

TinyAP: LoRa 환경을 위한 경량 응용프로토콜

유 명 한*, 김 상 경°

TinyAP: Lightweight Application Protocol for LoRa Environment

Myung-han Yu*, Sangkyung Kim°

요 약

대표적 IoT용 응용프로토콜인 MQTT-SN은 연결협상 절차에 따른 제어 메시지 수 증가와 비동기적인 슬립모드(Sleep mode) 지원으로 인해 에너지 효율성이 감소하여 LoRa Class A 단말의 적용에 적합하지 않다. 본 논문에서 제안하는 TinyAP 응용프로토콜은 MQTT-SN 대비 적은 메시지 수와 동기적인 슬립모드 지원을 통해 배터리 기반 저전력 단말에서의 에너지 효율을 향상시킬 수 있다. 시뮬레이션을 통해 MQTT-SN과 TinyAP의 성능을 비교하여 에너지 효율이 향상됨을 확인하고 실제 시스템을 구현하여 그 결과를 검증하였다.

Key Words : IoT, TinyAP, Application Protocol, MQTT-SN, LoRa

ABSTRACT

MQTT-SN is currently the most representative application protocol for IoT, but not suitable for LoRa Class A hardware due to the decrease of energy efficiency, which is caused by the connection negotiation procedure that increases total control messages and asynchronous sleep mode. TinyAP application protocol proposed in this paper can improve the energy efficiency of battery-based low-power devices by way of small control message overhead and developing synchronous sleep mode compared to MQTT-SN. We have compared the performance of TinyAP with that of MQTT-SN through simulation and real-world system implementation, and confirmed that energy efficiency is improved.

1. 서 론

소형 무선 센서로부터 가정용 가전제품에 이르기까지 인터넷에 연결되는 장치들의 수는 가파르게 상승하고 있다^[1]. IoT 플랫폼의 데이터 처리 능력과 별개로 단말 및 게이트웨이에 탑재되는 데이터 전송 프로토콜의 중요성 역시 커지고 있다. WSN(Wireless Sensor Network) 환경에서는 MQTT(Message Queuing Telemetry Transport protocol) 프로토콜이 많이 사용되어 왔으며^[2] 최근들어 IoT용 비 IP 장치들

의 통신에는 MQTT의 경량화 버전인 MQTT-SN(MQTT for Sensor Networks) 프로토콜^[3]이 사용되고 있다.

MQTT-SN은 토픽(Topic)에 대한 게시-구독(Publish-Subscribe) 형태를 가진 경량 메시징 프로토콜로 이전에 발표된 MQTT 프로토콜이 TCP/IP만을 지원하고^[4] 항상 Listen 모드에 놓임으로써 저전력 단말에서 요구하는 슬립모드(Sleep mode)를 지원하지 못하는 단점을 해결하기 위해 개발되었다^[3,5]. MQTT-SN은 비 신뢰적인 전송 프로토콜을 이용하며

* 본 연구는 2015년도 강릉원주대학교 장기혜의 과건연구 지원에 의하여 수행되었습니다.

• First Author : Gangneung Wonju National Univ., Dept. of Computer Science & Engineering, greatymh@gmail.com, 정희원

° Corresponding Author : Gangneung Wonju National Univ., Dept. of Computer Science & Engineering, skkim98@gwnu.ac.kr, 종신회원

논문번호 : KICS2017-09-286, Received September 30, 2017; Revised December 18, 2017; Accepted December 29, 2017

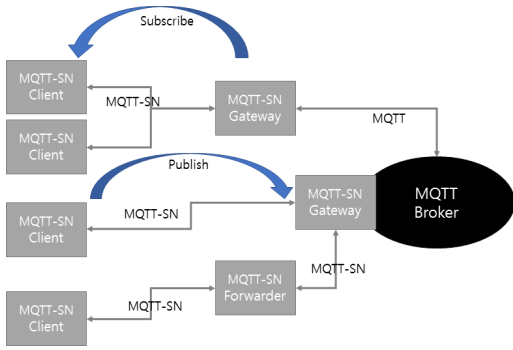


그림 1. MQTT-SN 기반 시스템의 구성[3]
Fig. 1. MQTT-SN system model

ZigBee 등의 비 IP 네트워크에서 사용할 수 있도록 설계되었다. MQTT-SN 기반 시스템의 구성을 그림 1에 나타내었다.

MQTT-SN의 게시-구독 형태는 특정 단말을 지정하여 액추에이터 제어 또는 센서 데이터 수집 명령 등의 데이터를 직접 전송할 수 없으며, 토픽에 대한 구독을 포함하는 연결협상 절차에 따라 제어 메시지의 수가 많아지므로^[6,7] 단순 서버-클라이언트 구조에 비해 에너지 효율성이 요구되는 저전력 장거리 전송에는 적합하지 않다. 따라서 최대 20km의 전송거리에서 0.3kbps 가량의 낮은 데이터 속도를 가지는 LoRa^[8] 등의 LPWAN (Low-Power Wide-Area Network) 환경에 적용하기 위해서는 보다 단순한 형태의 비 IP 통신용 프로토콜이 필요하다.

