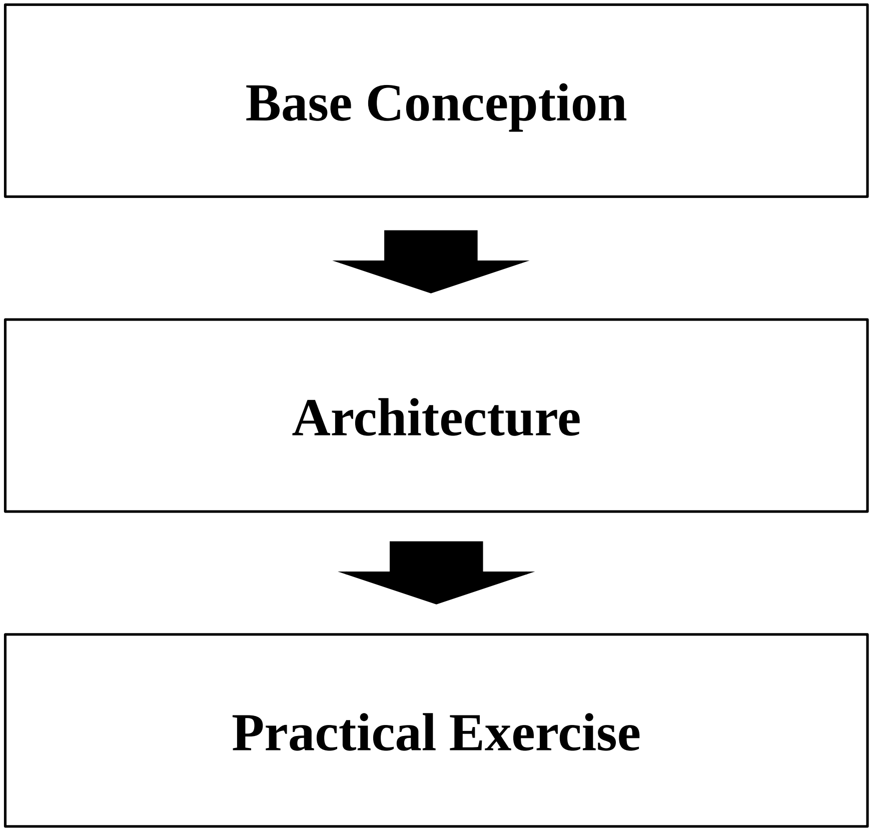
Energy Management System
Virtual Top Node
Overview

MIR Lab
http://mir.hanyang.ac.kr

Lecture Index



Base Conception

1. OpenADR
2. System Architecture

Architecture

3. EMS Overview
4. EMS Program Architecture
5. Package Explanation
6. Message Format

Practical Exercise

7. How to Execute MIR Program (EMS, VTN, EMA)
8. Experiment Procedure
9. Captured Screen as following Procedure

7. How to Execute MIR Program(VTN)

- **Prerequisite**
  - **Language**

    : Python(2.7.13), HTML(Web Page)
  - **Library**

    : UDP, ElementTree(XML Parser), BaseHTTPServer

7. How to Execute MIR Program(VTN)

- **Installation of Python 2.7**

```
mir@:~$ sudo apt-get update
mir@:~$ sudo apt-get install python
```

- **Modify Python XML Parser library**

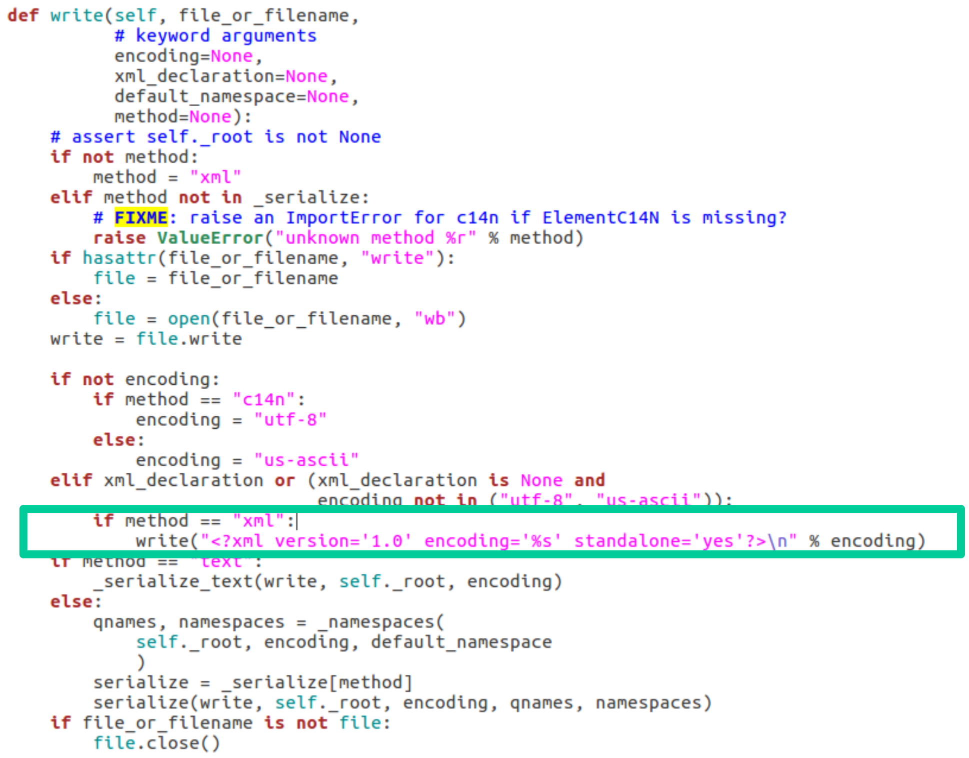
```
mir@:~$ cd /usr/lib/python2.7/xml/etree
mir@:~$ gedit ElementTree.py
```

next Page

7. How to Execute MIR Program(VTN)

- To match with VEN(EPRI) Data Format

  - Move to Line **812**



```
def write(self, file_or_filename,
          # keyword arguments
          encoding=None,
          xml_declaration=None,
          default_namespace=None,
          method=None):
    # assert self._root is not None
    if not method:
        method = "xml"
    elif method not in _serialize:
        # FIXME: raise an ImportError for c14n if ElementC14N is missing?
        raise ValueError("unknown method %r" % method)
    if hasattr(file_or_filename, "write"):
        file = file_or_filename
    else:
        file = open(file_or_filename, "wb")
    write = file.write

    if not encoding:
        if method == "c14n":
            encoding = "utf-8"
        else:
            encoding = "us-ascii"
    elif xml_declaration or (xml_declaration is None and
                             encoding not in ("utf-8", "us-ascii")):
        if method == "xml":
            write("<?xml version='1.0' encoding='%s' standalone='yes'?>\n" % encoding)
    if method == "text":
        _serialize_text(write, self._root, encoding)
    else:
        qnames, namespaces = _namespaces(
            self._root, encoding, default_namespace
            )
        serialize = _serialize[method]
        serialize(write, self._root, encoding, qnames, namespaces)
    if file_or_filename is not file:
        file.close()
```

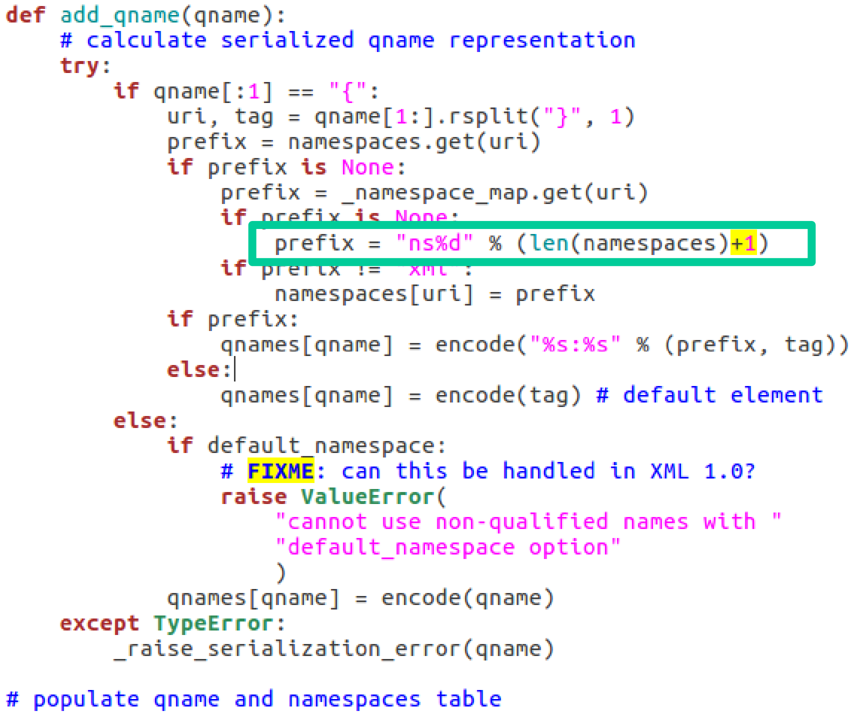
Change this line to

Write (<“?xml version=‘1.0’ encoding=‘%s’ standalone=‘yes’?>\n” % encoding)

7. How to Execute MIR Program(VTN)

- To match with VEN(EPRI) Data Format

  - Move to Line **856**



```
def add_qname(qname):
    # calculate serialized qname representation
    try:
        if qname[:1] == "{":
            uri, tag = qname[1:].rsplit("}", 1)
            prefix = namespaces.get(uri)
            if prefix is None:
                prefix = _namespace_map.get(uri)
                if prefix is None:
                    prefix = "ns%d" % (len(namespaces)+1)
                if prefix != "xml":
                    namespaces[uri] = prefix
            if prefix:
                qnames[qname] = encode("%s:%s" % (prefix, tag))
            else:
                qnames[qname] = encode(tag) # default element
        else:
            if default_namespace:
                # FIXME: can this be handled in XML 1.0?
                raise ValueError(
                    "cannot use non-qualified names with "
                    "default_namespace option"
                    )
            qnames[qname] = encode(qname)
    except TypeError:
        _raise_serialization_error(qname)

# populate qname and namespaces table
```

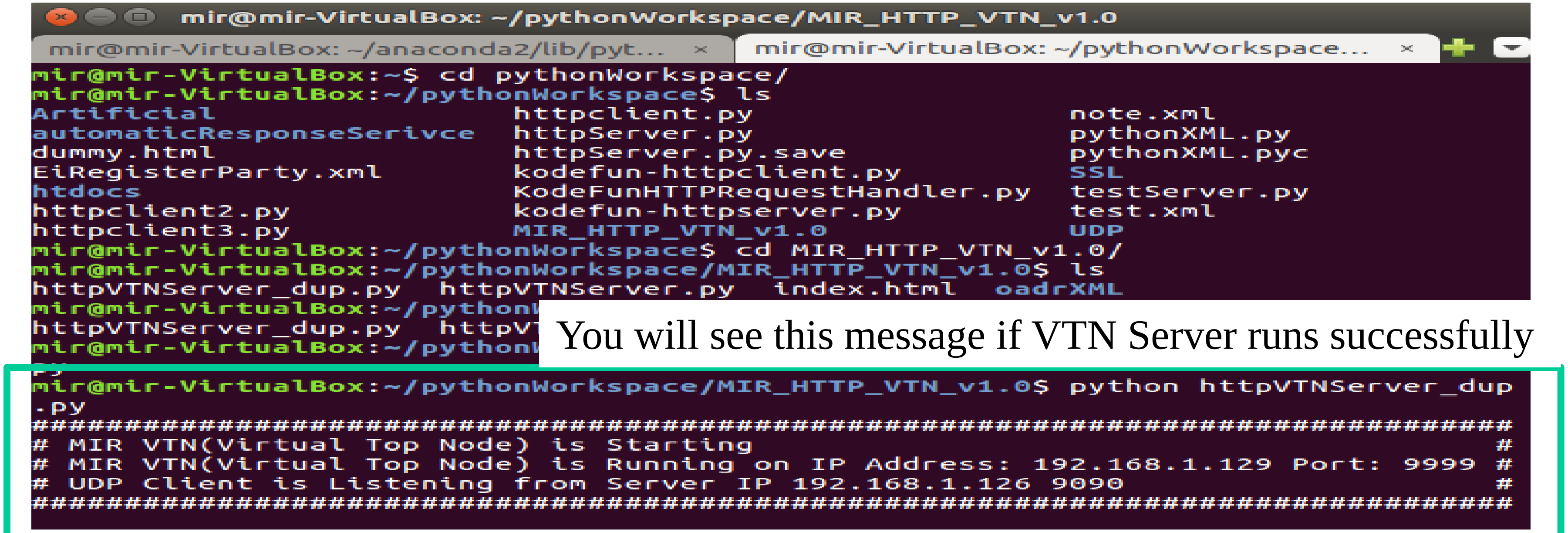
Change this line to
Prefix = “ns%d” % (len(namespaces)+1)

7. How to Execute MIR Program(VTN)

- Download VTN(MIR version)

- Move to Directory that VTN was downloaded

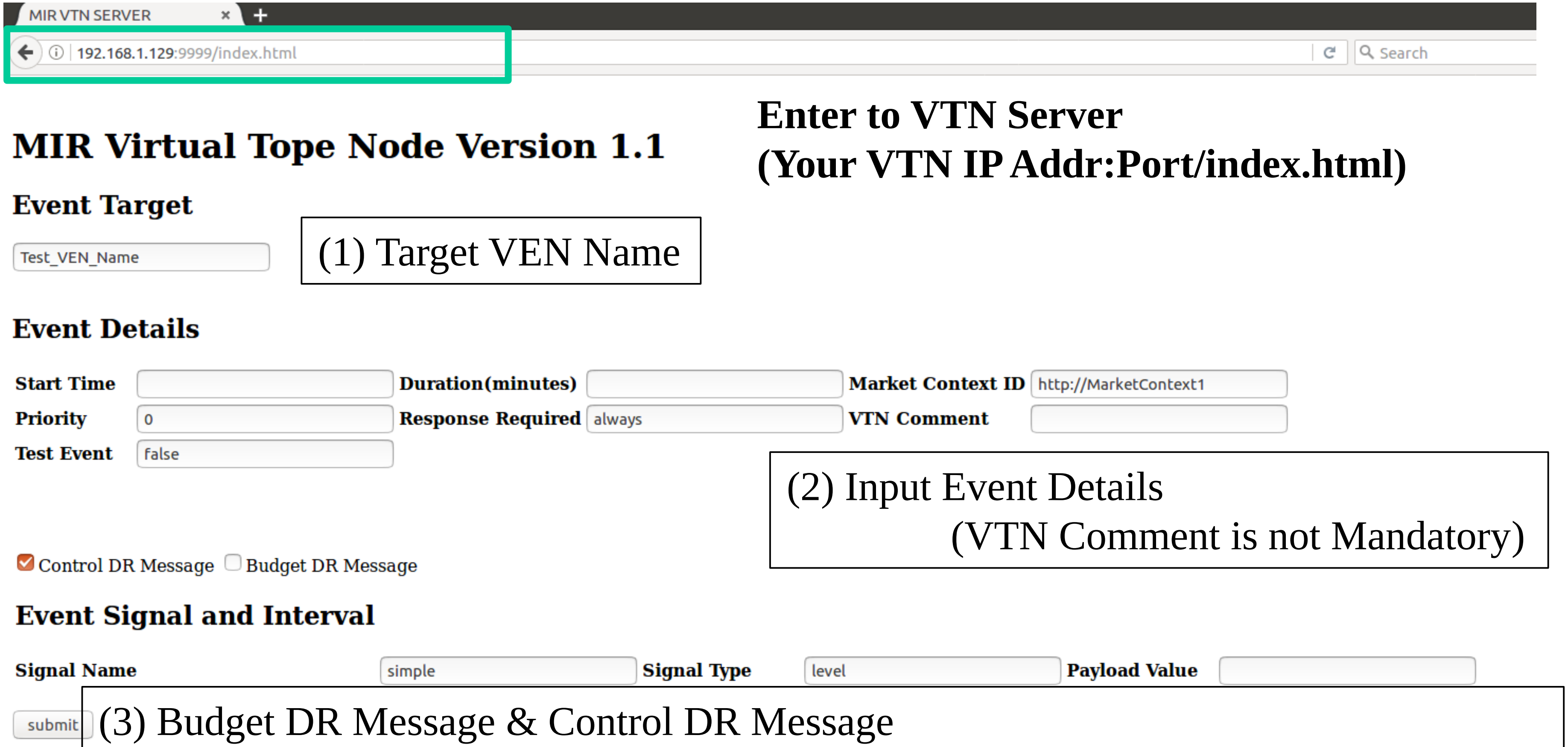
- mir@:~$ python httpVTNServer.py


mir@mir-VirtualBox: ~/pythonWorkspace/MIR_HTTP_VTN_v1.0
mir@mir-VirtualBox: ~/anaconda2/lib/pyt...
mir@mir-VirtualBox: ~/pythonWorkspace...

```
mir@mir-VirtualBox:~$ cd pythonWorkspace/
mir@mir-VirtualBox:~/pythonWorkspace$ ls
Artificial                httpclient.py                 note.xml
automaticResponseSerivce  httpServer.py                 pythonXML.py
dummy.html                httpServer.py.save            pythonXML.pyc
EiRegisterParty.xml       kodefun-httpclient.py         SSL
htdocs                    KodeFunHTTPRequestHandler.py  testServer.py
httpclient2.py            kodefun-httpserver.py         test.xml
httpclient3.py            MIR_HTTP_VTN_v1.0             UDP
mir@mir-VirtualBox:~/pythonWorkspace$ cd MIR_HTTP_VTN_v1.0/
mir@mir-VirtualBox:~/pythonWorkspace/MIR_HTTP_VTN_v1.0$ ls
httpVTNServer_dup.py  httpVTNServer.py  index.html  oadrXML
mir@mir-VirtualBox:~/pythonW
httpVTNServer_dup.py  httpV
mir@mir-VirtualBox:~/pythonW
py
mir@mir-VirtualBox:~/pythonWorkspace/MIR_HTTP_VTN_v1.0$ python httpVTNServer_dup
.py
################################################################################
# MIR VTN(Virtual Top Node) is Starting                                        #
# MIR VTN(Virtual Top Node) is Running on IP Address: 192.168.1.129 Port: 9999 #
# UDP Client is Listening from Server IP 192.168.1.126 9090                    #
################################################################################
```

You will see this message if VTN Server runs successfully

7. How to Execute MIR Program(VTN)



MIR Virtual Tope Node Version 1.1

Event Target

Test_VEN_Name

Event Details

Start Time		**Duration(minutes)**		**Market Context ID**	http://MarketContext1
Priority	0	**Response Required**	always	**VTN Comment**	
Test Event	false				

☑ Control DR Message ☐ Budget DR Message

Event Signal and Interval

Signal Name	simple	**Signal Type**	level	**Payload Value**	

Enter to VTN Server
(Your VTN IP Addr:Port/index.html)

(1) Target VEN Name

(2) Input Event Details
(VTN Comment is not Mandatory)

(3) Budget DR Message & Control DR Message

Budget DR Message → In case when you send Initial & Incentive & Negotiation Price
Control DR Message → Normally when you send Demand Response Message

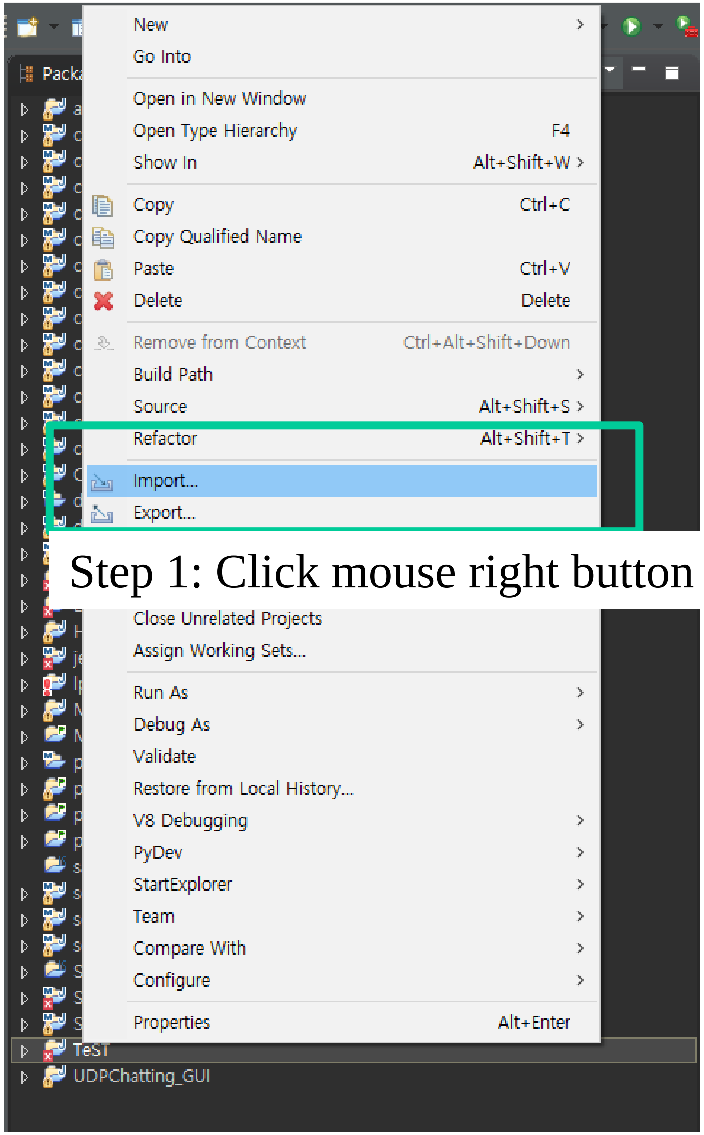
7. How to Execute MIR Program(EMS)

- **Prerequisite**
  - **Language**

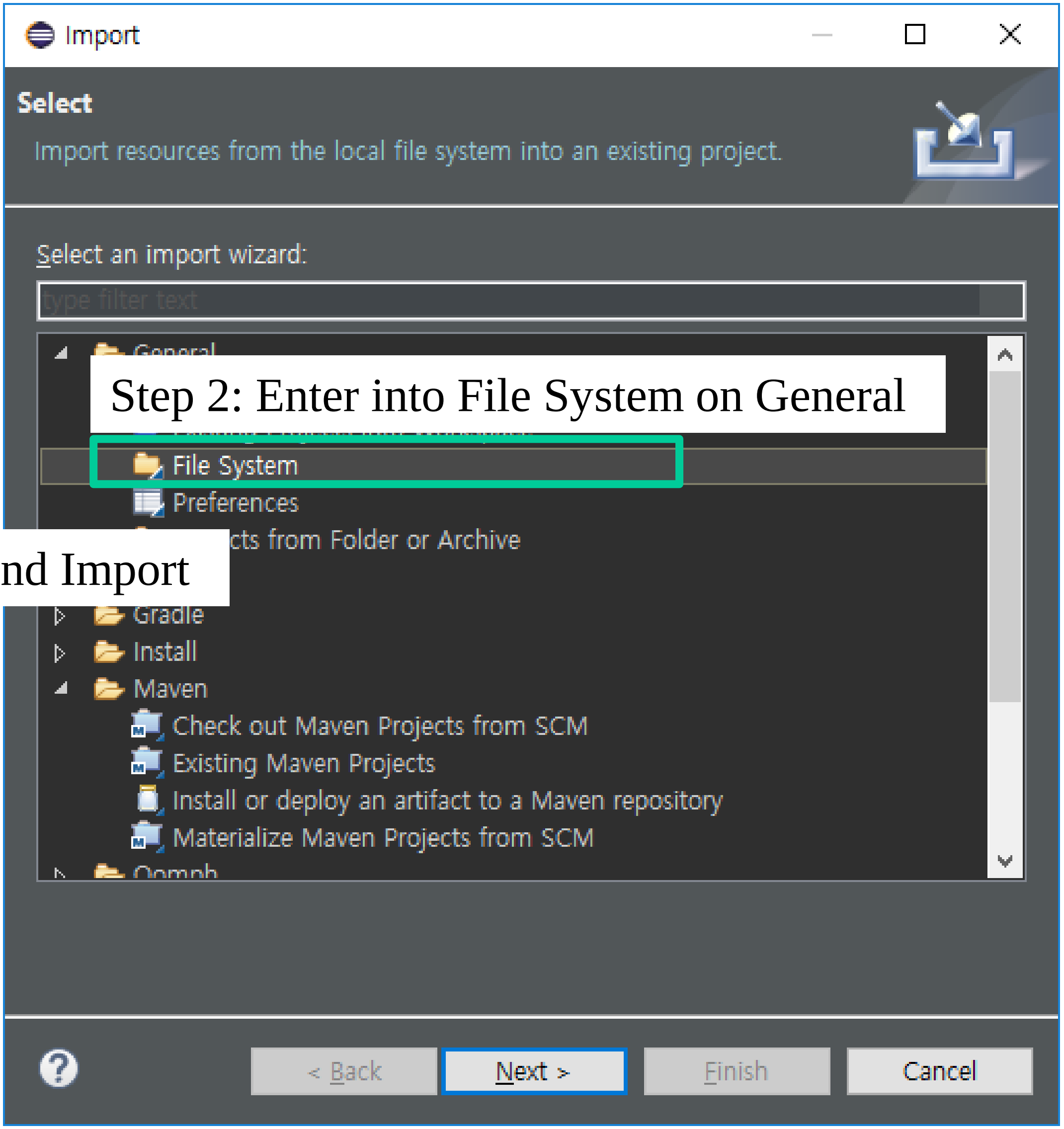
    : Java version 7, Maven
  - **Library**

    : californium-CoAP, paho-MQTT, JSON, x-chart

7. How to Execute MIR Program(EMS)



Step 1: Click mouse right button and Import

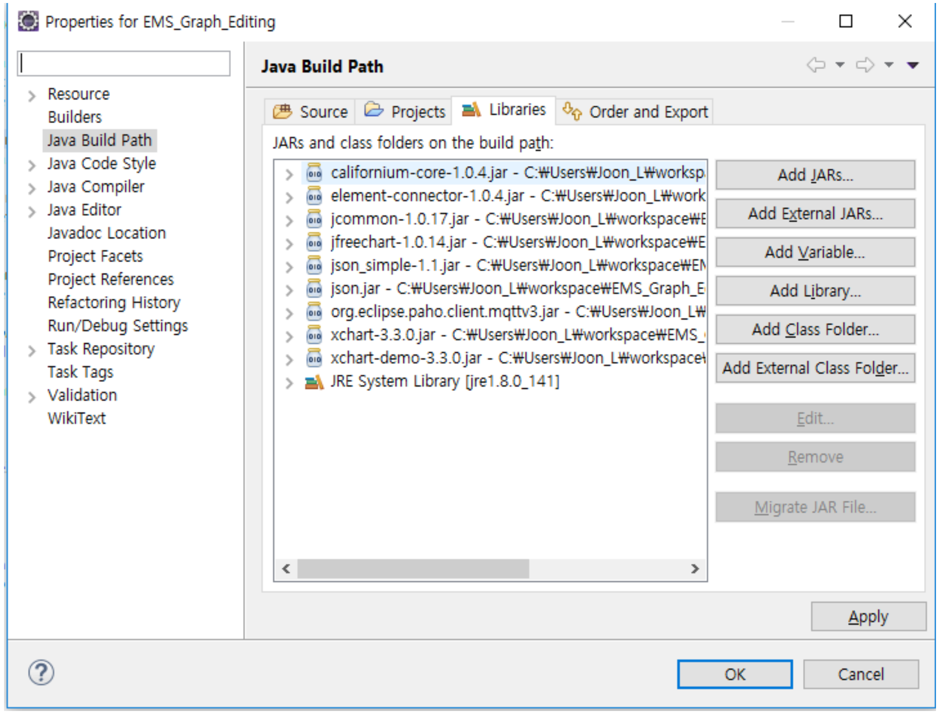


Import resources from the local file system into an existing project.

Step 2: Enter into File System on General

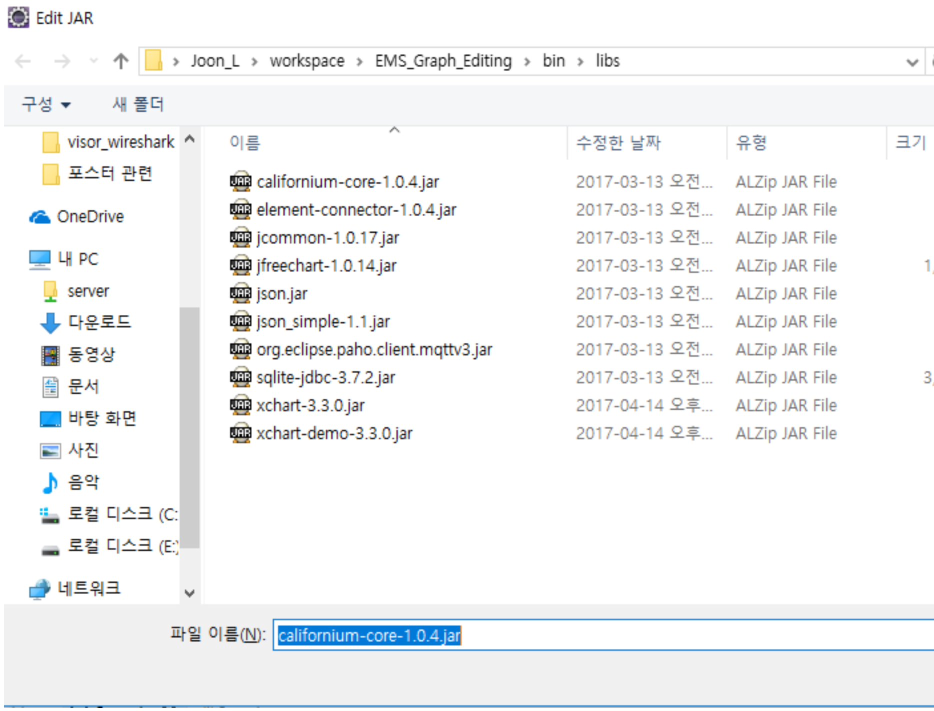
7. How to Execute MIR Program(EMS)

Library 문제 발생시 properties 들어 간 후에 해당 library 선택하여 경로 재설정



JARs and class folders on the build path:

- californium-core-1.0.4.jar - C:₩Users₩Joon_L₩worksp
- element-connector-1.0.4.jar - C:₩Users₩Joon_L₩work
- jcommon-1.0.17.jar - C:₩Users₩Joon_L₩workspace₩E
- jfreechart-1.0.14.jar - C:₩Users₩Joon_L₩workspace₩E
- json_simple-1.1.jar - C:₩Users₩Joon_L₩workspace₩EN
- json.jar - C:₩Users₩Joon_L₩workspace₩EMS_Graph_E
- org.eclipse.paho.client.mqttv3.jar - C:₩Users₩Joon_L₩
- xchart-3.3.0.jar - C:₩Users₩Joon_L₩workspace₩EMS_
- xchart-demo-3.3.0.jar - C:₩Users₩Joon_L₩workspace₩
- JRE System Library [jre1.8.0_141]

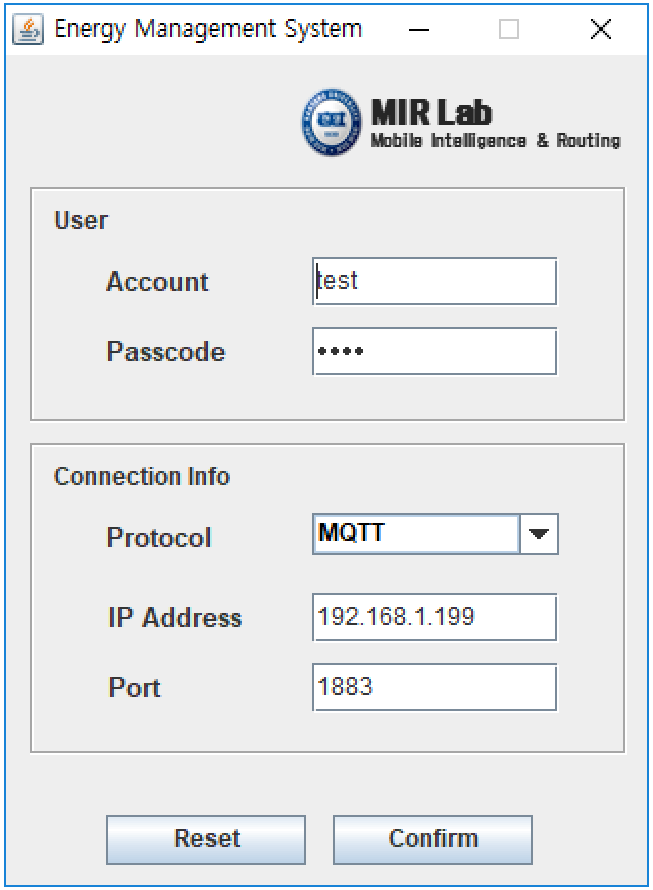


이름	수정한 날짜	유형	크기
californium-core-1.0.4.jar	2017-03-13 오전...	ALZip JAR File	
element-connector-1.0.4.jar	2017-03-13 오전...	ALZip JAR File	
jcommon-1.0.17.jar	2017-03-13 오전...	ALZip JAR File	
jfreechart-1.0.14.jar	2017-03-13 오전...	ALZip JAR File	1,
json.jar	2017-03-13 오전...	ALZip JAR File	
json_simple-1.1.jar	2017-03-13 오전...	ALZip JAR File	
org.eclipse.paho.client.mqttv3.jar	2017-03-13 오전...	ALZip JAR File	
sqlite-jdbc-3.7.2.jar	2017-03-13 오전...	ALZip JAR File	3,
xchart-3.3.0.jar	2017-04-14 오후...	ALZip JAR File	
xchart-demo-3.3.0.jar	2017-04-14 오후...	ALZip JAR File	

