TTA Standard

정보통신단체표준(국문표준)
TTAK.KO-10.1121-part1 제정일: 2018 년 12 월 19 일

에너지 전력 분야 사물인터넷 (e-IoT) - 제1부: 시스템 규격

Internet of Things in Electricity and Energy Domain(e-IoT) - part 1: System Specifications

TTA 한국정보통신기술협회
Telecommunications Technology Association

표준초안 검토 위원회	사물인터넷 네트워킹 프로젝트그룹(SPG12)				
표준안 심의 위원회	사물인터넷 특별기술위원회(STC1)				
	성명	**소 속**	**직위**	**위원회 및 직위**	**표준번호**
표준(과제) 제안	김영현	전력연구원	차장	-	TTAK.KO-10.1121-part1
표준 초안 작성자	오승훈	ETRI	책임	-	
	이형옥	ETRI	선임		
	고석갑	ETRI	선임		
	박명혜	전력연구원	부장		
	이병탁	ETRI	실장		
사무국 담당	이종화	TTA		-	

본 문서에 대한 저작권은 TTA에 있으며, TTA와 사전 협의 없이 이 문서의 전체 또는 일부를 상업적 목적으로 복제 또는 배포해서는 안 됩니다.

본 표준 발간 이전에 접수된 지식재산권 확약서 정보는 본 표준의 '부록(지식재산권 확약서 정보)'에 명시하고 있으며, 이후 접수된 지식재산권 확약서는 TTA 웹사이트에서 확인할 수 있습니다.
본 표준과 관련하여 접수된 확약서 외의 지식재산권이 존재할 수 있습니다.

발행인 : 한국정보통신기술협회 회장
발행처 : 한국정보통신기술협회
13591, 경기도 성남시 분당구 분당로 47
Tel : 031-724-0114, Fax : 031-724-0109
발행일 : 2018.12

서 문

1 표준의 목적

이 표준은 에너지 전력 분야 사물인터넷(e-IoT) 표준으로 특히 전력 시설에 구축될 e-IoT 디바이스(센서 및 액추에이터)를 연결하여 e-IoT 디바이스에서 측정된 정보를 수집하고, 제어할 수 있으며 수집된 측정 데이터를 e-IoT 플랫폼에 전달하는 기능을 제공하는 사물인터넷 서비스 규격을 정의한다.

2 주요 내용 요약

이 표준은 IETF CoAP, OMA LWM2M, oneM2M 표준을 근간으로 e-IoT 서비스를 지원할 수 있는 규격을 정의한다. e-IoT 디바이스와 e-IoT 게이트웨이, e-IoT 플랫폼 사이의 메시지 절차를 정의하였고 이를 위한 장치 및 리소스 식별자를 정의하였다.

3 인용 표준과의 비교

3.1 인용 표준과의 관련성

해당 사항 없음

3.2 인용 표준과 본 표준의 비교표

해당 사항 없음

Preface

1 Purpose

The standard is to define the specification for Internet of Things services in energy and electricity fields (e-IoT). Using it, e-IoT devices are connected to e-IoT gateway, which is able to collect the measured data from them and control them. The collected data is transferred to e-IoT platform.

2 Summary

The standard redefines the specification to support IoT services of the energy electric field based on the IETF CoAP, OMA LWM2M, and oneM2M standards. It defines the message procedure between e-IoT device, e-IoT gateway and e-IoT platform, and defines device and resource identifier for it.

3 Relationship to Reference Standards

- None.