본 논문에서는 LoRa Class A^[9]와 같은 배터리 구동형 LPWAN 단말에 적합한 TinyAP (Tiny Application Protocol) 를 제안한다. TinyAP는 단말의 위치가 고정적인 네트워크를 대상으로 하며, 단거리 및 중-장거리 네트워크에 모두 적용할 수 있다. MQTT-SN과 달리 연결 협상 절차가 없기 때문에 단말로부터의 데이터 수신 시 교환되는 제어 메시지의 양이 줄어든다. 사용자는 서버를 통해 특정 단말로 액추에이터 제어 또는 센서 데이터 수집 명령 등의 제어 메시지를 직접 전달할 수 있다. 또한 단말과 게이트웨이 간에 슬립모드 시간을 동기화하여 추가적인 제어 메시지의 송수신 없이도 단말이 미리 정한 시간에 게이트웨이로부터 데이터를 수신하여 에너지 효율을 향상시킬 수 있다. 시간 동기화는 절대시간(시스템 클럭)을 맞추는 대신 게이트웨이가 단말로부터 슬립 요청 메시지를 수신한 시각부터의 상대시간을 적용하여 해결하였다.

본 논문의 구성은 다음과 같다. 2장에서는 TinyAP

의 구조 및 상세한 동작을 기술하고 3장에서는 시뮬레이션을 통해 TinyAP와 MQTT-SN의 성능을 비교하여 분석하였다. 4장에서는 결론을 맺는다.

II. TinyAP

2.1 시스템 모델

TinyAP의 적용이 가능한 시스템의 모델을 그림 2에 나타내었다. 단말(Device)은 LoRa 인터페이스를 통해 게이트웨이(Gateway)에 접속하며, 게이트웨이는 TCP/IP를 기반으로 단말과 서버간, 또는 단말 상호간의 메시지를 중계하는 역할을 담당한다. 이는 MQTT-SN과 달리 전통적인 서버-클라이언트 구조를 나타낸다. 서버는 게이트웨이에서 수집되는 데이터를 데이터베이스에 보관하며, 사용자가 이용할 수 있도록 필요한 기능을 제공한다. 또한 사용자의 요청에 따라 게이트웨이를 거쳐 단말에 명령어를 전송하기도 한다. 사용자는 컴퓨터나 스마트폰 등의 사용자 디바이스를 이용하여 서버에 저장된 데이터를 모니터링하며, 서버와 게이트웨이를 거쳐 단말에 메시지를 전달할 수 있다.

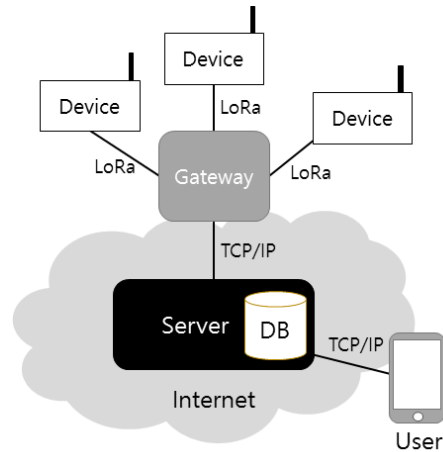


그림 2. TinyAP 시스템 모델
Fig. 2. TinyAP system model

2.2 메시지 형식

TinyAP에서 단말과 게이트웨이, 게이트웨이와 서버간 교환되는 메시지의 기본 구조를 그림 3과 같이 정의하였다.

메시지는 기본 헤더(Basic Header)와 메시지 부분(Message Part)으로 이루어진다. 기본 헤더는 모든 TinyAP 메시지의 앞부분에 공통적으로 포함되며 5바이트의 고정 길이를 가진다. 그림 4는 기본 헤더의 구

← 5 bytes →	← Variable length →
Basic Header	Message Part

그림 3. 메시지의 구조
Fig. 3. Message format

← 1 byte →	← 1 byte →	← 2 byte →	← 1 byte →
TYPE	LEN	SRC / DST (Address)	SEQ

그림 4. 기본 헤더 구조 (총 5바이트)
Fig. 4. Basic header format (5 bytes)

조이다.

기본 헤더의 TYPE 필드는 메시지의 방향과 종류를 나타낸다. 그림 5에 상세한 구조를 나타내었다. MSB가 0이면 메시지는 단말에서 게이트웨이 방향으로 전송되며, 1이면 게이트웨이에서 단말 방향으로 전송됨을 의미한다. 나머지 7비트로 메시지 종류를 표현한다. 각 메시지는 프로토콜의 동작에 따라 단말에서 게이트웨이 방향으로, 또는 게이트웨이에서 단말 방향으로 전송될 수 있으며, 상세한 절차는 2.3절에 기술하였다.

표 1은 TinyAP 동작을 위해 필수적인 메시지의 종류를 보여준다. DATA 메시지는 단말의 센서 데이터를 포함하며, 단말에서 서버로 또는 한 단말에서 게이트웨이를 거쳐 다른 단말로 전송될 수 있다. REQ_DATA 메시지는 서버가 단말로부터 주기적인 데이터 전송시간 외에 데이터 수신을 필요로 하는 경우, 게이트웨이로부터 특정 단말로 전송된다. 이를 수신한 단말은 DATA를 게이트웨이로 전송한다. SET_SLEEP 메시지는 단말에서 게이트웨이 방향으로만 전송되며, 슬립모드에 들어가기 전 현재 시간부터 주어진 시간 간격으로 깨어날 것을 지정하여 게이트웨이로 알리기 위해 사용된다. 단말은 슬립모드 해제 시 CLR_SLEEP 메시지를 이용하여 게이트웨이에 알린다.

