논문 20-45-04-10 The Journal of Korean Institute of Communications and Information Sciences '20-04 Vol.45 No.04
https://doi.org/10.7840/kics.2020.45.4.691

LPWAN 기반 IoT 재난 경보 시스템의 개선

유 명 한*, 김 상 경°

Improvement of IoT Disaster Alerting System Based on LPWAN

Myung-han Yu*, Sangkyung Kim°

요 약

기존의 LPWAN 기반 IoT 원격 측정 시스템을 재난 경보 시스템으로 활용할 경우, 센서로부터 수집된 데이터는 게이트웨이에서 네트워크 서버 및 어플리케이션 서버를 거쳐 재난경보 서버 및 모니터링 클라이언트로 송신된다. 따라서 경고 발신에 신속성을 요구하는 재난 경보 분야에 이용되기에는 약점이 존재했다. 본 논문에서는 경고 발신의 신속성을 확보하기 위해 게이트웨이가 LPWAN 패킷의 페이로드를 검사하여 긴급상황 경고 데이터에 한해 재난경보 서버와 모니터링 클라이언트로 직접 송신하는 재난경보 시스템을 제안하였다. 이에 대해 실제 환경에서의 적용가능성을 확인하기 위해 제안 시스템을 구현하였으며, 성능평가를 수행하였다.

키워드 : LPWAN, 사물인터넷, 재난 경보 시스템, Common Alerting Protocol, TinyAP
Key Words : LPWAN, IoT, Disaster Alerting System, Common Alerting Protocol, TinyAP

ABSTRACT

When using an existing LPWAN-based IoT telemetry system as a disaster alerting system, the data collected from the sensor is sent from the gateway through the network server and application server to the disaster alerting server and monitoring client. Thus, there was a weakness to be used in the field of disaster alerts requiring promptness to send alerts. In this paper, we propose a disaster alerting system that has a structure in which the gateway checks the payload of LPWAN packets and sends the emergency alert data directly to the disaster alerting server and monitoring client. In order to confirm the applicability in the real environment, the proposed system was implemented and performance evaluation was performed.

Ⅰ. 서 론

기존의 LPWAN 기반 IoT 원격 측정 시스템은 서버-클라이언트 구조를 가지며[1,2] UI에 표현하기 위한 각종 데이터를 HTTP REST 를 비롯한 여러 응용 프로토콜을 바탕으로 수신한다. 이러한 시스템 구조는 실시간 정보 전달 능력에 있어 한계를 가질 수 있으며, 특히 초 단위의 시간을 다투는 재난 상황, 그리고 이를 최대한 신속하게 알려야 하는 재난경보 분야에서는 개선이 필요하다.

대표적인 LPWAN인 LoRaWAN[3]의 경우, 기존의 원격 측정 시스템들은 현장의 센서로부터 게이트웨이(Gateway)로 수집된 데이터가 인터넷을 통해 네트워크 서버로 보내지고, 네트워크 서버는 데이터를 어플

◆ First Author : HighTech, greatymh@gmail.com, 선임연구원, 정회원
° Corresponding Author : Gangneung Wonju National Univ., Dept. of Computer Science & Engineering, skkim98@gwnu.ac.kr, 정교수, 종신회원
논문번호 : 201912-345-B-RN, Received December 21, 2019; Revised February 6, 2020; Accepted February 10, 2020

리케이션 서버로 보내며, 어플리케이션 서버에서는 경보 전파가 필요한 상황인지를 판단한 후, 최종적으로 재난 경보를 발령하는 구조를 가졌다.

본 논문에서 제안하는 시스템은 LPWAN 패킷의 페이로드를 검사하여 긴급상황 경고 데이터에 한해 네트워크/어플리케이션 서버와의 연동 이전 단계, 즉 게이트웨이에서 재난 경보 서버와 모니터링 클라이언트로 직접 송신하는 구조를 가짐으로써 보다 빠른 시간 내에 경보를 전달할 수 있다. 제안 시스템은 추가적인 시스템 증설에 관한 부담 없이도 효과적인 재난 경보 시스템을 구축할 수 있다.

본 논문의 순서는 다음과 같다. 2장에서 관련된 선행연구에 대해 검토하고 3장에서 모니터링 시스템의 구조와 구현사항에 대해 상세히 설명한다. 4장에서 시스템을 시험하여 그 성능이 개선되었음을 확인 및 검증한다. 5장에서 결론을 맺는다.