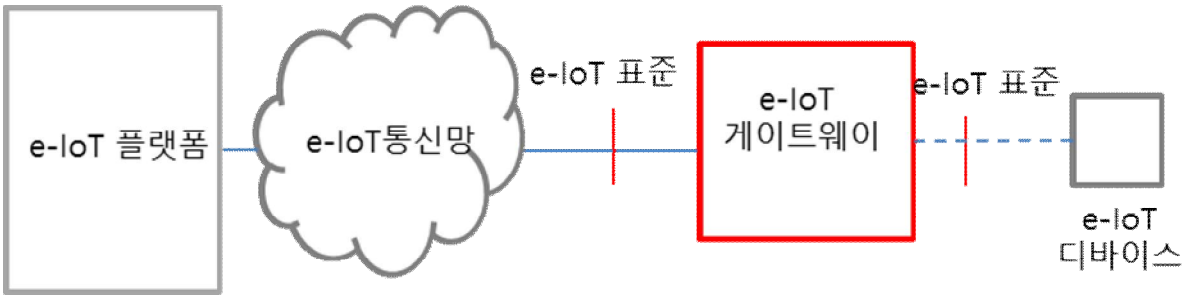
목 차

에너지 전력 분야 사물인터넷(e-IoT) - 제1부: 시스템 규격

(Internet of Things in Electricity and Energy Domain(e-IoT) - Part 1: System Specifications)

1 적용 범위

1.1 시스템 구성도

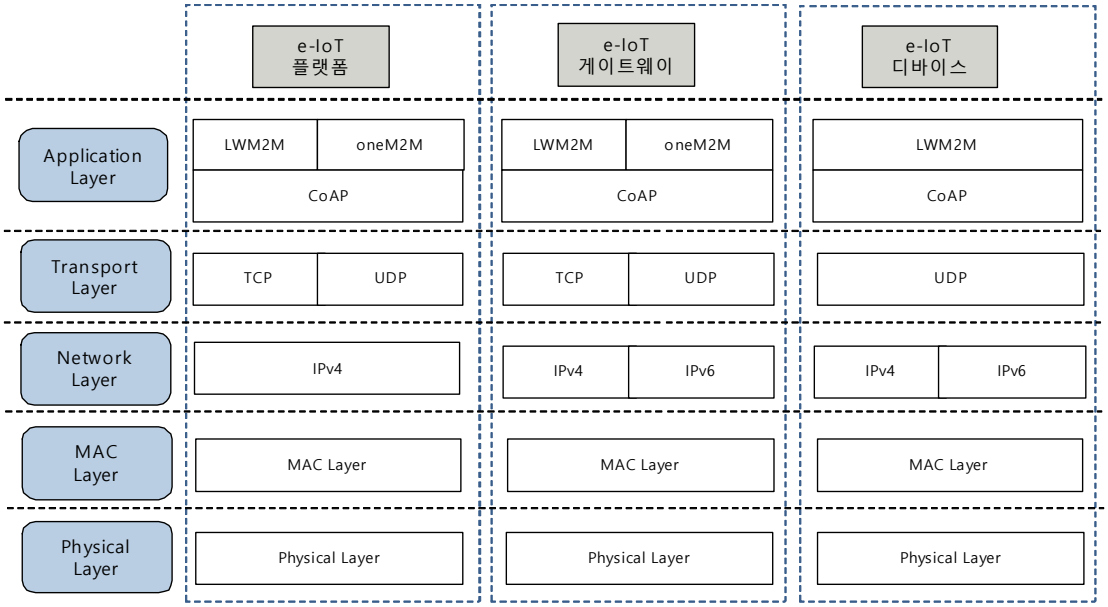


(그림 1-1) e-IoT 시스템 구성도

e-IoT 시스템은 종단의 전력 에너지 시설에 부착될 e-IoT 디바이스와 이들을 e-IoT 플랫폼에 연결해 주는 e-IoT 게이트웨이, 최종적으로 디바이스들의 모든 정보를 수집하고 디바이스들을 관리하며, 다양한 e-IoT 서비스를 가능하게 해 주는 e-IoT 플랫폼으로 구성된다.

1.2 시스템 계층 모델

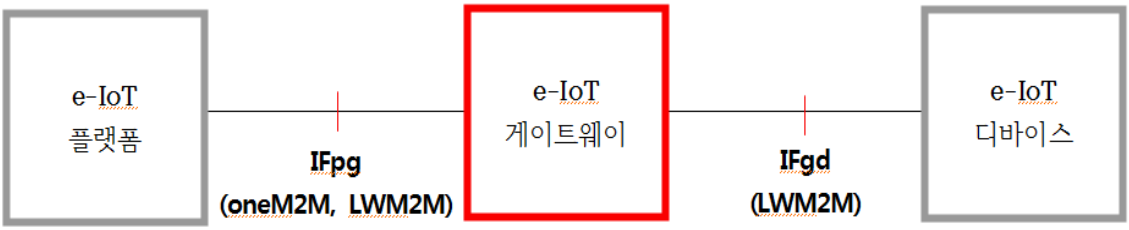
e-IoT 시스템 계층 모델은 OSI 계층 구조를 기본으로 한다.