REQ_CMD 메시지는 서버로부터의 요청에 따라 게이트웨이가 액추에이터 동작 또는 센싱 주기, 경고 임계치 등의 단말 설정값을 변경할 때 단말로 전송된다. 명령의 실행 결과는 RESP_CMD 메시지로 돌려받는다. REQ_ADDR 메시지는 단말이 네트워크에 처음

Direction	Message Type (0-127)						
MSB (7)	1	2	3	4	5	6	LSB (0)

그림 5. TYPE 필드 비트 정의
Fig. 5. Bit definition of TYPE field

표 1. 필수 메시지 종류 (TYPE 필드)
Table 1. Mandatory message types (TYPE field)

TYPE (Bit 0-6)	Message Type
1	DATA
2	REQ_DATA
10	SET_SLEEP
11	CLR_SLEEP
20	REQ_CMD
21	RESP_CMD
30	REQ_ADDR
31	RESP_ADDR
99	ACK

접속할 때, 새로운 단말 ID의 할당을 요청하기 위해 게이트웨이로 전송된다. 서버는 해당 단말에 할당할 ID를 선택한 후, 그 결과를 포함하는 RESP_ADDR 메시지를 게이트웨이를 통해 단말로 전송한다. ACK 메시지는 LoRa 환경에서 안정적인 메시지 전달을 위해 사용된다. 단말과 게이트웨이는 메시지를 오류 없이 수신한 경우 상대방으로 ACK 메시지를 전송한다. 전송 후 ACK 메시지를 수신하지 못한 경우 단말은 동일한 메시지를 2회 더 전송하며, 이후에도 ACK 메시지를 수신하지 못하면 해당 데이터는 전달에 실패한 것으로 판단한다.

그림 4의 기본 헤더에서 LEN 필드는 기본 헤더와 메시지 부분을 포함하는 메시지 전체의 길이를 나타낸다. LoRa MAC 규격^[10]에서 메시지 전체의 크기는 64바이트를 넘어설 수 없으므로 LEN 필드는 1바이트의 크기를 가진다. ADDRESS 필드는 단말이 게이트웨이로 메시지를 전송할 때는 출발지(SRC) 단말 ID로 사용되며, 게이트웨이가 단말로 메시지를 전달할 경우에는 목적지(DST) 단말 ID로 사용된다. 2바이트 길이를 가지며, 0부터 65,535까지의 단말 ID를 나타낼 수 있다. SEQ 필드는 단말별 메시지의 순서번호를 나타낸다. 단말은 REQ_ADDR 메시지를 게이트웨이로 전송할 때 순서번호 값을 1로 정하여 시작하며, 단말과 게이트웨이간 메시지 송·수신시 1씩 증가한다. 이와 관련한 상세한 설명은 “8. 전송 확인 절차”에 기술하였다.

2.3 단말 ID 할당 절차

단말 ID를 가지지 않은 단말은 네트워크에 단말 ID 발급을 요청하고 서버에서 이를 할당받는다. 게이트웨이와 단말간의 단말 ID 할당 절차를 그림 6에 나타내

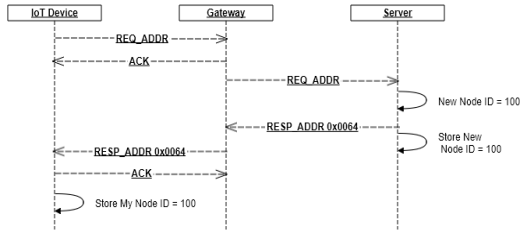


그림 6. 게이트웨이와 서버, 단말간의 단말 ID 할당 절차 예
Fig. 6. Example of ID assignment procedure among gateway, server and device

었다.

단말이 REQ_ADDR 메시지를 전송하면 이는 게이트웨이를 거쳐 서버로 전송된다. 서버는 단말 ID의 할당을 관리하며, 새 단말 ID (예:100)를 새로 할당한 후 RESP_ADDR 메시지를 이용하여 게이트웨이를 거쳐 단말로 전달한다. REQ_ADDR 메시지는 기본 헤더로만 구성된다. 서버에서 단말로 전송되는 RESP_ADDR 메시지의 메시지 부분 형식을 그림 7에 나타내었다. RESP_ADDR 메시지의 메시지 부분은 단말 ID 데이터로 구성된다. 만약 단말 ID 할당에 실패할 경우에는 ADATA 필드에 0이 설정된다. 이 경우, 단말은 ID 신규 할당 요청을 다시 보낼 수 있다.

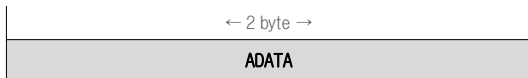


그림 7. RESP_ADDR 메시지 부분 형식
Fig. 7. Format of RESP_ADDR message part

2.4 데이터 전달 절차

TinyAP를 사용하는 단말에서 서버로 센서 데이터를 전송하는 과정을 그림 8에 나타내었다.

단말에서 측정된 센서 데이터는 게이트웨이로 전송되며 이를 수신한 게이트웨이는 LPWAN MAC에 포함된 CRC 점검 절차^[10]를 통해 데이터를 검증하고 이상이 없을 경우 단말 측으로 ACK 메시지를 보낸다. 게이트웨이는 수신한 데이터를 서버에 그대로 전달하

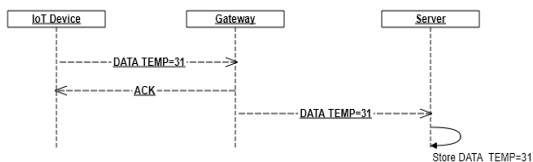


그림 8. 클라이언트에서 서버로의 센서 데이터 전송 예
Fig. 8. Example of data transmission from device to server

← 1 byte →	← Variable length →	← 2 byte →
DTYPE	DDATA	DDST (Optional)

그림 9. DATA 메시지 부분 형식
Fig. 9. DATA message part format

는 역할을 수행한다. 서버는 IP 주소를 통해 각 게이트웨이를 식별하고, 단말의 데이터를 데이터베이스에 저장한다.

