

## K-ONE 기술 문서 #14

# Cloud 환경 기반 LoRaWAN provisioning : Network Server , Gateway

Document No. K-ONE #1

Version 1.0

Date 2016-05-01

Author(s) 소재영

■ 문서의 연혁

버전	날짜	작성자	비고
초안 - 0.5	2016. 04. 15	소재영	
1.0	2016. 05. 08	소재영	

본 문서는 2015년도 정부(미래창조과학부)의 재원으로 정보통신  
기술진흥센터의 지원을 받아 수행된 연구임 (No. B0190-15-2012, 글로벌  
SDN/NFV 공개소프트웨어 핵심 모듈/기능 개발)

This work was supported by Institute for Information &  
communications Technology Promotion(IITP) grant funded by the  
Korea government(MSIP) (No. B0190-15-2012, Global SDN/NFV  
OpenSource Software Core Module/Function Development)

## 기술문서 요약

사물인터넷 (IoT , Internet of Things) 단말의 수가 기하급수적으로 늘어남에 따라 서버로 전송되는 데이터 트래픽의 양도 증가한다. 이에 따라 , 기존 네트워크 서버의 부하를 유연하고 지능적으로 관리 제어할 수 있는 기술이 필요하다. 사물인터넷을 위한 다양한 프로토콜이 존재 한다. 우리는 그 중 저 전력 무선 통신 장치들을 위한 LoRaWAN을 연구 대상으로 하며 여기서 소개하는 테스트베드는 향후 연구에 기초가 되는 역할을 수행하게 될 것이다.

본 문서는 LoRaWAN이나 OpenStack을 잘 모르는 사용자도 간단하게 테스트베드 구축을 할 수 있도록 하는데 초점을 두고 작성 하였다. 따라서 서버나 게이트웨이의 세부적인 동작 이해를 위해서는 LoRa Alliance에서 공개한 specification을 참조 하는 것이 좋다. 또한 테스트베드 구축을 위해 사용한 장치와 게이트웨이의 경우 어플리케이션 개발을 할 수 있도록 관련 소스코드가 공개되어 있으므로 세부동작은 사용자가 원하는 대로 수정하여 사용할 수 있다.

---

## Contents

### K-ONE #1. Cloud 환경 기반 LoRaWAN provisioning

1. OpenStack 기반 LoRaWAN 테스트베드 구성 .....	6
1.1. HW 시스템 환경 .....	6
1.2. SW 시스템 환경 .....	6
2. 인터페이스 정의 .....	8
2.1. 프로토콜 스택 .....	8
2.1.1. UDP , TCP .....	9
2.2. 메시지 프로토콜 .....	10
2.2.1. Gateway <-> Network Server .....	10
2.2.2. LoRaWAN Controller <-> Servers , gateway .....	13
2.2.3. Network Server <-> Application Server .....	15
2.3. JSON 프로토콜 .....	16
2.3.1. Gateway <-> Network Server .....	16
2.3.2. Controller -> Servers , gateway .....	19
2.3.3. Network Server -> Application Server .....	19
3. LoRaWAN Network Server .....	19
3.1. 기능 .....	19
3.2. HW 시스템 환경 .....	20
3.3. SW 시스템 환경 .....	20
3.4. 운용방법 및 검증 .....	21
4. LoRaWAN Gateway .....	22
4.1.1. 기능 .....	22
4.1.2. HW 시스템 환경 .....	22
4.1.3. SW 시스템 환경 .....	23
4.1.4. 운용방법 및 검증 .....	24

---

## 표 목차

표 1 테스트베드에 사용어 서버 스펙 .....	6
표 2 local.conf .....	7
표 3 TCP port .....	10
표 4 UDP port .....	10
표 5 PUSH_DATA format .....	10
표 6 PUSH_ACK format .....	11
표 7 PULL_RESP format .....	11
표 8 TX_ACK format .....	12
표 9 PULL_DATA format .....	12
표 10 PULL_ACK format .....	13
표 11 HELLO_GW format .....	13
표 12 HELLO_NS format .....	14
표 13 HELLO_APP format .....	15
표 14 APP_DATA_UP format .....	15
표 15 APP_DATA_DOWN format .....	16
표 16 stat JSON object .....	16
표 17 rxpk JSON object .....	17
표 18 txpk JSON object .....	18
표 19 JSON object between Controller and servers,gateway .....	19
표 20 JSON object between Network Server and Application Server .....	19
표 21 Network Server용 flavor .....	20

---

## 그림 목차

그림 1 테스트베드 하드웨어 구성 .....	6
그림 2 테스트베드 SW 시스템 환경 .....	6
그림 3 communication diagram .....	8
그림 4 Protocol Stack .....	9
그림 5 Upstream GWMP sequence diagram .....	11
그림 6 Downstream GWMP sequence diagram .....	12
그림 7 Gateway PULL_DATA diagram .....	13
그림 8 HELLO_GW diagram .....	14
그림 9 HELLO_NS diagram .....	14
그림 10 HELLO_AS diagram .....	15
그림 11 APP_DATA diagram .....	16
그림 12 Network Server 시스템 환경 .....	20
그림 13 heat template의 network server 셸 스크립트 .....	21
그림 14 Network Server의 구동확인 .....	22
그림 15 Network Server 구동 로그 .....	22
그림 16 LoRa IoT Station .....	23
그림 17 Hardware architecture of LoRa IoT Station .....	23
그림 18 Gateway SW 시스템 환경 #1 .....	23
그림 19 Gateway SW 시스템 환경 #2 .....	24
그림 20 Gateway log #1 .....	25
그림 21 Gateway log #2 .....	25