(그림 1-2) e-IoT 시스템의 프로토콜 계층 모델

1.3 규격 범위

본 표준은 e-IoT 디바이스와 연결되어 정보를 수집하고 e-IoT 플랫폼에 표준화된 형태의 데이터를 전달해 주며 플랫폼의 제어 메시지를 전달 받아 e-IoT 디바이스를 표준화된 형태로 제어하는 e-IoT 게이트웨이 표준화 모델과 연동 규격을 정의한다.
e-IoT 디바이스와 e-IoT 게이트웨이 간 인터페이스(IFgd)와 e-IoT 게이트웨이와 e-IoT 플랫폼간 인터페이스(IFpg) 통신 프로토콜 메시지 규격에 대해 정의한다. e-IoT 디바이스는 oneM2M 규격의 복잡성과 하드웨어 제약을 고려할 때 CoAP 기반의 LWM2M 정보모델 표준을 따르는 장치로 정의한다.



(그림 1-3) 프로토콜 측면에서 본 적용 범위

본 표준은 oneM2M, OMA LWM2M, IETF CoAP 표준을 근간으로 e-IoT 서비스를 실현하기 위한 서비스 절차를 정의하였고 게이트웨이와 디바이스 장치를 oneM2M과 LWM2M 표준의 정보 모델로 정의하였다. 장치 식별을 위한 OID기반의 체계를 정의하였으며, 더불어 리소스들을 프로파일화 하여 OID기반의 식별자로 부여하는 체계를 정의하였다.
e-IoT 서비스 구현에 필요한 아래와 같은 추가 사항들을 정의하였다.

- POST를 이용한 LWM2M의 주기적인 보고 방식 정의
- Sleep 기법을 이용한 저전력 디바이스 제어를 위한 Sleep node 제어 절차 정의

- 리소스 프로파일을 기술하기 위한 LWM2M JSON Type 추가 정의
- CoAP Confirmable/Non-confirmable Notification를 설정하기 위한 LWM2M Attribute 추가 정의

1.4 인터페이스 정의

1.4.1 IFpg

e-IoT 통신망을 통하여 e-IoT 플랫폼과 e-IoT 게이트웨이가 통신하기 위한 인터페이스이다. e-IoT 게이트웨이는 유선 또는 무선 통신 인터페이스를 장착해야 하고 유무선 물리적 계층과 무관하게 IP 기반 인터넷 연결성이 보장되어야 한다.

1.4.2 IFgd

e-IoT 게이트웨이와 e-IoT 디바이스가 통신하기 위한 인터페이스이다. e-IoT 게이트웨이는 유선, 또는 무선 통신 인터페이스를 장착해야 하고 유무선 물리적 계층과 무관하게 IP 기반 인터넷 연결성이 보장되어야 한다.

2 인용 표준

해당 사항 없음

3 용어 정의

3.1 e-IoT
에너지 전력 분야 사물인터넷 기술

3.2 e-IoT 디바이스
에너지 전력 분야의 현장에 설치된 e-IoT 종단 단말을 의미하며 대부분 센서, 또는 Actuator 장치를 의미

3.3 e-IoT 게이트웨이
e-IoT 디바이스들이 연결되어 있으며, e-IoT 표준에 의해 디바이스로부터 정보를 수집하고e-IoT 플랫폼에 전달해 주는 역할을 담당하는 장치

3.4 e-IoT 플랫폼
e-IoT 디바이스의 정보를 포함하여 측정되는 모든 정보를 저장·가공하며, 역으로 원격에서 e-IoT 디바이스를 제어할 수 있는 종합적인 에너지 사물인터넷 서버군