파일 이름(N): californium-core-1.0.4.jar

해당 properties 에서 에러 해결 후에 프로젝트 clean 및 refresh

7. How to Execute MIR Program(EMS)



User

Account: test

Passcode: ••••

Connection Info

Protocol: MQTT

IP Address: 192.168.1.199

Port: 1883

Step 3: To execute MIR Energy Management System, you have to execute source code on *EmsMainClass.java.* Then you would see window as you see left side

User Account : test
Passcode : 1234

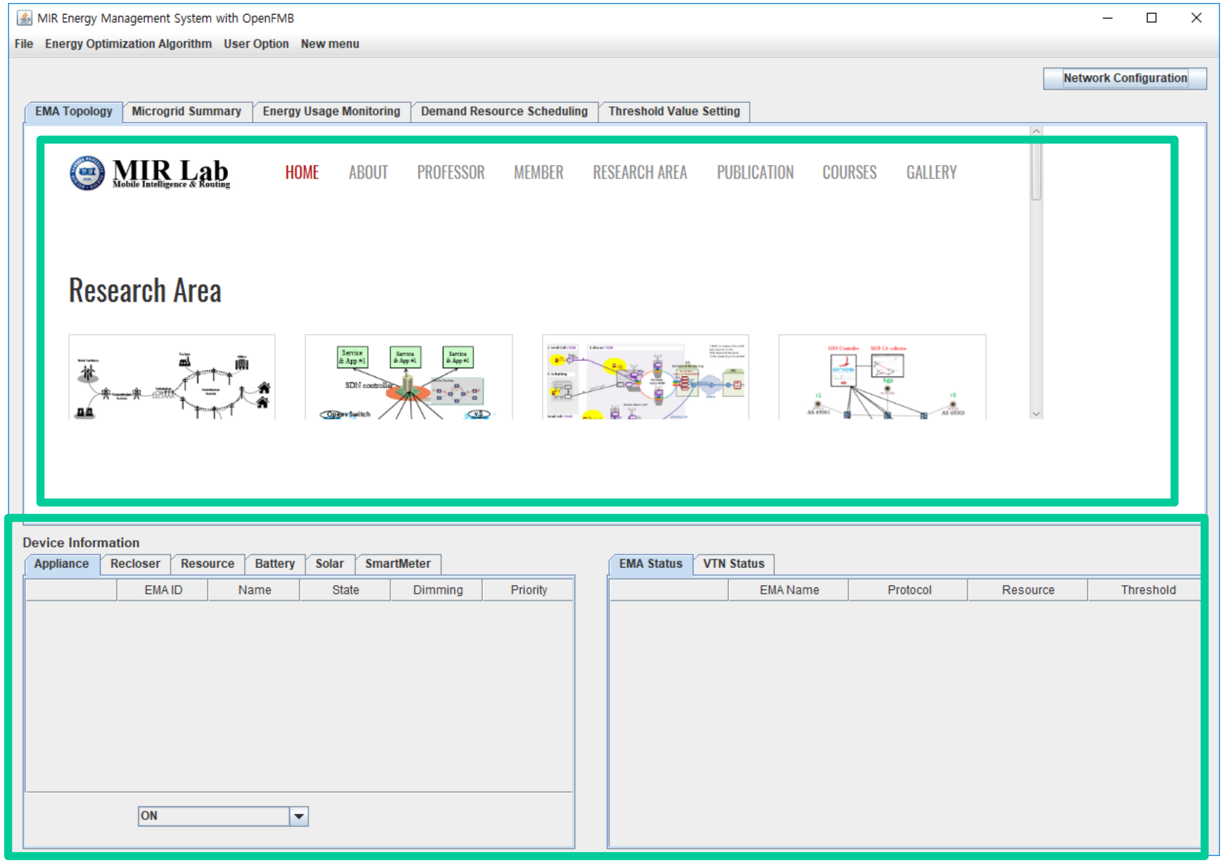
Connection Configuration

Protocol : MQTT, CoAP, UDP(*You can choose*)

IP Address:

MQTT – Should input Broker IP&Port
CoAP & UDP – Don’t need to input (EMS use as Server)

7. How to Execute MIR Program(EMS)



	EMA ID	Name	State	Dimming	Priority

	EMA Name	Protocol	Resource	Threshold

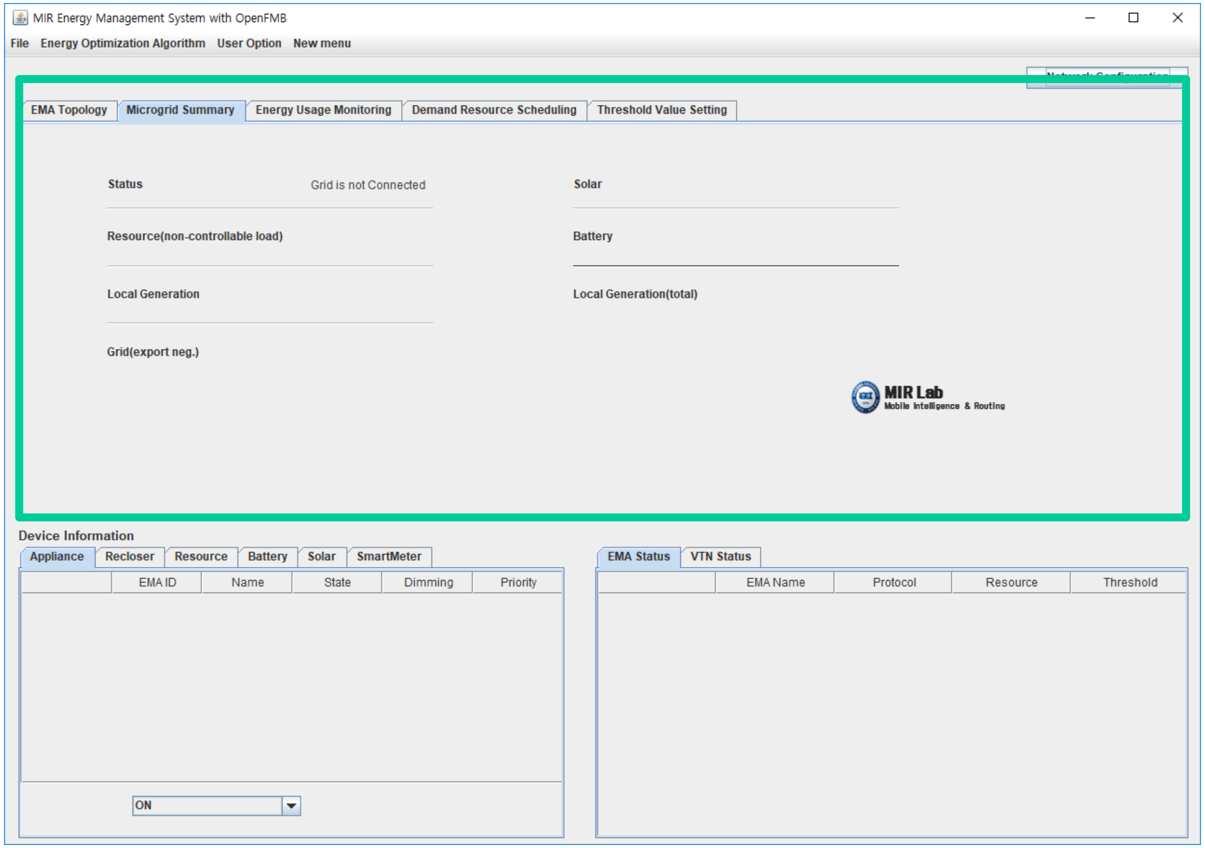
Step 4:

EMA, VTN
Topology Tab(Future Work)

To show the topology graph
Will use *JAVA FX* or *Spring*

Step 5:
EMA, VTN Status, Appliance(e.g. LED), Smart Meter and Micro grid Panel

7. How to Execute MIR Program(EMS)



Status: Grid is not Connected

Resource(non-controllable load)

Local Generation

Grid(export neg.)

Solar

Battery

Local Generation(total)

	EMA ID	Name	State	Dimming	Priority

	EMA Name	Protocol	Resource	Threshold

Step 6:
Micro grid Summary Panel

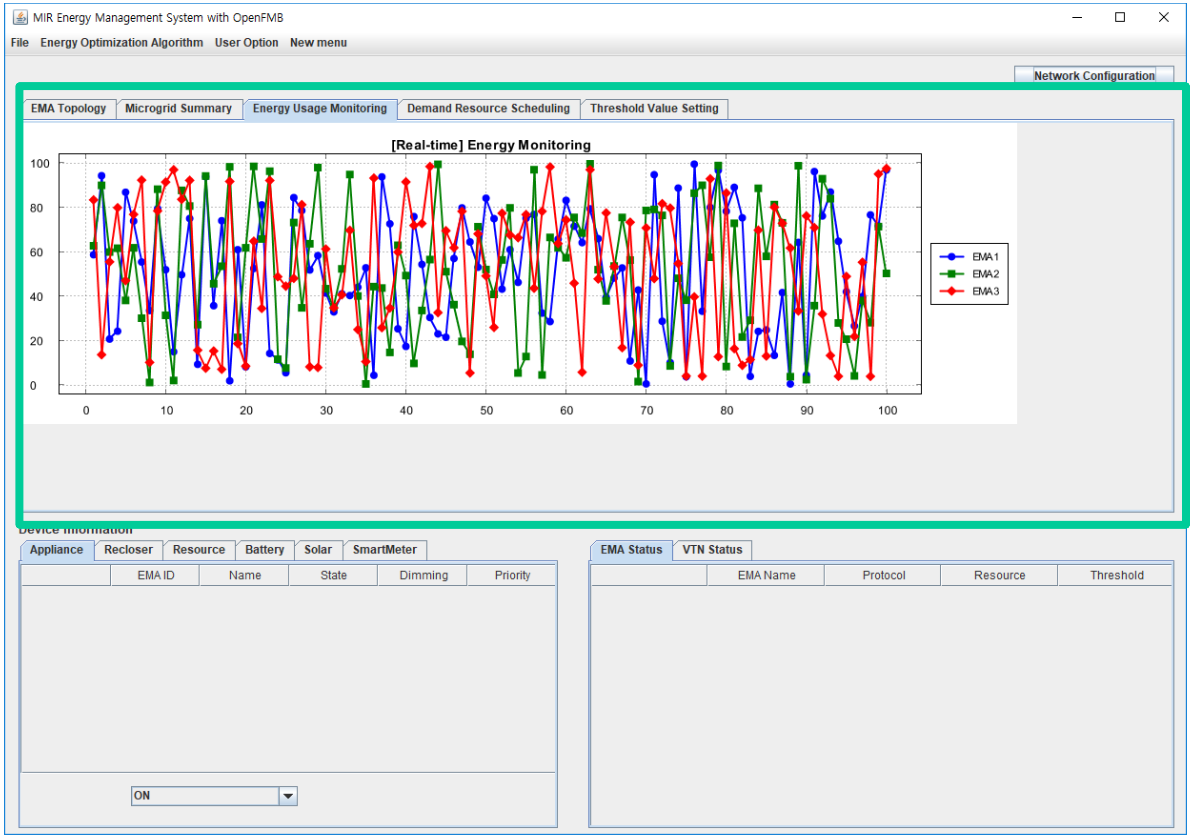
It is possible to check

Generated Energy
Energy Storage System Value
Photovoltaic Value

Consuming Energy
Resource

Export Energy
Grid

7. How to Execute MIR Program(EMS)



	EMA ID	Name	State	Dimming	Priority

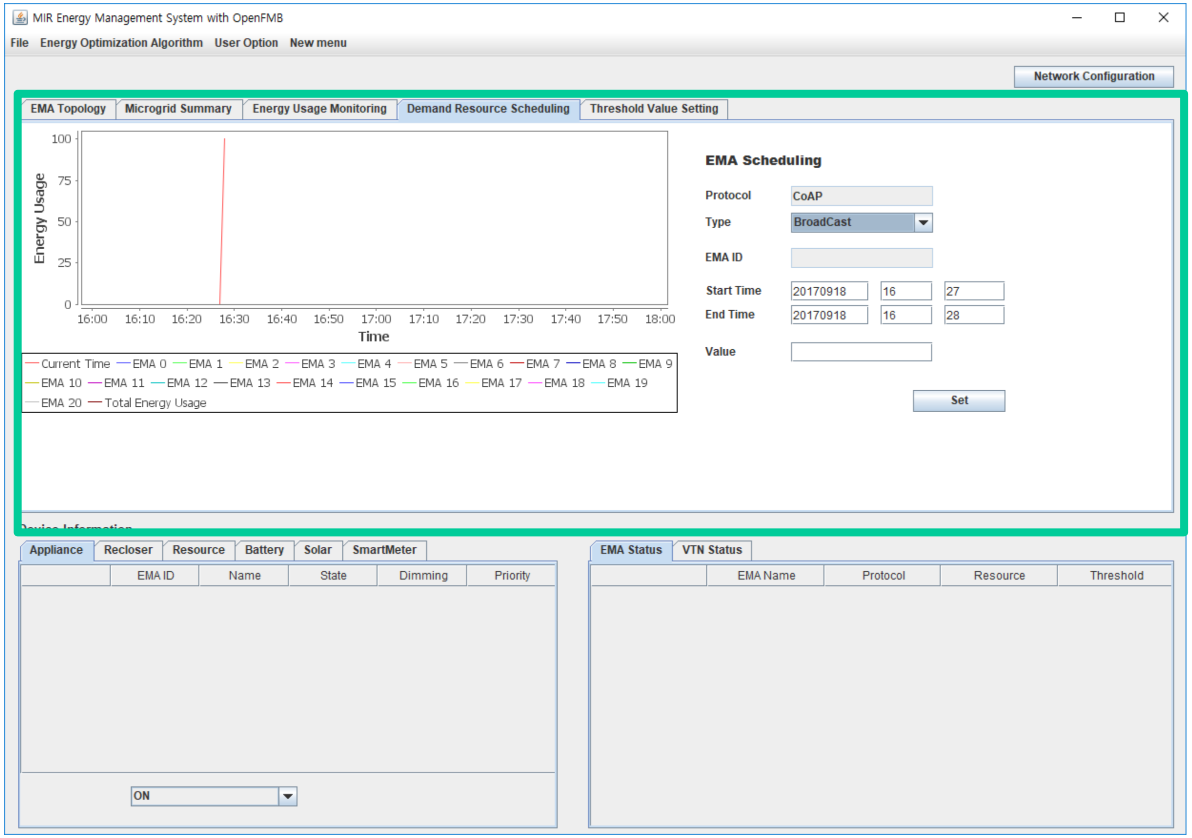
ON

	EMA Name	Protocol	Resource	Threshold

Step 7:

Real Time Energy Monitoring Graph Node will be 20s (Not real Data Now, Currently it shows random data)

7. How to Execute MIR Program(EMS)



EMA Scheduling

Protocol: CoAP

Type: BroadCast

EMA ID:

Start Time: 20170918 16 27

End Time: 20170918 16 28

Value:

Set

	EMA ID	Name	State	Dimming	Priority

ON

	EMA Name	Protocol	Resource	Threshold

Step 8:
EMA Scheduling

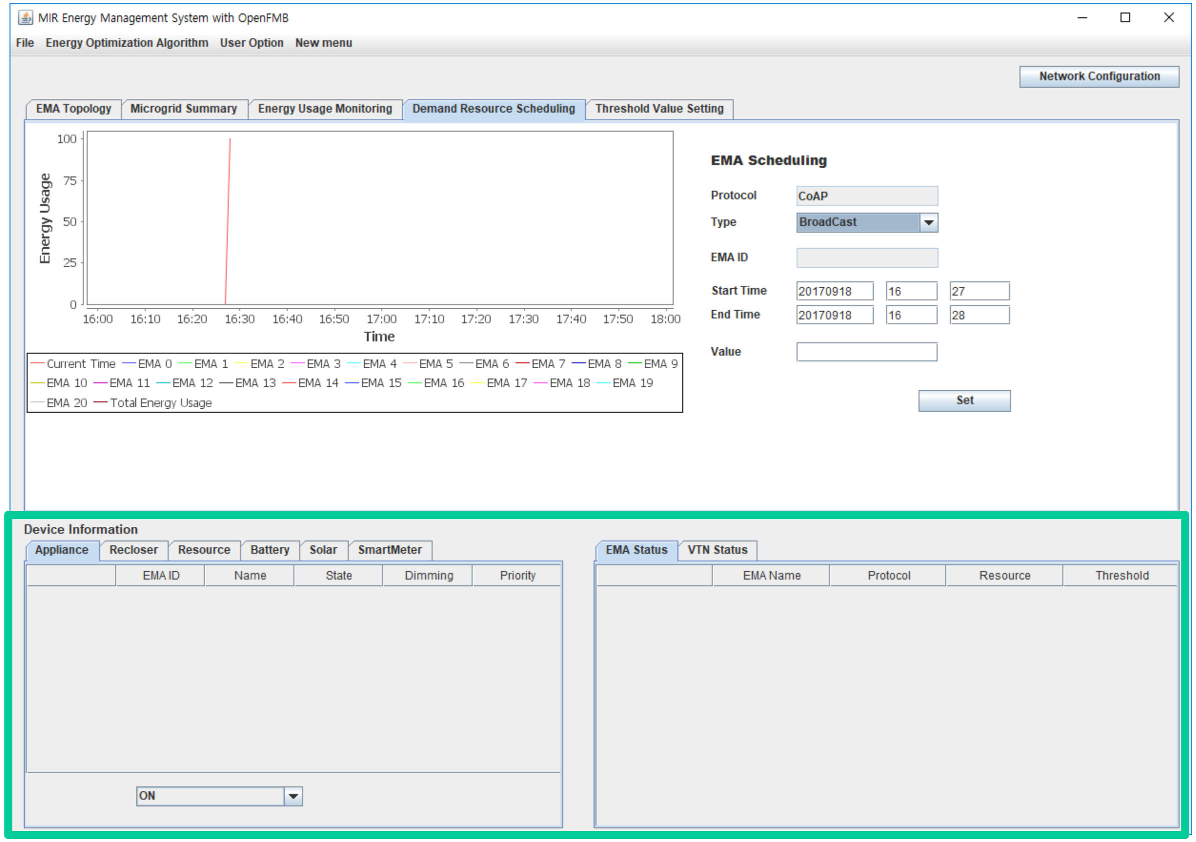
Type of Options:

(1) Broadcast(MQTT)
(2) Multicast(MQTT)
(3) Unicast(MQTT, CoAP)
→ CoAP send Message as Push mechanism

Start Time:
Event Start Time

End Time
Event End Time

7. How to Execute MIR Program(EMS)



EMA Scheduling

Protocol: CoAP

Type: BroadCast

EMA ID:

Start Time: 20170918 16 27

End Time: 20170918 16 28

Value:

Device Information

	EMA ID	Name	State	Dimming	Priority

ON

	EMA Name	Protocol	Resource	Threshold

Step 9 :

Monitoring Tables(Left)

Appliance(LED)
Recloser(One of OpenFMB things)
Resource
Batter
Solar
Smart Meter

Monitoring Tables(Right)

EMA
VTN

7. How to Execute MIR Program(VEN)

(1) Install : Putty

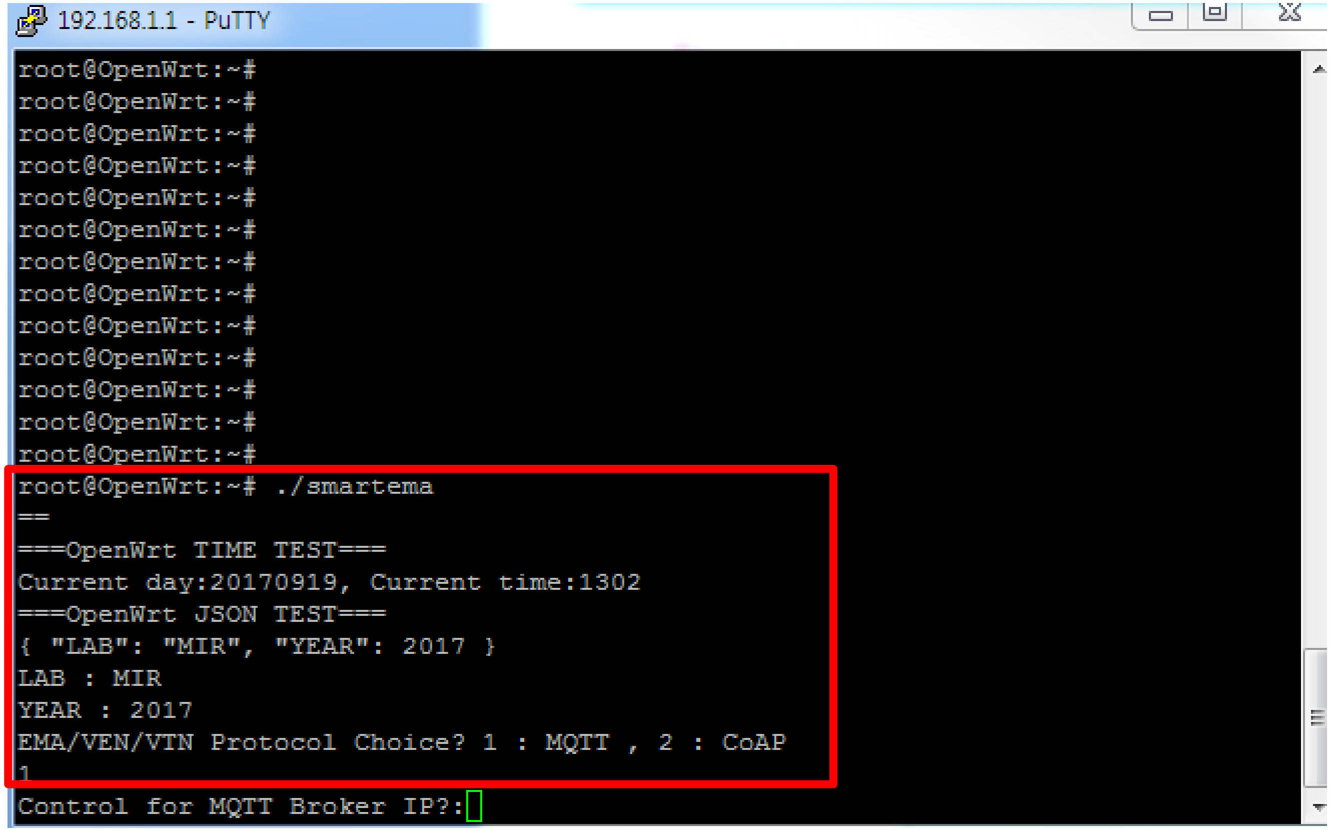
(2) OPENWRT_MIR(38)로 Wi-Fi 연결

(3) Putty를 통해 192.168.1.102 ssh로 접속한다.

(4) 접속 후 smartema 파일을 실행한다.

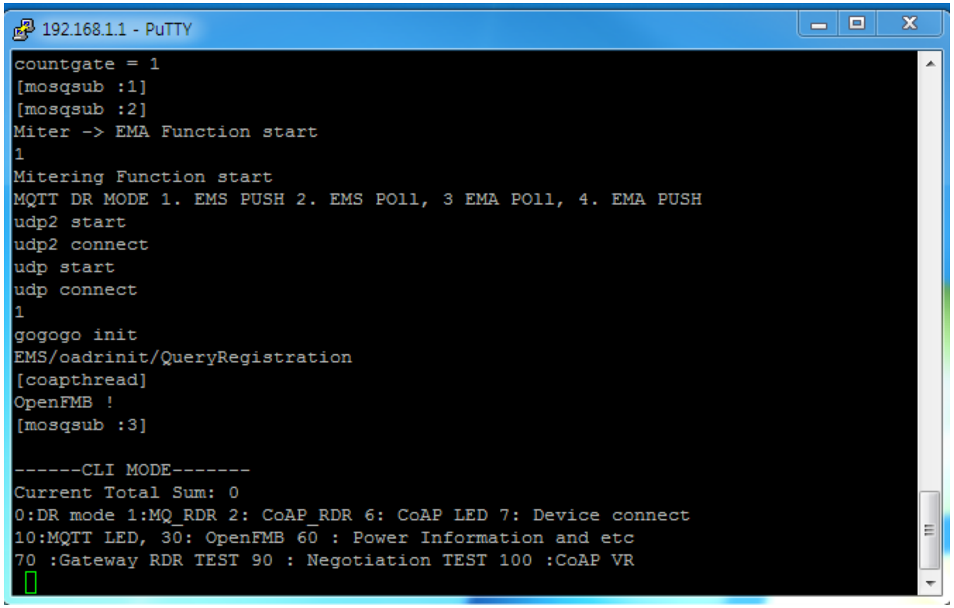
7. How to Execute MIR Program(VEN)

OpenWRT에서 smartema를 실행하면 1.MQTT, 2. CoAP를 선택할 수 있다.
MQTT는 Broker IP를 입력하고 CoAP는 EMS Server IP를 입력한다.



```
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~#
root@OpenWrt:~# ./smartema
==
===OpenWrt TIME TEST===
Current day:20170919, Current time:1302
===OpenWrt JSON TEST===
{ "LAB": "MIR", "YEAR": 2017 }
LAB : MIR
YEAR : 2017
EMA/VEN/VTN Protocol Choice? 1 : MQTT , 2 : CoAP
1
Control for MQTT Broker IP?:
```

7. How to Execute MIR Program(VEN)



```
countgate = 1
[mosqsub :1]
[mosqsub :2]
Miter -> EMA Function start
1
Mitering Function start
MQTT DR MODE 1. EMS PUSH 2. EMS POll, 3 EMA POll, 4. EMA PUSH
udp2 start
udp2 connect
udp start
udp connect
1
gogogo init
EMS/oadrinit/QueryRegistration
[coapthread]
OpenFMB !
[mosqsub :3]

------CLI MODE-------
Current Total Sum: 0
0:DR mode 1:MQ_RDR 2: CoAP_RDR 6: CoAP LED 7: Device connect
10:MQTT LED, 30: OpenFMB 60 : Power Information and etc
70 :Gateway RDR TEST 90 : Negotiation TEST 100 :CoAP VR
```

11 : ON / 15 : OFF (앞번호 Gcon-아두이노)
0: DR Mode 선택
1 :MQTT RDR 요청/ 2 CoAP RDR 요청
6: CoAP LED LED On/Off/Dimming
10 :MQTT LED On/Off/Dimming
40(50): 모두 On/Off
7 :Device을 직접 연결할 수 있음
70 :RDR TEST
90 :Gateway Negotiation
100 :CoAP 가상 디바이스 생성
60: 정보를 볼수 있음(전력 정보 / 가격정보 / Device 등록정보/ 알고리즘 정보 등)
30 :OpenFMB 분산전원 컨트롤

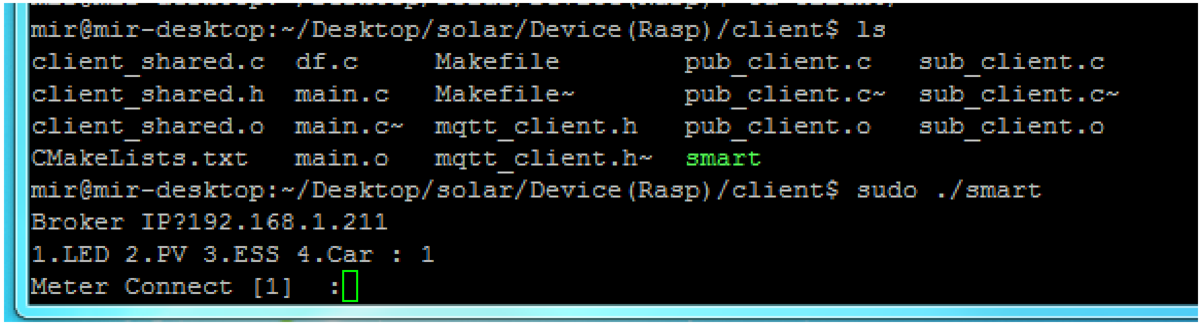
7. How to Execute MIR Program(DEVICE)

(1) Putty를 통해 192.168.1.211(Device IP) ssh로 접속한다.

(2) cd Desktop/solar/Device(Rasp)/client 로 이동한다.

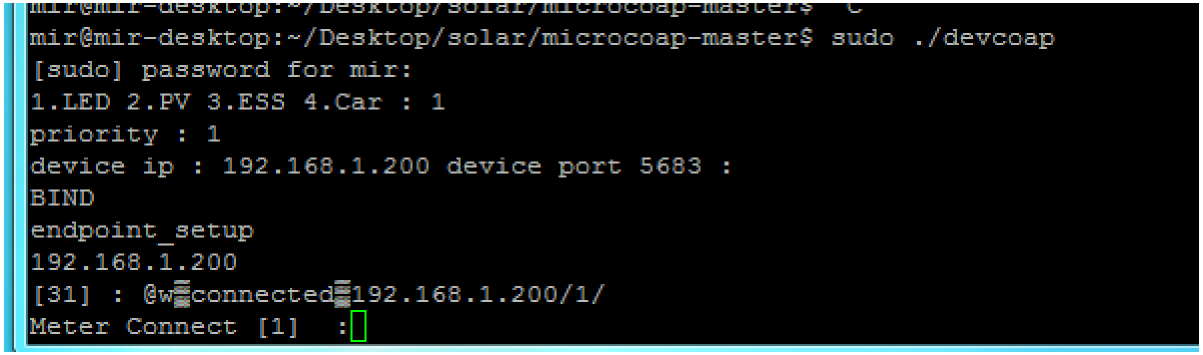
(3) 해당 경로에서 smart 파일을 root 권한으로 실행한다 → sudo ./smart

MQTT Device
1. Broker IP 입력
2. Type 선택
3. Meter 연결 선택



```
mir@mir-desktop:~/Desktop/solar/Device(Rasp)/$ cd client/
mir@mir-desktop:~/Desktop/solar/Device(Rasp)/client$ ls
client_shared.c  df.c     Makefile        pub_client.c   sub_client.c
client_shared.h  main.c   Makefile~       pub_client.c~  sub_client.c~
client_shared.o  main.c~  mqtt_client.h   pub_client.o   sub_client.o
CMakeLists.txt   main.o   mqtt_client.h~  smart
mir@mir-desktop:~/Desktop/solar/Device(Rasp)/client$ sudo ./smart
Broker IP?192.168.1.211
1.LED 2.PV 3.ESS 4.Car : 1
Meter Connect [1]   :
```

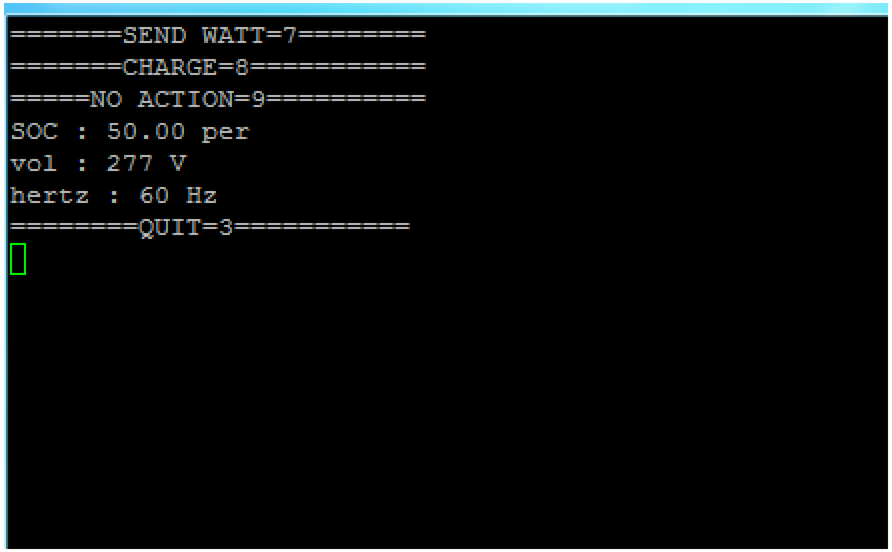
CoAP Device
1. Type 선택
2x. Meter 연결 선택



```
mir@mir-desktop:~/Desktop/solar/microcoap-master$ ^C
mir@mir-desktop:~/Desktop/solar/microcoap-master$ sudo ./devcoap
[sudo] password for mir:
1.LED 2.PV 3.ESS 4.Car : 1
priority : 1
device ip : 192.168.1.200 device port 5683 :
BIND
endpoint_setup
192.168.1.200
[31] : @w▒connected▒192.168.1.200/1/
Meter Connect [1]   :
```

MIR MQTT Broker IP: 166.104.28.49

7. How to Execute MIR Program(Device)



```
=======SEND WATT=7========
=======CHARGE=8===========
=====NO ACTION=9==========
SOC : 50.00 per
vol : 277 V
hertz : 60 Hz
========QUIT=3===========
```

ESS 시뮬레이터 Control
7. 방전(전력공급)
8. 충전
9 정지

8. List of Experiments

1. **VTN & VEN (OpenADR)**
   - Data Traffic
   - Poll-Response Time
   - DR
   - RDR

****

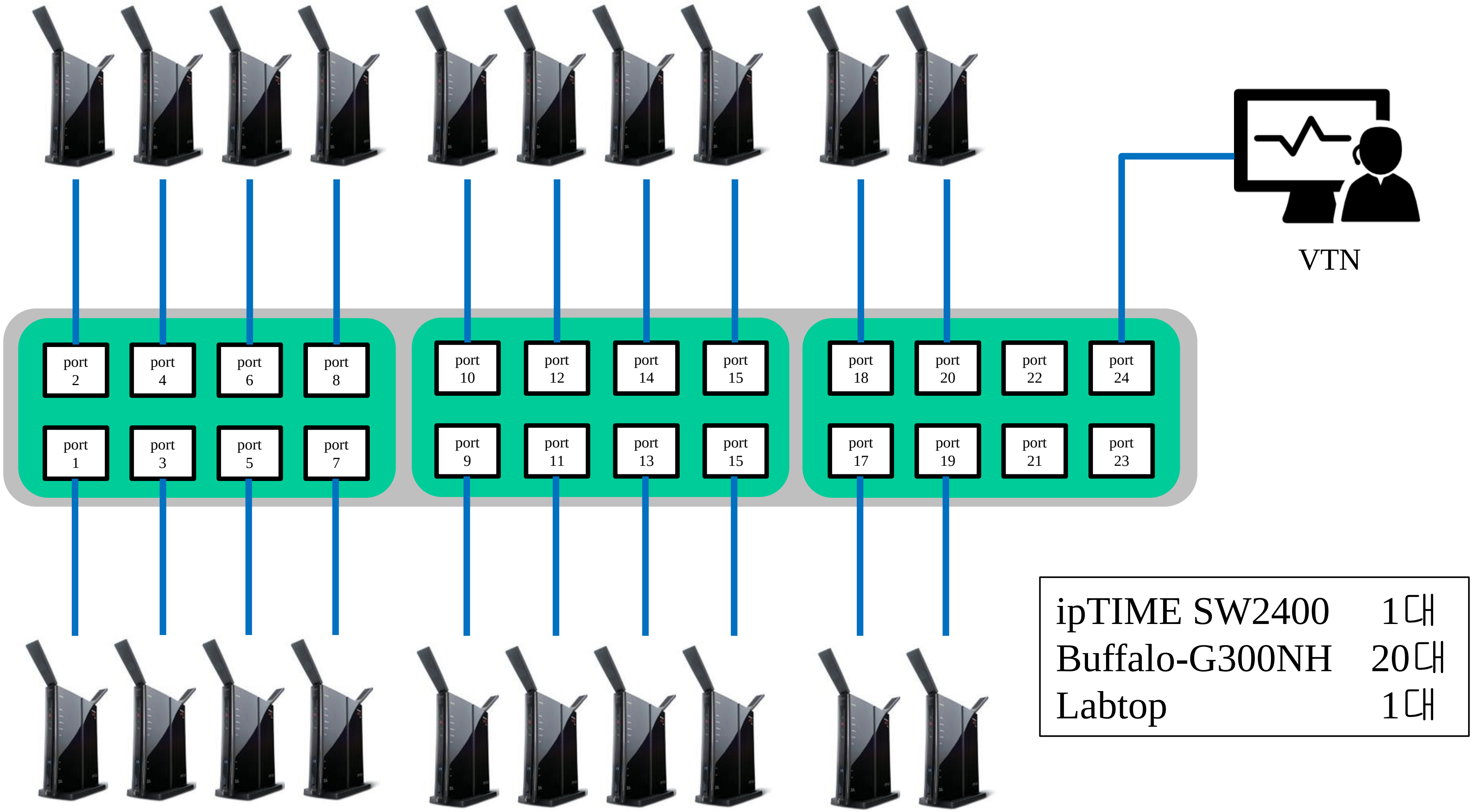
본 실험에서 진행되는 VTN은
EPRI에서 오픈소스로 제공하는 VTN으로 진행된다.
EPRI VTN 설치 방법은 동일 경로 내에 있는
Appendix. How to install EPRI VTN를 참고하면 된다.