## 1. OpenStack 기반 LoRaWAN 테스트베드 구성

### 1.1. HW 시스템 환경

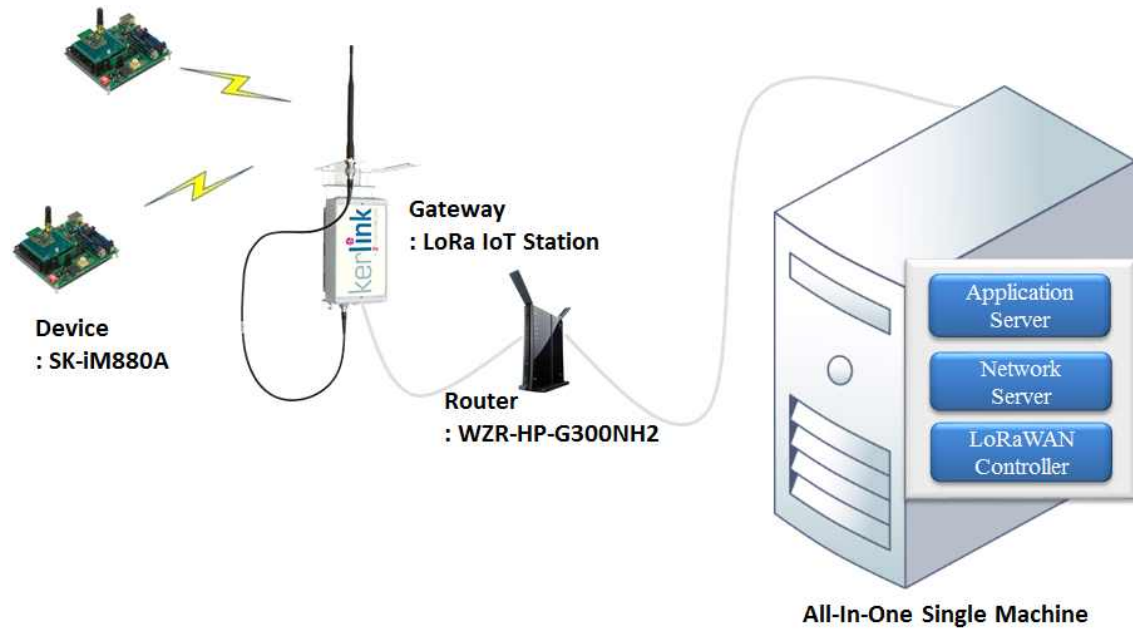


그림 1 테스트베드 하드웨어 구성

Model	SC-743
CPU	Intel(R) Xeon(R) CPU E5-2650 v2, 2.60 GHz, 8-cores
RAM	64GB
SDD	480GB

표 1 테스트베드에 사용어 서버 스펙

### 1.2. SW 시스템 환경

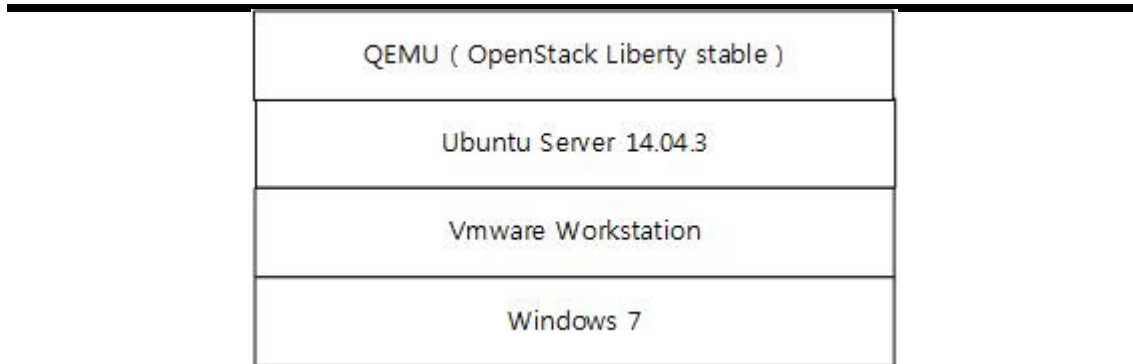


그림 2 테스트베드 SW 시스템 환경

o OpenStack은 All-In-One Single Machine형태로 compute node , controller node , network node를 단일 ubuntu server에 설치하였다. 아래는 설치에 사용한 devstack용 local.conf 파일이다.

표 2 local.conf

```
[[local | localrc]]
HOST_IP=192.168.10.19
SERVICE_HOST=192.168.10.19
MYSQL_HOST=192.168.10.19
RABBIT_HOST=192.168.10.19
GLANCE_HOSTPORT=192.168.10.19:9292
ADMIN_PASSWORD=secrete
DATABASE_PASSWORD=secrete
RABBIT_PASSWORD=secrete
SERVICE_PASSWORD=secrete

# Do not use Nova-Network
disable_service n-net

# Enable Neutron
ENABLED_SERVICES+=,heat,h-eng,h-api,h-api-cfn,h-api-cw,q-svc,q-dhcp,q-meta,q
-agt,q-l3

## Neutron options
Q_USE_SECGROUP=True
FLOATING_RANGE="192.168.10.0/24"
```



```
FIXED_RANGE="10.0.0.0/24"  
Q_FLOATING_ALLOCATION_POOL=start=192.168.10.170,end=192.168.10.180  
PUBLIC_NETWORK_GATEWAY="192.168.10.100"  
Q_L3_ENABLED=True  
PUBLIC_INTERFACE=eth0  
  
# Open vSwitch provider networking configuration  
Q_USE_PROVIDERNET_FOR_PUBLIC=True  
OVS_PHYSICAL_BRIDGE=br-ex  
PUBLIC_BRIDGE=br-ex  
OVS_BRIDGE_MAPPINGS=public:br-ex
```

## 2. 인터페이스 정의

- o 본 장에서는 Gateway 와 Network Server간 , Network Server 와 Application Server 간 , LoRaWAN Controller 와 Server들 간의 인터페이스를 정의한다. 특히 Gateway 와 Network Server간 인터페이스 GWMP ( Gateway message protocol ) 는 SEMTECH에서 공개한 'LoRaWAN Network Server Demonstration: Gateway to Server Interface Definition' 문서와 동일하다.