4 약어

ASN	Application Service Node
ADN	Application Dedicated Node
CSE	Common Service Entity
e-IoT	Energy(&Electricity) IoT
IFpg	Interface between Platform and Gateway
IFgd	Interface between Gateway and Device
DTLS	Datagram Transport Layer Security(Protocol)

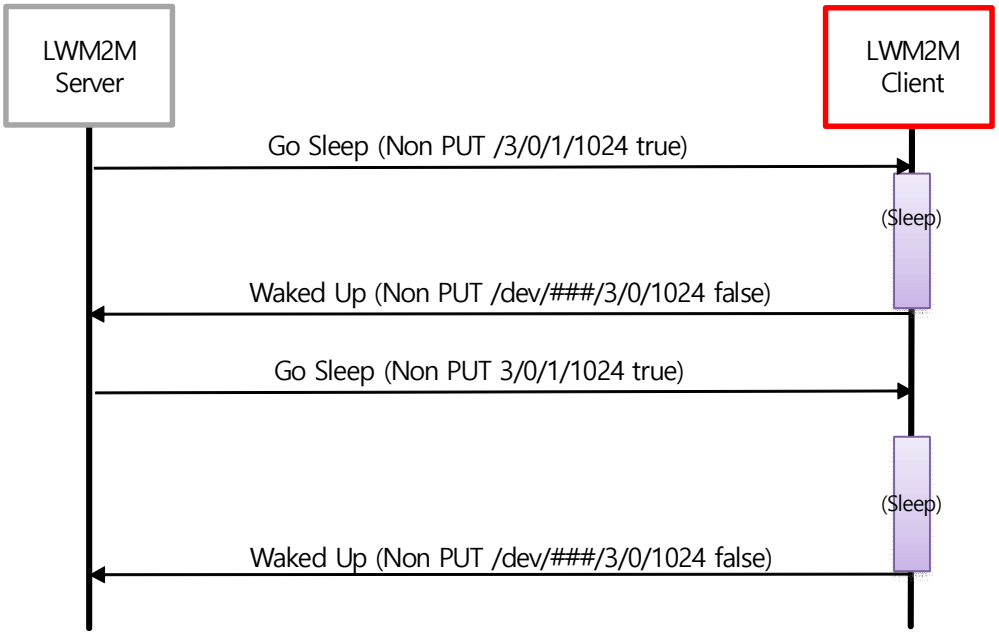
5 e-IoT LWM2M 요소기술

5.1 개요

e-IoT 서비스 제공을 위해 OMA LWM2M 기본 규격에 추가된 요소 기술을 정의한다.

5.2 Sleep Node Control

LWM2M 서버는 'Go Sleep', 'Waked Up' 기능을 통해서 클라이언트의 슬립(Sleep)를 제어할 수 있다. Sleep 제어는 Device(3) Object의 Sleep(1024) 리소스를 이용한다.
'Go Sleep'은 서버 입장에서 클라이언트에게 전송할 메시지가 없는 경우 슬립을 허가하는 의미이며, "/3/0/1024" 리소스를 이용해서 값을 true로 수정 요청하는 메시지이다. 이 메시지를 수신한 클라이언트는 꼭 슬립하는 것은 아니다. 클라이언트 자체의 슬립 메커니즘에 의해서 슬립할 수 있다.
'Waked Up'은 클라이언트가 슬립에서 깨어나면 서버에게 알리는 것이며, 서버에 등록된 "/dev/###/3/0/1024" 리소스를 이용해서 값을 false로 수정 요청하는 메시지이다.
이 메시지를 수신한 서버는 큐(Queue) 모드로 저장되어 있는 메시지를 해당 클라이언트에 전송한다.



(그림 5-1) LWM2M 디바이스 슬립(Sleep) 제어 기능

5.3 CoAP 메소드 매핑

LWM2M 동작과 CoAP 메소드 매핑은 다음과 같다. 본 규격에서 추가 정의된 'Go Sleep'과 'Waked Up' LWM2M 동작을 아래 테이블과 같이 CoAP 메소드에 매핑한다.