DATA 메시지의 메시지 부분 길이는 측정 대상 센서에 따라 가변적이다. 그림 9에 DATA 메시지의 메시지 부분 형식을 나타내었다.

DTYPE은 센서의 종류를 나타내며, 각 구현레벨에서 미리 지정할 필요가 있다. 표 2에 자주 사용되는 센서에 대한 DTYPE 및 DDATA 포맷을 프로토콜 차원에서 미리 정의하였다. DDATA 필드에는 표 2의 정의에 따라 가변적으로 내용을 삽입할 수 있다. 예를 들어, Wind Direction(0x1E) 데이터의 경우 길이는 총 4바이트이다. DDST는 단말에서 서버가 아닌 다른 단말로 데이터를 전송할 때 사용되며 목적지 단말의 ID 값을 가진다. 데이터의 목적지가 서버인지 또는 다른 단말인지는 DTYPE 필드의 MSB(bit 7)로 결정된다. MSB가 1이면 다른 단말로, 0이면 서버로 전송함을 의미한다.

표 2. 센서 종류 및 측정 값 포맷
Table 2. Sensor types and measured value format

Data Type	DTYPE (Bit 0-6)	DDATA	
Temperature	0x1A	2-byte Float	
Humidity	0x1B	2-byte Float	
Integrated Temp-Humi	0x1C	2-byte Temp	2-byte Humi
Wind speed	0x1D	2-byte Float	
Wind direction	0x1E	ASCII (4-byte Max)	
Light(cd.)	0x2A	2-byte Unsig. Float	
Light(lm.)	0x2B	2-byte Unsig. Float	
Fire detection	0x2C	1-byte Boolean	
Motion detection	0x2D	1-byte Boolean	
Door open	0x2E	1-byte Boolean	

2.5 REQ_DATA 메시지를 통한 데이터 수집 절차

REQ_DATA 메시지는 게이트웨이에서 단말로 미리 정해진 시간 외에 데이터를 받을 필요성이 생길 경우 사용한다. 데이터 요청 시 메시지 부분은 받으려는

← 1 byte →	← 1 byte →	← Variable length →
STYPE	WTIME	Condition

그림 10. REQ_DATA 메시지 부분
Fig. 10. REQ_DATA message part format

데이터(센서)의 종류, 데이터의 조건, 데이터 수신 예상시간으로 구분된다. 요청을 통한 데이터 수집은 게이트웨이로부터 단말로 단말이 깨어있는 시간, 즉 슬립모드 시간 이외에만 요청된다. 만약 단말이 슬립모드에 들어갔을 경우 게이트웨이는 단말이 깨어날 때까지 대기하였다가 요청을 전송한다. REQ_DATA 메시지의 메시지 부분을 그림 10에 나타내었다.

메시지 부분의 STYPE는 수신 받기를 원하는 센서의 종류이며 사전에 정의된 센서 종류 코드를 사용한다. WTIME은 데이터 수신 예상시간으로 서버가 데이터를 수신하기 위해 대기 가능한 시간을 의미하며, 해당 분 단위 이내로 데이터를 수신 받아야 한다는 조건이다. 0일 경우 무한정 대기한다. 이후는 데이터 조건(Condition)으로, 해당 수치범위를 초과할 경우 또는 미달할 경우의 조건이 위치한다. 조건은 정수 또는 실수형 숫자와 사칙연산 기호로 이루어질 수 있다. 데이터 수집을 요청 받은 상대 단말에서의 응답은 2.4절의 DATA 메시지 형태로 전송한다.

2.6 명령 및 설정 업데이트 절차

게이트웨이에서 단말로 명령 또는 설정 값을 송신하기 위해 REQ_CMD 메시지가 사용된다. 서버에서 클라이언트로 REQ_CMD 메시지를 전송하는 과정을 그림 11에 나타내었다. 서버에서 게이트웨이를 거쳐 명령어 데이터를 전송하면 단말은 해당 명령을 수행한 후, 명령 수행에 대한 결과 값으로 RESP_CMD 메시지를 송신한다. 명령 요청에서 메시지 부분은 보낼 명령의 종류, 명령 코드, 명령의 내용, 리턴 인디케이터(indicator)로 구분된다.

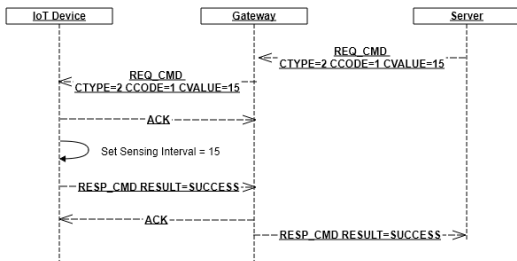


그림 11. 서버에서 클라이언트로의 명령어 전송 예
Fig. 11. Example of command transmission from server to device

그림 12에 명령 요청의 메시지 부분을 나타내었다. CTYPE는 명령의 종류로서 액추에이터 ON/OFF 제어는 0, PWM(Pulse Width Modulation) 제어는 1, 설정 값 변경은 2, 리셋 등 추가 제어 명령은 3으로 설정된다. 실제 명령의 종류는 단말의 구현에 따라 달라질 수 있다. CCODE는 명령 코드로, 예를 들어 액추에이터 ON/OFF일 경우 어느 액추에이터에 대해 명령을 실행할 것인지에 대한 정보가 포함된다. 단말 설정값 변경일 경우 설정을 원하는 항목의 번호가 포함된다. CVALUE는 명령의 내용으로 단말에 대해 설정할 값이나 제어할 액추에이터의 값을 나타낸다. 예를 들어 단말의 센싱 주기 설정 값 변경일 경우 분 단위의 센싱 주기 설정 값이 포함된다.