### 2.1. 프로토콜 스택

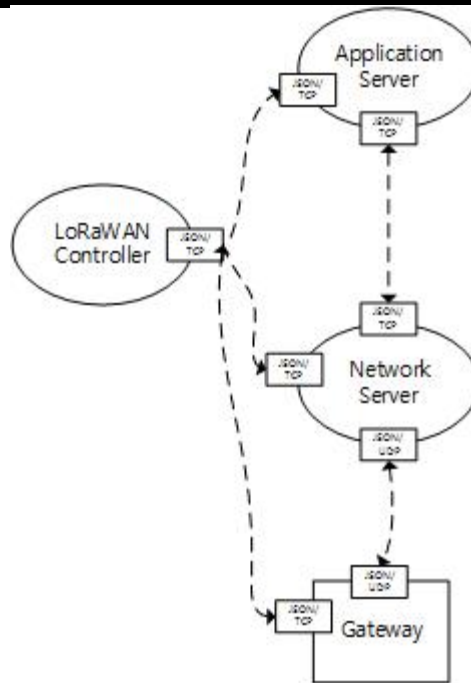


그림 3 communication diagram

JSON String	
LoRa Gateway Message Protocol	LoRa Server Message Protocol
UDP	TCP
Internet Protocol	

그림 4 Protocol Stack

### 2.1.1. UDP , TCP

o Gateway와 Network Server를 제외한 모든 통신은 TCP로 이뤄진다. 통신에 사용

되는 포트를 정리하면 아래의 표와 같다.

표 3 TCP port

	LoRaWAN Controller destination port	Network Server destination port
Network Server	1683	
Application Server	1682	1681
Gateway	1684	UDP

표 4 UDP port

GWMP message type	Direction	Gateway UDP source port	Network Server UDP port	Gateway UDP destination port
PUSH_DATA	To server	Arbitrary	1680	
PUSH_ACK	To gateway			PUSH_DATA 메시지의 source port
PULL_DATA	To server	Arbitrary		
PULL_ACK	To gateway			PULL_DATA 메시지의 source port
PULL_RESP	To gateway			가장최근 PULL_DATA메시 지의 source port

- o 표2는 서버간 TCP포트, LoRaWAN Controller와 gateway간 TCP포트를, 표3은 gateway와 Network Server간의 메시지별 UDP port를 나타낸다.

## 2.2. 메시지 프로토콜

### 2.2.1. Gateway <-> Network Server

- o PUSH\_DATA 메시지는 gateway에서 Network Server로 메시지를 보낼 때 사용되는 메시지이다. Payload는 JSON Object로 구성되며 Device가 보내는 데이터 뿐만 아니라 무선통신에서 발생한 파라미터 정보, Gateway 상태정보 등이 포함된다.

표 5 PUSH\_DATA format

Offset	Number of octets	Function	Value
0	1	Protocol version	0x01 or 0x02
1	2	Token	Arbitrary value set by Gateway
3	1	PUSH_DATA identifier	0x00
4	8	Gateway EUI	
12		JSON object	

o PUSH\_ACK 메시지는 Network Server가 PUSH\_DATA 메시지를 정상적으로 수신했을 때 gateway로 보내는 메시지이다. PUSH\_ACK 메시지에는 응답하고자 하는 PUSH\_DATA 메시지의 token이 포함되어야 한다.

표 6 PUSH\_ACK format

Offset	Number of octets	Function	Value
0	1	Protocol version	0x01 or 0x02
1	2	Token	Token of the PUSH_DATA
3	1	PUSH_ACK identifier	0x01

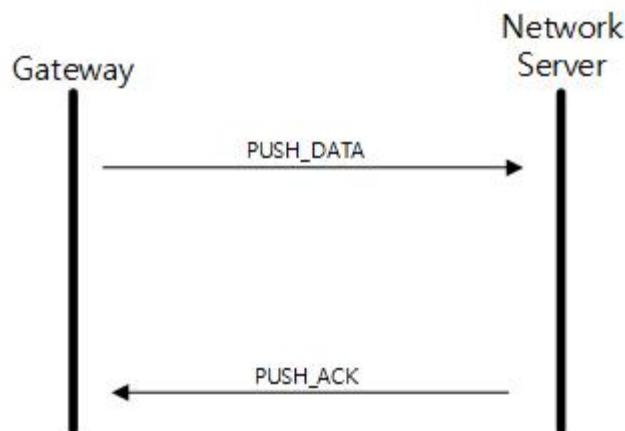


그림 5 Upstream GWMP sequence diagram

o PULL\_RESP 메시지는 Network Server에서 Gateway로 메시지를 보낼 때 사용되는 메시지이다. Protocol이 V2인 경우 PUSH\_ACK 메시지에는 응답하고자 하는 PUSH\_DATA 메시지의 token이 포함되어야 한다.