<표 5-1> LWM2M 메소드 매핑

LWM2M 동작 파라미터	CoAP 메소드
Register	POST
Update	POST
De-register	DELETE
Read	GET
Discover	GET
Write	PUT POST
Write Attribute	PUT
Execute	POST
Create	POST
Delete	DELETE
Observer	GET
Notify	Asynchronous Response
Cancel Observation	GET
Go Sleep	PUT
Waked Up	PUT

5.4 정보 모델 속성

본 표준에서는 OMA에서 정의한 Object(0~1023)와 제3의 표준 단체(IPSO Alliance)에서 정의한 오브젝트(Object)를 기반으로 정보 모델을 정의한다. 예를 들어 에너지 IoT용으로 전력회사 내부에 사용할 오브젝트들은 20000대 번호에 할당한다.
LWM2M 기반의 정보 모델, 즉 오브젝트와 오브젝트 인스턴스(Object Instance), 리소스(Resource)에 다음과 같은 속성값을 갖는다.
다음 속성값은 서버가 데이터 관리에 도움이 될 수 있는 메타 정보를 제공한다. 아래 차원 속성값은 해당 오브젝트 인스턴스에 구현된 리소스 인스턴스의 수를 의미한다. 아래 데이터 보고/통보와 관련된 속성과는 달리, 다수의 리소스 인스턴스가 존재하는 경우는 필수적으로 클라이언트 측에서 제공되어야 한다.

<표 5-2> 필수 LWM2M 속성

속성 이름	Core Link Param	Attachment	Assignation Level	Type
Dimension	dim	Resource	Resource Level	Integer [0:255]

다음 속성값들은 데이터 보고/통보(Notification)와 관련되어 있다. dim 속성과 달리 선택적으로 제공된다.

<표 5-3> 선택적 LWM2M 속성

속성 이름	Core Link Param	Attachment	Assignation Level	Type
최소 주기	pmin	Resource	Resource Level Object Instance Level Object Level	Integer
최대 주기	pmax	Resource	Resource Level Object Instance Level Object Level	Integer
Greater Than	gt	Resource	Resource Level	Float
Less Than	lt	Resource	Resource Level	Float
Step	st	Resource	Resource Level	Float
Notification CoAP Message Type	ntype	Resource	Resource Level Object Instance Level Object Level	Boolean

특히 본 표준에서는 Notification CoAP Message Type을 추가 정의한다. 이 값이 True인 경우 Confirmable CoAP 메시지, False인 경우 Non-confirmable CoAP 메시지로 보고 메시지를 전송한다. 게이트웨이는 주기적 보고의 보고 메시지를 전송함에 있어 해당 Resource의 “ntype” 속성값을 확인한 후에 Confirmable/non-confirmable 메시지를 선택하여 전송한다.

5.5 데이터 형식

Resource 정보 전달 데이터 형식은 Plain-text와 link-format, octet-stream, JSON, TLV를 지원한다. 각 형식별 할당된 번호는 다음과 같다. 아래는 ‘Data Format(IANA Media Type): Numeric Content-Formats[CoAP]’ 형식으로 표현된다.

• Plain text(text/plain; charset=utf-8): 0

- Core Link Param(application/link-format): 40
- Opaque(application/octet-stream): 42(RFC 2045, RFC 2046)
- JSON(application/vnd.oma.lwm2m+json: 11543
- TLV(application/vnd.oma.lwm2m+tlv: 11542

LWM2M에서 정의한 JSON 변수는 다음과 같다. 특히 본 표준에서는 OBJ-RSC(Object Resource) 프로파일에서 사용할 'Extra Type' 변수를 아래와 같이 추가 정의한다.

<표 5-4> LWM2M JSON 변수 정의

속성	JSON 변수		설명
Base Name	bn		모든 Resource의 루트가 되는 위치
Base Time	bt		기본이 되는 현재 시간으로, Time 변수가 bt 값을 기본으로 상대 값을 갖는다.
Resource Array	e		Resource 리스트
	Array Parameters		
	Name	n	Resource instance의 경로 이름
	Time	t	초 단위, bt를 기준으로 상대적인 시간 값
	Float Value	v	Int, float, time인 경우
	Boolean Value	bv	JSON Boolean으로0, 1, true, false 값을 가질 수 있다.
	ObjectLink Value	ov	Resource Data type이 Objlnk인 경우
	String Value	sv	Boolean, Int, float, time 이외의 경우
	Extra Type	ex	Executable, Opaque type의 Resource를 표현, 게이트웨이와 디바이스의 OBJ-RSC 프로파일에만 사용됨, 값은 0으로 표현한다.