Ⅱ. 관련연구

2.1 재난 경보 프로토콜

긴급상황의 전파에 필요한 정보전달 방법을 통일하기 위해 CAP (Common Alerting Protocol)[4]가 표준화되었으며 최근 들어, 이를 바탕으로 한 다양한 응용과 연구가 진행되고 있다.

[1]에서는 IoT 기반의 위험 모니터링 시스템을 제안하였다. MySQL 데이터베이스와 MQTT 프로토콜에 기반한 모니터링 시스템은 전통적인 재난 경보 시스템의 구조를 설명하고 있다. 이는 데이터베이스에 저장된 정보를 웹페이지를 통해 표시하며, 별도로 이메일로도 통보하는 방법을 사용한다. 이러한 시스템은 전통적인 IoT 모니터링 시스템의 한계를 나타낸다. 센서로부터 게이트웨이로 전달된 데이터는 원격지의 서버를 거쳐 데이터베이스에 저장된다. 모니터링 시스템은 주기적으로 REST API 등의 방법을 통해 데이터를 가져와 표시하거나 서버 측에서 이메일 또는 SMS, CAP 등의 방법으로 경고상황을 전파하게 된다. 이는 신속한 경보 전파 목적으로 이용하기 힘들다.

[5]에서는 CAP에 기반하여 확장된 형식의 프로토콜인 CAP-PER로 재난경보를 발령하는 방법에 대해 구현하고 실제 필드에서 시험하였다. 일반적으로 국내에서 사용되는 휴대전화 문자 발신 방법이 아닌 RDS(Radio Data System)을 이용하여 구현하였으며 사람이 개입하여 경보를 발생하는 시스템이며 원격측정 시스템을 기반으로 사람 없이 자동으로 경보를 발령하는 IoT 시스템의 구조와는 차이가 있다.

[6]에서는 디지털 셋톱박스(STB)에서 CAP를 구현하고 수신하는 능력을 시험하였다. T-DMB 기반으로 CAP 데이터를 수신하여 만약 셋톱박스가 꺼져 있더라도 백그라운드에서 대기하다가 메시지 수신 시 빠르게 깨어나 사용자에게 알림을 전달하는 구조를 채택하였다.

이외에도 다양한 구현 사례가 존재할 수 있으며, 본 논문에서는 LPWAN의 일종인 LoRaWAN 기반 IoT 시스템을 바탕으로 위험 감지 시 자동으로 재난 방송을 송출하는 원격 측정 시스템에 대하여 구현한다.

2.2 LPWAN 기반 IoT 원격 측정 시스템

기존의 LPWAN 기반 원격측정 시스템의 경우 신속한 재난정보 전파를 위해서는 다음과 같은 부분이 약점으로 판단된다.

우선 센서로부터 데이터가 전송되어 게이트웨이를 거쳐 서버로 수집되는 구조의 문제이다. LoRaWAN의 경우 센서에서 수신되는 모든 데이터는 일차적으로 업링크(Uplink)로 원격지 네트워크 서버에 전달된다. 네트워크 서버에서 인증값 검사와 데이터 형식 등에 대하여 확인 후 올바른 데이터일 경우 어플리케이션 서버로 전달하며, 게이트웨이로 수신 완료 응답을 다운링크(Downlink)로 내려보내주는 구조이다[3]. 이러한 구조는 송/수신 과정에서 CRC 오류가 발생한 패킷의 재전송 절차 외에도 게이트웨이와 네트워크/어플리케이션 서버 사이의 통신 품질에 따라 소요되는 시간이 늘어날 수 있다.

[7]에 따르면 게이트웨이와 네트워크 서버가 100Base-T 이더넷으로 연결된 환경에서 게이트웨이에서 송신한 데이터가 네트워크 서버를 거쳐 어플리케이션 서버에 도달하는 데 걸리는 평균 소요시간은 약 0.624초로 나타났다. 이는 이더넷 대신 LTE 등 무선통신망의 이용, 게이트웨이와 네트워크/어플리케이션 서버간의 데이터 전송 프로토콜의 종류 등에 따라 1초 이상까지 늘어날 수 있다.

또한 경보를 전파하는 과정에서도 [1-2]와 같은 시스템은 우선 어플리케이션 서버가 데이터 수집 서버로 데이터를 전송하고 이를 데이터베이스에 저장한 뒤, 다시 이 정보를 모니터링 클라이언트가 주기적으로 수신받아 경보를 표출하는 구조를 가진다. 이는 추가적인 시간 지연의 원인이 될 수 있다.