표 7 PULL\_RESP format

Offset	Number of octets	Function	Value
0	1	Protocol version	0x01 or 0x02

1	2	Unused(V1)/Token(V2)	Token of the PULL_DATA
3	1	PULL_DATA identifier	0x03
4		JSON object	

- o TX\_ACK 메시지는 GWMP V2에서만 사용되는 메시지로 Gateway에서 Network Server로 PULL\_RESP 메시지를 정상 수신했음을 알리는 메시지이다. Token 필드에는 응답하고자하는 PULL\_RESP의 token값이 포함되어야 한다.

표 8 TX\_ACK format

Offset	Number of octets	Function	Value
0	1	Protocol version	0x01 or 0x02
1	2	Token	Token of the PULL_RESP
3	1	TX_ACK identifier	0x05
4		JSON object	

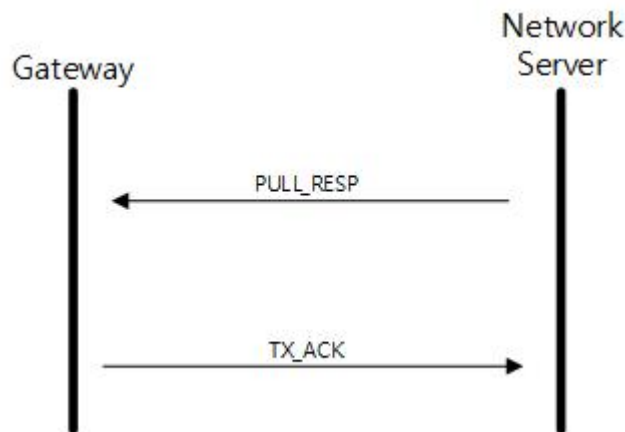


그림 6 Downstream GWMP sequence diagram

- o PULL\_DATA 메시지는 Network Server가 PULL\_RESP 메시지를 보낼 때 사용할 UDP 포트를 알려주기 위한 메시지이다.

표 9 PULL\_DATA format

Offset	Number of octets	Function	Value
--------	------------------	----------	-------

0	1	Protocol version	0x01 or 0x02
1	2	Token	Arbitrary value set by Gateway
3	1	TX_ACK identifier	0x05
4		JSON object	

- o PULL\_ACK 메시지는 Network Server가 PULL\_DATA를 수신했음을 알리는 메시지이다. Token 필드에는 응답하고자하는 PULL\_DATA의 token값이 포함되어야 한다.

표 10 PULL\_ACK format

Offset	Number of octets	Function	Value
0	1	Protocol version	0x01 or 0x02
1	2	Token	Token of PULL_DATA
3	1	TX_ACK identifier	0x05
4		JSON object	

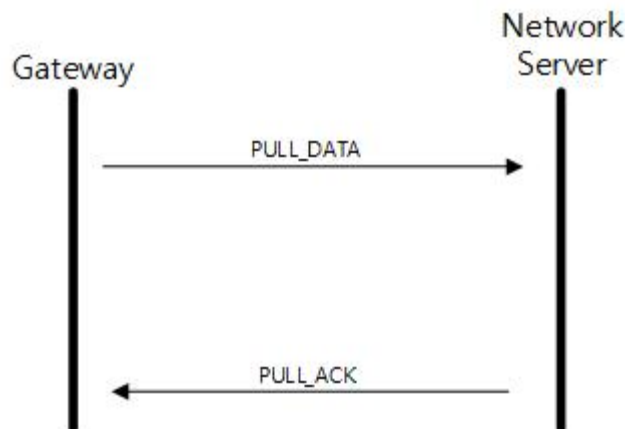


그림 7 Gateway PULL\_DATA diagram

## 2.2.2. LoRaWAN Controller <-> Servers , gateway

- o HELLO\_GW 메시지는 LoRaWAN Controller가 Gateway에게 Network Server의 IP 주소를 알려주기 위한 메시지이다.

표 11 HELLO\_GW format

Offset	Number of octets	Function	Value
0	4	total payload length	
1	2	Protocol version	0
3	1	HELLO_GW identifier	0x02
4		JSON object	

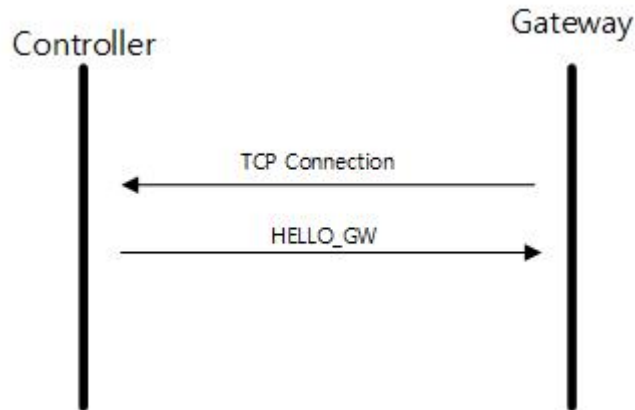


그림 8 HELLO\_GW diagram

- o HELLO\_NS 메시지는 LoRaWAN Controller가 Network Server 에게 Gateway 와 Application Server의 IP 주소 , Application Server에 해당하는 APP EUI 를 알려 주기 위한 메시지이다. APP EUI는 Application Server를 구분하는 식별자로 사용 된다.