Base Name과 Base Time 속성 값은 선택적으로 포함할 수 있으나, Resource Array는 필수로 포함하여야 한다. 즉, 배열 형태로 값을 표현해야하기 때문에 JSON 변수로 'e' 파라미터를 꼭 포함해서 작성되어야 한다. 아래 예제와 같이 Name 변수와 함께 Float Value, Boolean Value, Object Link Value, String Value 중 하나를 가져야 한다. Time 변수는 Base Time을 기준으로 상대적인 시간을 표현할 때 넣을 수 있으나 필수적인 것은 아니다. 아래 예제는 게이트웨이의 OBJ-RSC 프로파일 예시이다. '/3/0/4' reboot Resource와 '/5/0/0' 펌웨어 패키지 Resource를 아래와 같이 "ex" Type으로 표현할 수 있다. 이때 값은 0으로 설정한다.

```
{“e”:[
      {"n":"1/0/0","v":1},
      {"n":"1/0/1","v":30},

      …
      {"n":"3/0/1","sv":"LWM2M Gateway"},
      {"n":"3/0/2","sv":"000000001"},
        {"n":"3/0/3","sv":"0.3"},
        {"n":"3/0/4","ex":0},

        …
      {"n":"4/0/4/0","sv":""},
        {"n":"5/0/0","ex":0},
      {"n":"5/0/1","sv":“ftp://iot.kepco.co.kr/pkg”},

      …

      {"n":"7/0/8","v":10}]
}
```

