT를 기반으로 한

- [1] Shelby, Zach, et al. "The Constrained Application Protocol (CoAP)", IETF RFC 7252, Jun. 2014.
- [2] 고석갑, 박일균, 손승철, 이병탁, "IETF CoAP 기반 센서 접속 프로토콜 기술 동향", ETRI, 전자통신동향분석, 2013년 12월
- [3] Lina Yao, Quan Z. Sheng, Schahram Dustdar, "Web-Based Management of the Internet of Things", IEEE Computer Society, p.10-17 Jan./Feb. 2015
- [4] microcoap, GitHub (<https://github.com/1248/microcoap>)
- [5] libcoap, C-Implementation of CoAP, Sourceforge(<http://sourceforge.net/projects/libcoap/>)
- [6] Highcharts, Solid gauge, highcharts (<http://www.highcharts.com/demo/gauge-solid>)