표 12 HELLO\_NS format

Offset	Number of octets	Function	Value
0	4	total payload length	
1	2	Protocol version	0
3	1	HELLO_GW identifier	0x03
4		JSON object	

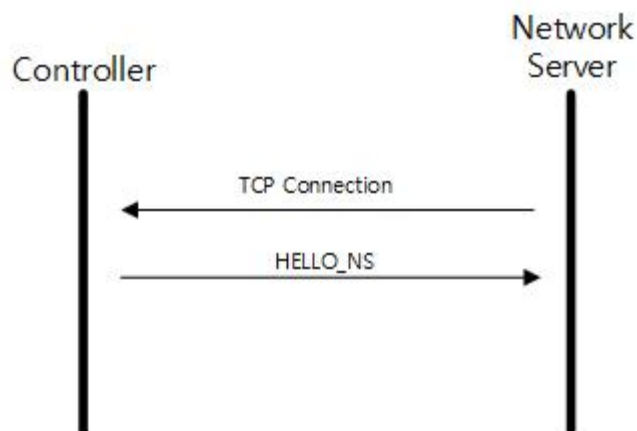


그림 9 HELLO\_NS diagram

- o HELLO\_APP 메시지는 LoRaWAN Controller가 Application Server에게 Network Server의 IP 주소를 알려주는 메시지이다.

표 13 HELLO\_APP format

Offset	Number of octets	Function	Value
0	4	total payload length	
1	2	Protocol version	0
3	1	HELLO_GW identifier	0x04
4		JSON object	

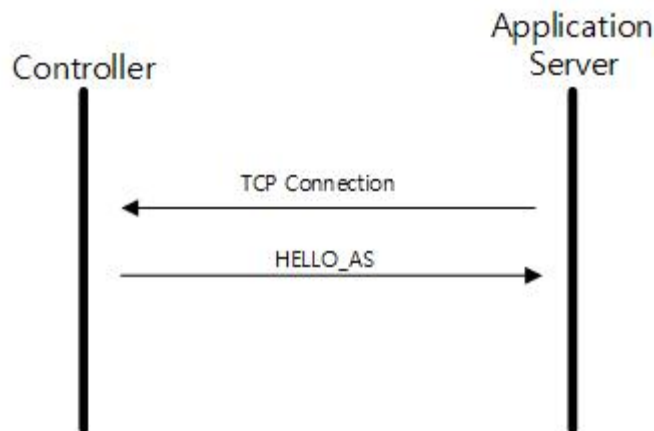


그림 10 HELLO\_AS diagram

### 2.2.3. Network Server <-> Application Server

- o APP\_DATA\_UP 메시지는 Network Server에서 Application Server로 메시지를 보낼 때 사용하는 메시지이다.

표 14 APP\_DATA\_UP format

Offset	Number of octets	Function	Value
0	1	Protocol Version	0
1	1	APP_DATA_UP identifier	0



2	4	Device Address	0x00
6		JSON object	

- o APP\_DATA\_DOWN 메시지는 Network Server에서 Application Server로 메시지를 보낼 때 사용하는 메시지이다.

표 15 APP\_DATA\_DOWN format

Offset	Number of octets	Function	Value
0	1	Protocol Version	
1	1	APP_DATA_DOWN identifier	0
2	4	Device Address	0x01
6		JSON object	

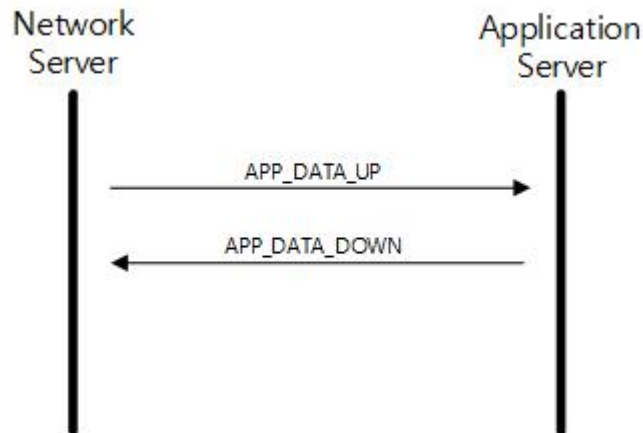


그림 11 APP\_DATA diagram

## 2.3. JSON 프로토콜

### 2.3.1. Gateway <-> Network Server

- o stat ( Gateway -> Network Server )

표 16 stat JSON object

Name	Required	Type	Function
time	No	string	UTC system time of the

			gateway. (ISO 8601)
lati	No	float	The latitude of the gateway
long	No	float	The longitude of the gateway
alti	No	signed integer	The altitude of the gateway
rxnb	No	unsigned integer	The number of radio frames received
rxok	No	unsigned integer	The number of radio frames received with correct CRC
rxfw	No	unsigned integer	The number of radio frames forwarded to network server
ackr	No	unsigned integer	Proportion of radio frames acknowledged
dwnb	No	unsigned integer	The number of radio frames received
txnb	No	unsigned integer	The number of radio frames transmitted

o rxpk ( Gateway -> Network Server )

표 17 rxpk JSON object

Name	Required	Type	Function
time	No	String	UTC system time of the receipt of the LoRa frame (ISO 8601)
tmst	Yes	unsigned integer	The value of the gateway internal time counter
freq	Yes	unsigned float	The frequency of the signal
chan	Yes	unsigned integer	Concentrator "IF" channel
rfch	Yes	unsigned integer	Concentrator radio frequency chain
stat	Yes	signed integer	The result of the gateway's CRC test on the frame
modu	Yes	string	The modulation technique used
datr	Yes	string	Datarate identifier
codr	Yes	string	ECC code rate
rssi	Yes	signed integer	The measured received signal strength in dBm
lsnr	Yes	signed float	The measured received signal to noise ratio in dBm
size	No	unsigned integer	the number of octets in

			the received frame
data	Yes	string	The frame payload encoded into Base64

o txpk ( Network Server -> Gateway )