(그림 5-2) 게이트웨이의 OBJ-RSC 프로파일 예시

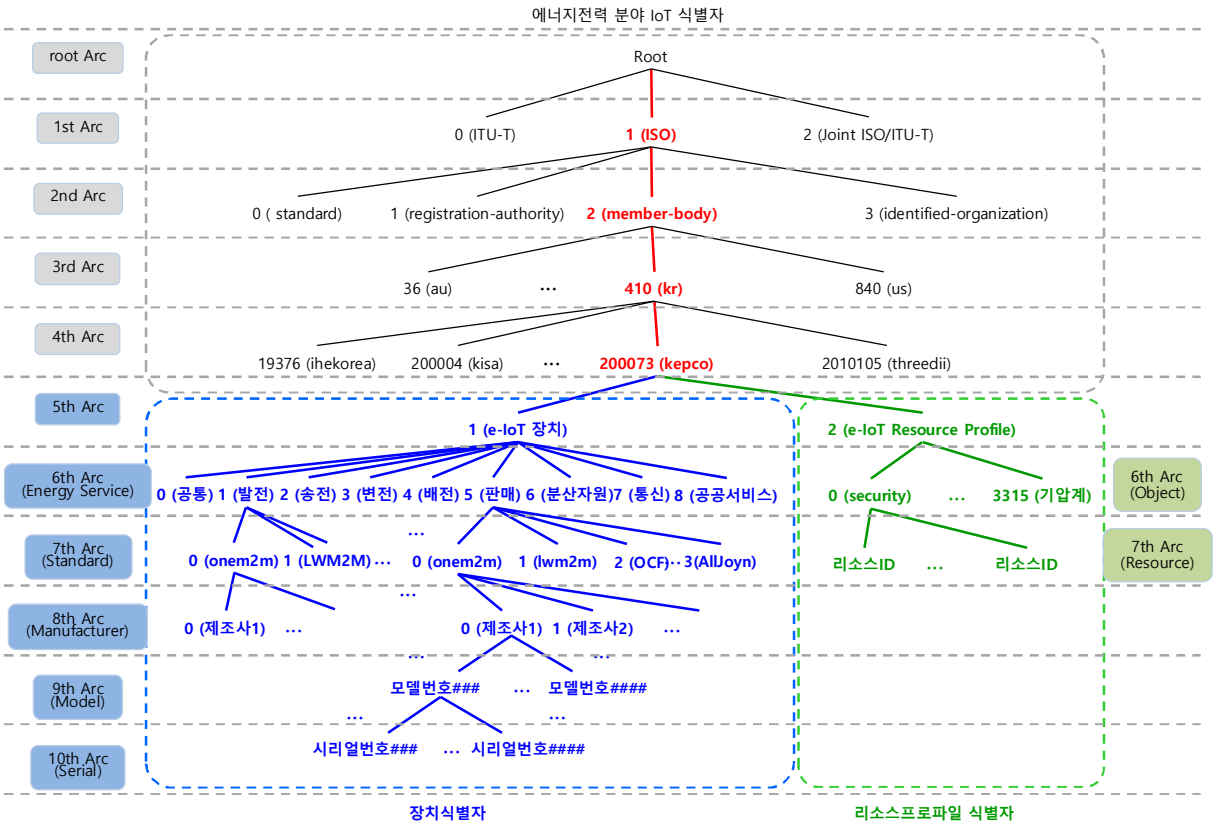
6 식별자

6.1 개요

본 표준에서는 e-IoT 장치를 식별하는 목적으로 유일성을 보장할 수 있는 OID 체계를 따른다.
※ KS X4105, 정보기술 - 개방형 시스템간 상호 접속 - 객체식별자(OID)의 구성과 등록 절차

OID 기반의 식별자는 기관 식별자, IoT 장치 식별자, IoT Resource 프로파일 식별자로 구성된다. 기관 식별자는 장치와 Resource 식별자 체계에서 공통으로 적용되는 상위 기본 식별 체계이고, 장치 식별자는 게이트웨이와 디바이스를 식별하기 위한 목적으로 사용되며 Resource 프로파일 식별자는 해당 기관의 IoT 장치내 사용되는 Resource를 의미적으로 표현하기 위해 사용된다.

예를 들어 대표적인 전력회사인 한국전력의 경우 e-IoT 장치 식별자, e-IoT Resource 프로파일 식별자 체계를 하나의 트리로 표현하며 다음과 같다.

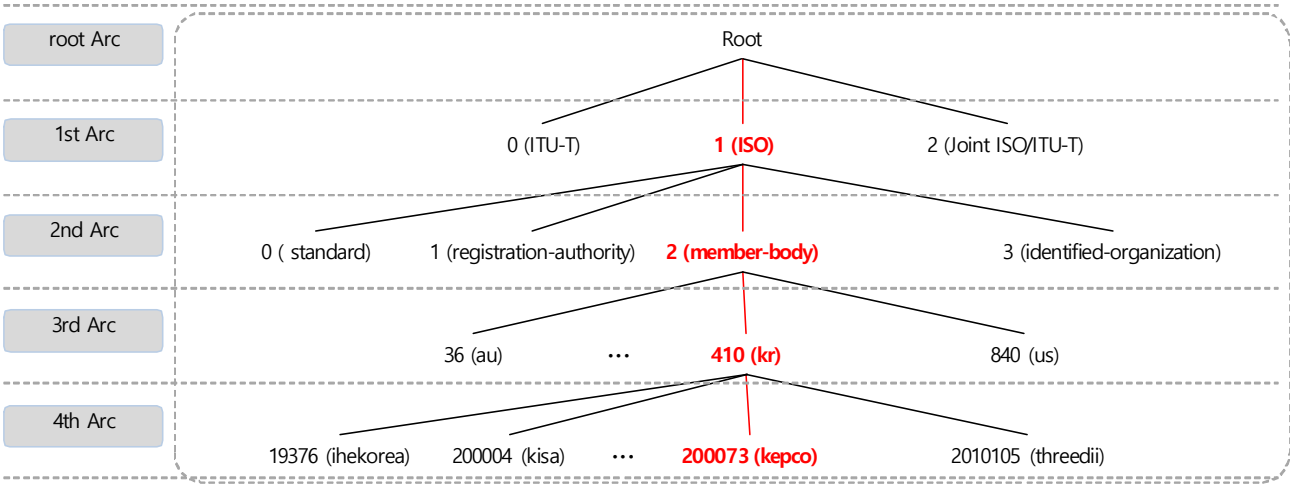


(그림 6-1) OID 기반 e-IoT 장치 식별자 체계 개요 예시(한국전력의 경우)

6.2 기관 식별자

6.2.1 개요

OID 기반의 식별자로 특정 기관을 나타내는 식별자이다. 예를 들어 한국 전력은 아래와 같이 200073 식별자를 부여 받았다.