그림 13에 RESP_CMD 메시지의 메시지 부분 형식을 나타내었다. 앞의 3가지 필드에 대한 정의는 REQ_CMD 메시지의 형식과 동일하나, 메시지 부분의 마지막에 1바이트의 CSTATUS 필드가 위치한다. 해당 필드의 값에서 0은 성공, 이외의 값은 실패를 나타낸다. CSTATUS에 상세한 에러 코드를 포함할 수 있다.

← 1 byte →	← 1 byte →	← Variable length →
CTYPE	CCODE	CVALUE

그림 12. REQ_CMD 메시지 부분 형식
Fig. 12. REQ_CMD message part format

← 1 byte →	← 1 byte →	← Variable length →	← 1 byte →
CTYPE	CCODE	CVALUE	CSTATUS

그림 13. RESP_CMD 메시지 부분 형식
Fig. 13. RESP_CMD message part format

2.7 슬립모드 지정 및 해제 절차

TinyAP에서는 전송되는 메시지 수를 줄이기 위해 단말이 슬립모드에서 깨어나도 게이트웨이에 별도의 메시지를 보내지 않는다. 대신 단말은 슬립모드에 들어가기 전 자신의 슬립/웨이크 업(Wakeup) 주기(분 단위)를 SET_SLEEP 메시지를 이용해 게이트웨이로 통보한다. 게이트웨이가 단말에 데이터를 전달할 필요가 있을 경우 응용에 따라 미리 설정된 슬립모드 시간 동안 기다린 뒤 데이터를 전송한다. 슬립모드 시간은 응용 및 환경에 따라 다를 수 있다.

그림 14에 슬립모드 지정에 따른 데이터의 전송절차를 나타내었다.

SET_SLEEP 메시지로 단말이 깨어날 시간이 지정되어 있을 경우, 해당 슬립모드 시간동안 서버에서 전

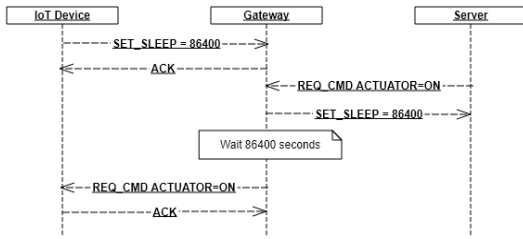


그림 14. 슬립모드 지정에 따른 데이터 전송절차 예
Fig. 14. Example of data transmission during sleep mode

송된 메시지는 단말에 전송되지 않고 게이트웨이에 보관되었다가 단말이 깨어난 이후에 전송된다. 만약 단말이 깨어난 뒤 게이트웨이로부터 통신이 5초간 수신되지 않으면 다시 지정된 시간동안 슬립모드로 들어간다. 이는 MQTT-SN의 슬립모드 지원 방법 중 단말이 깨어날 경우 전송하는 PINGREQ 명령과 같이^[1] 단말에서 게이트웨이로 보낼 데이터가 없음에도 불필요한 웨이크 업 확인 및 재연결 요청 메시지를 보내는 것을 막는다. 그림 15에 SET_SLEEP 메시지의 메시지 부분 형식을 나타내었다.

메시지 부분의 첫 2 바이트는 주기에 대한 설정 값 (SPERIOD)으로 분 단위의 상대시간으로 설정한다. 게이트웨이는 메시지 수신 시점으로부터 웨이크 업 시간을 카운트한다. 상대 시간을 사용함으로써 게이트웨이와 단말간에 별도의 시간 동기화 절차 없이 게이트웨이는 단말의 웨이크 업 시간에 대기 중인 메시지들을 단말로 전송할 수 있다. 게이트웨이는 메시지 전송 후 웨이크 업 시간을 새롭게 카운트한다. 슬립모드 동작이 누적되는 동안 단말과 게이트웨이 간 상대적 시간 동기가 흐트러질 수 있기 때문에 단말 및 게이트웨이는 상호간 메시지 송수신이 발생한 후 슬립모드 타이머를 다시 설정한다. SIND 필드는 일회성 메시지 인디케이터로서 0일 경우 해당 설정이 주기적임을 나타낸다. 필드의 값이 1이면 설정이 일회성으로 WPERIOD에 지정한 시간 후 깨어난 단말은 이후 다시 SET_SLEEP 메시지를 게이트웨이로 보낼 때까지 슬립모드에 들어가지 않고 계속 깨어 있음을 의미한다.



그림 15. SET_SLEEP 메시지 부분 형식
Fig. 15. SET_SLEEP message part format

2.8 전송 확인 절차

TinyAP의 모든 메시지에는 SEQ(순서번호)가 포함된다. SEQ는 1바이트 크기를 가지며, 1에서 시작하여

모듈로(Modulo) 255 연산으로 동작한다. 만약 게이트웨이 또는 단말이 받은 메시지의 시퀀스 정보를 검사하여 자신이 가진 최종 시퀀스 정보로부터 1을 더한 것과 다르거나 LoRa MAC에 포함된 CRC 점검 절차^[10] 수행 결과 잘못된 메시지로 판단되면 ACK 메시지를 되돌려주지 않는다.