표 18 txpk JSON object

Name	Required	Type	Function
imme	No	boolean	If true, the gateway is commanded to transmit the frame immediately
tmst	No	unsigned integer	If "imme" is not true and "tmst" is present, the gateway is commanded to transmit the frame when its internal timestamp counter equals the value of "tmst"
time	Yes	string	If "imme" is false or not present and "tmst" is not present, the gateway is commanded to transmit the frame at this time
freq	No	unsigned float	The centre frequency on when the frame is to be transmitted in units of MHz
rfch	Yes	unsigned integer	The antenna on which the gateway is commanded to transmit the frame
powe	No	signed integer	The output power which what the gateway is commanded to transmit the frame
modu	No	string	The modulation technique to be used
datr	No	string	Datarate identifier
codr	Yes	string	ECC code rate. "codr" comprises the string "k/n", where 'k' represents the carried bits and 'n' the total number of bits transmitted
ipol	Yes	bool	If true, commands gateway to invert the polarity of the transmitted bits

size	No	unsigned integer	The number of octets in the received frame
data	No	string	The frame payload, encoded into Base64
ncrc	No	bool	If not false, disable physical layer CRC generation by the transmitter

### 2.3.2. Controller -> Servers , gateway

o “NS” , “APP” , “GW” , “DEV”

표 19 JSON object between Controller and servers,gateway

Name	Required	Type	Function
ip_address	except “DEV”	string	ip address
APP_EUI	only for APP	string	Application identifier
DEV_EUI	only for DEV	string	Device identifier

### 2.3.3. Network Server -> Application Server

표 20 JSON object between Network Server and Application Server

Name	Required	Type	Function
fport	Yes	byte	key for encryption
data	Yes	float	Data of the device
long	Yes	float	The longitude of the gateway
lati	Yes	float	The latitude of the gateway
gwEUI	Yes	string	Gateway identifier

## 3. LoRaWAN Network Server

### 3.1. 기능

o Network Server는 LoRaWAN device와의 통신을 위한 가장 핵심적인 기능들을 수행 한다. 현재는 Device의 인증 ( Activation By Personalization ) , LoRaWAN device의 데이터 암호화 및 복호화 , uplink 통신(network server -> application server )시 해당 application server로 포워딩, downlink 통신( network server -> device )시 통신 상태를 고려한 gateway 선정, gateway 부하 모니터링 등의 기능

을 수행하고 있다.

- o 앞으로 추가되어야 할 기능으로는 첫 번째로 원활한 무선구간 통신을 위한 Class 별 MAC Command를 통한 device 제어 기능이 추가 되어야 한다. 그리고 device activation을 위한 OTAA ( Over-the-Air Activation )기능 또한 추가되어야 한다.

### 3.2. HW 시스템 환경

- o 테스트에 사용한 OpenStack flavor는 다음 표와 같다.

표 21 Network Server용 flavor

VCPUs	RAM(MB)	Root Disk(GB)	Ephemeral Disk(GB)	Swap Disk (MB)
1	1024	5	1	0

### 3.3. SW 시스템 환경

- o 테스트에 사용한 SW 시스템 환경은 다음과 같다.

Network Server
Java 1.8
Ubuntu Server 14.04 LTS ( Trusty Tahr )
QEMU ( OpenStack Liberty stable )
Ubuntu Server 14.04.3
Vmware Workstation
Windows 7

그림 12 Network Server 시스템 환경

### 3.4. 운용방법 및 검증

- o Horizon의 “ Create LoRaWAN Network ” 탭을 이용하여 망을 생성하면 ubuntu 계정의 디렉토리에 LoRaWAN\_heat\_script.yaml 파일이 생성되며 heat stack을 생성한다. LoRaWAN Controller는 입력받은 정보들을 토대로 연결을 허용해줄 gateway , application server 목록들을 Network Server로 전달한다. Network Server는 전달받은 목록들의 연결만 허용한다. 망을 생성하는데 사용한 heat template 에는 network server가 부팅이 완료된 후에 자동으로 구동할 수 있도록 쉘 스크립트를 추가 하였다.

```
ntw:
  type: OS::Nova::Server
  properties:
    name: LoRaWAN Network Server
    image: { get_param: image }
    flavor: { get_param: flavor }
    key_name: { get_param: key_name }
    networks:
      - port: { get_resource: ntw_port }
    user_data_format: RAW
    user_data: |
      #!/bin/sh
      echo start > /home/ubuntu/bootup.log
      sudo apt-get update
      sudo apt-get install sshpass -y
      sshpass -p qwer1234 scp -o StrictHostKeyChecking=no stack@192.168.10.19:/home/stack/ns.jar /home/ubuntu/ns.jar
      /etc/jre1.8.0_74/bin/java -jar /home/ubuntu/ns.jar 192.168.10.19 > /home/ubuntu/bootup.log
      echo end > /home/ubuntu/end.log
```

그림 13 heat template의 network server 쉘 스크립트

- o 추가한 쉘 스크립트는 SCP를 통해 구동에 필요한 network server JAR 파일을 받아오고 Controller의 주소를 argument로 구동을 하게 한다. Network server가 정상적으로 구동중인지를 확인하려면 현재 열려있는 포트를 조회하면 알 수 있다. 1680 ( gateway ) , 1683 ( Controller ) , 1681 ( Application Server ) 들이 목록에 보이면 연결이 정상적으로 맺어진 상태이다.

```
ubuntu@lorawan-network-server:~$ netstat -nap
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (servers and established)