****

2. **EMS & EMA**
   - Monitoring
     - Discovery (connect/disconnect time)
     - Status (report time)
   - Control

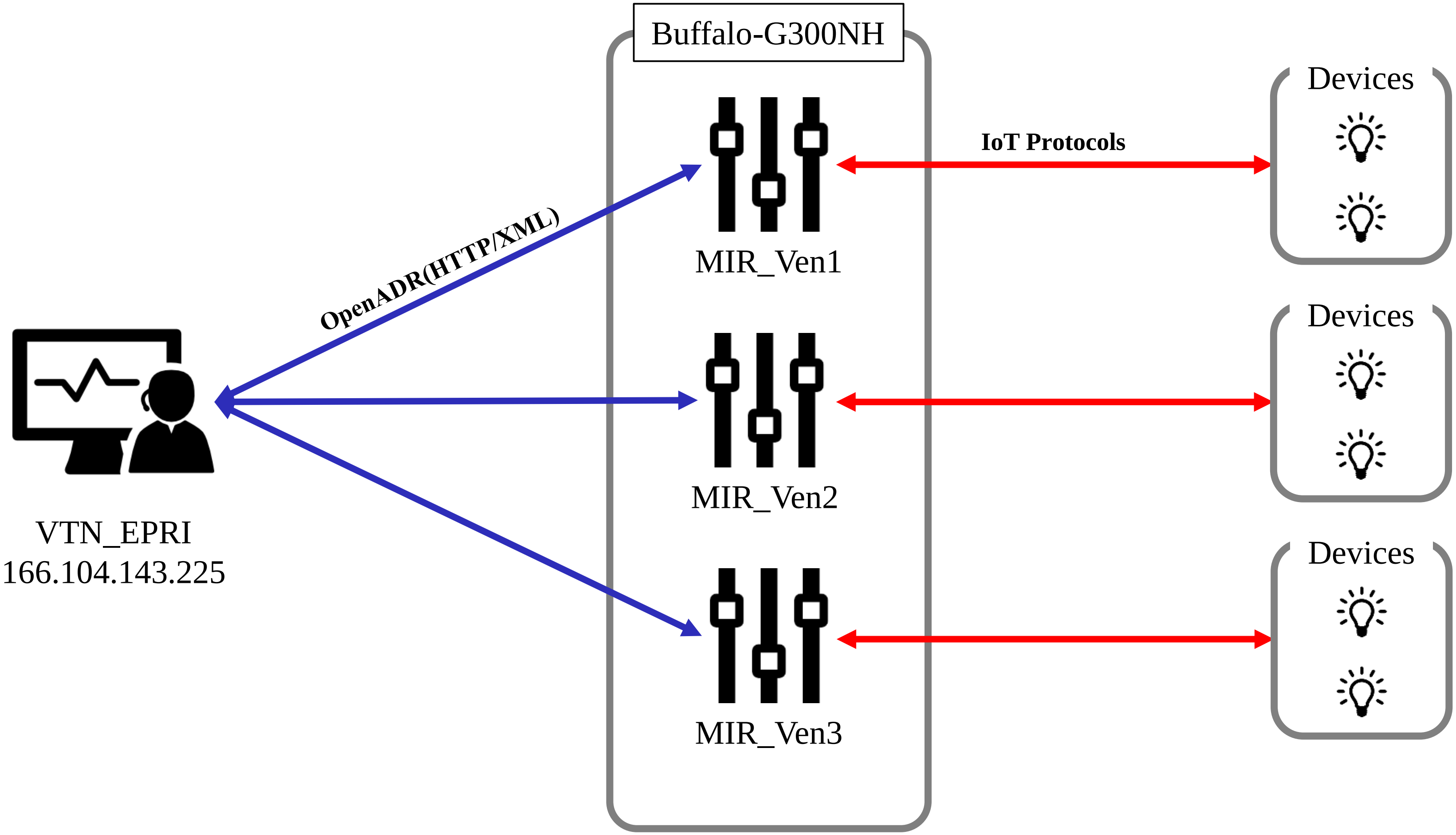
3. **EMS & OpenFMB(CoAP)**
   - Monitoring
     - Resource, Recloser, Energy Storage System, PV

8. Experiment Testbed(VTN-VEN)



ipTIME SW2400	1대
Buffalo-G300NH	20대
Labtop	1대

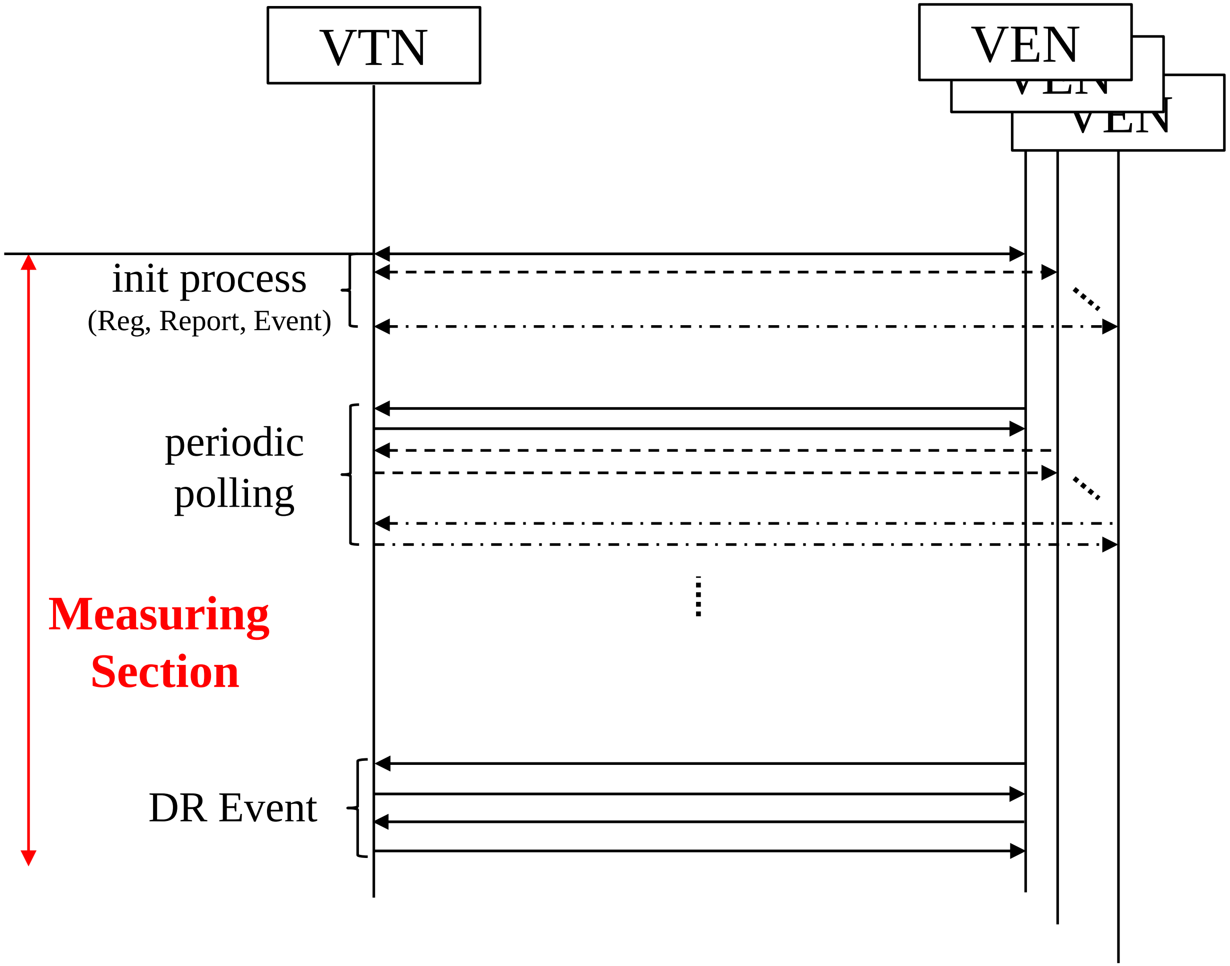
8. Experiment Testbed(VTN-VEN-Dev)



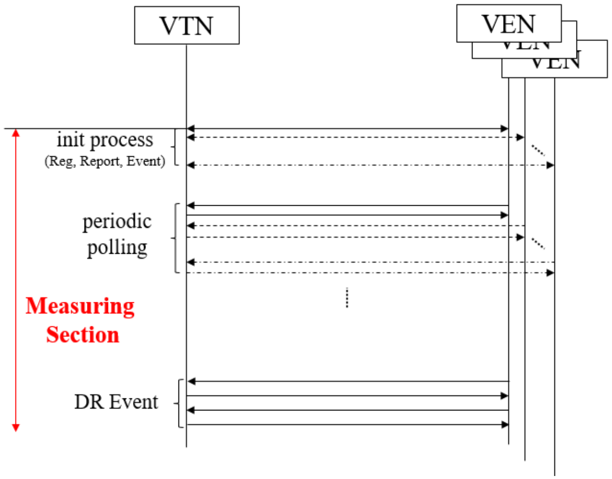
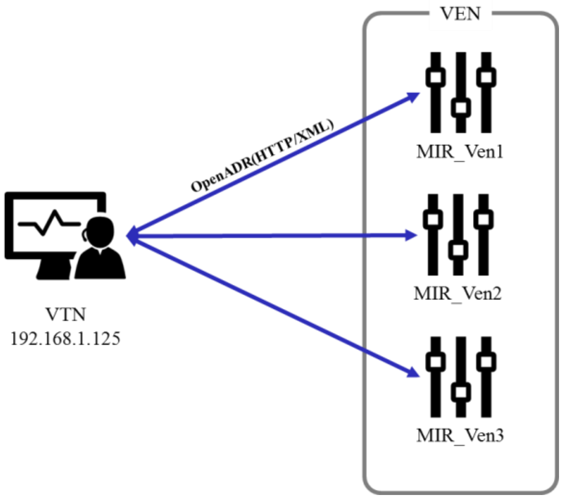
It can be extended to 20's

8. Message Flow(Data traffic)

1. Data traffic by the number of VEN

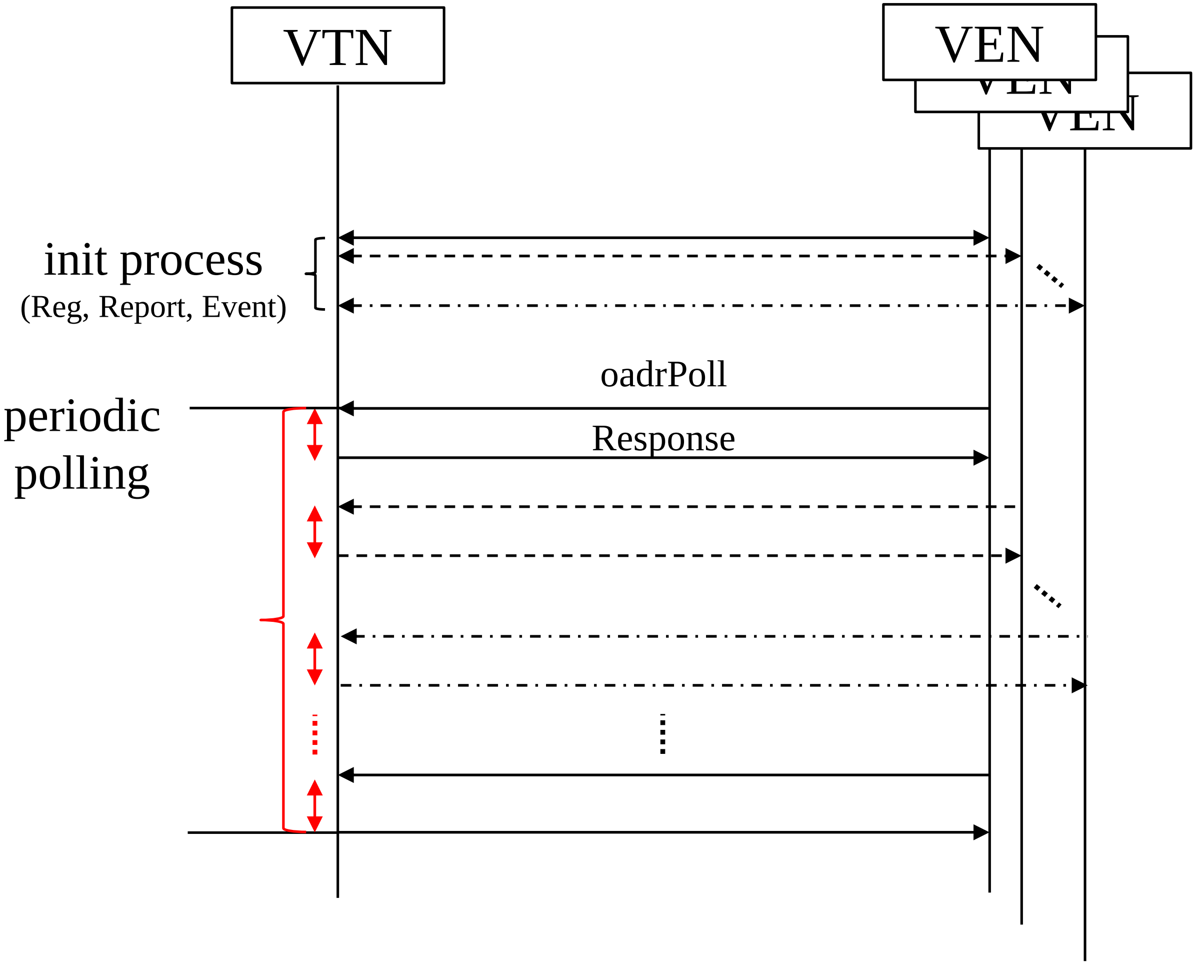


8. Experiment Procedure

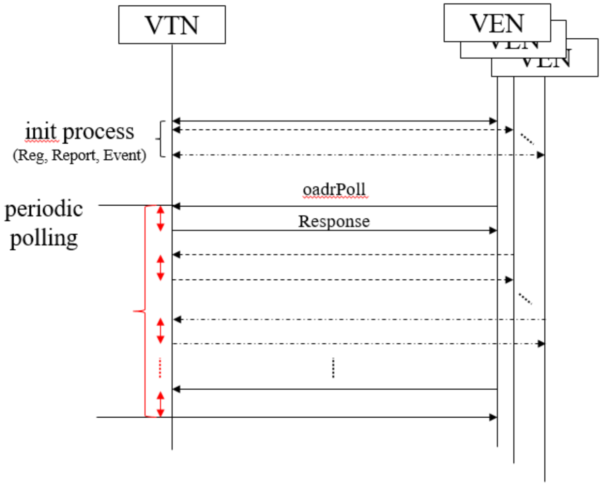
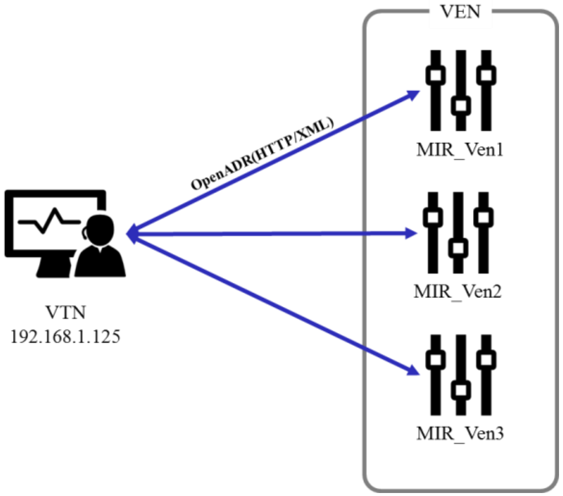
항목 번호		**시험 일자**		**시 험 자**	박헌일, 박현진
대 항 목		**중 항 목**		**소 항 목**	
목 적	VEN 개수에 따른 Data Traffic량 측정				
시험 절차 (시험 절차 또는 방법 작성)	1. 시험 구성도와 같이 시험 환경을 구성한다. 2. VTN을 실행시킨다. 3. Wireshark 실행한다. 4. 각 VEN을 실행한다. • VEN starter를 이용하여 각 VEN들을 가능한 동시에 실행 (대략 250ms 간격) 5. 초기 등록 과정 과 모든 VEN의 Poll 과정이 시작 후 30초까지 측정한다. 6. VTN에서 DR Event를 생성 및 Publish 한다. 7. 30초가 되면 Wireshark 측정을 종료한다. 8. VTN과 VEN을 종료한다. 9. 위의 과정을 VEN 개수를 1, 3, 9, 20으로 증가하며 반복한다.				
판정(측정) 기준			**시험 구성**(시험 구성도 및 관련 명령어 작성)		
 VTN VEN VEN VEN init process (Reg, Report, Event) periodic polling Measuring Section DR Event			 VEN MIR_Ven1 MIR_Ven2 MIR_Ven3 OpenADR(HTTP/XML) VTN 192.168.1.125		
판정			**비고**		

8. Message Flow(Poll-Response)

2. Polling Response Average Time by the number of VEN

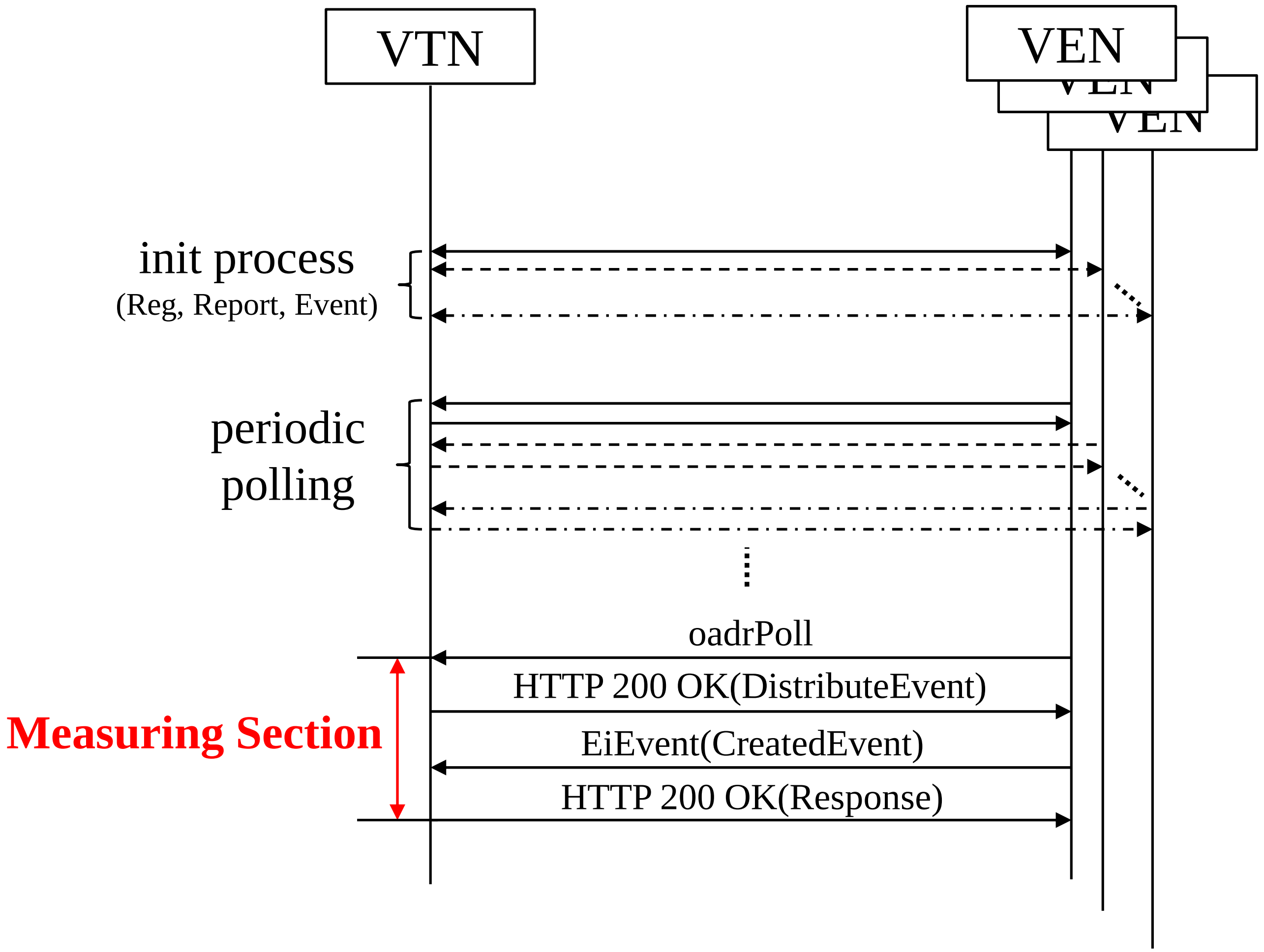


8. Experiment Procedure

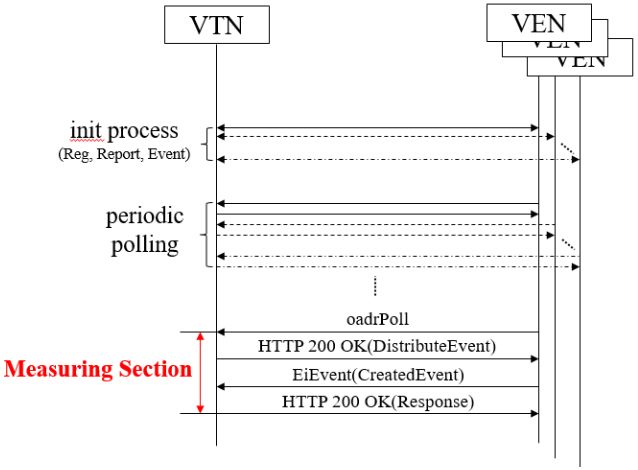
항목 번호		**시험 일자**		**시 험 자**	박헌일, 박현진
대 항 목		**중 항 목**		**소 항 목**	
목 적	VEN 개수에 따른 Poll-Response Average Time 측정				
시험 절차 (시험 절차 또는 방법 작성)	1. 시험 구성도와 같이 시험 환경을 구성한다. 2. VTN을 실행시킨다. 3. Wireshark 실행한다. 4. 각 VEN을 실행한다. • VEN starter를 이용하여 각 VEN들을 가능한 동시에 실행 (대략 250ms 간격) 5. 초기 등록 과정 과 모든 VEN의 Poll 과정이 시작 후 30초까지 측정한다. 6. VTN에서 DR Event를 생성 및 Publish 한다. 7. 30초가 되면 Wireshark 측정을 종료한다. 8. VTN과 VEN을 종료한다. 9. 위의 과정을 VEN 개수를 1, 3, 9, 20으로 증가하며 반복한다.				
판정(측정) 기준			**시험 구성**(시험 구성도 및 관련 명령어 작성)		
 VTN VEN VEN VEN init process (Reg, Report, Event) oadrPoll Response periodic polling			 VEN MIR_Ven1 MIR_Ven2 MIR_Ven3 OpenADR(HTTP/XML) VTN 192.168.1.125		
판정			**비고**		

8. Message Flow(DR Event)

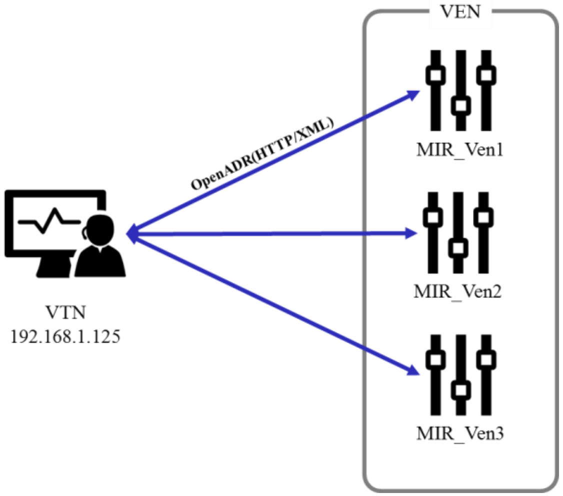
3. DR Event Response time by the number of VEN



8. Experiment Procedure

<table>
<tr><td>항목 번호</td><td></td><td>시험 일자</td><td></td><td>시 험 자</td><td>박헌일, 박현진</td></tr>
<tr><td>대 항 목</td><td></td><td>중 항 목</td><td></td><td>소 항 목</td><td></td></tr>
<tr><td>목 적</td><td colspan="5">VEN 개수에 따른 DR Event Response time</td></tr>
<tr><td>시험 절차
(시험 절차 또는 방법 작성)</td><td colspan="5">1. 시험 구성도와 같이 시험 환경을 구성한다.
2. VTN을 실행시킨다.
3. Wireshark 실행한다.
4. 각 VEN을 실행한다.
• VEN starter를 이용하여 각 VEN들을 가능한 동시에 실행 (대략 250ms 간격)
5. 초기 등록 과정 과 모든 VEN의 Poll 과정이 시작 후 30초까지 측정한다.
6. VTN에서 DR Event를 생성 및 Publish 한다.
7. 30초가 되면 Wireshark 측정을 종료한다.
8. VTN과 VEN을 종료한다.
9. 위의 과정을 VEN 개수를 1, 3, 9, 20으로 증가하며 반복한다.</td></tr>
<tr><td colspan="3">판정(측정) 기준</td><td colspan="3">시험 구성(시험 구성도 및 관련 명령어 작성)</td></tr>
<tr><td colspan="3">

VTN
VEN
VEN
VEN
init process
(Reg, Report, Event)
periodic
polling
oadrPoll
HTTP 200 OK(DistributeEvent)
EiEvent(CreatedEvent)
HTTP 200 OK(Response)
Measuring Section

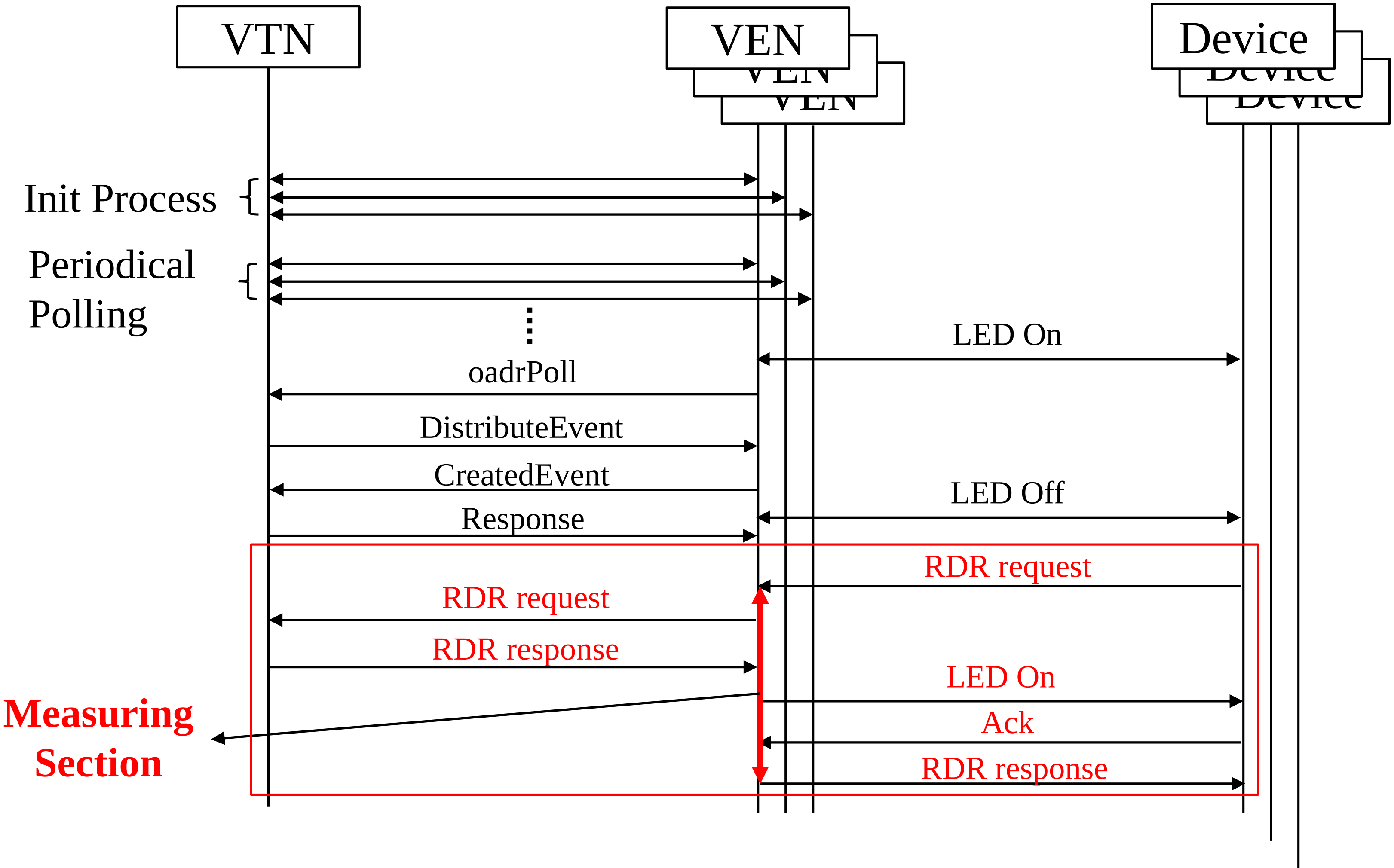
</td><td colspan="3">

VEN
MIR_Ven1
MIR_Ven2
MIR_Ven3
OpenADR(HTTP/XML)
VTN
192.168.1.125

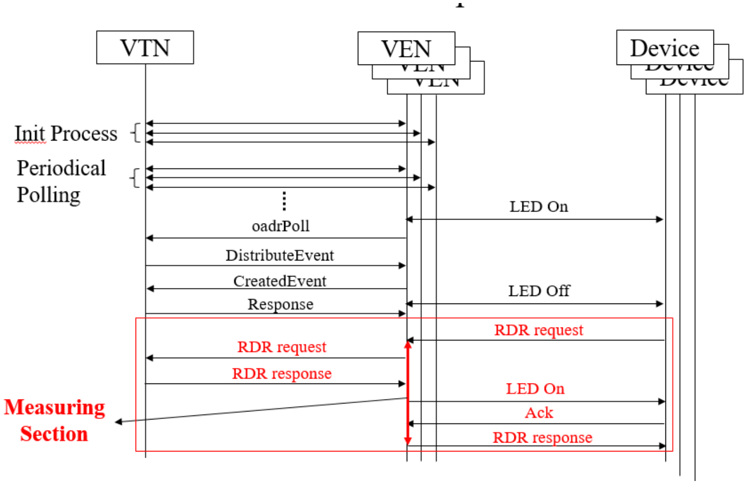
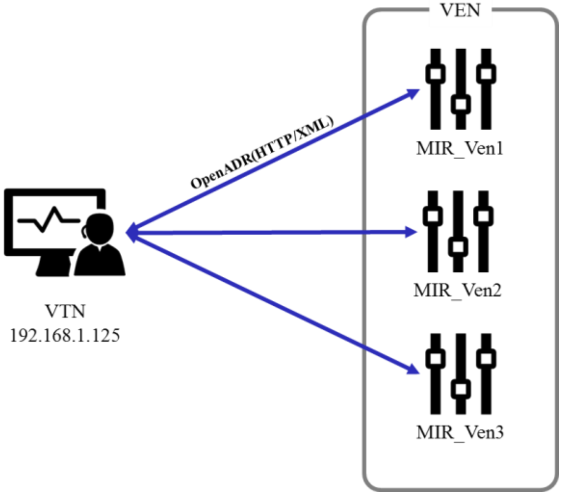
</td></tr>
<tr><td>판정</td><td colspan="2"></td><td>비고</td><td colspan="2"></td></tr>
</table>