본 논문에서는 게이트웨이가 수신된 센서 데이터의 임계치 범위를 초과하였는지 판단하여 경보를 발생해야 할 상황에는 CAP 서버로 직접 메시지를 전송하여 경보상황을 전파하게 된다. 이 때 전송되는 CAP 경보

정보의 형식 또한 텍스트 형태의 XML 포맷에 기반하므로 데이터의 양이 커지며, 전송 시간 지연의 원인이 되므로 TinyAP 프로토콜[9]에 기반한 경량 패킷 구조를 사용하였다.

Ⅲ. IoT 재난 경보 시스템의 구현

3.1 원격 측정 시스템의 구성

실시간 IoT 모니터링 시스템의 전체 구성은 그림 1과 같다. 구성요소는 크게 디바이스와 게이트웨이, 네트워크 서버(NS), 어플리케이션 서버(AS), 데이터 수집 서버 (CS), 모니터링 클라이언트, CAP 서버로 구성된다.

지역적으로 분산되어 있는 센서들은 각 지역의 게이트웨이에 연결된다. 게이트웨이는 LoRaWAN 게이트웨이를 기준으로 구현하였으며, CAP 서버와의 통신을 위한 서브프로세스가 실행된다. 이를 TinyAP PUSH 서비스로 명명하였다. 이는 Packet Forwarder[10]에서 발신하는 UDP 패킷을 캡처하여 재난상황을 자동으로 통보하는 역할을 수행한다.

어플리케이션 서버에는 게이트웨이의 TinyAP PUSH 서비스를 위한 임계치 및 범위 정보 등의 설정값을 제공할 수 있는 소켓 서버가 구성된다. 이를 TinyAP PUSH Helper 로 칭한다.

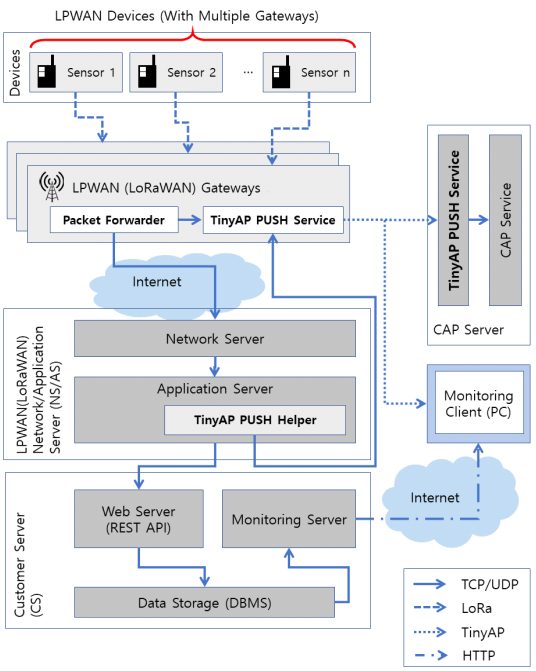

그림 1. IoT 재난 경보 시스템 구성도
Fig. 1. The structure of IoT disaster alerting system

CAP 서버와 게이트웨이간 직접 연결을 위해 별도의 VPN을 이용하여 보안 터널을 구축하며, 이는 하드웨어 또는 소프트웨어적인 방법으로 구성 가능하다. 이는 공공망을 위한 LoRaWAN 구축 시 권장되는 사항으로[8] 본 논문에서는 OpenVPN[11] 소프트웨어로 구현하였다.

CAP 서버는 실제 발송 단계에서 이동통신사와 XMPP 기반의 통신을 수행하도록 구성한다. 하나의 CAP 서버는 여러 개의 게이트웨이들과 통신을 수행할 수 있다. CAP 서버의 위치와 개수는 본 논문의 범위 밖이며, 이동통신사와의 연동 및 그 이후의 부분은 구현한 범위가 아니므로 논문의 내용에서 제외한다.

3.2 시스템 구현 사항

시스템 구현에 사용된 게이트웨이와 네트워크/어플리케이션 서버, 데이터 수집 서버, CAP 서버, 모니터링 클라이언트의 하드웨어 및 소프트웨어적인 제원을 표 1에 나타내었다.