SEQ는 단말이 처음 게이트웨이에 접속하여 메시지 전송을 시작할 때 1로 설정된다. 게이트웨이는 각 단말별로 SEQ를 관리하며, 해당 단말로 ACK가 아닌 메시지를 전송할 때 SEQ를 증가시킨다. ACK 메시지는 기본 헤더로만 구성되며, SEQ 필드에는 정상적으로 수신한 메시지의 SEQ 번호가 포함된다. 만약 SEQ가 0일 경우에는 해당 메시지에 대해 ACK 메시지를 요구하지 않은 것으로 판단하여 ACK 메시지를 보내지 않는다.

III. 구현 및 실험

TinyAP의 동작 검증을 위하여 실제 LoRa Class A 하드웨어를 구현하여 시험하였다. 사용한 하드웨어 및 개발환경은 그림 16 및 표 3과 같다.

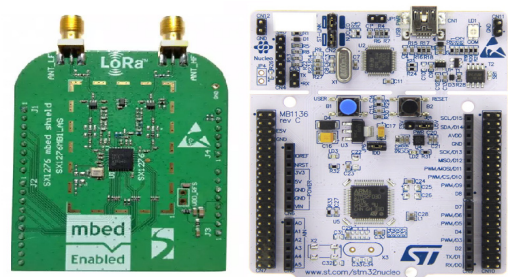


그림 16. 구현 하드웨어 (좌) SX1276MB1LAS / (우) Nucleo-L476
Fig. 16. Hardware used (Left) SX1276MB1LAS / (Right) Nucleo-L476

표 3. 하드웨어 및 개발환경
Table 3. Hardware and Development environment

Environment	Model and software used
LoRa Radio Board	Semtech SX1276MB1LAS
Mainboard (MCU)	STmicroelectronics Nucleo-L476
Compiler	Keil uVision IDE
Language	C99
Gateway	Semtech LoRa IOT Starter Kit SX1301 Concentrator

실험은 그림 17 및 표 4와 같이 야외에서 진행하였다. 게이트웨이 하나에 두 개의 단말을 연결하였으며, 측정된 단말들과 게이트웨이간 거리는 약 4.1km였다. 해당 구간은 아무런 장애물이 없는 open site이다.

시험과정에서 단말 ID 발급, 데이터 송·수신, 슬립 모드 기능 등이 정상적으로 동작됨을 확인하였다. 성능 평가를 위한 시뮬레이션 파라미터를 구하기 위해 디지털 멀티미터를 이용하여 단말의 에너지 소비량을 측정하였다. AA 배터리 두 개를 직렬 연결한 단말에서 5바이트 길이의 메시지를 14dBm의 출력으로 62,000회 연속 전송한 후의 배터리 잔량을 확인하여 1개의 메시지 송신에 의한 에너지 소비량을 계산하였다. 최초 단말의 전압은 3.137V이었으며 모든 메시지들의 전송 완료 후 전압은 2.53V이었다. 즉, 매 전송당 0.00979mV가 소비되었다.

이를 바탕으로 시뮬레이션 환경을 구축하고 실험하

였다. TinyAP와 MQTT-SN간의 성능 비교를 위해 게이트웨이 하나에 2개의 단말을 연결하는 것을 가정하였다. 게이트웨이는 단말로부터 데이터를 수집하고 서버로부터 받은 데이터를 단말로 전달한다. 서버는 데이터를 수신하여 저장하거나 사용자의 요청 데이터를 게이트웨이를 거쳐 단말로 전송하는 역할을 한다. MQTT-SN의 QoS 레벨은 2로 가정하였으며, 메시지는 단말과 게이트웨이 간에 1:1 형태로 게시-구독 방식을 사용하여 공유하였다. TinyAP 및 MQTT-SN 프로토콜을 사용하는 각 단말은 동일한 주기로 슬립모드에서 깨어날 때마다 5바이트 페이로드 크기의 데이터 메시지를 게이트웨이로 전송하도록 하였으며 게이트웨이로부터 단말로의 데이터 전달은 수신 메시지 5개당 1회 간격으로, 동일한 5바이트 페이로드 크기를 설정하였다.

시뮬레이션에서 에너지 소비량은 구현을 통한 시험에서의 환경과 동일하게 단말의 전압 레벨을 3.137V로 시작하여 5바이트 길이의 메시지에 대해 1회 전송당 0.00979mV의 에너지를 감소시켰으며, 메시지의 길이에 따라 5바이트를 초과하는 메시지는 1바이트당 0.001mV의 에너지를 더 감소시켰다. 이는 메시지의 길이가 길어질수록 전송시간이 늘어나며, 패킷 손실 등으로 인한 재전송 등 전체적인 에너지 소비가 증가함을 고려한 것이다. 해당 조건으로 단말의 에너지가 2.53V에 도달할 때까지 시험하였다.

수신과 송신에 필요한 에너지는 동일하게 설정하였으며, 실험 결과는 프로토콜별로 2개 단말의 평균값으로 산출하였다. 모든 시뮬레이션 결과 그래프의 X축은 전체 메시지 중 단말에서 송·수신된 데이터 메시지의 수를 기준으로 하였다. 시뮬레이션을 통해 측정된 에너지 소비량을 그림 18에 나타내었다.