8. Message Flow(RDR)

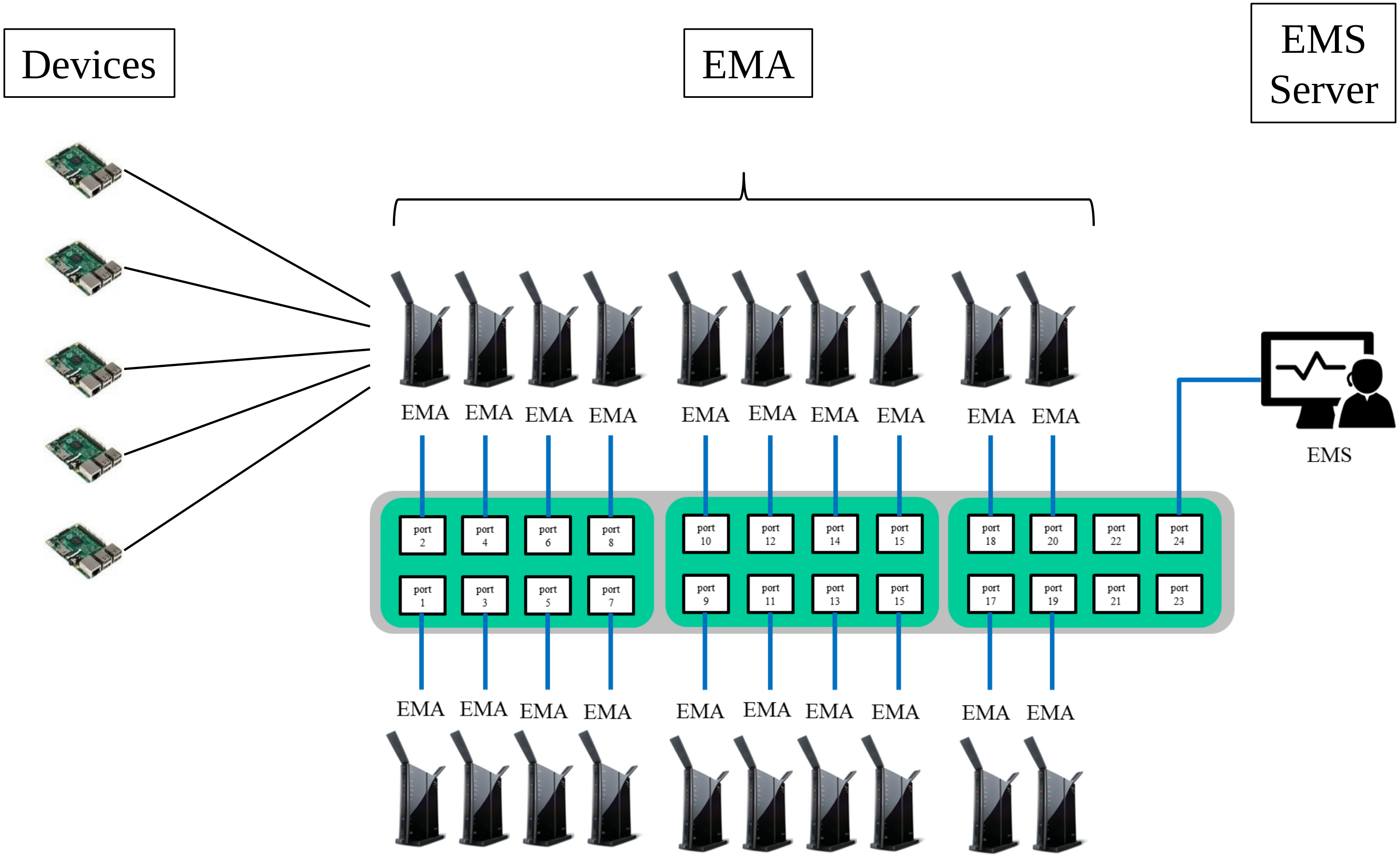
- VEN 개수에 따른 RDR-Response time



8. Experiment Procedure

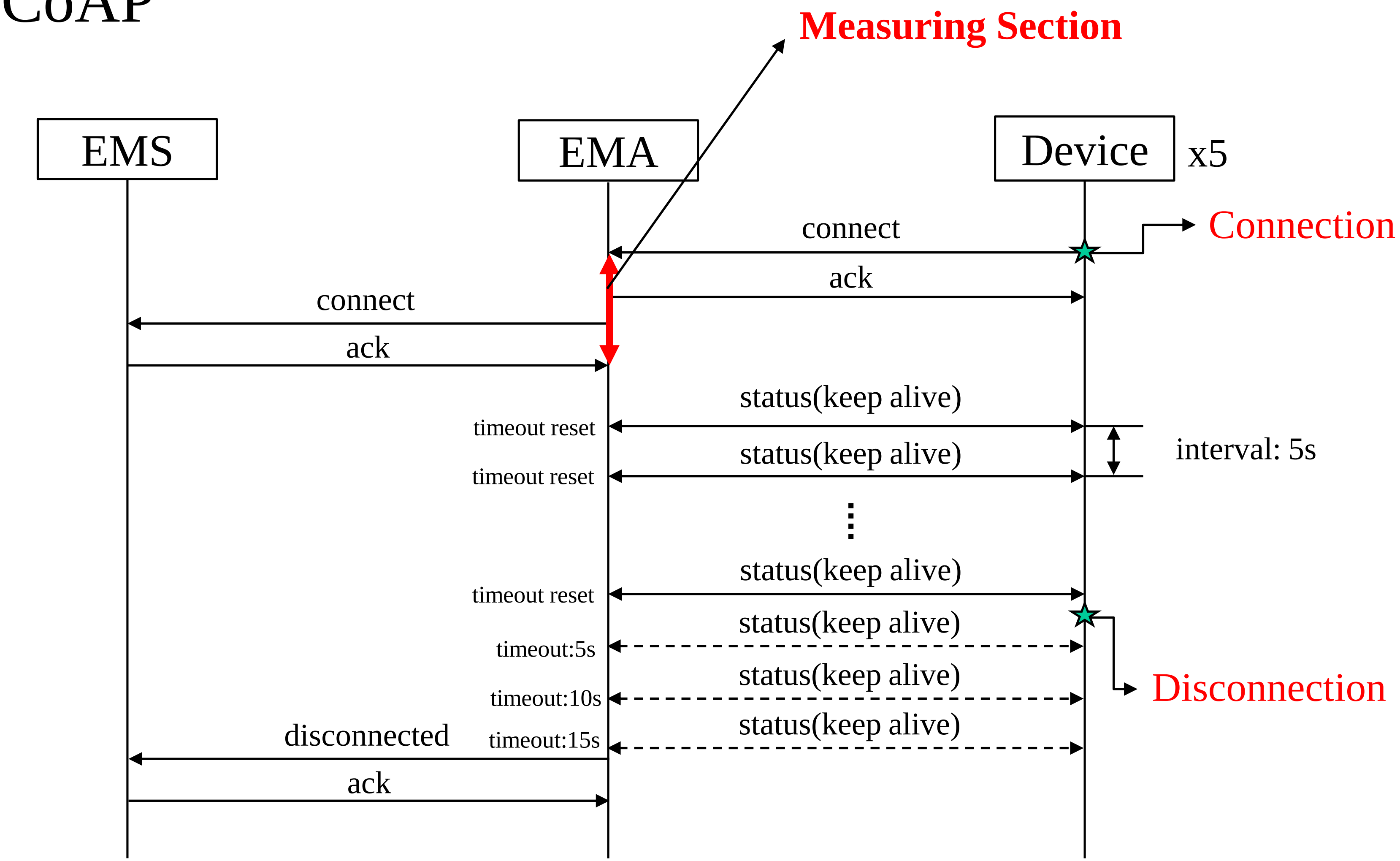
<table>
<tr><td><b>항목 번호</b></td><td></td><td><b>시험 일자</b></td><td></td><td><b>시 험 자</b></td><td colspan="2">박헌일, 박현진</td></tr>
<tr><td><b>대 항 목</b></td><td></td><td><b>중 항 목</b></td><td></td><td><b>소 항 목</b></td><td colspan="2"></td></tr>
<tr><td><b>목 적</b></td><td colspan="6"></td></tr>
<tr><td><b>시험 절차</b>
(시험 절차 또는 방법 작성)</td><td colspan="6">1. 시험 구성도와 같이 시험 환경을 구성한다.
2. 아두이노 디바이스를 킨다.
3. 측정장비를 킨다.
4. VTN, EMS을 킨다.
5. 와이어샤크를 킨다.
6. 게이트웨이 프로그램을 킨다.(옵션에 RDR이벤트를 체크한다.)
7. 게이트웨이의 마진을 50으로 세팅한 후아무 디바이스나 1개 킨다.
8. 디바이스의 스위치를 눌러 이벤트 발생 후 측정
9. 30초가 되면 Wireshark 측정을 종료한다.
10. VTN과 VEN을 종료한다.
11. 위의 과정을 VEN 개수를 1, 3, 9, 20으로 증가하며 반복한다.</td></tr>
<tr><td colspan="3"><b>판정(측정) 기준</b></td><td colspan="4"><b>시험 구성</b>(시험 구성도 및 관련 명령어 작성)</td></tr>
<tr><td colspan="3">
</td><td colspan="4">
</td></tr>
<tr><td><b>판정</b></td><td colspan="2"></td><td><b>비고</b></td><td colspan="3"></td></tr>
</table>

8. Experiment Testbed(EMS-EMA)



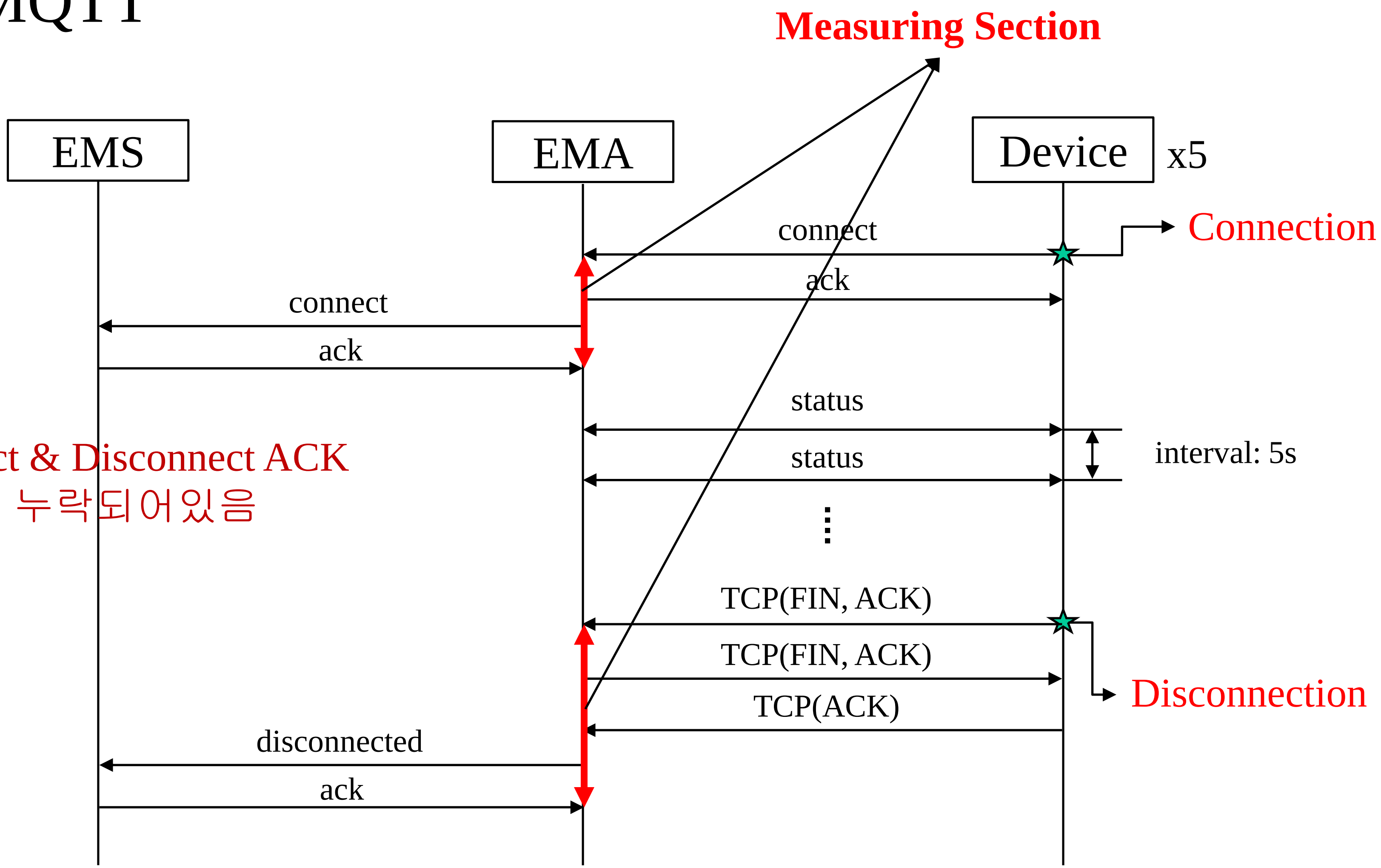
8. Message Flow(Discovery)

- CoAP



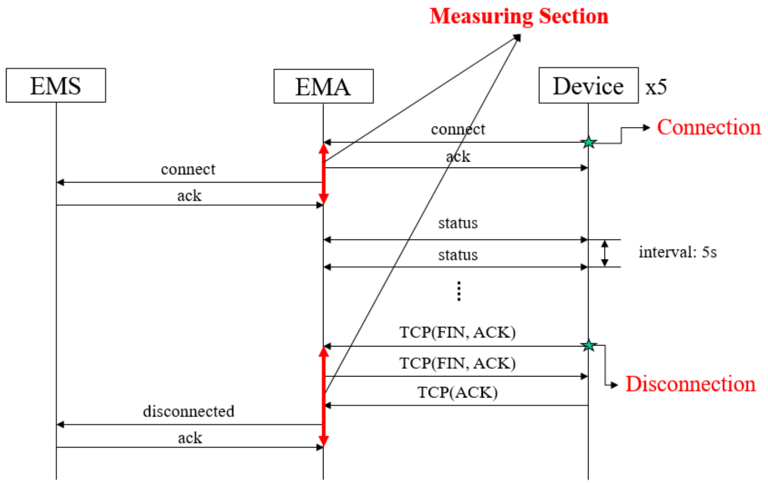
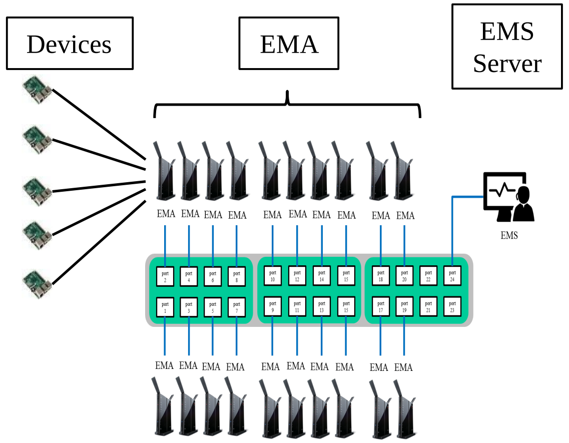
8. Message Flow(Discovery)

- MQTT



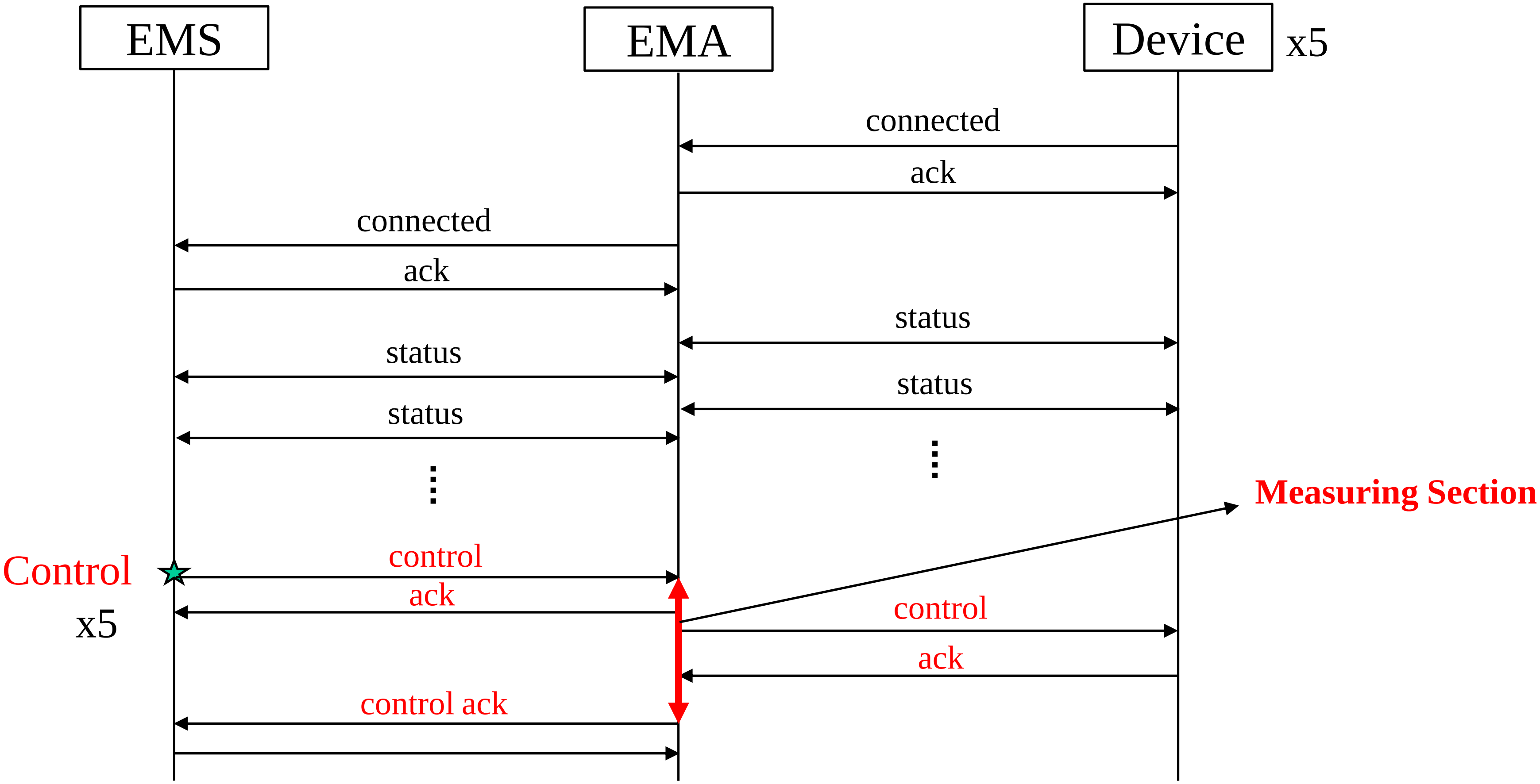
MS Connect & Disconnect ACK
부분이 지금 누락되어있음

8. Experiment Procedure

<table>
<tr><td><b>항목 번호</b></td><td></td><td><b>시험 일자</b></td><td></td><td><b>시 험 자</b></td><td>박헌일, 박현진</td></tr>
<tr><td><b>대 항 목</b></td><td></td><td><b>중 항 목</b></td><td></td><td><b>소 항 목</b></td><td></td></tr>
<tr><td><b>목 적</b></td><td colspan="5"></td></tr>
<tr><td><b>시험 절차</b>
(시험 절차 또는 방법 작성)</td><td colspan="5">1. EMA 1, 3, 9, 20개를 실행.
2. EMS 실행. (재실행 필요)
3. EMA에서 tcpdump 시작.
4. tcpdump -i br-wan -vvv port 5683 -w ~/test.pcap(<b>패킷 캡쳐 예시</b>)
5. EMA 동시에 작동
6. End- Device 5개 실행.
7. 약 5초간 대기
8. EMS에서 End - Device 5대 LED On Signal
9. 5초 후 EMS LED Off Signal
10. Device 종료 후 Disconnect확인
11. Tcpdump Packet Capturing 종료.</td></tr>
<tr><td colspan="3"><b>판정(측정) 기준</b></td><td colspan="3"><b>시험 구성</b>(시험 구성도 및 관련 명령어 작성)</td></tr>
<tr><td colspan="3">
</td><td colspan="3">
</td></tr>
<tr><td><b>판정</b></td><td colspan="2"></td><td><b>비고</b></td><td colspan="2"></td></tr>
</table>

8. Message Flow(Control-EMS)

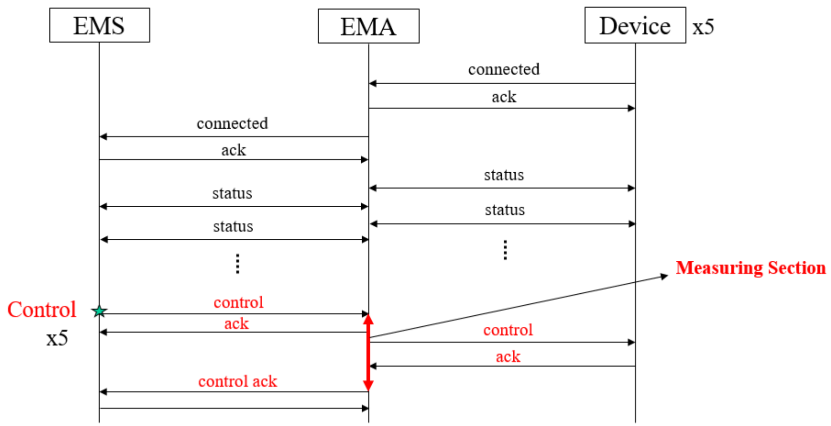
- CoAP, MQTT

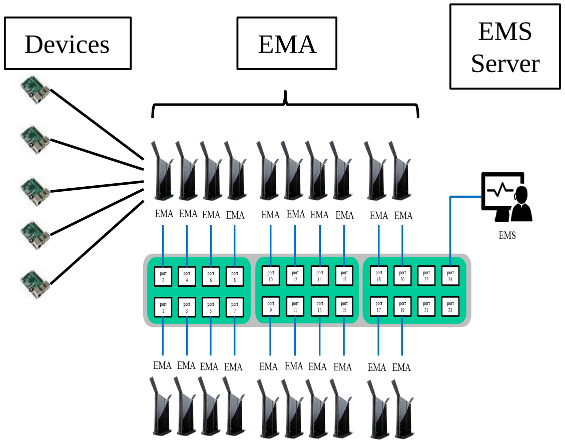


8. Experiment Procedure

항목 번호		**시험 일자**		**시 험 자**	박헌일, 박현진
대 항 목		**중 항 목**		**소 항 목**	
목 적					
시험 절차 (시험 절차 또는 방법 작성)	1. EMA 1, 3, 9, 20개를 실행. 2. EMS 실행. (재실행 필요) 3. EMA에서 tcpdump 시작. 4. tcpdump -i br-wan -vvv port 5683 -w ~/test.pcap(**패킷 캡쳐 예시**) 5. EMA 동시에 작동 6. End- Device 5개 실행. 7. 약 5초간 대기 8. EMS에서 End - Device 5대 LED On Signal 9. 5초 후 EMS LED Off Signal 10. Device 종료 후 Disconnect확인 11. Tcpdump Packet Capturing 종료				

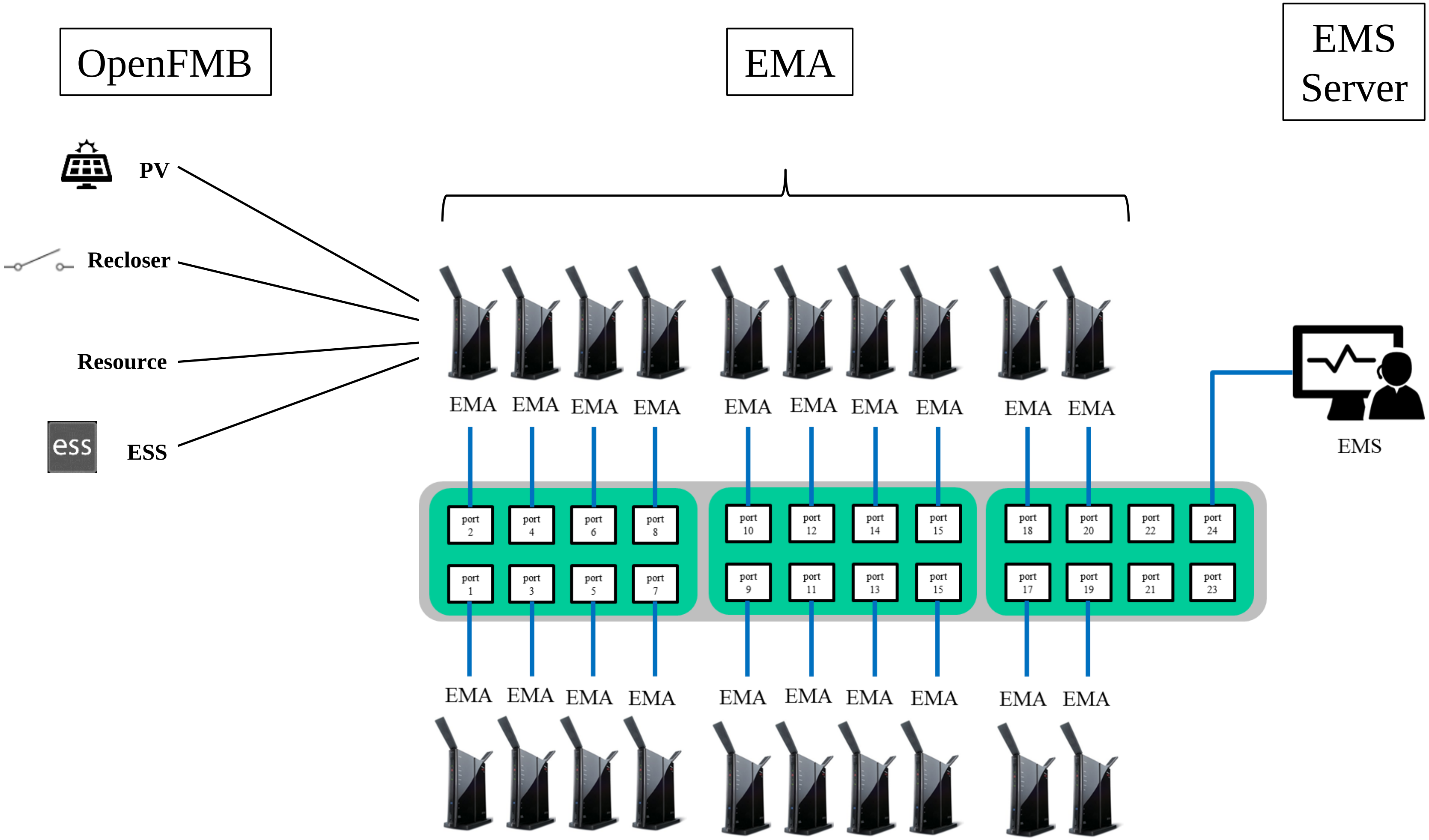
판정(측정) 기준	**시험 구성**(시험 구성도 및 관련 명령어 작성)





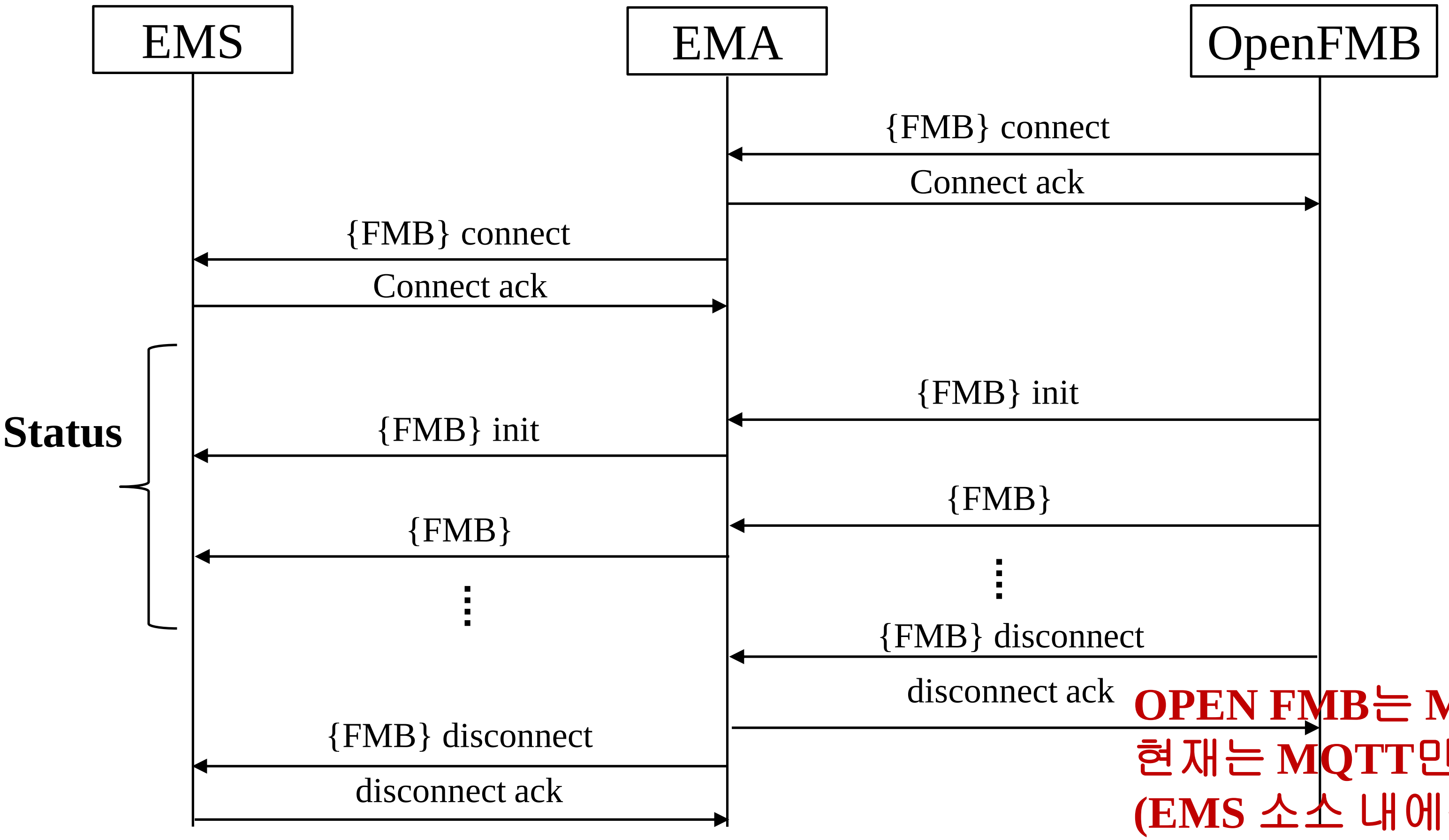
판정		**비고**	

8. Experiment Testbed (EMS-EMA-OpenFMB)



Message Flow(EMS-EMA-OpenFMB)