표 1. 구현에 사용된 하드웨어 및 소프트웨어
Table 1. Hardware and software used in the implementation

Element	Specification
Gateway	Platform: Raspberry Pi 3B LoRaWAN Concentrator: SX1301 AP2 Language: Python 3 Packet Forwarder: Semtech UDP Packet Forwarder
Network Server	OS: CentOS 7 x64 Application: Chirpstack Network Server
Application Server	OS: CentOS 7 x64 Application: Chirpstack Application Server
Customer Server	OS: CentOS 7 x64 Language: PHP 6
CAP Server	OS: Windows 10 Language: PHP 5 Library: CAP PHP Library
Web Server, Monitoring Server	OS: CentOS 7 x64 Language: PHP 6 Database: MySQL
Monitoring Client(PC)	Platform: x64 OS: Windows 10 Web Browser: Chrome

3.2.1 게이트웨이의 센서 데이터 임계치 검사

Packet Forwarder는 UDP 포트로 데이터를 발신하며 이는 일반적으로 게이트웨이 내에서 루프백 인터페이스(lo)를 통해 Gateway Bridge 서비스[12] 등에 수신된다. 이러한 구조를 가지는 이유는 여러 Bridge 서비스에 대해 동시 다발적으로 데이터를 전송하기 위함으로 TinyAP PUSH 구현을 위한 센서 데이터의 수집 방법으로 해당 UDP 데이터를 활용한다.

Packet Forwarder가 전달하는 UDP 데이터의 JSON 구조를 그림 2에 나타내었다.

데이터는 신호 세기 및 모듈레이션 정보, 무선 채널 등의 부가정보를 담고 있으며, 긴급 여부를 확인하기 위해서는 센서가 보낸 페이로드에 해당하는 "data" 필드를 복호화하여 사용한다. 이는 base64로 인코딩된 형태이다.

TinyAP PUSH 서비스는 UDP를 통해 원격지 어플리케이션 서버로부터 경보 발생에 필요한 임계치의 정보 및 경보를 전달할 CAP 서버의 주소를 수신한다. 수신한 경보 임계치 관련 정보는 별도의 파일로 저장하며, 1시간 또는 30분 단위로 원격지 어플리케이션 서버와 다시 통신하여 갱신할 수 있다. 이는 네트워크의 크기와 포함된 센서의 종류 및 수량, 게이트웨이와 네트워크/어플리케이션 서버간의 대역폭, 전체 게이트웨이의 수에 따라 다르게 설정한다. 해당 데이터의 형식을 그림 3에 나타내었다.

게이트웨이는 저장한 설정파일의 정보를 바탕으로

```
{"rxpk":[{
   "time":"2013-03-31T16:21:17.528002Z",
   "tmst":3512348611,
   "chan":2,
   "rfch":0,
   "freq":866.349812,
   "stat":1,
   "modu":"LORA",
   "datr":"SF7BW125",
   "codr":"4/6",
   "rssi":-35,
   "lsnr":5.1,
   "size":32,
   "data":"-DS4CGaDCdG+48eJNM3Vai-zDpsR71Pn9CPA9uCON84"
   }]
}
```

그림 2. Packet Forwarder에서 수신되는 데이터 형식[10]
Fig. 2. Format of data received by Packet Forwarder[10]

```
{"config":{
  "cap_addr": "10.8.0.1", // Address of CAP Server
  "cap_port": 3000,       // UDP Port of CAP Server
  "underflow": -4,      // Minimum Sensor Data Value
  "overflow": 4         // Maximum Sensor Data Value
  }
}
```

그림 3. TinyAP PUSH 서버 환경설정 데이터 형식
Fig. 3. Data format for TinyAP PUSH server configuration

수신된 데이터의 경보 발생 여부를 지속적으로 확인하며 만약 경보가 발생하면 CAP 서버로 즉시 알림을 발송한다.

3.2.2 게이트웨이의 위치 정보 이용

LoRaWAN 게이트웨이의 규격 사항에는 GPS 등의 GNSS(Global Navigation Satellite System)를 통해 좌표정보와 시간정보를 수신하도록 규정되어 있다[13]. 이를 CAP를 위한 위치정보 파라미터로 이용할 수 있으므로 데이터를 수집하여 함께 전송한다. GNSS 위치 정보는 Packet Forwarder에서 수신되는 UDP 메시지에 포함되므로 이를 수신할 필요가 있다.

그림 4에 JSON 형태로 수신한 Packet Forwarder의 GNSS 위치 정보 형식을 나타내었다[10].

이를 CAP 메시지 포맷으로 표현하면 그림 5와 같다. <circle> 범위는 시스템상에서 미리 규정한 위치에 따르며, 해당 데이터는 서버측에서 센서 데이터 임계치를 수신할 때 함께 수신한다.