```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	-
tcp	0	0	20.2.0.12:22	192.168.10.19:40073	ESTABLISHED	-
tcp6	0	0	:::1681	:::*	LISTEN	-
tcp6	0	0	:::22	:::*	LISTEN	-
tcp6	0	0	20.2.0.12:60019	192.168.10.19:1683	ESTABLISHED	-
tcp6	0	0	20.2.0.12:1681	192.168.10.178:39783	ESTABLISHED	-
tcp6	0	0	20.2.0.12:1681	192.168.10.179:52242	ESTABLISHED	-
udp	0	0	0.0.0.0:25100	0.0.0.0:*	-	-
udp	0	0	0.0.0.0:68	0.0.0.0:*	-	-
udp6	0	0	:::38186	:::*	-	-
udp6	0	0	:::1680	:::*	-	-

```
Active UNIX domain sockets (servers and established)
```

그림 14 Network Server의 구동확인

- o 구동관련 로그를 실시간으로 확인하려면 Network Server 프로세스를 kill 명령어로 종료시킨 후, SCP로 받아온 ns.jar파일을 controller 주소로 argument로 하여 재 실행 하면 관련 로그를 확인할 수 있다.

```
ubuntu@lorawan-network-server:~$ java -jar ns.jar 192.168.10.19
[main] manager IP :192.168.10.19
Manager is connected
gwstatelistsize :0
devstatelistsize :0
appstatelistsize :0
[MANAGER]msglength - 44
[MANAGER]msg from manager
[MANAGER] HELLO MSG (new GW)
[MANAGER]json-{"GW":[{"ip_address":"192.168.10.2"}]}
[MANAGER]msglength - 135
[MANAGER]msg from manager
[MANAGER] HELLO MSG (new APP)
[MANAGER]json-{"APP":[{"ip_address":"192.168.10.178","APP_EUI":"0,0,0,0,0,0,0,0"}, {"ip_address":"192.168.10.179","APP_EUI":"0,0,0,0,0,0,0,0"}]}
```

그림 15 Network Server 구동 로그

## 4. LoRaWAN Gateway

### 4.1.1. 기능

- o LoRaWAN Gateway는 device와 Network Server사이에서 단순히 데이터를 포워딩하는 역할을 수행한다. LoRaWAN Controller가 HELLO\_GW 메시지를 통해 데이터를 포워딩해야할 목적지의 주소를 전달하고 이를 수신한 Gateway는 해당하는 목적지의 주소로 device의 데이터를 포워딩 한다. 이때 무선 통신 구간의 상태를 짐작할 수 있는 파라미터 값들을 추가로 JSON형태로 추가하여 Network Serrver로 포워딩 한다. 현재 데이터 포워딩을 제외하고 특별한 기능을 수행하고 있지는 않다.

### 4.1.2. HW 시스템 환경

o kerlink 사의 LoRa IoT Station 868MHz



그림 16 LoRa IoT Station

#### 4.1.3. SW 시스템 환경

o Packet\_forwarder application은 Semtech에서 제공하는 LoRa HAL을 기반으로 동작한다. Packet\_forwarder application과 LoRa HAL은 github에 공개되어 있으며 추가로 device관련 예제 코드도 공개되어있다.