- CoAP, MQTT



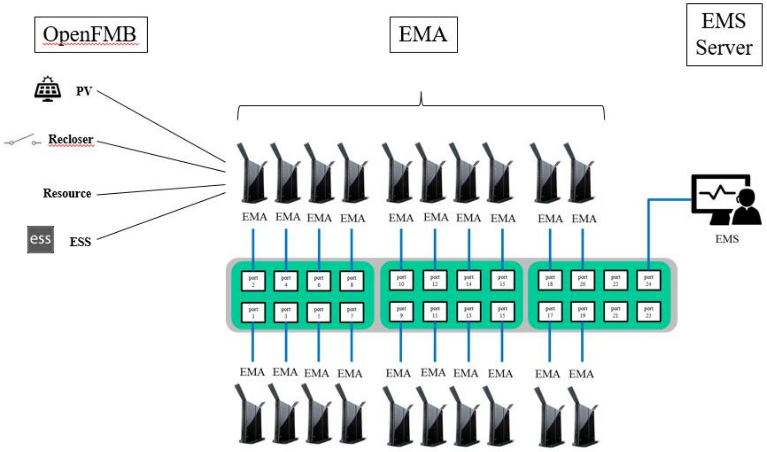
****Notice****
Recloser
ESS
PV
Resource

{FMB}의 의미는 상위 4개 항목을 통합한 것

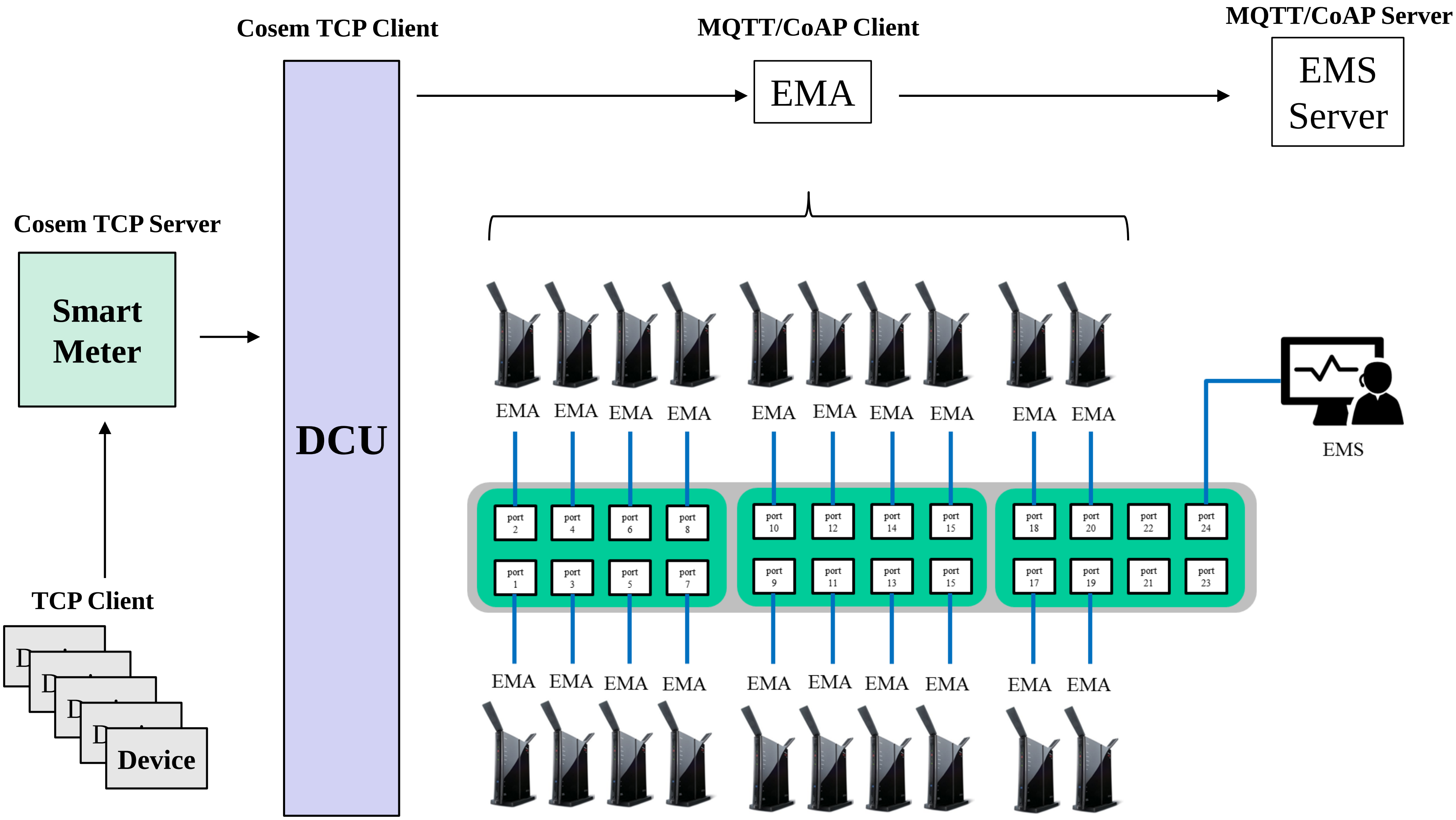
e.g.
ESSconnect

OPEN FMB는 MQTT 기반이기
현재는 MQTT만 구현되어있다.
(EMS 소스 내에는 CoAP 도 포함
실제 EMA와 연동 테스트를 해보

8. Experiment Procedure

항목 번호		**시험 일자**		**시 험 자**	박헌일, 박현진
대 항 목		**중 항 목**		**소 항 목**	
목 적					
시험 절차 (시험 절차 또는 방법 작성)	1. EMA 1개 실행, 2. EMS 실행. (재실행 필요) 3. EMA에서 tcpdump 시작. 4. tcpdump -i br-wan -vvv port 5683 -w ~/test.pcap(**패킷 캡쳐 예시**) 5. EMA 동시에 작동 6. OpenFMB 모듈 실행 7. 약 5초간 대기 8. 수신되는 FMB 모듈들의 Status를 EMS에서 확인 9. OpenFMB 종료 및 EMA 종류 후 Disconnect확인 10. Tcpdump Packet Capturing 종료				
판정(측정) 기준			**시험 구성**(시험 구성도 및 관련 명령어 작성)		
OpenFMB와 Smart Meter는 최근에 연동된 부분이기 때문에 실험에 대한 판정 기준이 존재하지 않는다. 그러므로 현재 상황에서는 EMS 와 OpenFMB를 연동하는 부분 까지만 실험을 할 수 있다.			 OpenFMB EMA EMS Server PV Recloser Resource ess ESS EMA EMA EMA EMA EMA EMA EMA EMA EMA EMA port 2 port 4 port 6 port 8 port 10 port 12 port 14 port 15 port 18 port 20 port 22 port 24 port 1 port 3 port 5 port 7 port 9 port 11 port 13 port 15 port 17 port 19 port 21 port 23 EMA EMA EMA EMA EMA EMA EMA EMA EMA EMA EMS		
판정			**비고**		

8. Experiment Testbed (EMS-EMA-Smart Meter)



Cosem TCP Client

MQTT/CoAP Client

MQTT/CoAP Server

EMA

EMS Server

Cosem TCP Server

Smart Meter

DCU

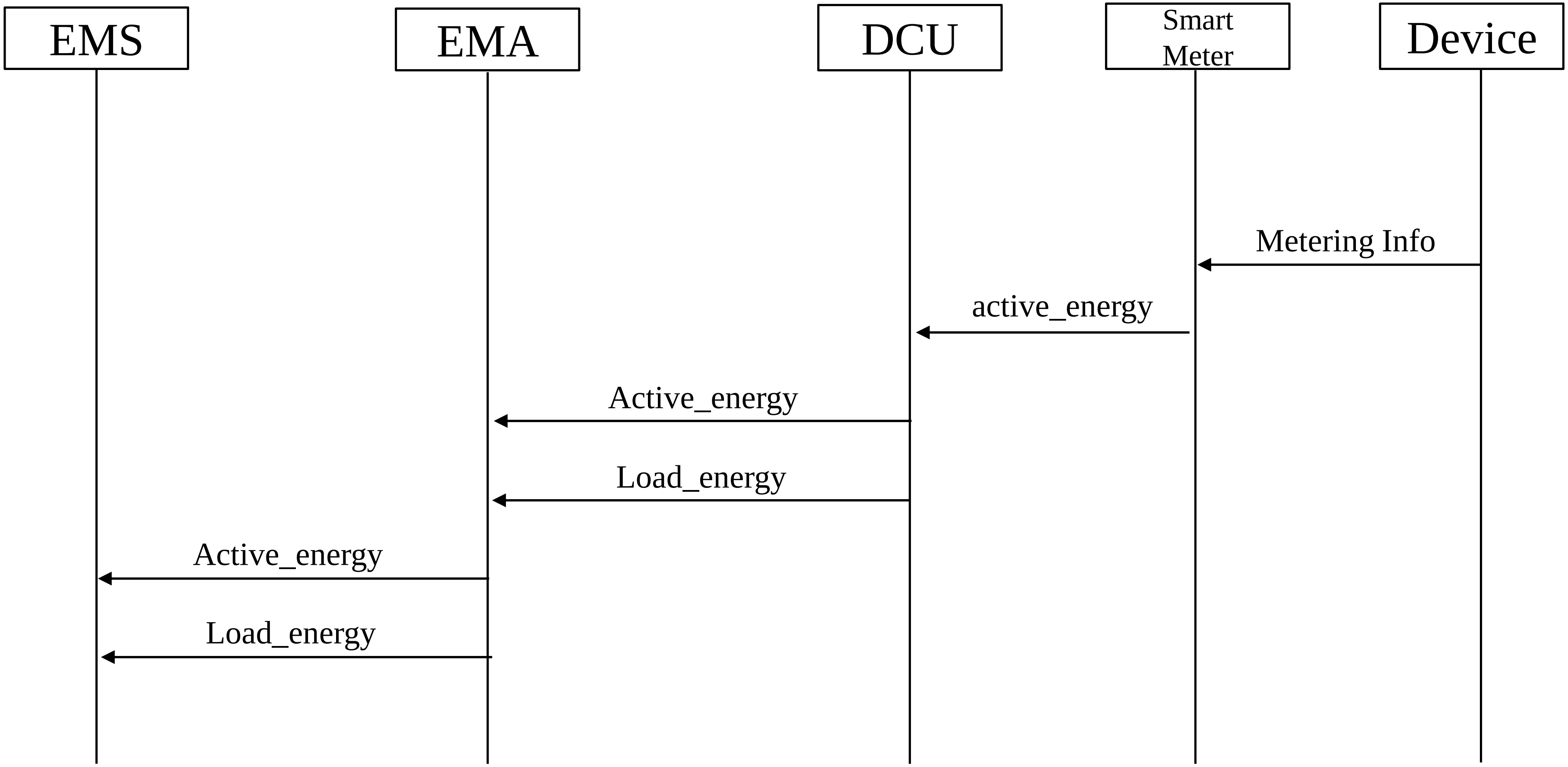
TCP Client

Device

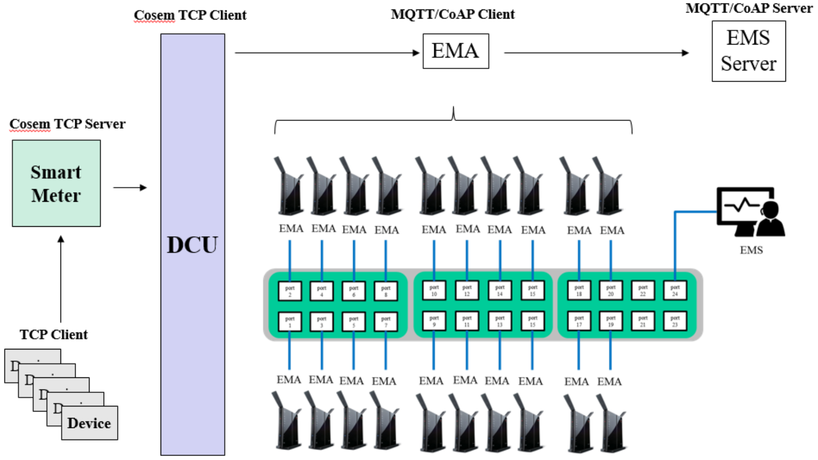
EMS

8. Message Flow(EMS-EMA-DCU-Smart Meter-Device)

- CoAP, MQTT

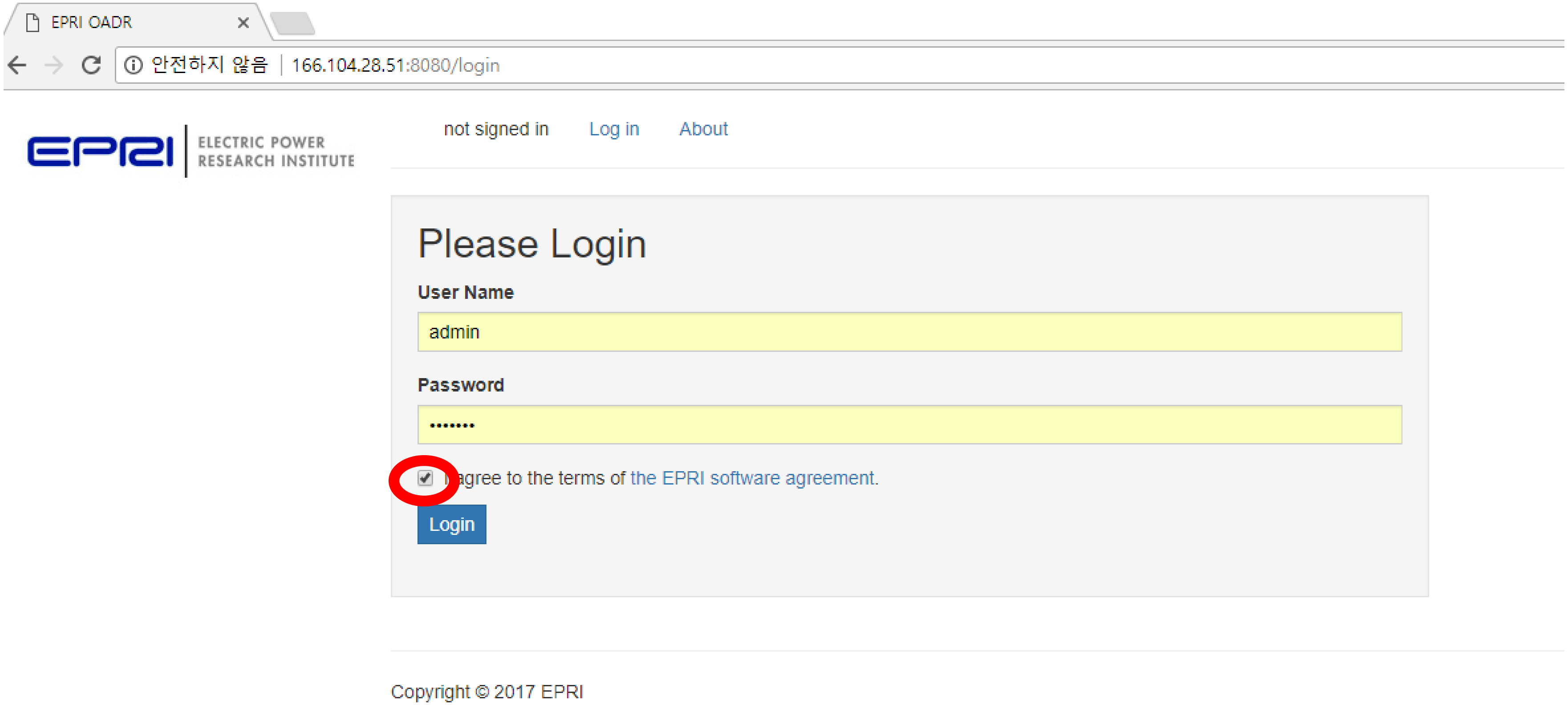


8. Experiment Procedure

항목 번호		**시험 일자**		**시 험 자**	박헌일, 박현진
대 항 목		**중 항 목**		**소 항 목**	
목 적					
시험 절차 (시험 절차 또는 방법 작성)	1. End-Device 실행. 2. Smart Meter 실행 3. DCU 실행 4. EMA 실행 5. EMS 실행 6. 약 5초간 대기				
판정(측정) 기준			**시험 구성**(시험 구성도 및 관련 명령어 작성)		
OpenFMB와 Smart Meter는 최근에 연동된 부분이기 때문에 실험에 대한 판정 기준이 존재하지 않는다. 그러므로 현재 상황에서는 EMS 와 OpenFMB를 연동하는 부분 까지만 실험을 할 수 있다.			 Cosem TCP Client MQTT/CoAP Client MQTT/CoAP Server EMA EMS Server Cosem TCP Server Smart Meter DCU TCP Client Device EMA EMA EMA EMA EMA EMA EMA EMA EMA EMA port 2 port 4 port 6 port 8 port 10 port 12 port 14 port 15 port 18 port 20 port 22 port 24 port 1 port 3 port 5 port 7 port 9 port 11 port 13 port 15 port 17 port 19 port 21 port 23 EMA EMA EMA EMA EMA EMA EMA EMA EMA EMA EMS		
판정			**비고**		

9. Captured Screen as following Instruction

9. VTN-VEN DR Test



EPRI OADR

안전하지 않음 | 166.104.28.51:8080/login

ELECTRIC POWER RESEARCH INSTITUTE

not signed in Log in About

Please Login

User Name

admin

Password

•••••••

☑ I agree to the terms of the EPRI software agreement.

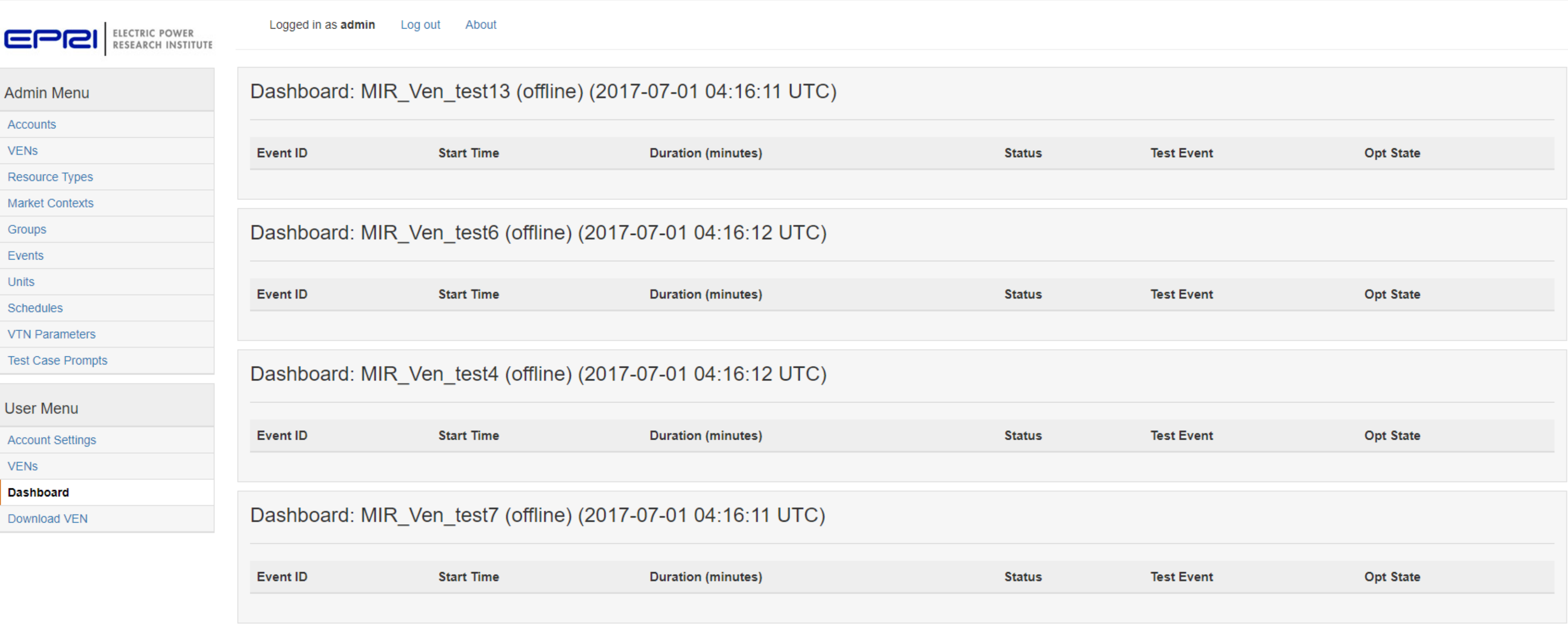
Login

Copyright © 2017 EPRI

Login in the VTN ID : admin // Password : testing

Checkbox must be Checked

9. VTN-VEN DR Test



Dashboard: MIR_Ven_test13 (offline) (2017-07-01 04:16:11 UTC)

Event ID	Start Time	Duration (minutes)	Status	Test Event	Opt State

Dashboard: MIR_Ven_test6 (offline) (2017-07-01 04:16:12 UTC)

Event ID	Start Time	Duration (minutes)	Status	Test Event	Opt State

Dashboard: MIR_Ven_test4 (offline) (2017-07-01 04:16:12 UTC)

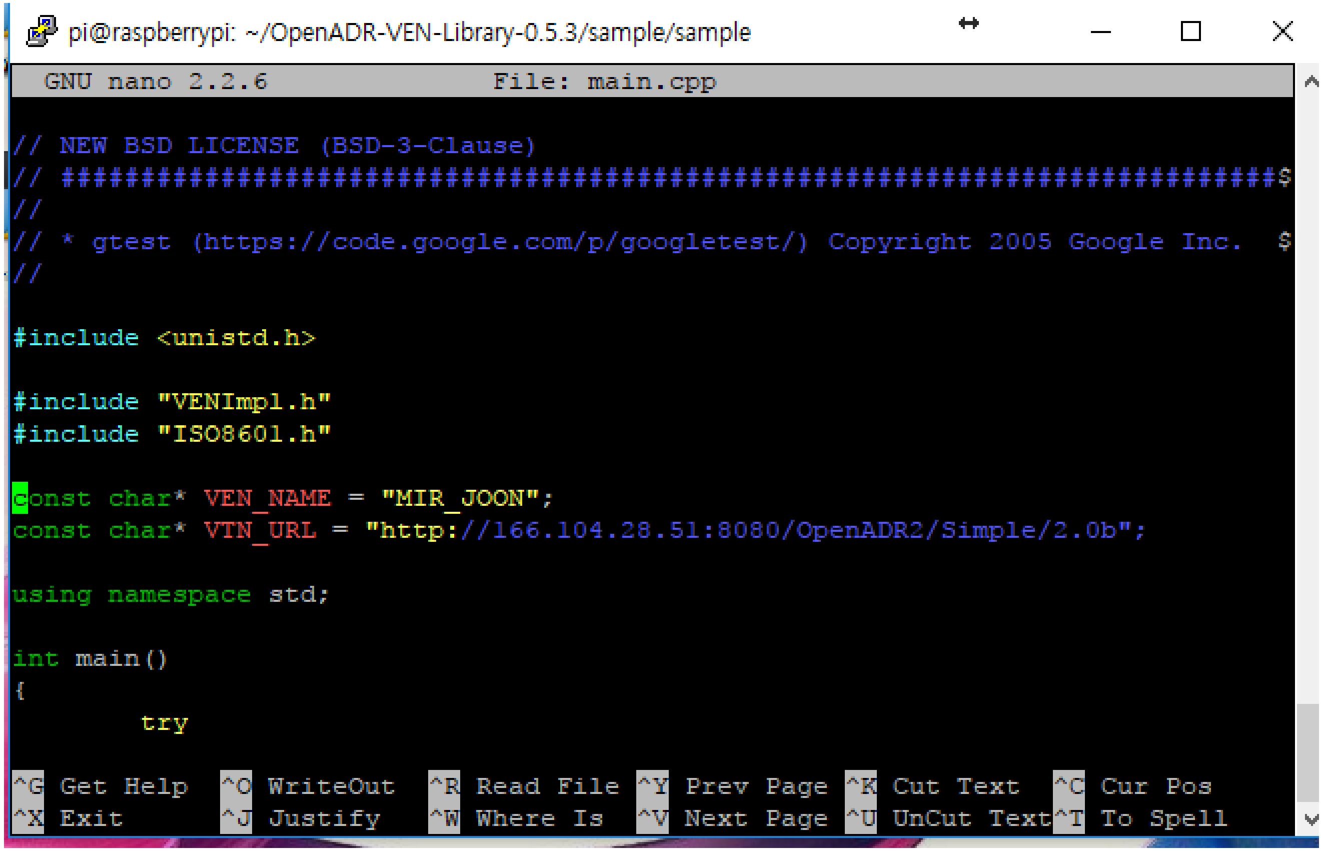
Event ID	Start Time	Duration (minutes)	Status	Test Event	Opt State

Dashboard: MIR_Ven_test7 (offline) (2017-07-01 04:16:11 UTC)

Event ID	Start Time	Duration (minutes)	Status	Test Event	Opt State

Login Success

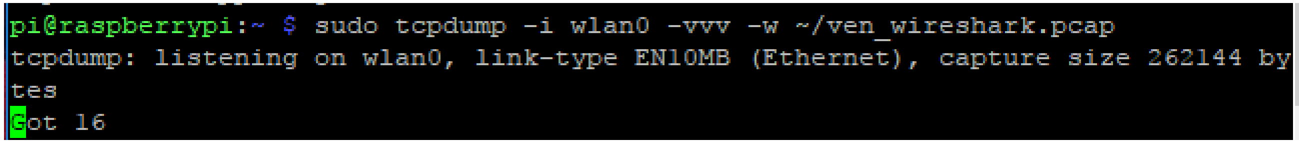
9. VTN-VEN DR Test



```
  GNU nano 2.2.6                    File: main.cpp

// NEW BSD LICENSE (BSD-3-Clause)
// ##############################################################################$
//
// * gtest (https://code.google.com/p/googletest/) Copyright 2005 Google Inc.   $
//

#include <unistd.h>

#include "VENImpl.h"
#include "ISO8601.h"

const char* VEN_NAME = "MIR_JOON";
const char* VTN_URL = "http://166.104.28.51:8080/OpenADR2/Simple/2.0b";

using namespace std;

int main()
{
        try

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

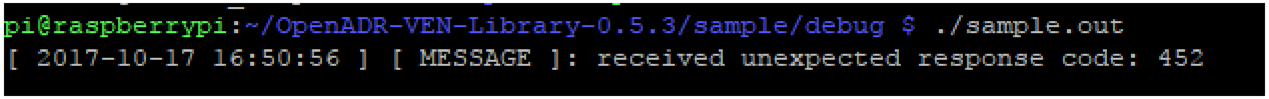
Create VEN_Name
URL must be the VTN address

9. VTN-VEN DR Test



```
pi@raspberrypi:~ $ sudo tcpdump -i wlan0 -vvv -w ~/ven_wireshark.pcap
tcpdump: listening on wlan0, link-type EN10MB (Ethernet), capture size 262144 by
tes
Got 16
```

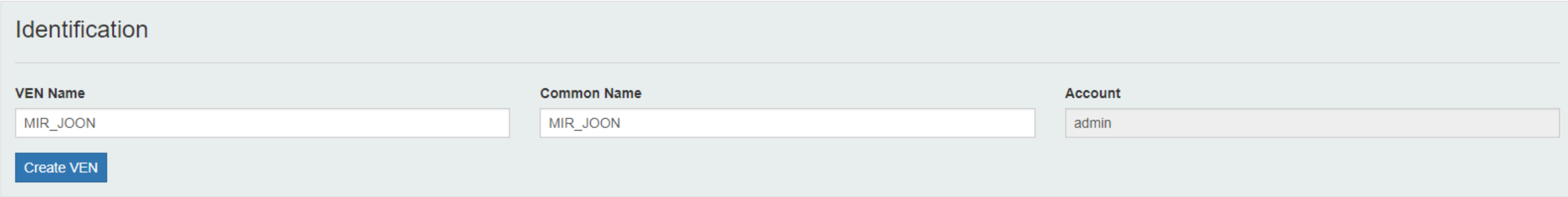
Run the tcpdump in the VEN Putty.
Duplicate session of putty is needed.