3.2.3 게이트웨이와 CAP 서버의 상호작용

게이트웨이와 CAP 서버간의 통신은 VPN을 통해 이루어진다. LTE 등 공공 인터넷망에서의 게이트웨이

```
<alert>
 <status>Actual</status>
 <msgType>Alert</msgType>
 <scope>Public</scope>
 <info>
  <area>
   <circle>46.24000, 3.25230 0</circle>
  </area>
 </info>
</alert>
```

그림 4. CAP 메시지 포맷
Fig. 4. CAP message format

```
{"stat":{
        "time":"2014-01-12 08:59:28 GMT",
        "lati":46.24000,
        "long":3.25230,
        "alti":145,
        "rxnb":2,
        "rxok":2,
        "rxfw":2,
        "ackr":100.0,
        "dwnb":2,
        "txnb":2
    }
}
```

그림 5. Packet Forwarder에서 수신되는 GNSS 위치 정보 형식[10]
Fig. 5. GNSS geolocation format received by Packet Forwarder[10]

운영을 위해서는 보안 연결이 필요하며 특히, 대부분의 경우 VPN을 이용하여 터널을 구성할 것을 권장하고 있다. 본 논문에서는 소프트웨어 VPN인 OpenVPN을 이용하여 VPN 서버와 클라이언트를 구현하였다.

게이트웨이에서 CAP 서버 측으로 전송에 사용되는 모든 데이터는 XML 포맷이 아닌 TinyAP 포맷을 이용한다. 이는 텍스트 기반인 XML 포맷 대신 약 20 바이트 크기를 가지는 TinyAP를 이용함으로써 데이터의 크기를 줄여 경보 전달 속도를 최대한 빠르게 하기 위함이다.

이를 위해 CAP 서버측에 TinyAP PUSH 수신용 프로세스를 구성하고 항시 실행한다. CAP 서버측의 TinyAP PUSH 메시지 수신용 프로세스는 수신한 시점에서 루프백 인터페이스(lo)를 통해 CAP 메시지를 전송한다. 이후, 이동통신사의 SMS 서비스 또는 기타 서비스 (T-DMB, FM 라디오 RDS 데이터, 전광판 등) 로 경보 신호가 발신될 수 있다.

TinyAP PUSH의 게이트웨이와 CAP 서버간의 메시지 흐름을 그림 6에 나타내었다.

TinyAP PUSH는 UDP에 기반하며, 전송 경로상 손실을 고려하여 3회를 연속으로 전송한다. 만약 ACK 메시지가 도달하지 않을 경우 다시 3회를 연속으로 전송하며, 전체 과정을 5회 반복한다. UDP를 사용하고 여러 번 중복하여 전송하는 이유는 보다 빠른 메시지 전달에 그 목적이 있으며, CAP 서버측에서는 중복되는 메시지가 여러 번 도착하는 것에 대비하여 최초로 도착한 패킷 이후 중복되는 동일한 시퀀스의 패킷에 대해서는 처리하지 않는다.

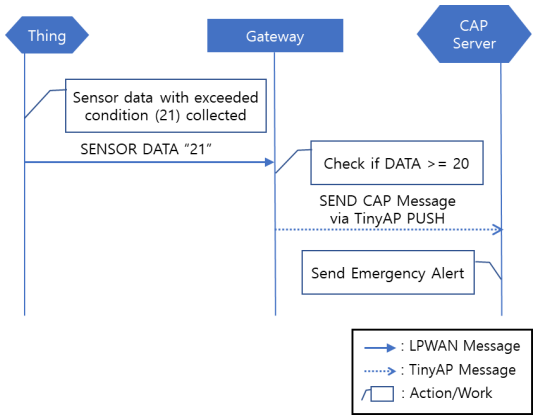

그림 6. TinyAP PUSH 메시지 흐름도
Fig. 6. TinyAP PUSH message flow

3.2.4 게이트웨이와 모니터링 클라이언트의 상호작용

게이트웨이에서 모니터링 클라이언트로의 연결은 CAP와 유사한 PUSH 구조를 바탕으로 동작한다. 메시지는 웹 소켓(web socket)[14]을 바탕으로 동작하며 TinyAP 포맷의 메시지를 게이트웨이가 전송함으로써 경보 정보를 모니터링 클라이언트와 공유한다. 모니터링 클라이언트와 게이트웨이간의 메시지 흐름도를 그림 7에 나타내었다.