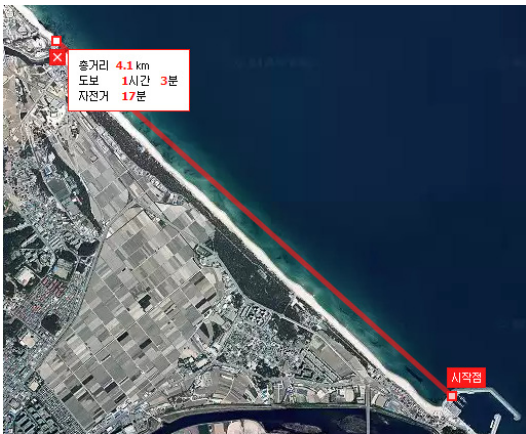


그림 17. TinyAP 필드 테스트 구간 (4.1km)
Fig. 17. Field test site for TinyAP (4.1km)

표 4. 시험환경
Table 4. Test environment

Environment	Parameters
Gateway Count	1
Sensor Node Count	2
Distance	4.1km
Wireless Frequency	922.1MHz / 922.3MHz ISM-band
Wireless TX Power	14dBm
Wireless Antenna	2dBi Dipole
LoRa Configuration	Spreading Factor 12, Bandwidth 125kHz, 4/5 Coding Rate

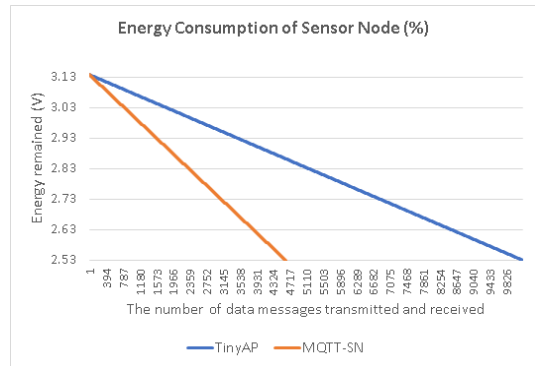


그림 18. 프로토콜별 단말 에너지 소비량 비교
Fig. 18. Comparison of energy consumption between two protocols

배터리 잔량이 2.53V에 도달할 때까지 TinyAP 단말들은 10,200회의 데이터 메시지를 송·수신하였으며, MQTT-SN 단말들은 4,618회의 데이터 메시지를 송·수신하여 약 45.27%의 에너지 효율이 개선되는 것으로 나타났다. 이는 두 프로토콜들 간에 연결협상 및 네트워크 유지, 데이터를 교환하기 위해 요구되는 추가적인 제어 메시지 전송횟수와 메시지 길이의 차이에 의한 것으로 분석된다.

웨이크 업 - 단말에서 게이트웨이로 데이터 전송 - 슬립모드 진입을 하나의 트랜잭션이라고 할 때 트랜잭션 당 송수신되는 메시지 수와 바이트 수를 표 5에 비교하여 나타내었다.

반대 방향으로 웨이크 업 - 게이트웨이에서 단말로 데이터 전송 - 슬립모드 진입의 경우 트랜잭션 당 송수신 메시지 수와 바이트 수를 표 6에 나타내었다.

시뮬레이션 결과 TinyAP는 MQTT-SN 대비 전반적인 제어 메시지의 양과 이에 따른 에너지 소비량 측면에서 우세하였다. TinyAP의 경우 슬립모드 동기화 이후 데이터 메시지의 전송 및 슬립모드 재진입 시 DATA 메시지와 ACK 메시지만이 교환되나 MQTT-SN은 연결설정 및 해제 과정이 필요하며 게이트웨이에서 대기 중인 데이터를 확인하고 수신하기 위해 PING REQ와 PING RESP 메시지를 주고받아야 하므로 메시지의 수는 늘어나게 된다³⁾.

그림 19는 데이터 메시지 송·수신 시 교환된 총 메시지 수를 나타낸다.

그림 20은 데이터 메시지 송·수신 시 교환된 제어 메시지를 포함한 메시지들의 총 바이트 수를 보여준다.

표 5. 메시지 수 및 바이트 수 비교표 (node to gateway)
Table 5. Comparison table of the number of messages and bytes per transaction (node to gateway)

	MQTT-SN	TinyAP
Messages per transaction	6	2
Bytes per transaction	31	16

표 6. 메시지 수 및 바이트 수 비교표 (gateway to node)
Table 6. Comparison table of the number of messages and bytes per transaction (gateway to node)

	MQTT-SN	TinyAP
Messages per transaction	3	2
Bytes per transaction	17	16

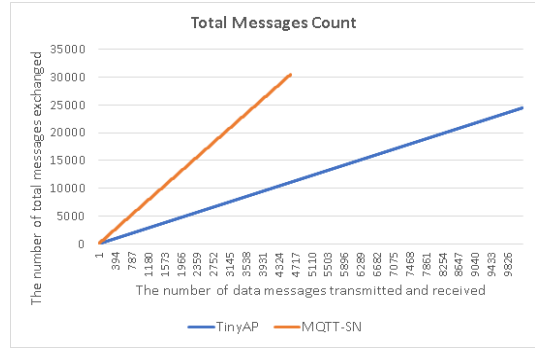


그림 19. 교환된 데이터 메시지 대비 총 메시지 수
Fig. 19. The number of total messages according to data messages exchanged