```
pi@raspberrypi:~/OpenADR-VEN-Library-0.5.3/sample/debug $ ./sample.out
[ 2017-10-17 16:50:56 ] [ MESSAGE ]: received unexpected response code: 452
```

Run the VEN Process
If this error comes, you must be forgot identification in the VTN.

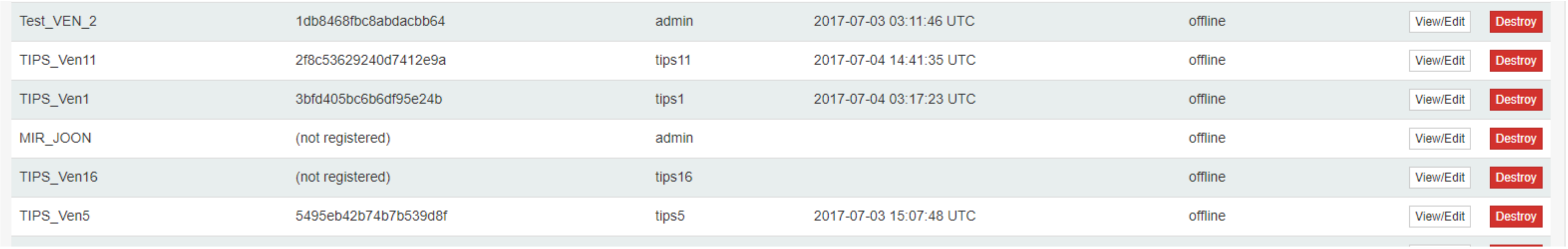
9. VTN-VEN DR Test



Identification

VEN Name	Common Name	Account
MIR_JOON	MIR_JOON	admin

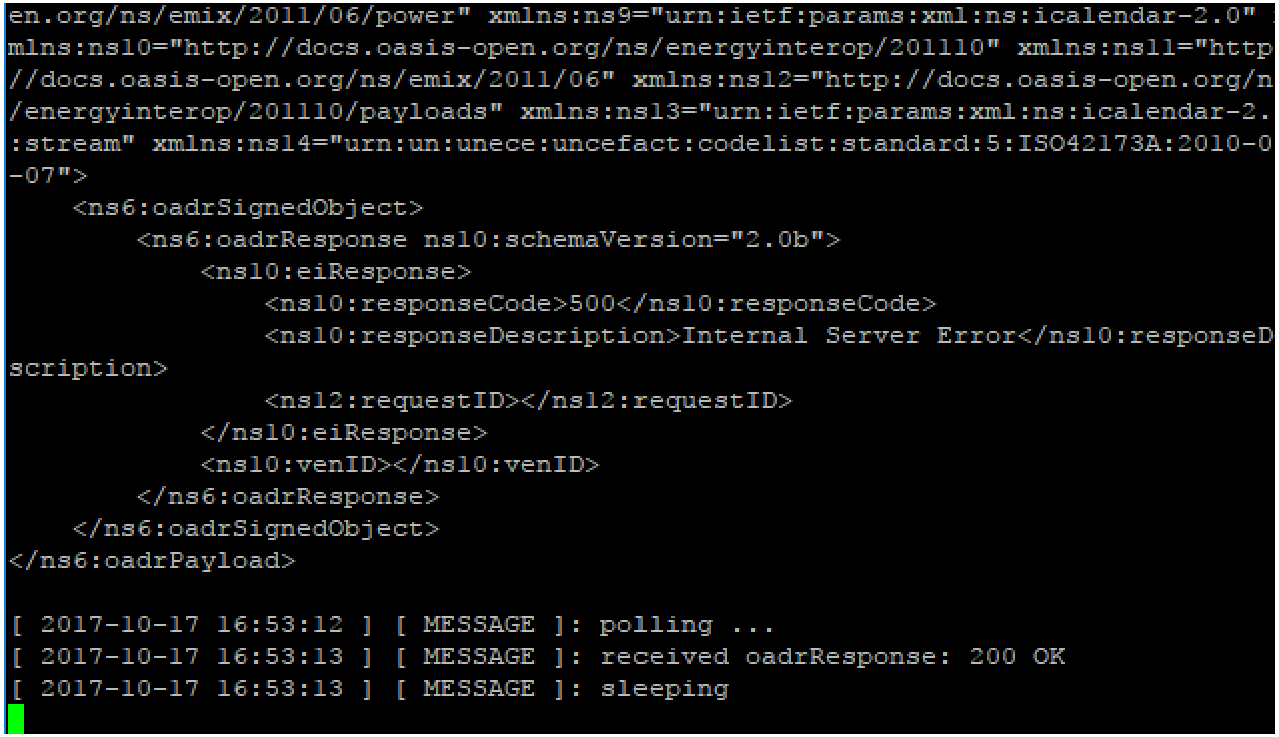
Make Ven Name and common Name.
Ven name should be same with VEN name which is settled in VEN



Test_VEN_2	1db8468fbc8abdacbb64	admin	2017-07-03 03:11:46 UTC	offline	View/Edit	Destroy
TIPS_Ven11	2f8c53629240d7412e9a	tips11	2017-07-04 14:41:35 UTC	offline	View/Edit	Destroy
TIPS_Ven1	3bfd405bc6b6df95e24b	tips1	2017-07-04 03:17:23 UTC	offline	View/Edit	Destroy
MIR_JOON	(not registered)	admin		offline	View/Edit	Destroy
TIPS_Ven16	(not registered)	tips16		offline	View/Edit	Destroy
TIPS_Ven5	5495eb42b74b7b539d8f	tips5	2017-07-03 15:07:48 UTC	offline	View/Edit	Destroy

If VEN name is settled successfully,
I can See my VEN information in VENs

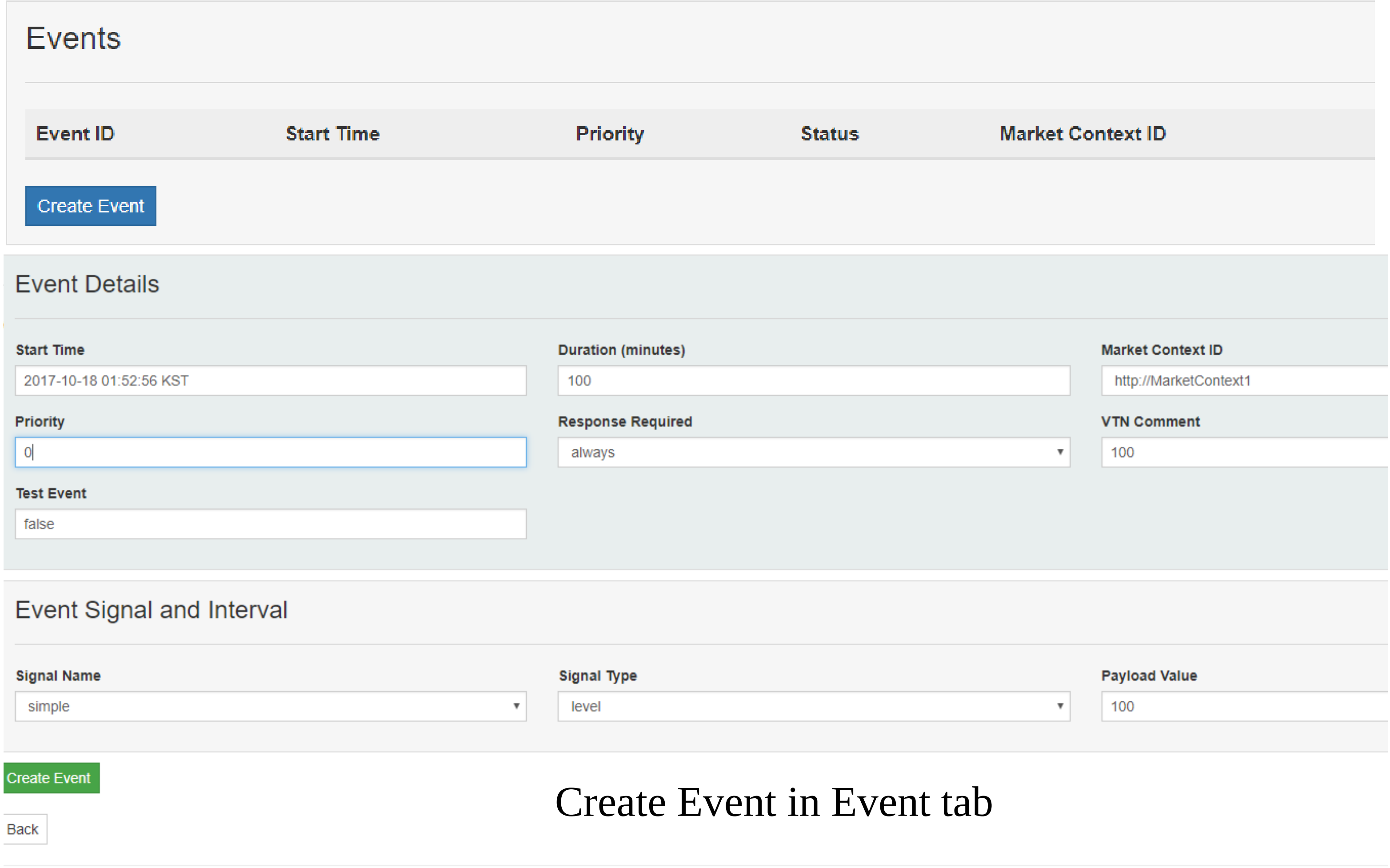
9. VTN-VEN DR Test



```
en.org/ns/emix/2011/06/power" xmlns:ns9="urn:ietf:params:xml:ns:icalendar-2.0"
mlns:ns10="http://docs.oasis-open.org/ns/energyinterop/201110" xmlns:ns11="http
//docs.oasis-open.org/ns/emix/2011/06" xmlns:ns12="http://docs.oasis-open.org/n
/energyinterop/201110/payloads" xmlns:ns13="urn:ietf:params:xml:ns:icalendar-2.
:stream" xmlns:ns14="urn:un:unece:uncefact:codelist:standard:5:ISO42173A:2010-0
-07">
    <ns6:oadrSignedObject>
        <ns6:oadrResponse ns10:schemaVersion="2.0b">
            <ns10:eiResponse>
                <ns10:responseCode>500</ns10:responseCode>
                <ns10:responseDescription>Internal Server Error</ns10:responseD
scription>
                <ns12:requestID></ns12:requestID>
            </ns10:eiResponse>
            <ns10:venID></ns10:venID>
        </ns6:oadrResponse>
    </ns6:oadrSignedObject>
</ns6:oadrPayload>

[ 2017-10-17 16:53:12 ] [ MESSAGE ]: polling ...
[ 2017-10-17 16:53:13 ] [ MESSAGE ]: received oadrResponse: 200 OK
[ 2017-10-17 16:53:13 ] [ MESSAGE ]: sleeping
```

Run sample.out.

9. VTN-VEN DR Test



Events

Event ID	Start Time	Priority	Status	Market Context ID

Event Details

Start Time: 2017-10-18 01:52:56 KST

Duration (minutes): 100

Market Context ID: http://MarketContext1

Priority: 0

Response Required: always

VTN Comment: 100

Test Event: false

Event Signal and Interval

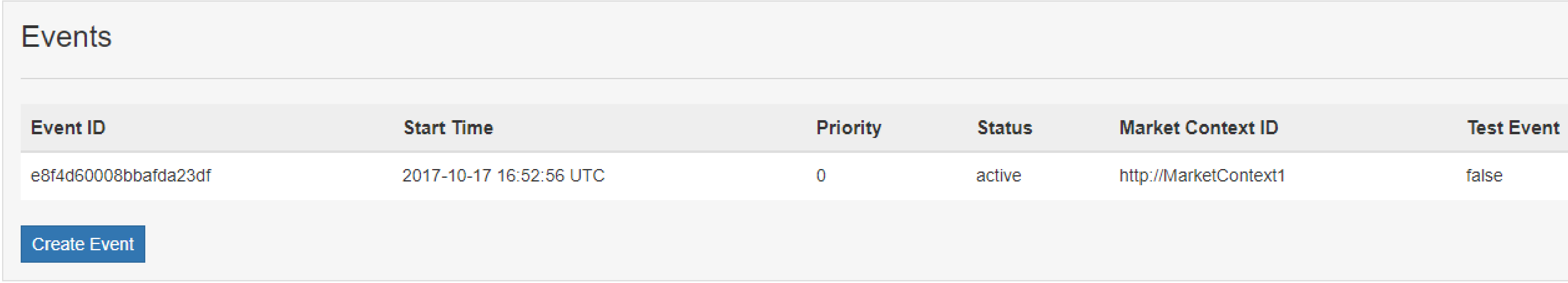
Signal Name: simple

Signal Type: level

Payload Value: 100

Create Event in Event tab

9. VTN-VEN DR Test

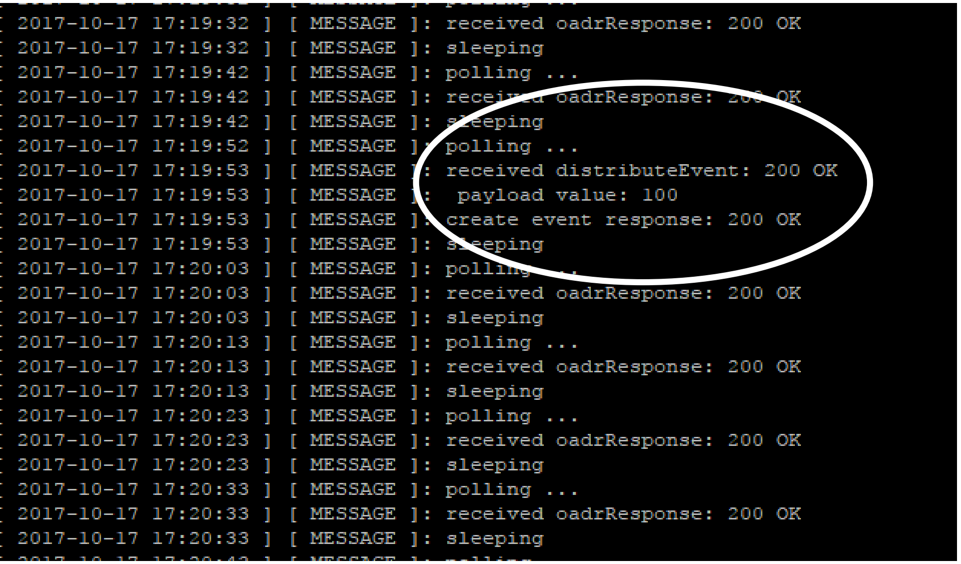


Events

Event ID	Start Time	Priority	Status	Market Context ID	Test Event
e8f4d60008bbafda23df	2017-10-17 16:52:56 UTC	0	active	http://MarketContext1	false

Copyright © 2017 EPRI

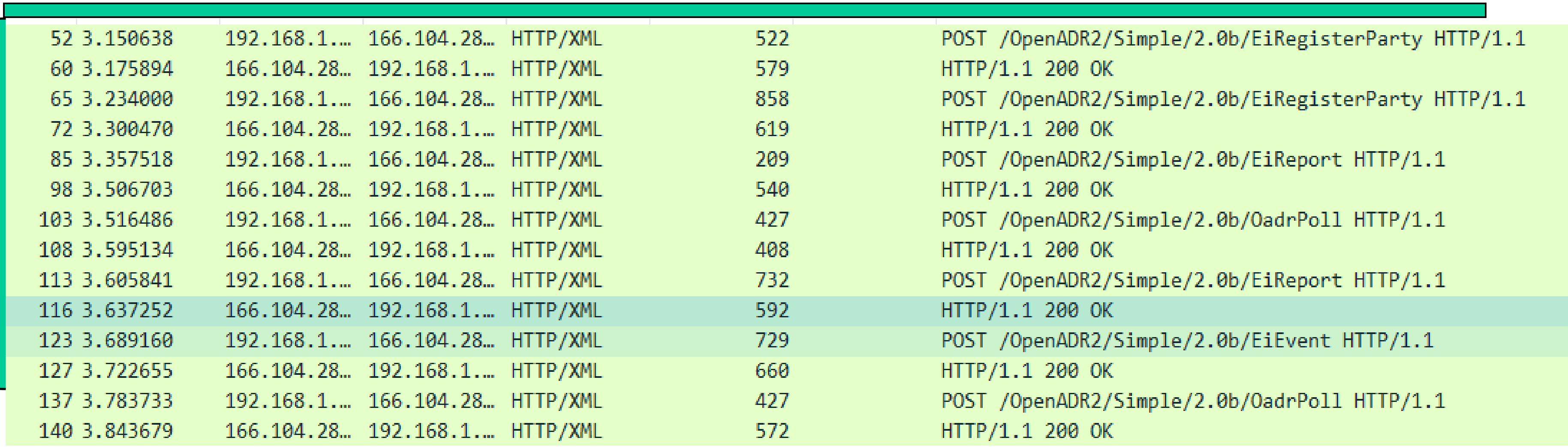
I can see the created event in Events tab



```
2017-10-17 17:19:32 ] [ MESSAGE ]: received oadrResponse: 200 OK
2017-10-17 17:19:32 ] [ MESSAGE ]: sleeping
2017-10-17 17:19:42 ] [ MESSAGE ]: polling ...
2017-10-17 17:19:42 ] [ MESSAGE ]: received oadrResponse: 200 OK
2017-10-17 17:19:42 ] [ MESSAGE ]: sleeping
2017-10-17 17:19:52 ] [ MESSAGE ]: polling ...
2017-10-17 17:19:53 ] [ MESSAGE ]: received distributeEvent: 200 OK
2017-10-17 17:19:53 ] [ MESSAGE ]:  payload value: 100
2017-10-17 17:19:53 ] [ MESSAGE ]: create event response: 200 OK
2017-10-17 17:19:53 ] [ MESSAGE ]: sleeping
2017-10-17 17:20:03 ] [ MESSAGE ]: polling ...
2017-10-17 17:20:03 ] [ MESSAGE ]: received oadrResponse: 200 OK
2017-10-17 17:20:03 ] [ MESSAGE ]: sleeping
2017-10-17 17:20:13 ] [ MESSAGE ]: polling ...
2017-10-17 17:20:13 ] [ MESSAGE ]: received oadrResponse: 200 OK
2017-10-17 17:20:13 ] [ MESSAGE ]: sleeping
2017-10-17 17:20:23 ] [ MESSAGE ]: polling ...
2017-10-17 17:20:23 ] [ MESSAGE ]: received oadrResponse: 200 OK
2017-10-17 17:20:23 ] [ MESSAGE ]: sleeping
2017-10-17 17:20:33 ] [ MESSAGE ]: polling ...
2017-10-17 17:20:33 ] [ MESSAGE ]: received oadrResponse: 200 OK
2017-10-17 17:20:33 ] [ MESSAGE ]: sleeping
```

VEN receives Event from VTN

9. VTN-VEN Wireshark



52	3.150638	192.168.1.…	166.104.28…	HTTP/XML	522	POST /OpenADR2/Simple/2.0b/EiRegisterParty HTTP/1.1
60	3.175894	166.104.28…	192.168.1.…	HTTP/XML	579	HTTP/1.1 200 OK
65	3.234000	192.168.1.…	166.104.28…	HTTP/XML	858	POST /OpenADR2/Simple/2.0b/EiRegisterParty HTTP/1.1
72	3.300470	166.104.28…	192.168.1.…	HTTP/XML	619	HTTP/1.1 200 OK
85	3.357518	192.168.1.…	166.104.28…	HTTP/XML	209	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
98	3.506703	166.104.28…	192.168.1.…	HTTP/XML	540	HTTP/1.1 200 OK
103	3.516486	192.168.1.…	166.104.28…	HTTP/XML	427	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
108	3.595134	166.104.28…	192.168.1.…	HTTP/XML	408	HTTP/1.1 200 OK
113	3.605841	192.168.1.…	166.104.28…	HTTP/XML	732	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
116	3.637252	166.104.28…	192.168.1.…	HTTP/XML	592	HTTP/1.1 200 OK
123	3.689160	192.168.1.…	166.104.28…	HTTP/XML	729	POST /OpenADR2/Simple/2.0b/EiEvent HTTP/1.1
127	3.722655	166.104.28…	192.168.1.…	HTTP/XML	660	HTTP/1.1 200 OK
137	3.783733	192.168.1.…	166.104.28…	HTTP/XML	427	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
140	3.843679	166.104.28…	192.168.1.…	HTTP/XML	572	HTTP/1.1 200 OK

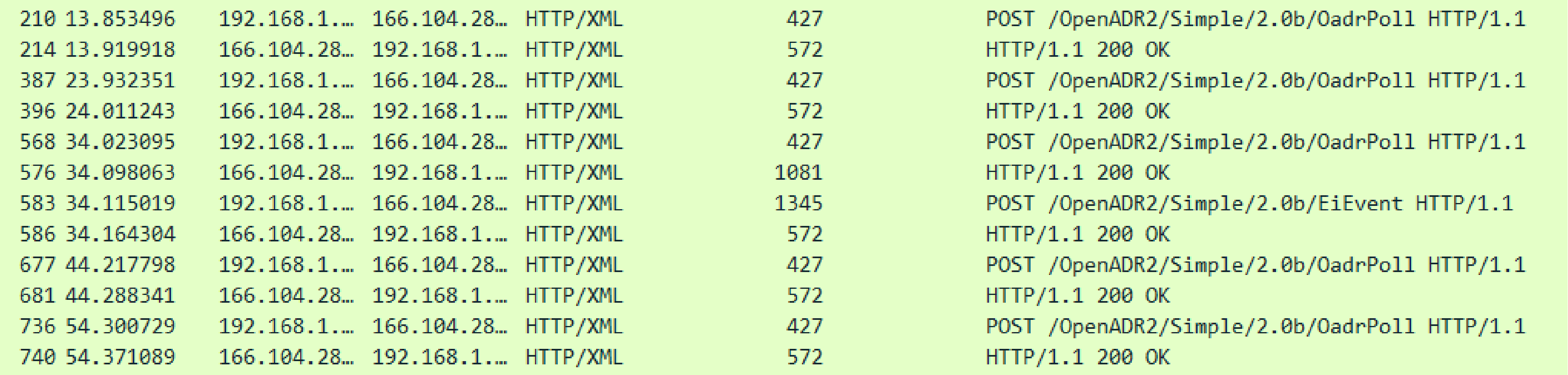
VEN makes connection with VTN.

VEN sends

- RegisterParty (Registration Starts)
- EiReport
- OadrPoll
- EiReport
- Eievent (Registration ends)
- OadrPoll

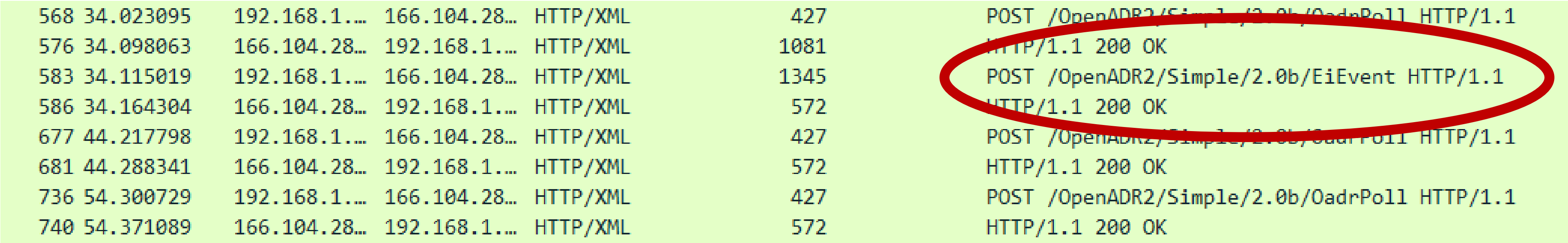
and each receives 200 OK

9. VTN-VEN Wireshark



```
210 13.853496   192.168.1.…  166.104.28…  HTTP/XML     427   POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
214 13.919918   166.104.28…  192.168.1.…  HTTP/XML     572   HTTP/1.1 200 OK
387 23.932351   192.168.1.…  166.104.28…  HTTP/XML     427   POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
396 24.011243   166.104.28…  192.168.1.…  HTTP/XML     572   HTTP/1.1 200 OK
568 34.023095   192.168.1.…  166.104.28…  HTTP/XML     427   POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
576 34.098063   166.104.28…  192.168.1.…  HTTP/XML    1081   HTTP/1.1 200 OK
583 34.115019   192.168.1.…  166.104.28…  HTTP/XML    1345   POST /OpenADR2/Simple/2.0b/EiEvent HTTP/1.1
586 34.164304   166.104.28…  192.168.1.…  HTTP/XML     572   HTTP/1.1 200 OK
677 44.217798   192.168.1.…  166.104.28…  HTTP/XML     427   POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
681 44.288341   166.104.28…  192.168.1.…  HTTP/XML     572   HTTP/1.1 200 OK
736 54.300729   192.168.1.…  166.104.28…  HTTP/XML     427   POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
740 54.371089   166.104.28…  192.168.1.…  HTTP/XML     572   HTTP/1.1 200 OK
```

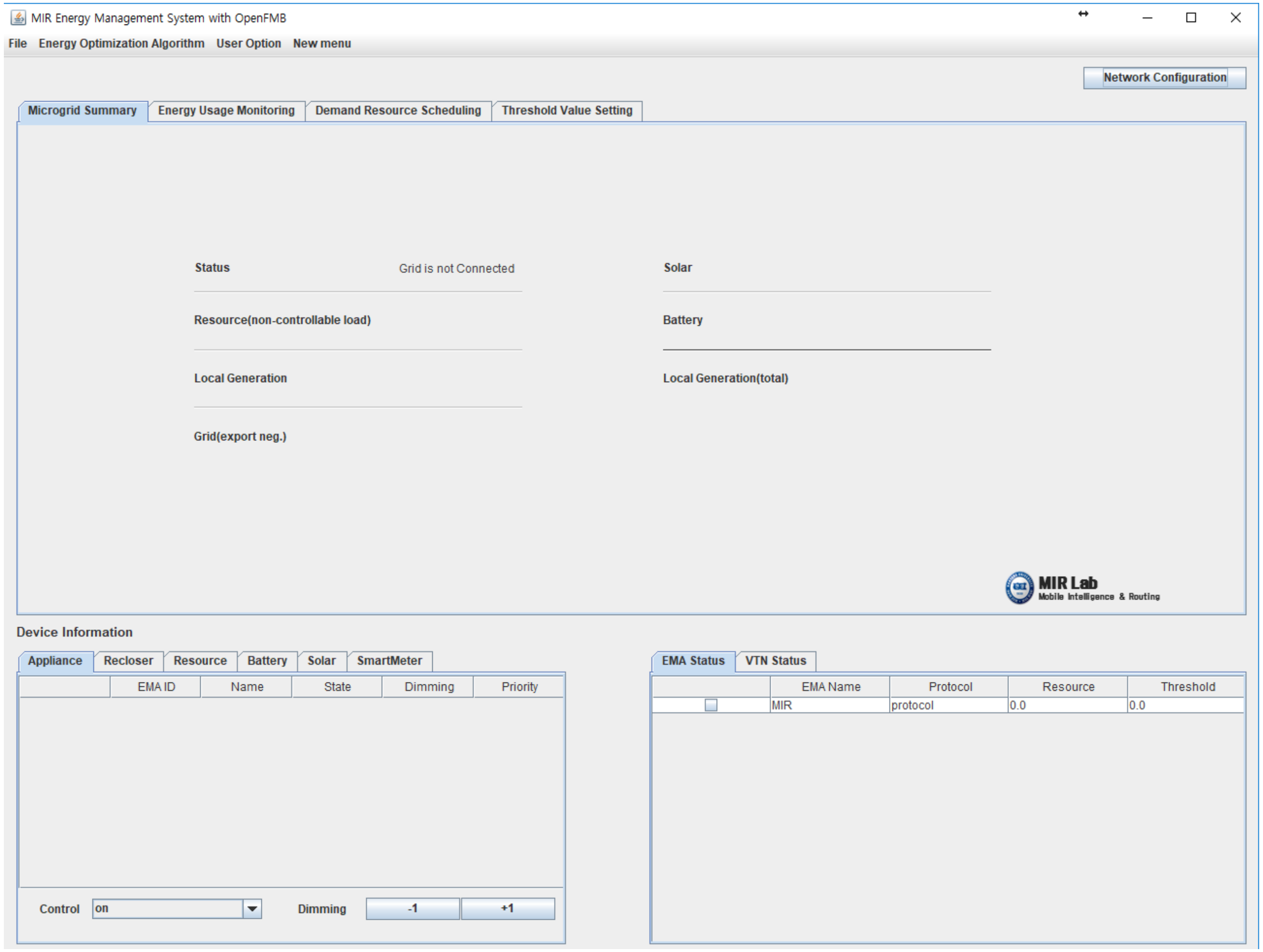
Poll Messages sends every 10 seconds



```
568 34.023095   192.168.1.…  166.104.28…  HTTP/XML     427   POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
576 34.098063   166.104.28…  192.168.1.…  HTTP/XML    1081   HTTP/1.1 200 OK
583 34.115019   192.168.1.…  166.104.28…  HTTP/XML    1345   POST /OpenADR2/Simple/2.0b/EiEvent HTTP/1.1
586 34.164304   166.104.28…  192.168.1.…  HTTP/XML     572   HTTP/1.1 200 OK
677 44.217798   192.168.1.…  166.104.28…  HTTP/XML     427   POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
681 44.288341   166.104.28…  192.168.1.…  HTTP/XML     572   HTTP/1.1 200 OK
736 54.300729   192.168.1.…  166.104.28…  HTTP/XML     427   POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
740 54.371089   166.104.28…  192.168.1.…  HTTP/XML     572   HTTP/1.1 200 OK
```

EiEvent Has come to VEN and sends 200 OK Back (66.241 ms)

9. EMS-EMA - MQTT



Status Grid is not Connected

Solar

Resource(non-controllable load)

Battery

Local Generation

Local Generation(total)

Grid(export neg.)

Device Information

	EMA ID	Name	State	Dimming	Priority

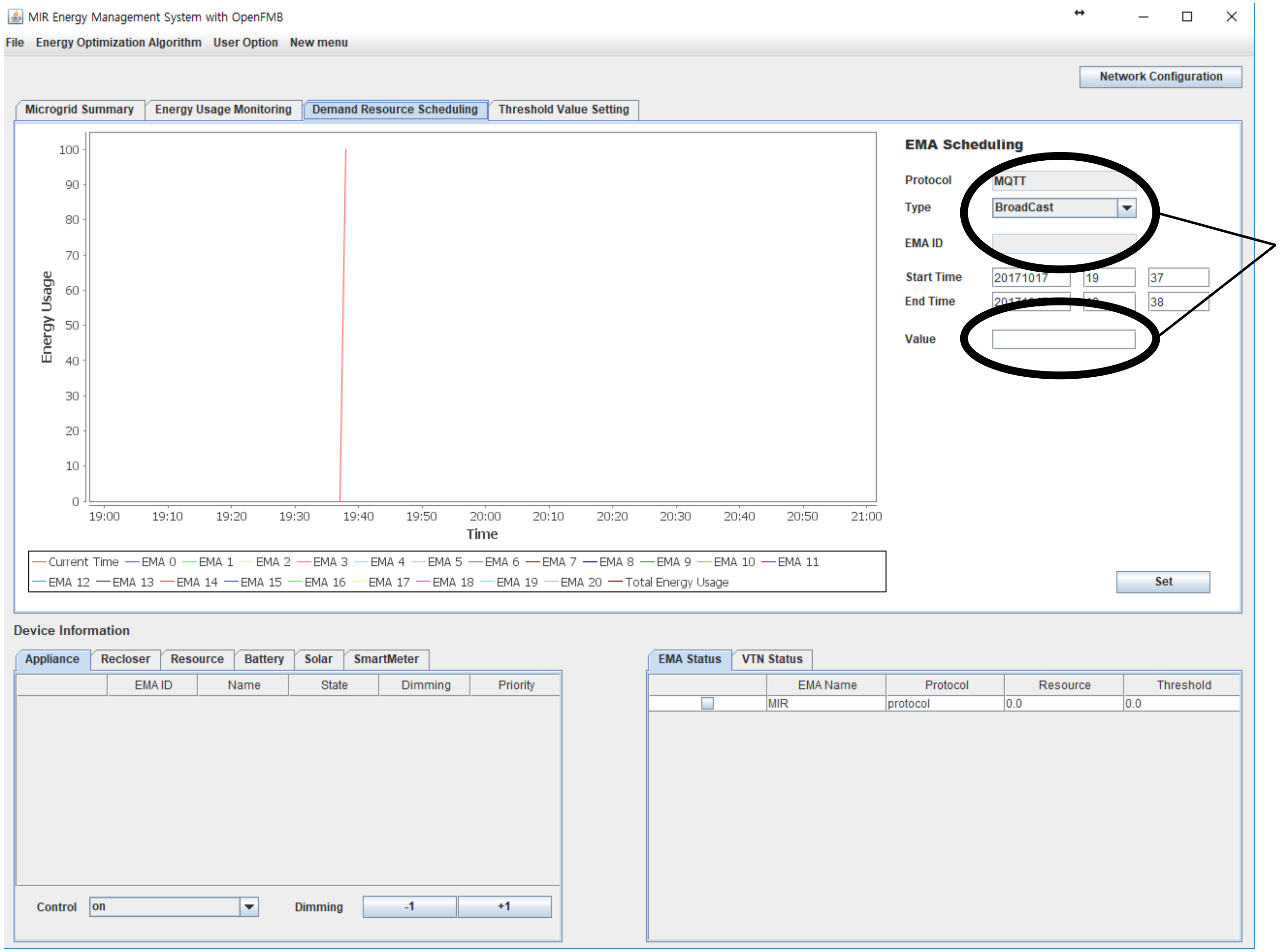
Control on

Dimming

	EMA Name	Protocol	Resource	Threshold
□	MIR	protocol	0.0	0.0

EMS – 실행 BASE

9. EMS-EMA - MQTT



MIR Energy Management System with OpenFMB

EMA Scheduling

Protocol: MQTT

Type: BroadCast

EMA ID:

Start Time: 20171017 19 37

End Time: 20171017 19 38

Value:

Device Information

	EMA ID	Name	State	Dimming	Priority

Control: on

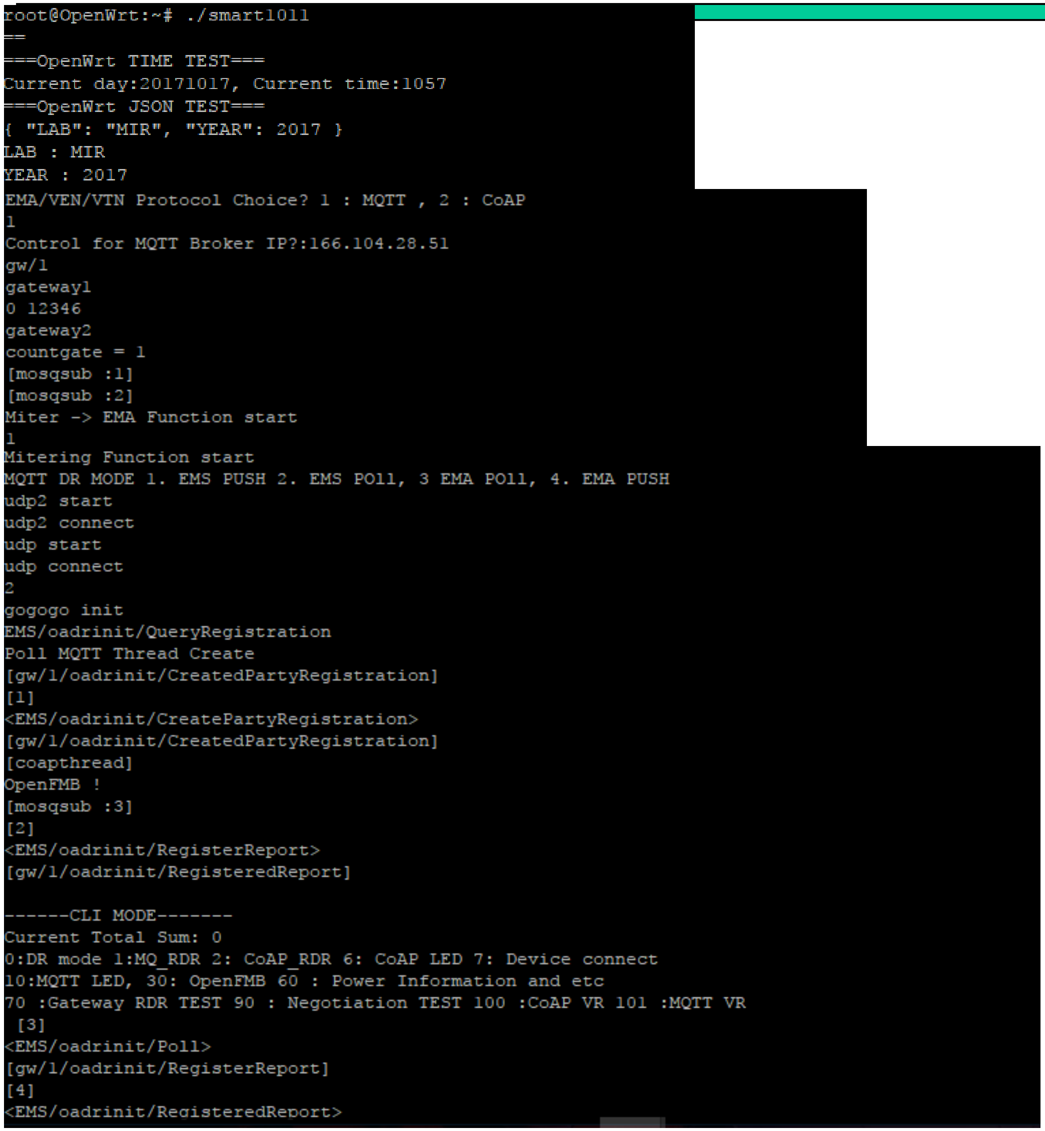
	EMA Name	Protocol	Resource	Threshold
□	MIR	protocol	0.0	0.0

EMS – 실행 BASE

- DR을 내리기 위한 TAB

이 곳을 확인, 이후 사용

9. EMS-EMA - MQTT



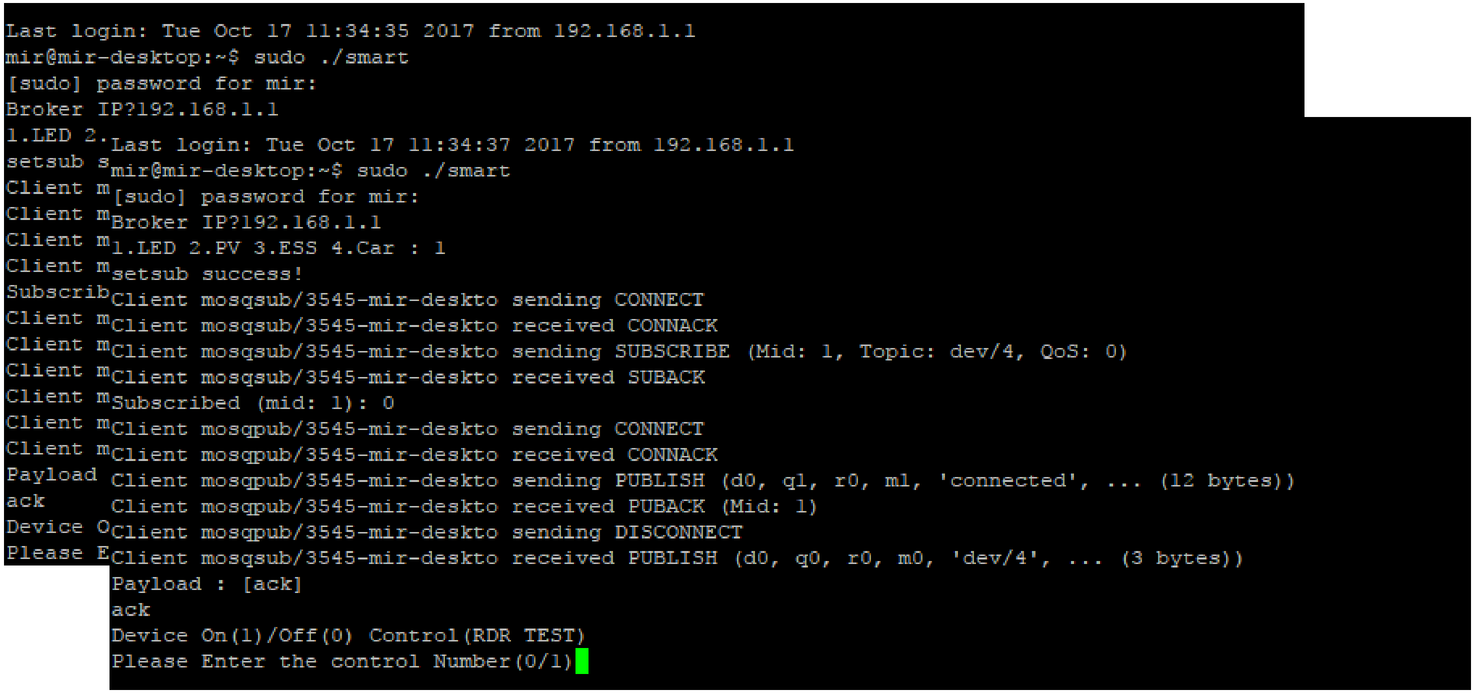
```
root@OpenWrt:~# ./smart1011
==
===OpenWrt TIME TEST===
Current day:20171017, Current time:1057
===OpenWrt JSON TEST===
{ "LAB": "MIR", "YEAR": 2017 }
LAB : MIR
YEAR : 2017
EMA/VEN/VTN Protocol Choice? 1 : MQTT , 2 : CoAP
1
Control for MQTT Broker IP?:166.104.28.51
gw/1
gateway1
0 12346
gateway2
countgate = 1
[mosqsub :1]
[mosqsub :2]
Miter -> EMA Function start
1
Mitering Function start
MQTT DR MODE 1. EMS PUSH 2. EMS POll, 3 EMA POll, 4. EMA PUSH
udp2 start
udp2 connect
udp start
udp connect
2
gogogo init
EMS/oadrinit/QueryRegistration
Poll MQTT Thread Create
[gw/1/oadrinit/CreatedPartyRegistration]
[1]
<EMS/oadrinit/CreatePartyRegistration>
[gw/1/oadrinit/CreatedPartyRegistration]
[coapthread]
OpenFMB !
[mosqsub :3]
[2]
<EMS/oadrinit/RegisterReport>
[gw/1/oadrinit/RegisteredReport]