모니터링 클라이언트는 각 게이트웨이의 소켓 주소를 미리 가지고 있어야 하며, 네트워크/어플리케이션 서버에서 HTTP를 통해 수신한다. 모니터링 클라이언트는 실행되는 동안 모든 게이트웨이와 웹 소켓 연결을 유지하여야 한다. 만약 경보해야 할 수준의 데이터가 수집되면 게이트웨이는 모니터링 클라이언트에 직접 메시지를 보내며, 각 클라이언트들은 이를 수신하

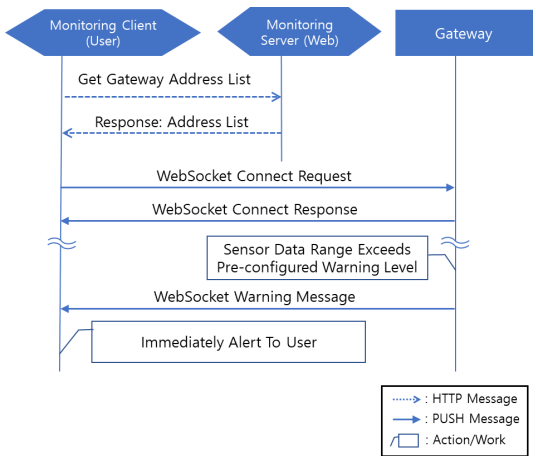

그림 7. 웹 소켓 메시지 흐름도
Fig. 7. Web socket message fow

는 즉시 사용자에게 경고한다.

3.2.5 모니터링 클라이언트 구현

모니터링 클라이언트의 성능을 확인하고 동작여부를 점검하기 위해 사용자 인터페이스를 구현하였다. 구현에는 자바스크립트를 사용하였으며, 웹 페이지 구성을 위해 Bootstrap 프레임워크와 Javascript 및 jQuery 라이브러리, PHP 언어를 사용하였다. 완성된 사용자 인터페이스의 스크린샷을 그림 8에 나타내었다.

모니터링 클라이언트는 화면에 센서의 위치와 최근 데이터를 나타낼 수 있으며, 각 게이트웨이와 웹소켓 연결을 유지하여 경보 발생 시 신속하게 사용자에게 알릴 수 있도록 구현하였다. 웹소켓은 연결이 끊어질 경우 자동으로 재연결할 수 있도록 구성하였다. 게이트웨이의 숫자가 많아질 경우 발생할 수 있는 성능저하에 대해 검토한 결과 실제 60개까지의 게이트웨이 동시 연결은 문제 없이 수용 가능한 것으로 나타났다.

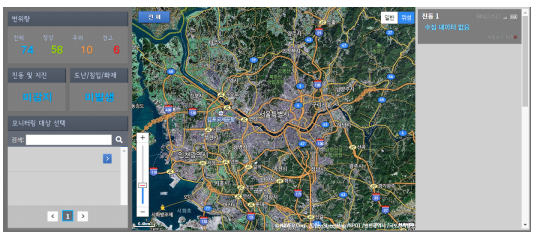

그림 8. 모니터링 클라이언트 사용자 인터페이스 스크린샷
Fig. 8. Monitoring client user interface screenshot

Ⅳ. 실험 및 검증

제안하는 시스템과 기존 시스템의 성능을 다음과 같이 평가하였다. 기존 시스템은 REST API를 기반으로 한 일반적인 모니터링 시스템을 가정하고 구현하였으며, MQTT를 통해 LoRaWAN 게이트웨이와 네트워크 서버가 연동되는 구조를 가진다. 제안하는 시스템은 이 구조에서 게이트웨이와 CAP 서버, 게이트웨이와 모니터링 클라이언트간에 VPN을 추가로 구성하고 직접 연결이 가능하도록 구성하였다.

4.1 CAP 경보 전파시간 평가

게이트웨이에서 센서로부터 수신한 데이터가 경보가 필요한 데이터일 경우를 가정하고, 센서가 데이터를 송신한 시간으로부터 CAP 발신이 이루어지기까지의 시간을 실제 환경에서 측정하였다. 실험은 LOS (Line-Of-Sight)를 기준으로 수행하였다. 1km 거리에 위치한 진동감지 센서가 게이트웨이로 진동발생 경보를 발신하도록 하였으며, 게이트웨이와 네트워크 서버 간은 LTE 통신망으로 연결되었다. 게이트웨이에서 CAP 서버 또한 동일한 LTE 통신망 하에 VPN을 이용하여 연결하였다. 모든 시스템의 시간은 NTP[15]를 통해 동기화하였다.

그림 9에 제안하는 시스템과 기존 시스템의 측정결과를 비교하여 나타내었다. X축은 시험 횟수를, Y축은 측정된 시간을 나타낸다.

실험 결과 제안하는 시스템의 전파시간이 기존 시스템보다 빠른 것으로 나타났다. 제안하는 시스템은 최대 243ms가 더 빨랐으며, 최소 169ms 더 빨랐다. 평균적으로는 194ms 더 빠른 것으로 나타났다.