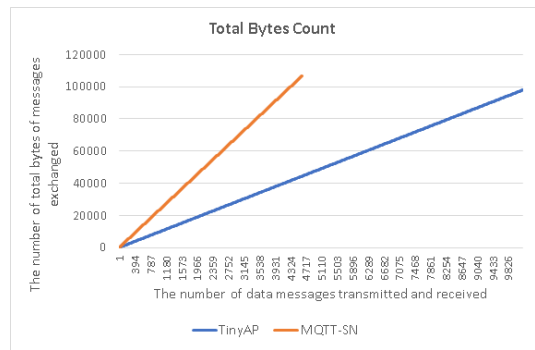


그림 20. 교환된 메시지들의 총 바이트 수
Fig. 20. The number of total bytes for messages exchanged

이러한 결과는 TinyAP의 동기화된 슬립모드 지원에 의한 차이도 있지만, 네트워크 연결 및 유지 절차가 게시-구독 구조를 가진 MQTT-SN보다 단순하므로 동일한 데이터의 전송을 위해 제어 메시지를 적게 전송하기 때문으로 분석된다.

IV. 결 론

본 논문에서는 LoRa와 같은 저전력 소비가 요구되는 환경에서 사용 가능한 새로운 IoT용 응용 프로토콜인 TinyAP를 제안하였다. TinyAP는 기존의 대표적 응용 프로토콜인 MQTT-SN과 달리 연결 협상 절차 없이 데이터를 수신하여 교환되는 제어 메시지들의 수를 줄이고 동기화된 슬립 모드 기능을 적용하여 에너지 효율을 높일 수 있다. 또한, 서버로부터 특정 단말로 직접 제어 메시지를 전달할 수 있으며 모든 전송되는 메시지에 대한 전송확인 기능을 통해 신뢰성을 제고하였다. TinyAP의 기능은 단말과 게이트웨이를

구현하여 검증하였다. MQTT-SN과의 성능 비교를 위하여 시뮬레이션을 수행하였으며 에너지 효율이 향상됨을 확인하였다. 이는 네트워크에 오가는 전체 메시지의 수를 줄임과 동시에 단말이 슬립모드 상태에서 깨어날 경우 발생하는 불필요한 메시지를 줄이고 최소한의 메시지만으로 통신을 수행하기 때문으로 분석된다.

한편, 메시지 전달 과정에서 메시지 손상이나 손실 발생 시 처리 과정은 TinyAP와 MQTT-SN이 다르지 않아 프로토콜 성능 개선을 위해서 재전송 메커니즘 등 오류제어에 관한 추후 연구를 진행할 예정이다. 또한, 본 논문에서는 단말 관점에서 에너지 효율성만을 시뮬레이션을 통해 평가하였으나 메시지 전달 지연, 메시지 전달률 등 게이트웨이를 포함한 네트워크의 성능 분석이 요구되는데 게이트웨이에 연결된 단말들의 수 및 단말과 게이트웨이 간 거리에 따른 시뮬레이션을 수행할 계획이다. 향후 LoRa, Sigfox 등의 LPWAN 환경에서 배터리로 구동되는 초소형 단말이 보다 많이 보급될 것으로 예상되므로 다양한 실제 환경에서 구현한 시스템을 시험하여 시스템의 안정성을 확보하고 성능 및 기능 개선을 위한 연구를 추진할 필요가 있다.

References

[1] P. Ganguly, "Selecting the right IoT cloud platform," in *IEEE IOTA*, pp. 316-320, Pune, India, Jan. 2016.

[2] V. Karagiannis, P. Chatzimisios, F. Vázquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," *Trans. IoT and Cloud Computing*, vol. 3, no. 1, pp. 11-17, Sept. 2015.

[3] A. Stanford-Clark and H. L. Truong, *MQTT for sensor networks (MQTT-SN) protocol specification*, version 1.2 (2013), Retrieved Sept. 15, 2017, from http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf.

[4] D. Locke, *Mq telemetry transport (mqtt) v3.1 protocol specification* (2010), Retrieved Sept. 15, 2017, from <https://www.ibm.com/developerworks/library/ws-mqtt/>

[5] D. Thangavel, X. Ma, A. Valera, H. Tan, and C. K. Tan, "Performance evaluation of MQTT

and CoAP via a common middleware," in *ISSNIP IEEE 2014*, pp. 1-6, Singapore, Apr. 2014.

[6] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, and H. Hashim, "A comparison of lightweight communication protocols in robotic applications," *IEEE IRIS 2015*, pp. 400-405, Langkawi, Malaysia, Oct. 2015.

[7] Y. Jang, J. Shim, and S. Park, "Analysis standardized of IoT-based low-power-light-weight protocol," *J. KIICE*, vol. 20, no. 10, pp. 1895-1902, Oct. 2016.

[8] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, MDPI, 2016.

[9] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 60-67, Nov. 2016.

[10] N. Sornin, M. Luis, T. Eirich, and T. Kramp, *LoRa MAC specification* (2014), Retrieved Dec. 15, 2017, from <https://github.com/Lora-net/LoRaMac-node/raw/v2.2/Doc/LoRa%20MAC%20Specification.pdf>.

유 명 한 (Myung-han Yu)



2010년 : 강릉원주대학교 컴퓨터공학과 학사
2011년 : 강원대학교 전자공학과 석사
2012년 : 강릉원주대학교 컴퓨터공학과 박사수료
<관심분야> IoT, USN, Wireless Networking

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