------CLI MODE-------
Current Total Sum: 0
0:DR mode 1:MQ_RDR 2: CoAP_RDR 6: CoAP LED 7: Device connect
10:MQTT LED, 30: OpenFMB 60 : Power Information and etc
70 :Gateway RDR TEST 90 : Negotiation TEST 100 :CoAP VR 101 :MQTT VR
 [3]
<EMS/oadrinit/Poll>
[gw/1/oadrinit/RegisterReport]
[4]
<EMS/oadrinit/RegisteredReport>
```

Putty를 통하여
OPENWRT 접속 후에 EMA 실행
Protocol MQTT 선택

Broker IP 입력하여
Broker에 접속한다.

EMS의 DR 모드 선택
(PUSH or POLL)

9. EMS-EMA - MQTT

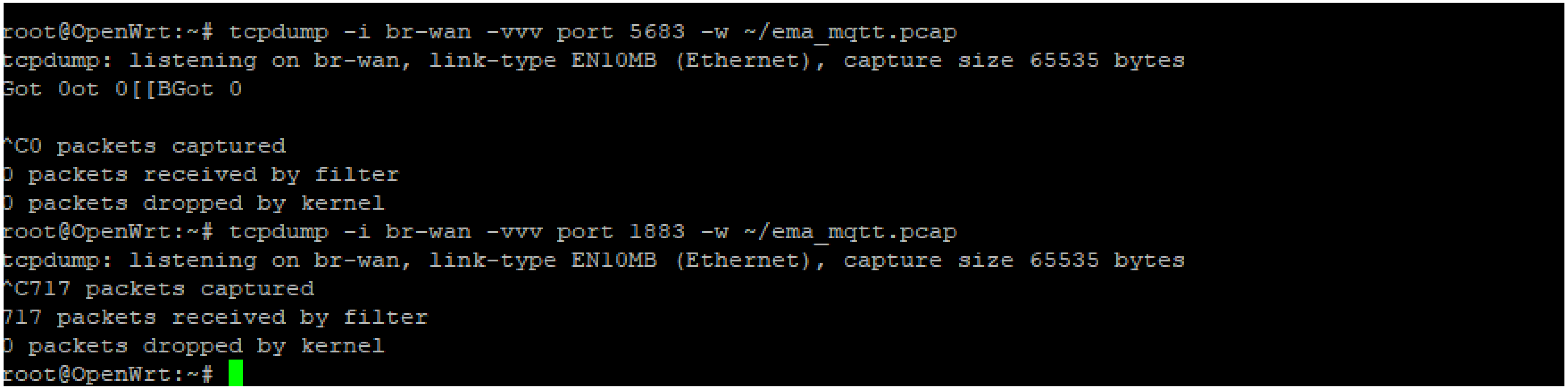


```
Last login: Tue Oct 17 11:34:35 2017 from 192.168.1.1
mir@mir-desktop:~$ sudo ./smart
[sudo] password for mir:
Broker IP?192.168.1.1
1.LED 2.
setsub s
Client m
Client m
Client m
Client m
Subscrib
Client m
Client m
Client m
Client m
Client m
Client m
Payload
ack
Device O
Please E
```

```
Last login: Tue Oct 17 11:34:37 2017 from 192.168.1.1
mir@mir-desktop:~$ sudo ./smart
[sudo] password for mir:
Broker IP?192.168.1.1
1.LED 2.PV 3.ESS 4.Car : 1
setsub success!
Client mosqsub/3545-mir-deskto sending CONNECT
Client mosqsub/3545-mir-deskto received CONNACK
Client mosqsub/3545-mir-deskto sending SUBSCRIBE (Mid: 1, Topic: dev/4, QoS: 0)
Client mosqsub/3545-mir-deskto received SUBACK
Subscribed (mid: 1): 0
Client mosqpub/3545-mir-deskto sending CONNECT
Client mosqpub/3545-mir-deskto received CONNACK
Client mosqpub/3545-mir-deskto sending PUBLISH (d0, q1, r0, m1, 'connected', ... (12 bytes))
Client mosqpub/3545-mir-deskto received PUBACK (Mid: 1)
Client mosqpub/3545-mir-deskto sending DISCONNECT
Client mosqsub/3545-mir-deskto received PUBLISH (d0, q0, r0, m0, 'dev/4', ... (3 bytes))
Payload : [ack]
ack
Device On(1)/Off(0) Control(RDR TEST)
Please Enter the control Number(0/1)
```

Putty를 통하여 Device에 접속
해당 기기는 LED 이므로 LED를 선택한다.
이후 해당 putty를 통하여 기기를 직접 control 할 수 있다.

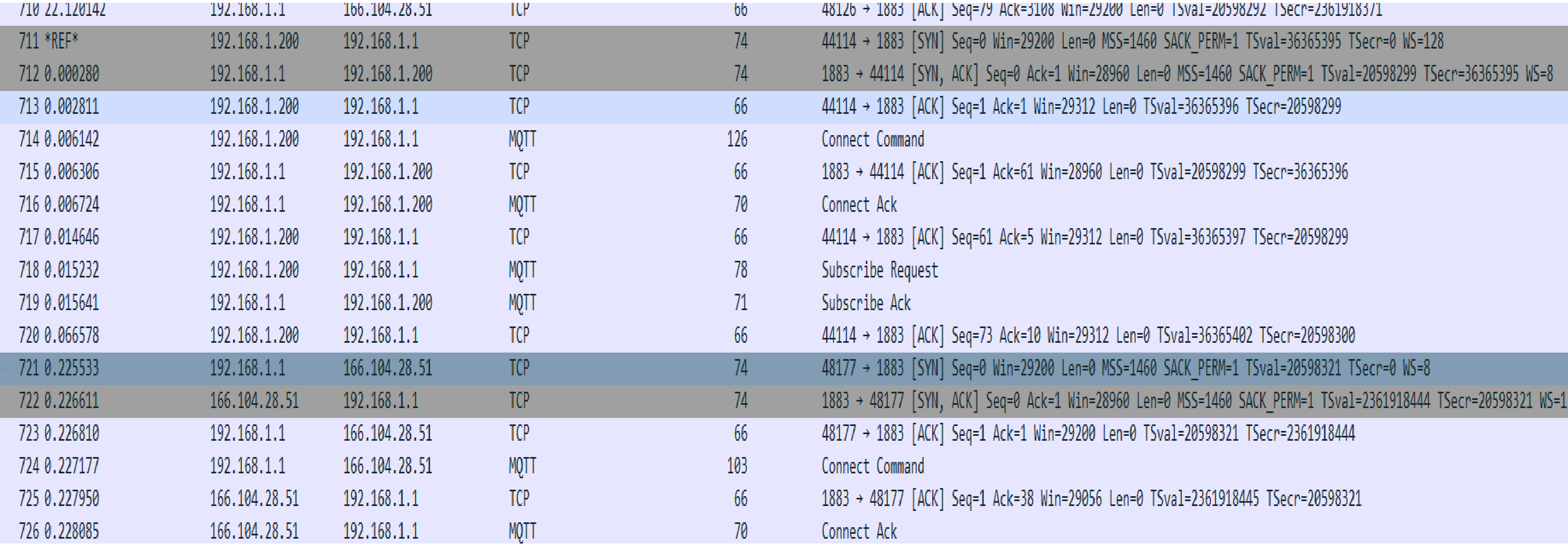
9. EMS-EMA - MQTT



```
root@OpenWrt:~# tcpdump -i br-wan -vvv port 5683 -w ~/ema_mqtt.pcap
tcpdump: listening on br-wan, link-type EN10MB (Ethernet), capture size 65535 bytes
Got 0ot 0[[BGot 0

^C0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@OpenWrt:~# tcpdump -i br-wan -vvv port 1883 -w ~/ema_mqtt.pcap
tcpdump: listening on br-wan, link-type EN10MB (Ethernet), capture size 65535 bytes
^C717 packets captured
717 packets received by filter
0 packets dropped by kernel
root@OpenWrt:~#
```

MQTT EMA 가 실행되는 OPENWRT에 추가 접속.
Tcp dump를 찍는다.
MQTT 의 tcp dump 는 **1883**
종료 후 winscp를 통하여 pcap 파일을 가져와 wireshark 파일을 통하여 분석한다.

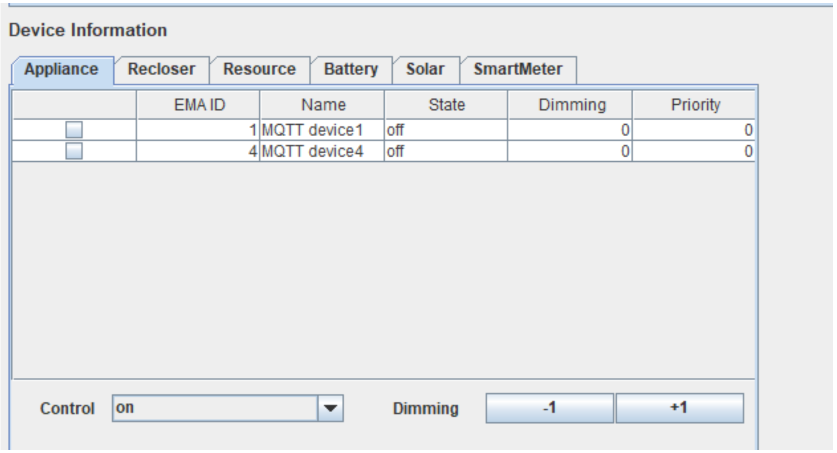
9. Device Connection



```
710 22.120142   192.168.1.1     166.104.28.51   TCP    66   48126 → 1883 [ACK] Seq=79 Ack=3108 Win=29200 Len=0 TSval=20598292 TSecr=2361918371
711 *REF*       192.168.1.200   192.168.1.1     TCP    74   44114 → 1883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=36365395 TSecr=0 WS=128
712 0.000280    192.168.1.1     192.168.1.200   TCP    74   1883 → 44114 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=20598299 TSecr=36365395 WS=8
713 0.002811    192.168.1.200   192.168.1.1     TCP    66   44114 → 1883 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=36365396 TSecr=20598299
714 0.006142    192.168.1.200   192.168.1.1     MQTT  126   Connect Command
715 0.006306    192.168.1.1     192.168.1.200   TCP    66   1883 → 44114 [ACK] Seq=1 Ack=61 Win=28960 Len=0 TSval=20598299 TSecr=36365396
716 0.006724    192.168.1.1     192.168.1.200   MQTT   70   Connect Ack
717 0.014646    192.168.1.200   192.168.1.1     TCP    66   44114 → 1883 [ACK] Seq=61 Ack=5 Win=29312 Len=0 TSval=36365397 TSecr=20598299
718 0.015232    192.168.1.200   192.168.1.1     MQTT   78   Subscribe Request
719 0.015641    192.168.1.1     192.168.1.200   MQTT   71   Subscribe Ack
720 0.066578    192.168.1.200   192.168.1.1     TCP    66   44114 → 1883 [ACK] Seq=73 Ack=10 Win=29312 Len=0 TSval=36365402 TSecr=20598300
721 0.225533    192.168.1.1     166.104.28.51   TCP    74   48177 → 1883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=20598321 TSecr=0 WS=8
722 0.226611    166.104.28.51   192.168.1.1     TCP    74   1883 → 48177 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2361918444 TSecr=20598321 WS=1
723 0.226810    192.168.1.1     166.104.28.51   TCP    66   48177 → 1883 [ACK] Seq=1 Ack=1 Win=29200 Len=0 TSval=20598321 TSecr=2361918444
724 0.227177    192.168.1.1     166.104.28.51   MQTT  103   Connect Command
725 0.227950    166.104.28.51   192.168.1.1     TCP    66   1883 → 48177 [ACK] Seq=1 Ack=38 Win=29056 Len=0 TSval=2361918445 TSecr=20598321
726 0.228085    166.104.28.51   192.168.1.1     MQTT   70   Connect Ack
```

Device Connect to EMA (TCP Connection)
EMA ACK to Device
Device Connect to EMA (MQTT Connect command)
EMA ACK to Device
EMA Send Connect Command to Broker
Broker ACKS, Connect ACK Returns

9. EMS-EMA - MQTT



Device Information

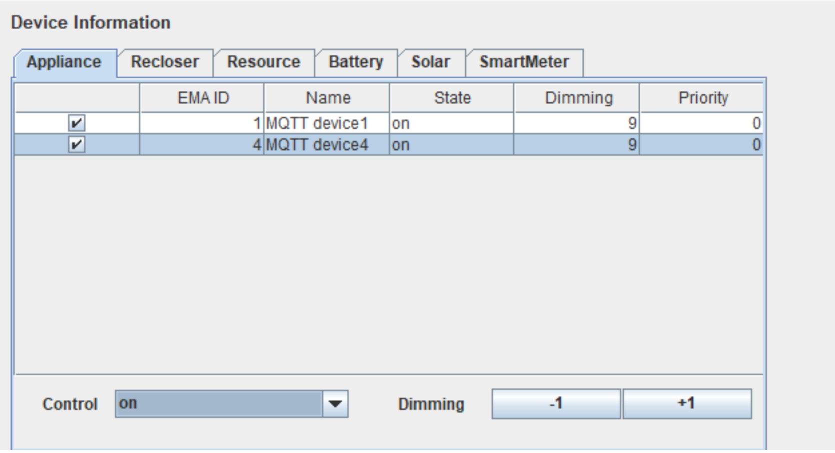
Appliance | Recloser | Resource | Battery | Solar | SmartMeter

	EMA ID	Name	State	Dimming	Priority
□	1	MQTT device1	off	0	0
□	4	MQTT device4	off	0	0

Control: on

Dimming: -1 | +1

Device가 EMA에 접속 시, EMS에 등록된다.



Device Information

Appliance | Recloser | Resource | Battery | Solar | SmartMeter

	EMA ID	Name	State	Dimming	Priority
☑	1	MQTT device1	on	9	0
☑	4	MQTT device4	on	9	0

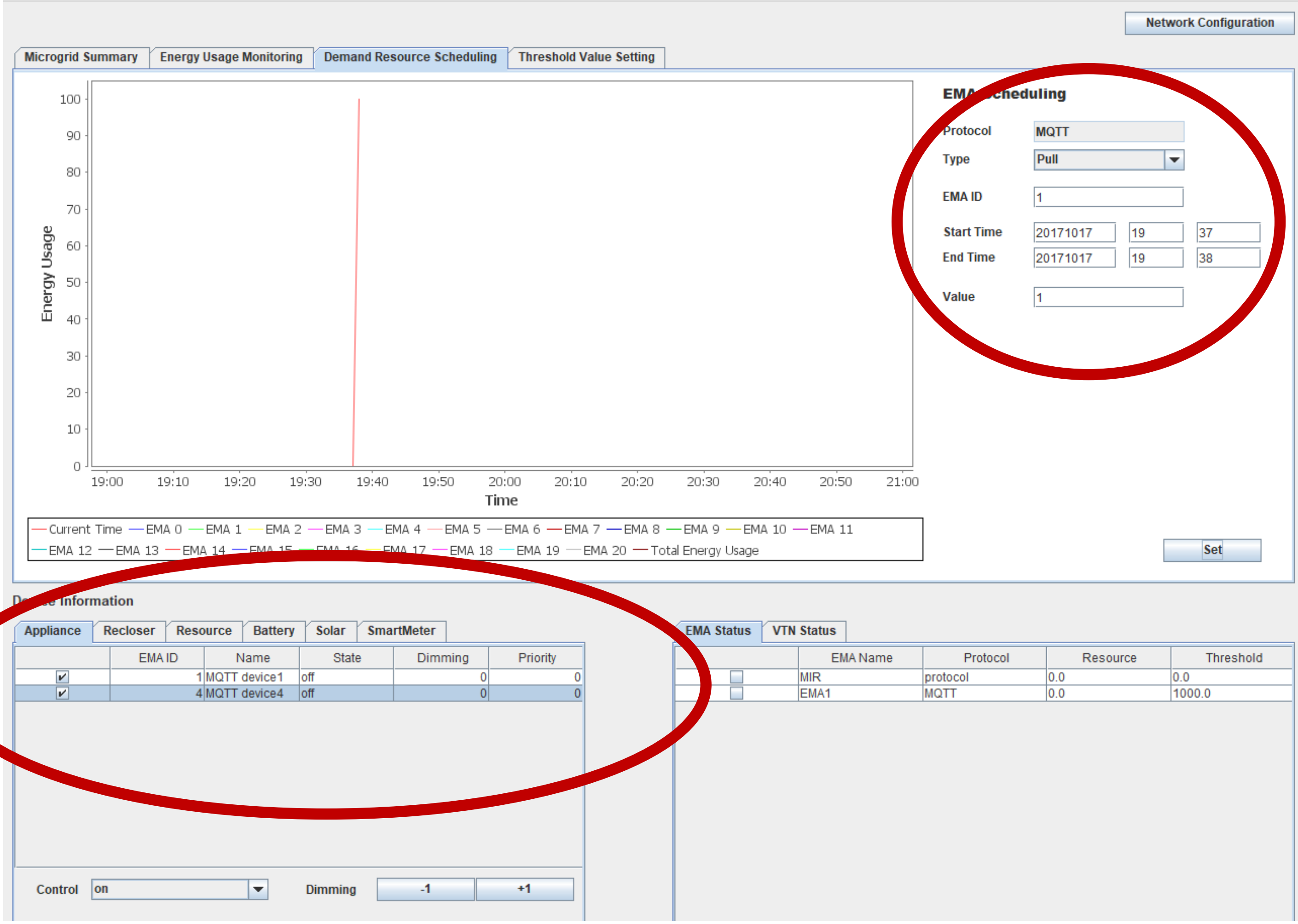
Control: on

Dimming: -1 | +1

EMS를 통하여 등록된 기기를 제어할 수 있다

On, Off 조작 가능

9. EMS-EMA - MQTT



EMA Scheduling

Protocol: MQTT

Type: Pull

EMA ID: 1

Start Time: 20171017 19 37

End Time: 20171017 19 38

Value: 1

Device Information

	EMA ID	Name	State	Dimming	Priority
☑	1	MQTT device1	off	0	0
☑	4	MQTT device4	off	0	0

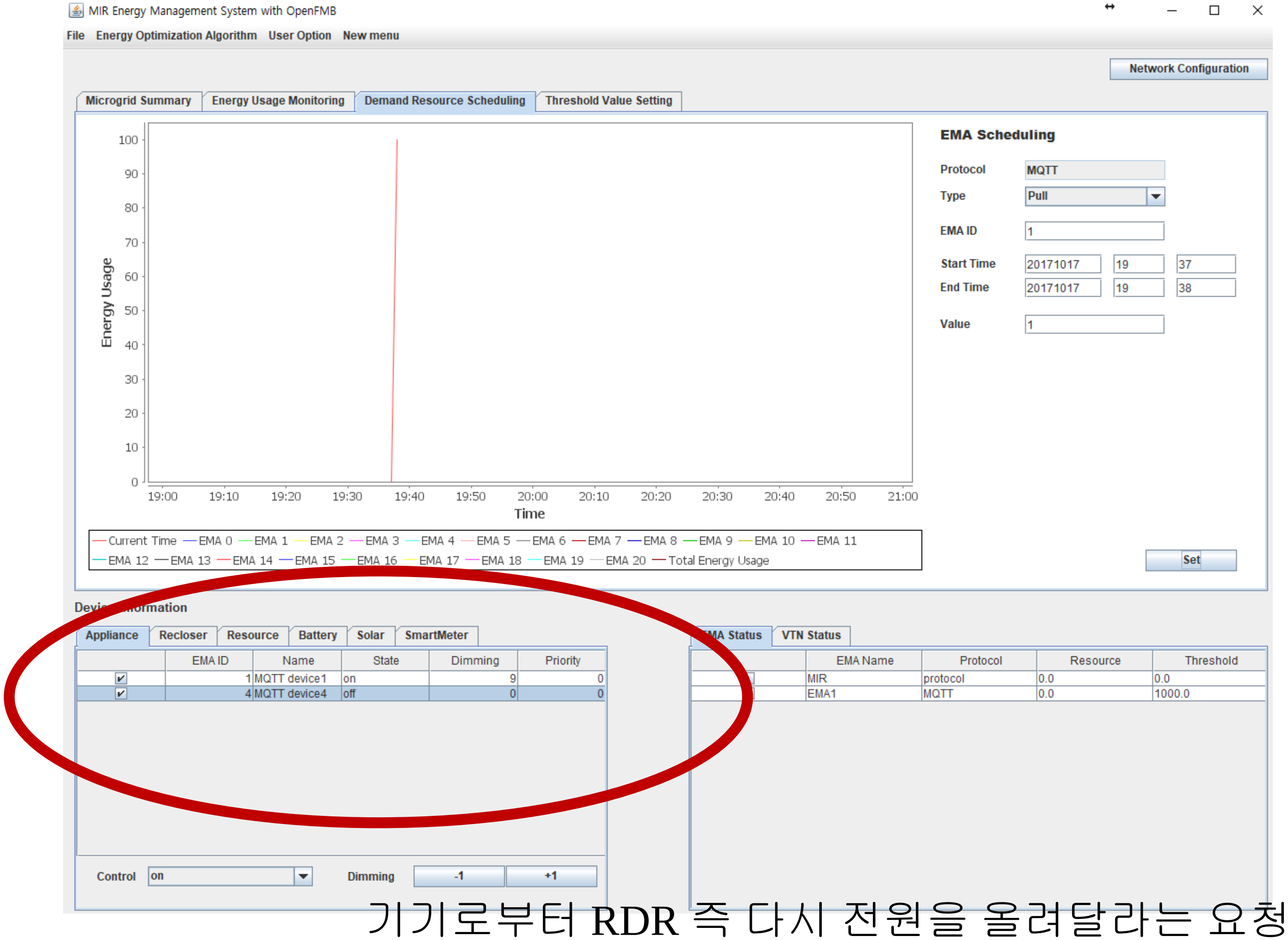
Control: on

	EMA Name	Protocol	Resource	Threshold
□	MIR	protocol	0.0	0.0
□	EMA1	MQTT	0.0	1000.0

MQTT Pull 방식
EMA에 직접 DR을
내릴 수 있다.
Value 값 제한을 두고,
해당 값이 넘어갈 경우
기기의 전원이 내려간

EMS가 DR을 내리자, 이에 따라 EMS에 등록된 EMA의 device의 전원이 내려간 것을 확인 할 수 있다.

9. EMS-EMA - MQTT



EMA Scheduling

Protocol: MQTT

Type: Pull

EMA ID: 1

Start Time: 20171017 19 37

End Time: 20171017 19 38

Value: 1

	EMA ID	Name	State	Dimming	Priority
☑	1	MQTT device1	on	9	0
☑	4	MQTT device4	off	0	0

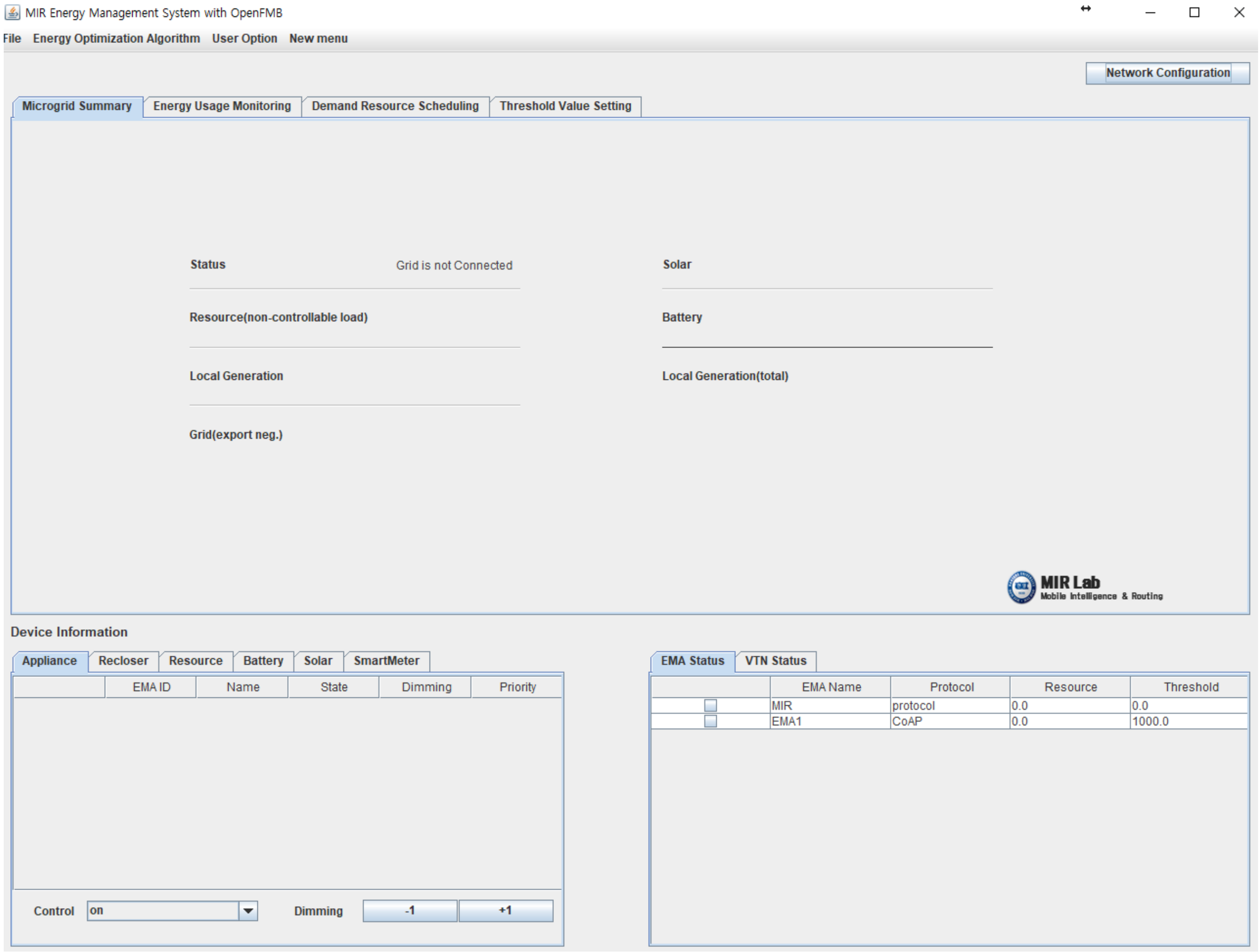
Control: on

Dimming

	EMA Name	Protocol	Resource	Threshold
	MIR	protocol	0.0	0.0
	EMA1	MQTT	0.0	1000.0

기기로부터 RDR 즉 다시 전원을 올려달라는 요청이 올 시에 전원이 다시 올라 간 것을 확인 할 수 있다.

9. EMS-EMA - CoAP



Status Grid is not Connected

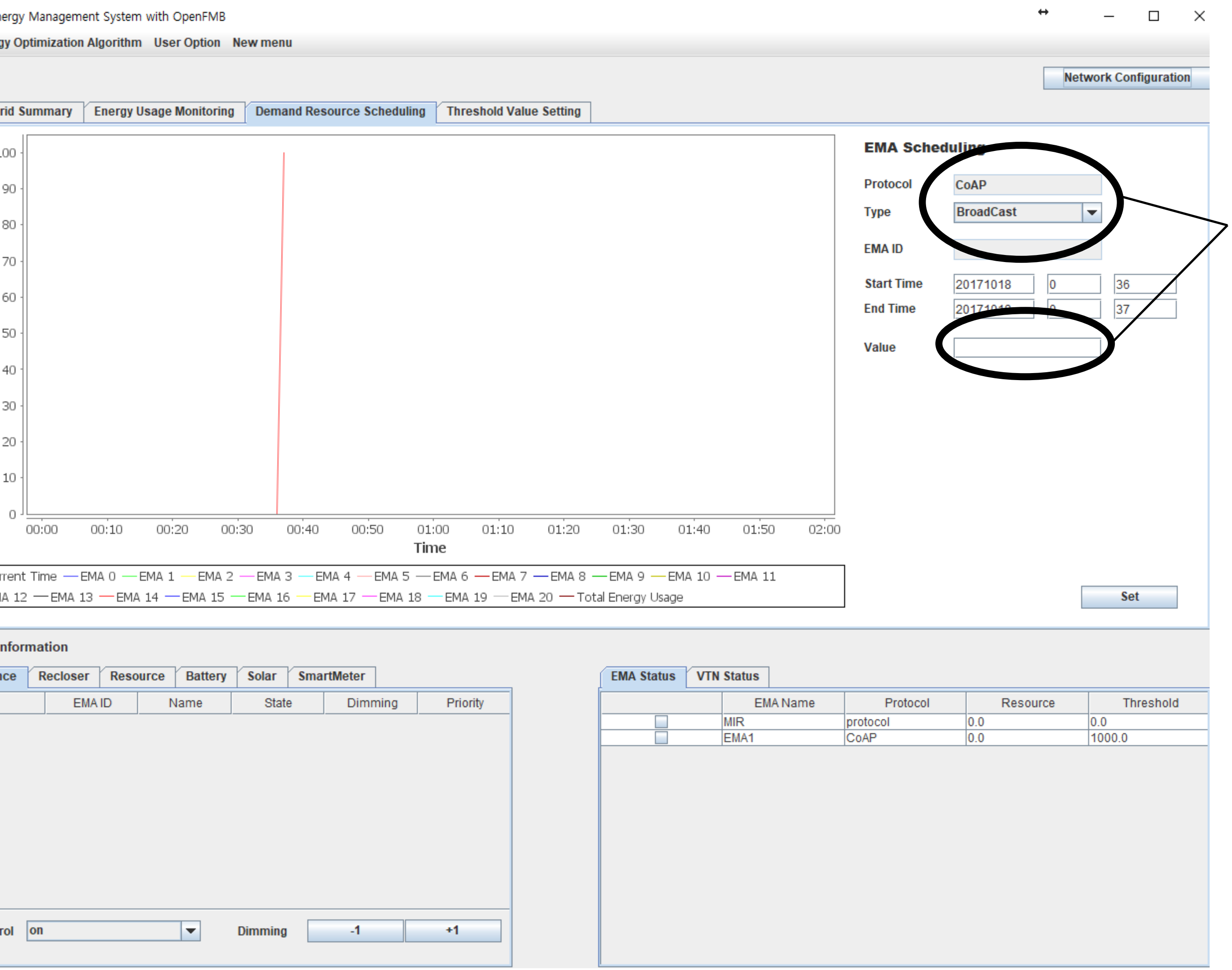
	EMA ID	Name	State	Dimming	Priority

Control on

	EMA Name	Protocol	Resource	Threshold
□	MIR	protocol	0.0	0.0
□	EMA1	CoAP	0.0	1000.0

EMS – 실행 BASE

9. EMS-EMA - CoAP



ergy Management System with OpenFMB
gy Optimization Algorithm
User Option
New menu
Network Configuration
rid Summary
Energy Usage Monitoring
Demand Resource Scheduling
Threshold Value Setting
.00
90
80
70
60
50
40
30
20
10
0
00:00
00:10
00:20
00:30
00:40
00:50
01:00
01:10
01:20
01:30
01:40
01:50
02:00
Time
rrent Time — EMA 0 — EMA 1 — EMA 2 — EMA 3 — EMA 4 — EMA 5 — EMA 6 — EMA 7 — EMA 8 — EMA 9 — EMA 10 — EMA 11
IA 12 — EMA 13 — EMA 14 — EMA 15 — EMA 16 — EMA 17 — EMA 18 — EMA 19 — EMA 20 — Total Energy Usage
Set
nformation
ce
Recloser
Resource
Battery
Solar
SmartMeter
EMA Status
VTN Status
rol
Dimming
-1
+1

EMA Scheduling

Protocol: CoAP

Type: BroadCast ▼

EMA ID:

Start Time: 20171018 | 0 | 36

End Time: 20171018 | 0 | 37

Value:

	EMA ID	Name	State	Dimming	Priority

on ▼

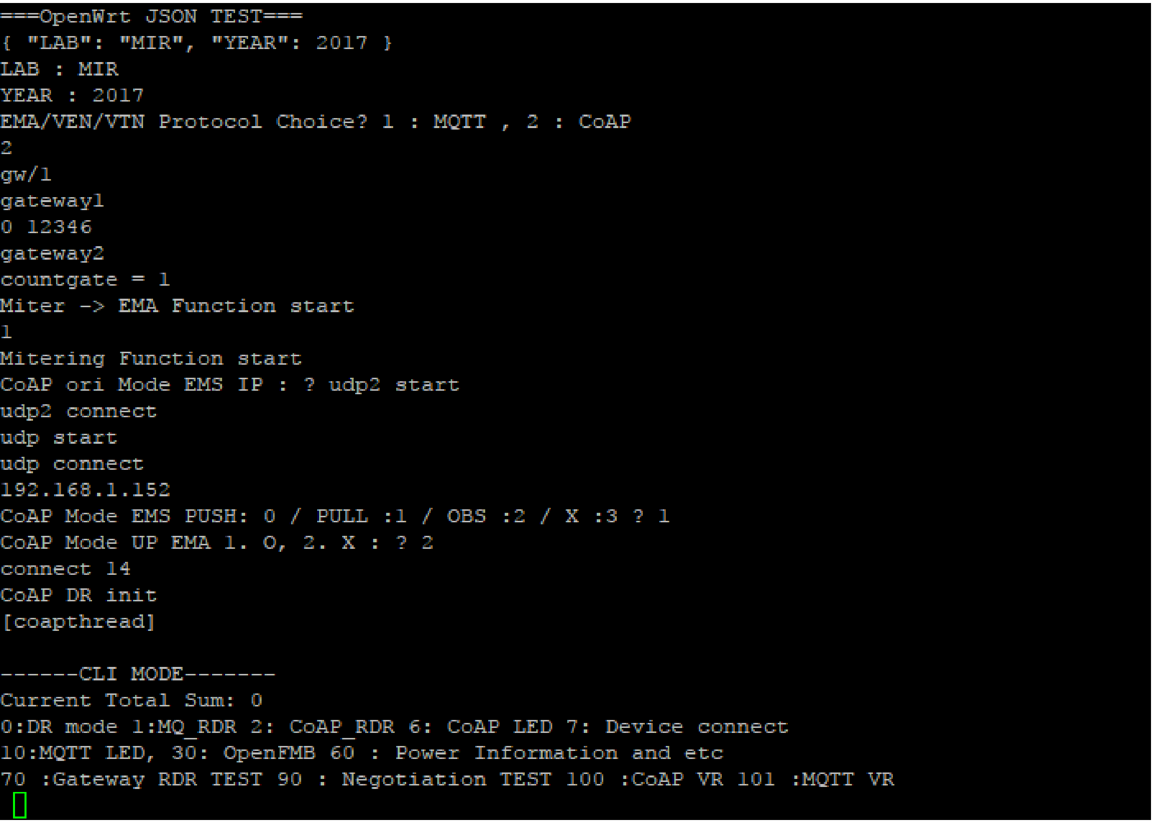
	EMA Name	Protocol	Resource	Threshold
□	MIR	protocol	0.0	0.0
□	EMA1	CoAP	0.0	1000.0

EMS – 실행 BASE

- DR을 내리기 위한 TAB

이 곳을 확인, 이후 사용

9. EMS-EMA - CoAP



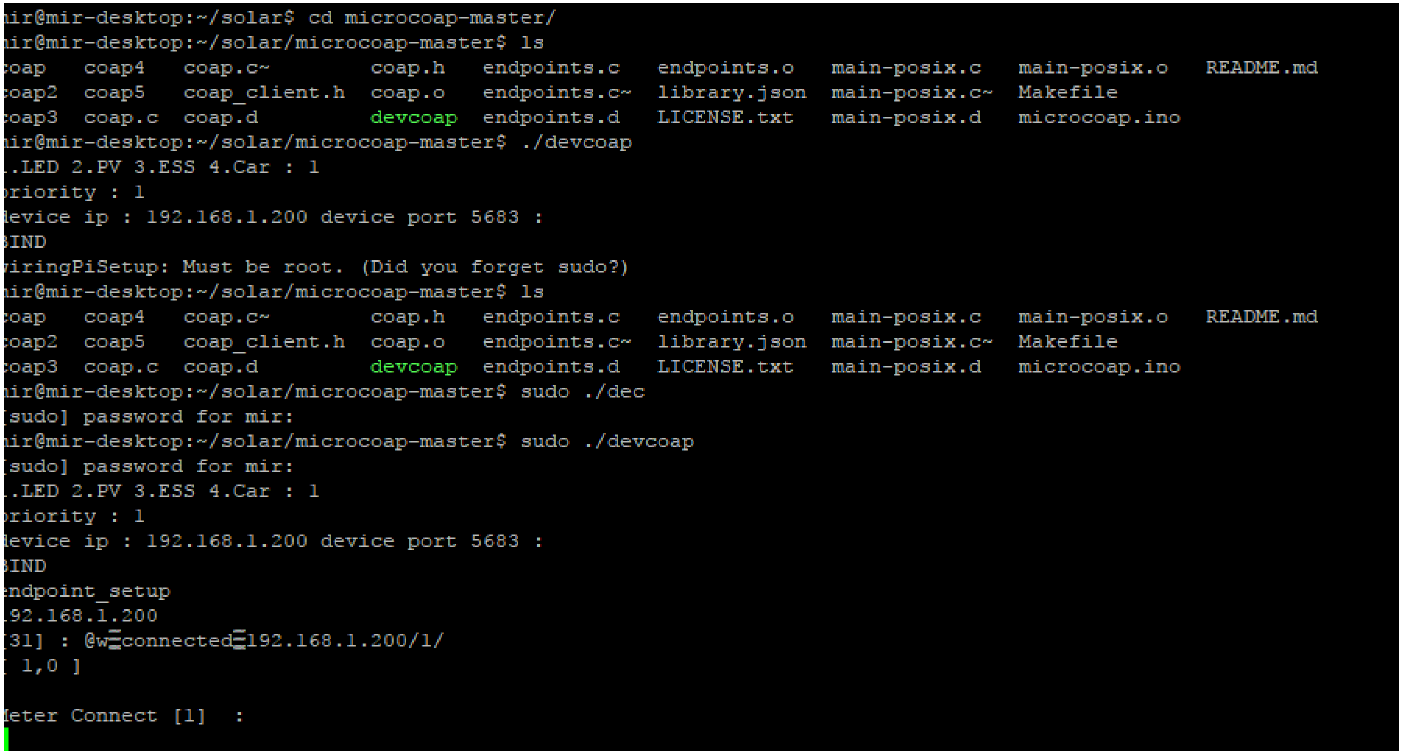
```
===OpenWrt JSON TEST===
{ "LAB": "MIR", "YEAR": 2017 }
LAB : MIR
YEAR : 2017
EMA/VEN/VTN Protocol Choice? 1 : MQTT , 2 : CoAP
2
gw/1
gateway1
0 12346
gateway2
countgate = 1
Miter -> EMA Function start
1
Mitering Function start
CoAP ori Mode EMS IP : ? udp2 start
udp2 connect
udp start
udp connect
192.168.1.152
CoAP Mode EMS PUSH: 0 / PULL :1 / OBS :2 / X :3 ? 1
CoAP Mode UP EMA 1. O, 2. X : ? 2
connect 14
CoAP DR init
[coapthread]