이러한 차이를 보이는 이유는 기존 시스템이 네트워크/어플리케이션 서버로 데이터를 전송하여 경보 발생 여부를 확인하고 이를 다시 CAP 서버에 전송하는 것과 달리 이 절차를 생략하고 게이트웨이에서 경보 발생 여부를 확인하고 CAP 서버로 바로 전송할 수 있기 때문이다.

측정 결과를 각 구간별로 세부적으로 살펴보면 게이트웨이에서 네트워크 서버로 전송하는 과정에서 평균 120ms 가량의 지연이 추가 발생하며, 네트워크 서버에서 어플리케이션 서버로 데이터를 전송하는 과정에서 평균 45ms, 마지막으로 어플리케이션 서버에서 데이터의 경보 발생 여부를 판별하는 단계에서 평균 29ms의 지연이 추가로 발생하였다.

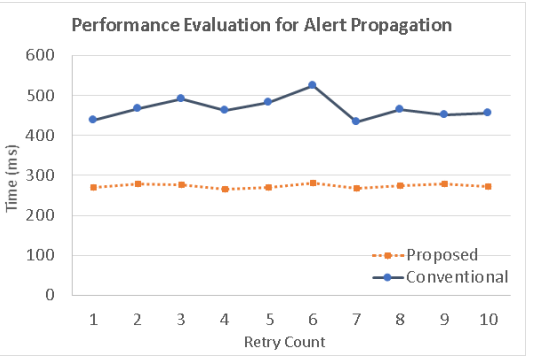

그림 9. 경보 발신시간 측정결과 비교
Fig. 9. Measurement result of Alert sending time

4.2 모니터링 클라이언트 경보전파기능 성능평가

동일한 환경에서 경보가 발생한 후 모니터링 클라이언트로 전달되기까지의 시간을 10회 측정하였다. 해당 결과를 그림 10에 나타내었다.

실험 결과 제안하는 시스템이 모니터링 시스템까지

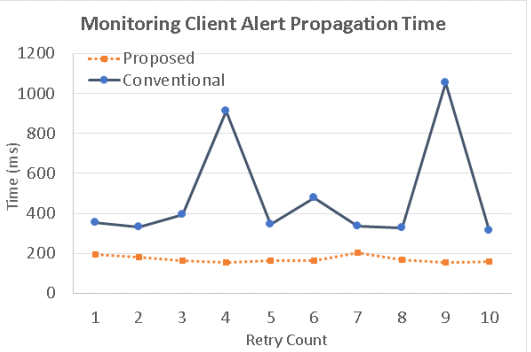

그림 10. 모니터링 클라이언트 경보전파기능 성능평가
Fig. 10. Monitoring client Alert propagation time

의 전파시간 부분에서 기존 시스템보다 빠른 것으로 나타났다. 위 그래프와 같은 차이를 보이는 이유는 기존 모니터링 시스템의 경우 네트워크/어플리케이션 서버로부터 데이터 수집 서버로 데이터가 보내져 데이터베이스에 저장되고 이를 다시 모니터링 클라이언트가 주기적으로 참조하여 경보 발생 여부를 확인하고 표출하는 것과 달리, 제안하는 시스템은 게이트웨이로부터 웹 소켓을 통해 모니터링 클라이언트가 직접 경보 데이터를 수신받기 때문이다.

또한 기존 시스템에서 사용하는 HTTP REST 기법의 특성으로 인해 만약 경보가 발생한 시간이 HTTP 요청이 이루어진 직후이면 다음 요청 사이클까지 경보가 발생하지 않는다. 이는 경보가 발생하는 시간이 불규칙하게 나타나는 원인이 된다. 제안하는 시스템은 웹 소켓을 통해 게이트웨이가 메시지를 즉시 발신하고 모니터링 클라이언트가 바로 수신하므로 불규칙한 지연현상이 적은 것으로 판단된다.

제안하는 시스템에서는 CAP 서버로의 경보 전파 시간은 평균 194ms 줄어들어 약 0.2초의 시간적 이득을 가진다. 이러한 전파시간 단축은 일반적인 서비스에는 큰 의미가 없을 수도 있으나 재난 상황에서의 경보 전파기능 특성상 유의미한 시간이 된다. 또한, 모니터링 클라이언트에서 기존 REST API 사용 시 발생하던 불규칙한 지연 현상이 개선되므로 실제로는 더 큰 전파시간 차이가 발생할 수 있다.