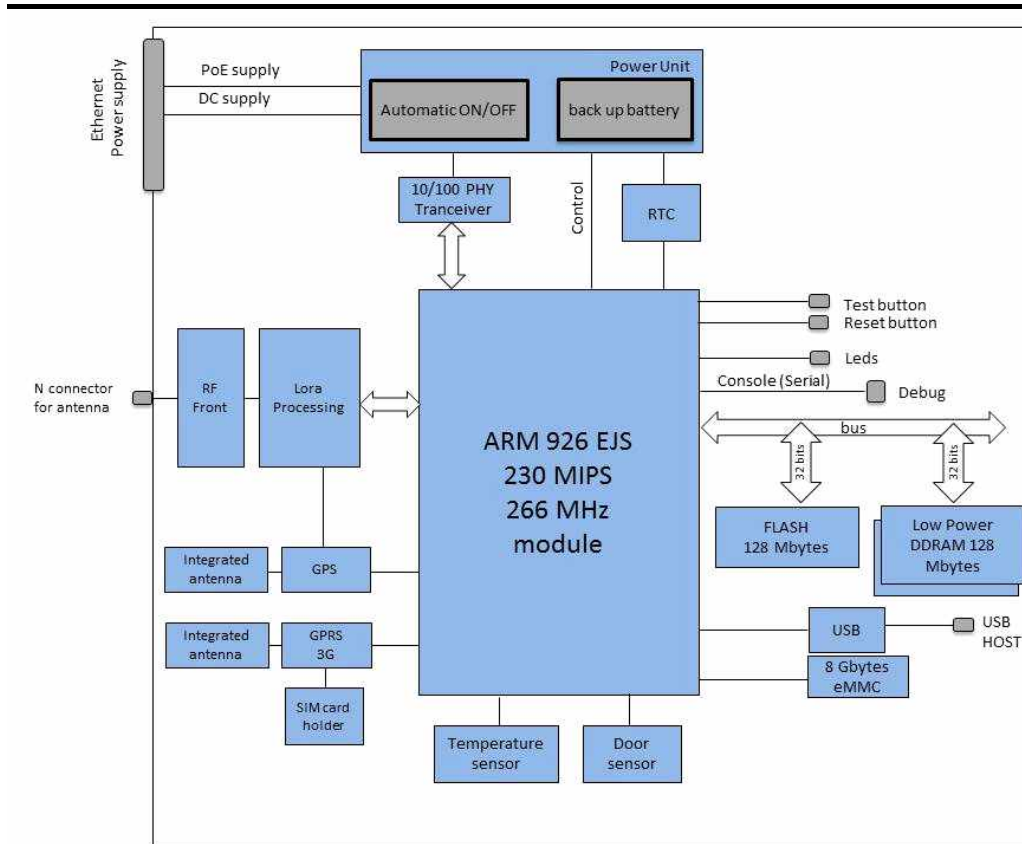


그림 17 Hardware architecture of LoRa IoT Station

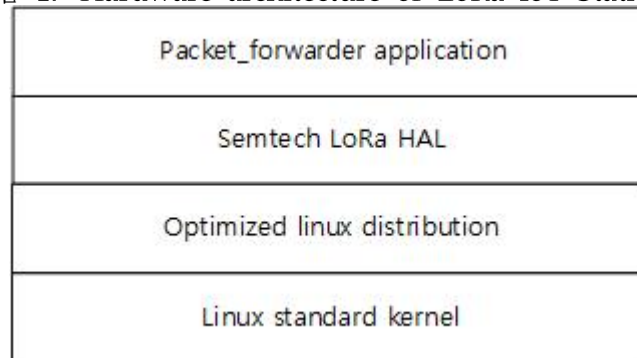


그림 18 Gateway SW 시스템 환경 #1

o kerlink IoT Station에서 사용되는 리눅스 커널과 펌웨어는 <http://wikierlink.fr/lora-station> 에 공개 되어 있으며 현재 실험환경 구성에 사용된 버전은 다음과 같다.

#### 4.1.4. 운용방법 및 검증

```
[root@Wirgrid_0806038d /root]# get_version -u -v
KERNEL_VER=3.10.37-klk-d26c52
PIC_VER=8.3
BOOTSTRAP_VER=
UBOOT_VER=1610
SCRIPT_VER=1610
INITRAMFS_VER=g4b3543a
FILESYSTEM_VER=2011.08-gdbb0f32
KNETD_VER="wirma2_v3.10 WAN_3.15 (Apr 14 2015-11:18:56)"
PROD_FW=wirmaV2_wirgrid_v2.2
LORABOARD_MANUFACTURER=00
LORABOARD_TYPE="868-27dBm"
LORABOARD_HWVERSION=05
LORABOARD_SERIALNO=00201F
```

그림 19 Gateway SW 시스템 환경 #2

- o Semtech에서 제공하는 순수한 Packet\_forwarder application의 경우 처음 실행하면 config 파일을 읽어서 데이터를 포워딩할 목적지를 정한다. 우리는 데이터의 목적지를 LoRaWAN Controller로부터 받아올 수 있도록 어플리케이션이 실행되면 Controller로 TCP 소켓통신을 하여 HELLO\_GW 메시지를 받는 절차를 추가 하였다. Controller로부터 HELLO\_GW 메시지를 정상적으로 수신한다면 아래와 같은 로그를 확인할 수 있다.

```
192.168.10.2 - PuTTY
INFO: try connection .
INFO: connection complete .
INFO: read size 44
,{"NS":{"ip_address":"192.168.10.177"}}INFO: 1server hostname or IP address is c
onfigured to "192.168.10.177"
INFO: 2server hostname or IP address is configured to "192.168.10.177"
INFO: [main] concentrator started, packet can now be received
WARNING: [gps] GPS out of sync, keeping previous time reference
WARNING: [gps] GPS out of sync, keeping previous time reference
INFO: [down] PULL_ACK received in 9 ms
```

그림 20 Gateway log #1

- o Device의 데이터와 gateway의 gps정보가 정상적으로 수신되면 아래와 같은 로그를 확인할 수 있다.

```
##### 2016-03-29 11:41:46 GMT #####  
### [UPSTREAM] ###  
# RF packets received by concentrator: 1  
# CRC_OK: 100.00%, CRC_FAIL: 0.00%, NO_CRC: 0.00%  
# RF packets forwarded: 1 (17 bytes)  
# PUSH_DATA datagrams sent: 2 (390 bytes)  
# PUSH_DATA acknowledged: 100.00%  
### [DOWNSTREAM] ###  
# PULL_DATA sent: 1 (100.00% acknowledged)  
# PULL_RESP(onse) datagrams received: 1 (167 bytes)  
# RF packets sent to concentrator: 1 (17 bytes)  
# TX errors: 0  
### [GPS] ###  
# Valid time reference (age: 1 sec)  
# GPS coordinates: latitude 37.62045, longitude 127.05935, altitude 52 m  
##### END #####  
INFO: [up] PUSH_ACK received in 2 ms
```

그림 21 Gateway log #2

## References

- [1] LoRaWAN Specification V1.0 , <https://www.lora-alliance.org/>
- [2] LoRaWAN Network Server Demonstration: Gateway to Server Interface Definition , <http://iot.semtech.com/>
- [3] Installation of OpenStack All-In-One Single Machine , <http://docs.openstack.org/developer/devstack/guides/single-machine.html>
- [4] Heat template examples  
,<https://github.com/openstack/heat-templates/tree/master/hot>

## *K-ONE* 기술 문서

- K-ONE 컨소시엄의 확인과 허가 없이 이 문서를 무단 수정하여 배포하는 것을 금지합니다.
- 이 문서의 기술적인 내용은 프로젝트의 진행과 함께 별도의 예고 없이 변경될 수 있습니다.
- 본 문서와 관련된 문의 사항은 아래의 정보를 참조하시길 바랍니다.  
(Homepage: <http://opennetworking.kr/projects/k-one-collaboration-project/wiki>, E-mail: [k1@opennetworking.kr](mailto:k1@opennetworking.kr))

작성기관: K-ONE Consortium  
작성년월: 2016/01