------CLI MODE-------
Current Total Sum: 0
0:DR mode 1:MQ_RDR 2: CoAP_RDR 6: CoAP LED 7: Device connect
10:MQTT LED, 30: OpenFMB 60 : Power Information and etc
70 :Gateway RDR TEST 90 : Negotiation TEST 100 :CoAP VR 101 :MQTT VR
```

Putty를 통하여
OPENWRT 접속 후에 EMA 실행
Protocol CoAP 선택

Broker IP 입력하여
Broker에 접속한다.

EMS의 DR 모드 선택
(PUSH or POLL)
CoAP Mode UP EMA → X 선택

9. EMS-EMA - CoAP



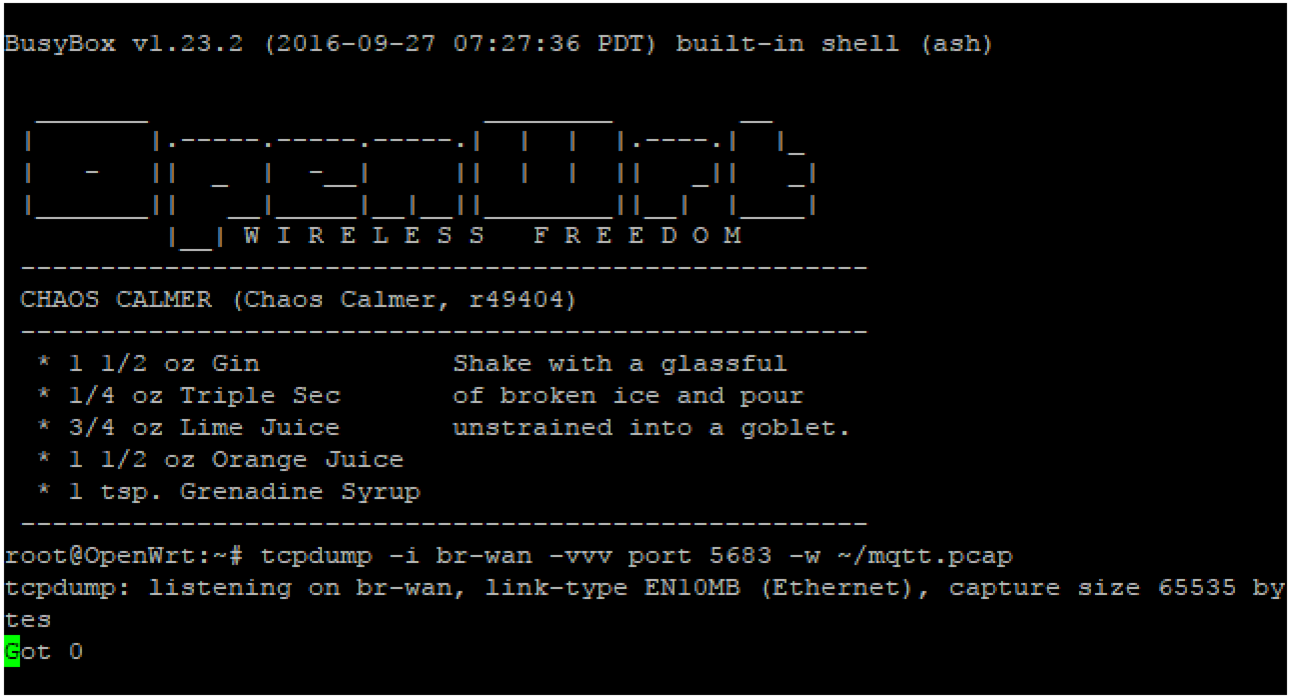
```
ir@mir-desktop:~/solar$ cd microcoap-master/
ir@mir-desktop:~/solar/microcoap-master$ ls
oap    coap4   coap.c~        coap.h    endpoints.c   endpoints.o   main-posix.c   main-posix.o   README.md
oap2   coap5   coap_client.h  coap.o    endpoints.c~  library.json  main-posix.c~  Makefile
oap3   coap.c  coap.d         devcoap   endpoints.d   LICENSE.txt   main-posix.d   microcoap.ino
ir@mir-desktop:~/solar/microcoap-master$ ./devcoap
.LED 2.PV 3.ESS 4.Car : 1
riority : 1
evice ip : 192.168.1.200 device port 5683 :
IND
iringPiSetup: Must be root. (Did you forget sudo?)
ir@mir-desktop:~/solar/microcoap-master$ ls
oap    coap4   coap.c~        coap.h    endpoints.c   endpoints.o   main-posix.c   main-posix.o   README.md
oap2   coap5   coap_client.h  coap.o    endpoints.c~  library.json  main-posix.c~  Makefile
oap3   coap.c  coap.d         devcoap   endpoints.d   LICENSE.txt   main-posix.d   microcoap.ino
ir@mir-desktop:~/solar/microcoap-master$ sudo ./dec
sudo] password for mir:
ir@mir-desktop:~/solar/microcoap-master$ sudo ./devcoap
sudo] password for mir:
.LED 2.PV 3.ESS 4.Car : 1
riority : 1
evice ip : 192.168.1.200 device port 5683 :
IND
ndpoint_setup
92.168.1.200
31] : @w=connected=192.168.1.200/1/
 1,0 ]

eter Connect [1]  :
```

Putty를 통하여 Device에 접속

해당 기기는 LED 이므로 LED를 선택한다.

이후 해당 putty를 통하여 기기를 직접 control 할 수 있다.

9. EMS-EMA - CoAP



```
BusyBox v1.23.2 (2016-09-27 07:27:36 PDT) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 CHAOS CALMER (Chaos Calmer, r49404)
 -----------------------------------------------------
  * 1 1/2 oz Gin            Shake with a glassful
  * 1/4 oz Triple Sec       of broken ice and pour
  * 3/4 oz Lime Juice       unstrained into a goblet.
  * 1 1/2 oz Orange Juice
  * 1 tsp. Grenadine Syrup
 -----------------------------------------------------
root@OpenWrt:~# tcpdump -i br-wan -vvv port 5683 -w ~/mqtt.pcap
tcpdump: listening on br-wan, link-type EN10MB (Ethernet), capture size 65535 by
tes
Got 0
```

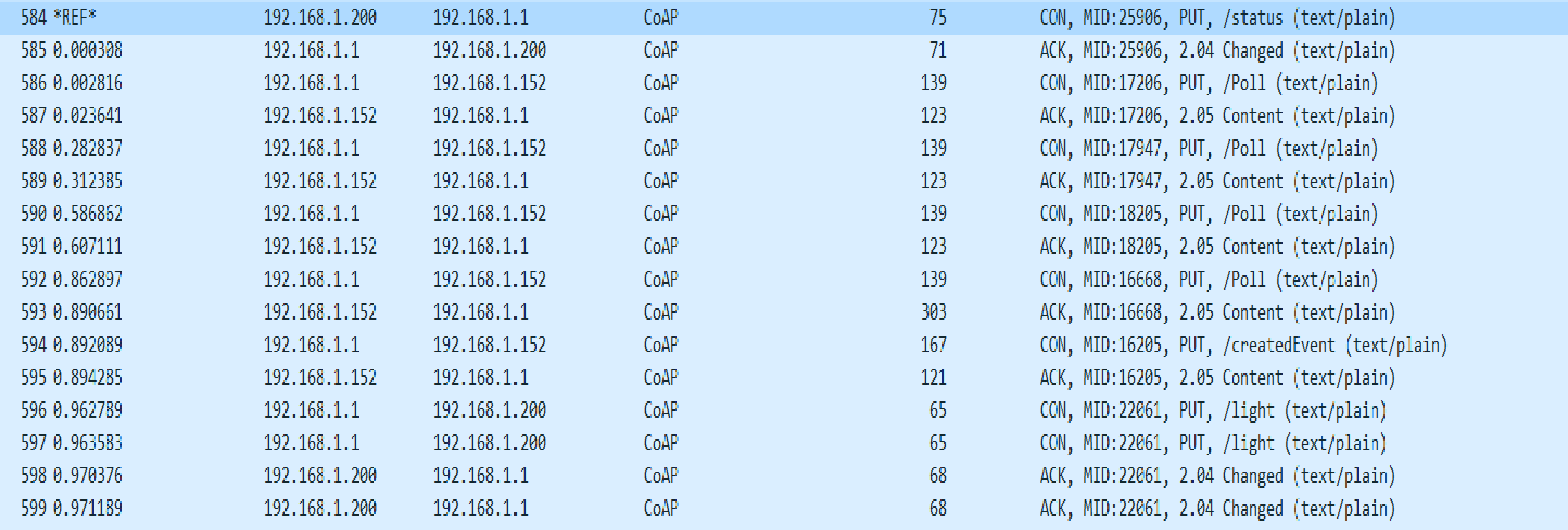
MQTT EMA 가 실행되는 OPENWRT에 추가 접속.

Tcp dump를 찍는다.

COAP 의 tcp dump 는 **5683**

종료 후 winscp를 통하여 pcap 파일을 가져와 wireshark 파일을 통하여 분석한다.

9. Device Connection

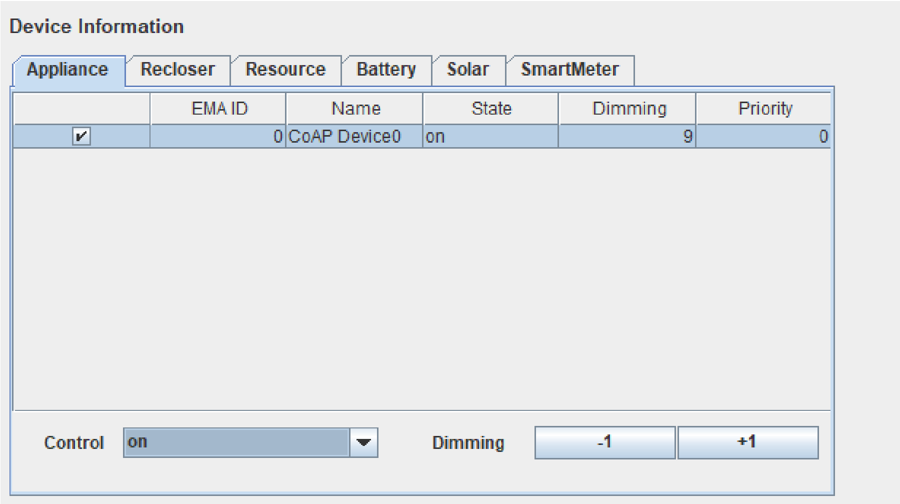


```
584 *REF*        192.168.1.200   192.168.1.1     CoAP    75    CON, MID:25906, PUT, /status (text/plain)
585 0.000308     192.168.1.1     192.168.1.200   CoAP    71    ACK, MID:25906, 2.04 Changed (text/plain)
586 0.002816     192.168.1.1     192.168.1.152   CoAP    139   CON, MID:17206, PUT, /Poll (text/plain)
587 0.023641     192.168.1.152   192.168.1.1     CoAP    123   ACK, MID:17206, 2.05 Content (text/plain)
588 0.282837     192.168.1.1     192.168.1.152   CoAP    139   CON, MID:17947, PUT, /Poll (text/plain)
589 0.312385     192.168.1.152   192.168.1.1     CoAP    123   ACK, MID:17947, 2.05 Content (text/plain)
590 0.586862     192.168.1.1     192.168.1.152   CoAP    139   CON, MID:18205, PUT, /Poll (text/plain)
591 0.607111     192.168.1.152   192.168.1.1     CoAP    123   ACK, MID:18205, 2.05 Content (text/plain)
592 0.862897     192.168.1.1     192.168.1.152   CoAP    139   CON, MID:16668, PUT, /Poll (text/plain)
593 0.890661     192.168.1.152   192.168.1.1     CoAP    303   ACK, MID:16668, 2.05 Content (text/plain)
594 0.892089     192.168.1.1     192.168.1.152   CoAP    167   CON, MID:16205, PUT, /createdEvent (text/plain)
595 0.894285     192.168.1.152   192.168.1.1     CoAP    121   ACK, MID:16205, 2.05 Content (text/plain)
596 0.962789     192.168.1.1     192.168.1.200   CoAP    65    CON, MID:22061, PUT, /light (text/plain)
597 0.963583     192.168.1.1     192.168.1.200   CoAP    65    CON, MID:22061, PUT, /light (text/plain)
598 0.970376     192.168.1.200   192.168.1.1     CoAP    68    ACK, MID:22061, 2.04 Changed (text/plain)
599 0.971189     192.168.1.200   192.168.1.1     CoAP    68    ACK, MID:22061, 2.04 Changed (text/plain)
```

Device Connect to EMA (COAP Connection)
EMA ACK to Device
EMA sends CON to EMS
EMS ACK to EMA

EMS & EMA continuing connection and send CON&ACK each other

9. EMS-EMA - CoAP

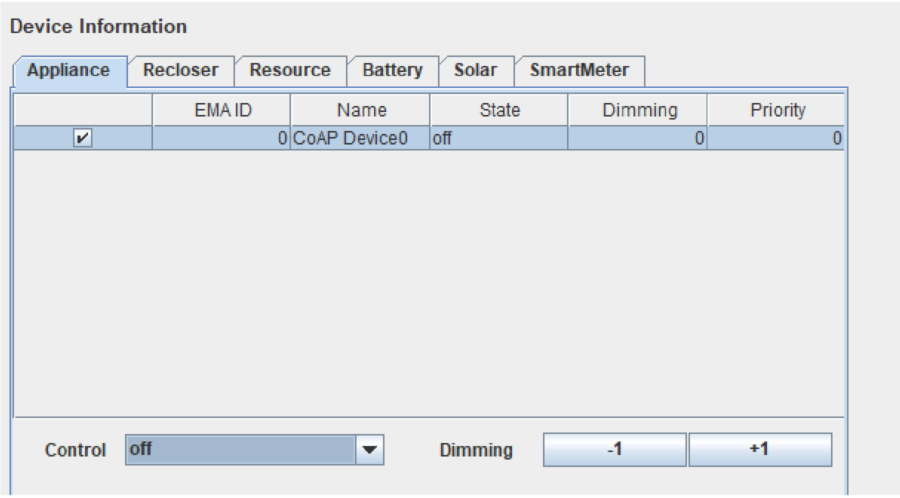


Device Information

☑	EMA ID	Name	State	Dimming	Priority
☑	0	CoAP Device0	on	9	0

Control: on ▾ Dimming

Device가 EMA에 접속 시, EMS에 등록된다.



Device Information

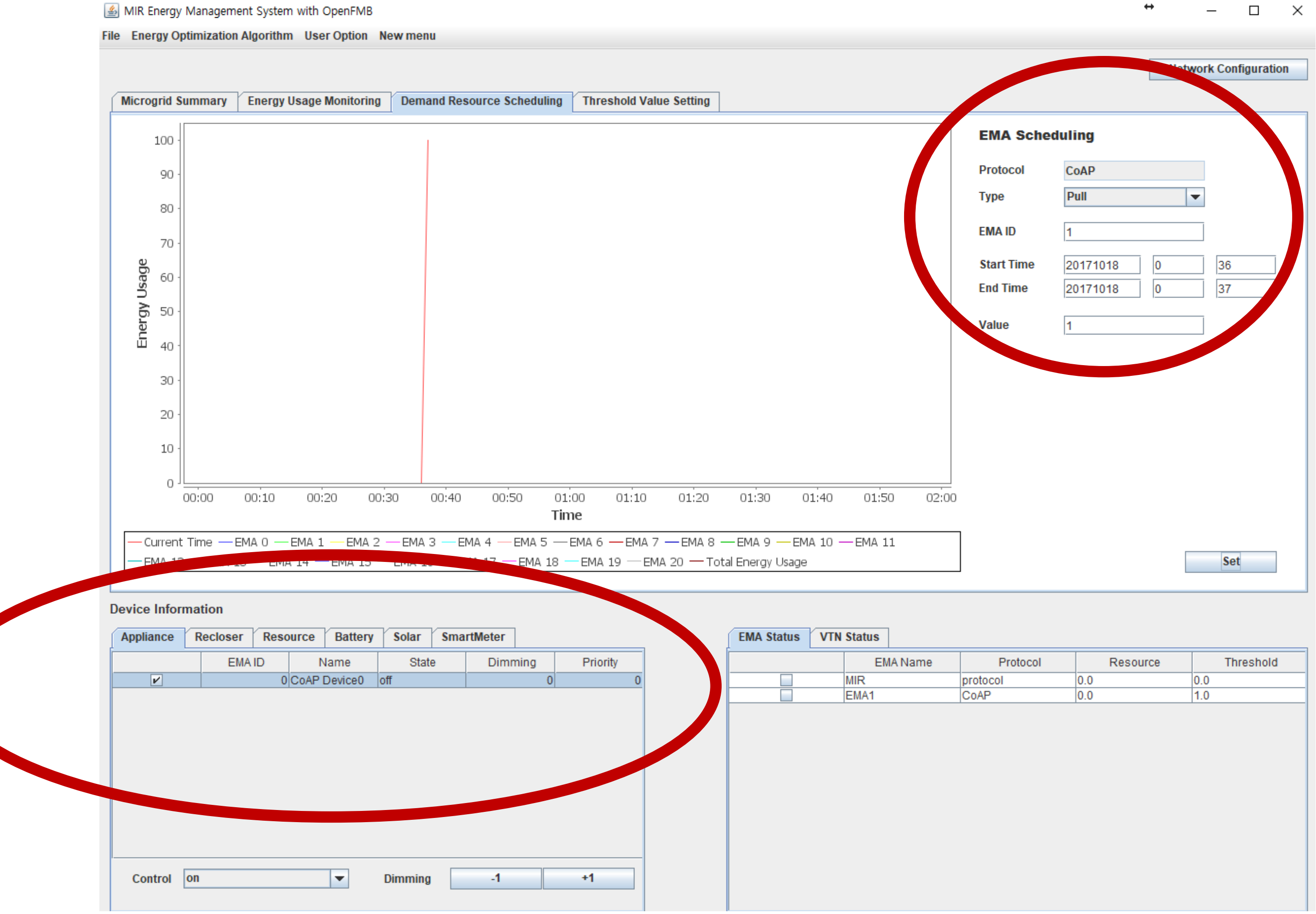
☑	EMA ID	Name	State	Dimming	Priority
☑	0	CoAP Device0	off	0	0

Control: off ▾ Dimming

EMS를 통하여 등록된 기기를 제어할 수 있ㄷ

On, Off 조작 가능

9. EMS-EMA - CoAP



EMA Scheduling

Protocol: CoAP

Type: Pull

EMA ID: 1

Start Time: 20171018 | 0 | 36

End Time: 20171018 | 0 | 37

Value: 1

	EMA ID	Name	State	Dimming	Priority
☑	0	CoAP Device0	off	0	0

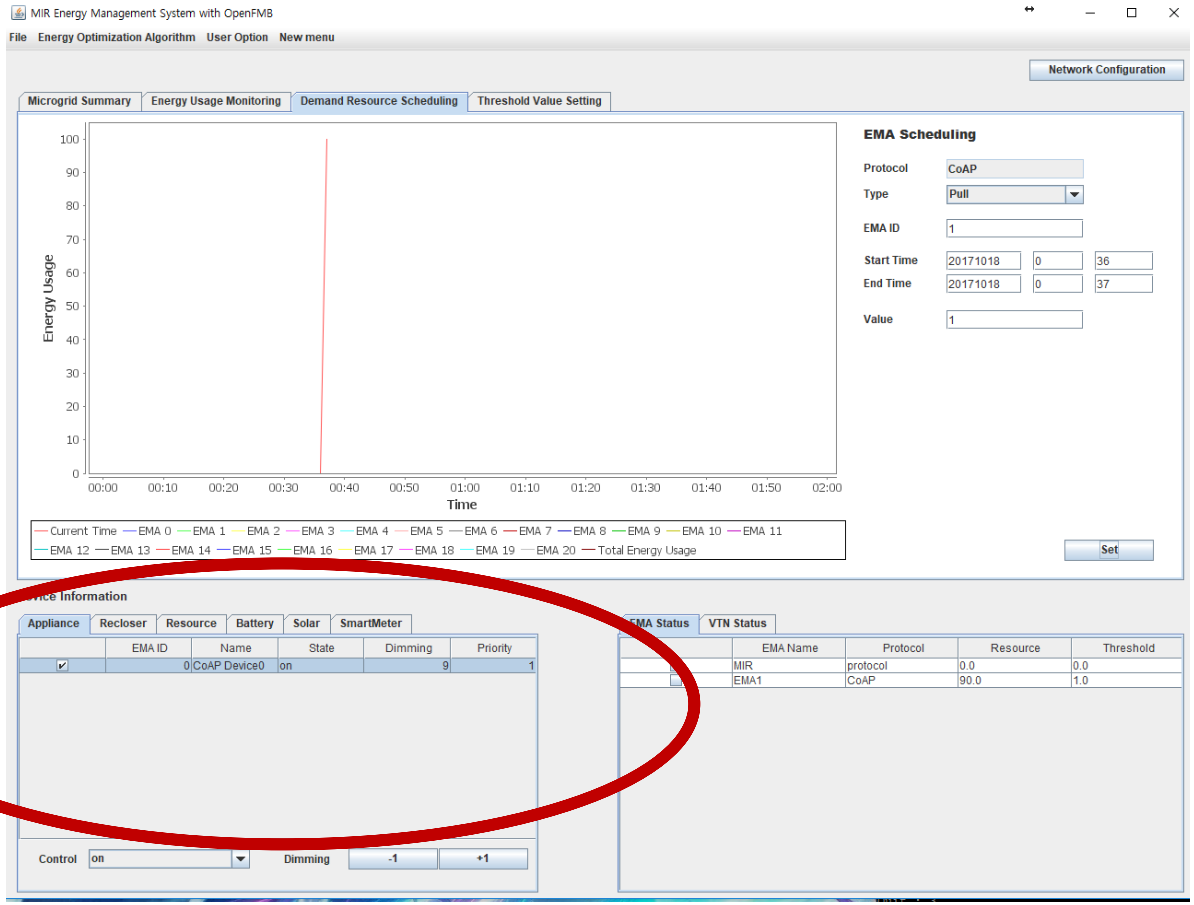
Control: on

	EMA Name	Protocol	Resource	Threshold
□	MIR	protocol	0.0	0.0
□	EMA1	CoAP	0.0	1.0

CoAP Pull 방식
EMA에 직접 DR을
내릴 수 있다.
Value 값 제한을 두고,
해당 값이 넘어갈 경우
기기의 전원이 내려간다

EMS가 DR을 내리자, 이에 따라 EMS에 등록된 EMA의 device의 전원이 내려간 것을 확인 할 수 있다.

9. EMS-EMA - CoAP



EMA Scheduling

Protocol: CoAP

Type: Pull

EMA ID: 1

Start Time: 20171018 | 0 | 36

End Time: 20171018 | 0 | 37

Value: 1

vice Information

	EMA ID	Name	State	Dimming	Priority
☑	0	CoAP Device0	on	9	1

Control: on　Dimming

	EMA Name	Protocol	Resource	Threshold
	MIR	protocol	0.0	0.0
□	EMA1	CoAP	90.0	1.0

기기로부터 RDR 즉 다시 전원을 올려달라는 요청이 올 시에 전원이 다시 올라 간 것을 확인 할 수 있다.