V. 결 론

향후 IoT 생태계에서는 LoRaWAN, NB-IoT, Sigfox, LTE-M 등 다양한 LPWAN 기술에 기반을 두어 배터리로 구동되는 초소형 단말이 더욱 활발하게 보급될 것으로 예상된다. 이러한 단말들이 각종 SoC 시설물이나 자연환경 등에 대량으로 설치되어 각종 위험상황 경보를 전파하는 상황이 올 것이며, 이 경우 경보의 전파시간은 매우 중요한 요소로 작용할 것이다.

이러한 실시간 경보 전파 분야에서 LPWAN 기반의 시스템이 가진 약점은 앞서 언급한 게이트웨이에서 서버간의 전송시간 지연과 더불어 센서에서 게이트웨이간의 무선 전송시간 지연 문제도 함께 지적된다. 향후 이 부분에 대한 고찰과 개선이 있다면 LPWAN 기반 경보 전파 시스템의 대중화가 더욱 빨라질 수 있을 것이다.

본 시스템 또한 향후 트렌드에 맞게 LPWAN 뿐만 아니라 보다 많은 무선통신기술 분야의 적용과 운영을 통해 그 적정성을 검증하고 분석하여 부족한 부분을 보완해 나가고자 한다. 또한 보다 다양한 센서, 네트워크 구조, 모니터링 클라이언트 기능에 대한 연구가 필요할 것으로 보인다.

References

[1] S. Paul and T. V. Sarath, “End to end IoT based hazard monitoring system,” in *Proc. ICIRCA 2018*, Jul. 2018.

[2] M. Yu and S. Kim, “Implementation of new IoT platform based on social relationship,” *J. KICS*, vol. 44, no. 11, pp. 2131-2145, 2019.

[3] LoRa Alliance, *LoRaWAN Specification 1.1*, Retrieved Dec. 12, 2019, from https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf

[4] OASIS Standard, *Common Alerting Protocol*, Retrieved Dec. 12, 2019, from http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html

[5] R. G. C. Rojas, et al., “Broadcasting system of an emergency alert protocol CAP-PER using the standard RDS,” *2018 IEEE BMSB*, Valencia, Spain, Jun. 2018.

[6] E. Kwon, et al., “Emergency-alert services framework using common alert protocol through cable Set-Top box,” *ICTC*, Jeju, South Korea, Oct. 2018.

[7] D. F. Carvalho, et al., “A test bench for evaluating communication delays in LoRaWAN applications,” *2018 IEEE Wrksp.*

on Metrology for Ind. 4.0 and IoT, Brescia, Italy, 2018.

[8] Ministry of the Interior and Safety, National Information Society Agency, *Government Internet of Things Guidelines*, 2019.

[9] M. Yu and S. Kim, "TinyAP: Lightweight application protocol for LoRa environment," *J. KICS*, vol. 43, no. 1, pp. 27-36, 2018.

[10] The Things Network, "*Semtech UDP Packet Forwarder*," Retrieved Dec. 12, 2019, from https://www.thethingsnetwork.org/docs/gateways/packet-forwarder/semtech-udp.html

[11] OpenVPN Technologies, Inc., "*VPN Software Solutions & Services For Business*," Retrieved Dec. 12, 2019, from https://openvpn.net/

[12] ChirpStack, *Chirpstack Gateway Bridge*, Retrieved Dec. 12, 2019, from https://www.chirpstack.io/gateway-bridge/overview/

[13] Semtech corporation, *Basic communication protocol between Lora gateway and server*, Retrieved Dec. 12, 2019, from https://github.com/Lora-net/packet_forwarder/blob/master/PROTOCOL.TXT

[14] I. Fette and A. Melnikov, "*Rfc 6455: The websocket protocol*," IETF, Dec. 2011.

[15] D. Mills, et al., "*RFC 5905: Network time protocol version 4: Protocol and algorithms specification*," Internet Engineering Task Force, 2010.

유 명 한 (Myung-han Yu)

2011년 : 강원대학교 전자공학과 석사

2018년 : 강릉원주대학교 컴퓨터공학과 박사

<관심분야> IoT, USN, Wireless Networking

[ORCID:0000-0003-3994-7130]

김 상 경 (Sangkyung Kim)

1985년 : 고려대학교 전자공학과 학사

1987년 : 고려대학교 전자공학과 석사

2002년 : 고려대학교 전자공학과 박사

1987년~1989년 : 삼성전자 주임연구원

1989년~2004년 : KT 선임연구원(부장)

2004년~현재 : 국립 강릉원주대학교 교수

<관심분야> Wireless Network Protocol, D2D Communications, IoT, Advanced Network Architecture

[ORCID:0000-0002-1410-9918]