(그림 6-2) 기관 IoT 식별자 체계 예시(한국전력의 경우, 4th Arc = 200073)

6.2.2 계층 구조

a) 1th Arc
표준 단체를 식별하는 계층

b) 2th Arc
표준 단체 Member-Body

c) 3th Arc
ISO 3166에서 규정한 국가 코드 National-Body
OID Repository 사이트에서 관리하고 있으며, 대한민국은 410.

d) 4th Arc
ISO의 국내 Member-Body 대한민국에서 관리하고 있는 기간에 할당된 번호 계층이다. 이 영역은 현재 국가기술표준원에서 관리하고 있다. 국가 표준 X 4105에 따라 다음과 같이 관리되고 있다.

- 10,000 ~ 99,999 : 미래 활용으로 예약됨
- 100,000 ~ 16,099,999 : 다음 기관에 할당될 영역
- 100,000 ~ 100,999 : 대한민국 국가 기관들 대상.
- 101,000 ~ 147,999 : 지방 정부 기관

- 148,000 ~ 199,999 : 예약된 구간
- 200,000 ~ 16,099,999 : 상위 기관 외 모든 기관

예를 들어 현재 한국전력은 “kepco” 명칭으로 200073 할당.

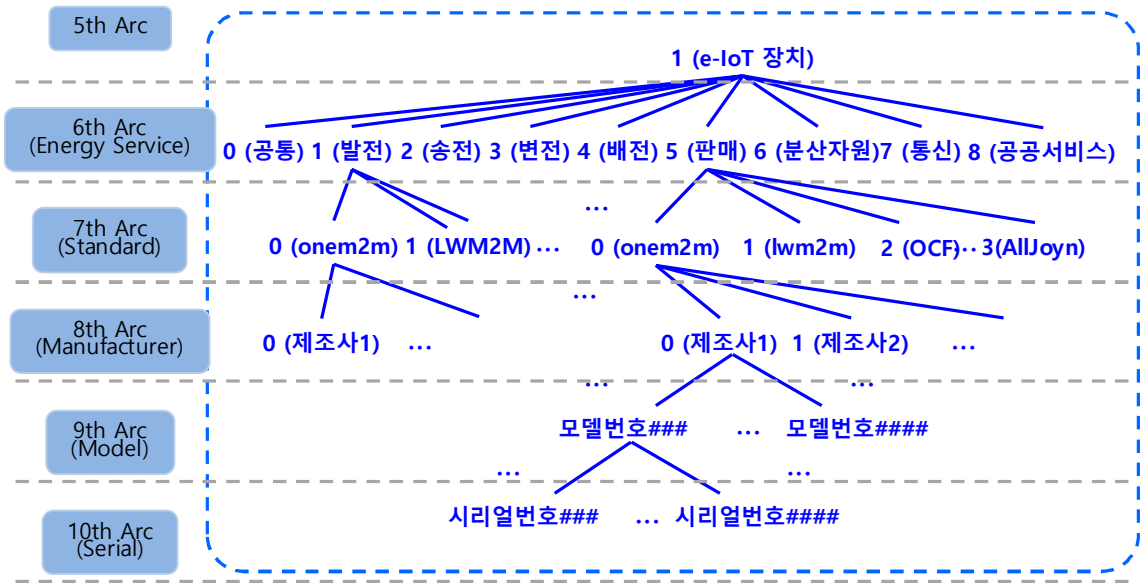
<표 6-1> 기관 OID 체계 (한국전력의 경우)

ID	설명
1	ISO
2	ISO Member-Body
410	대한민국 국가 코드
200073	전력회사 기관 (예시 한국전력)

6.3 장치 식별자

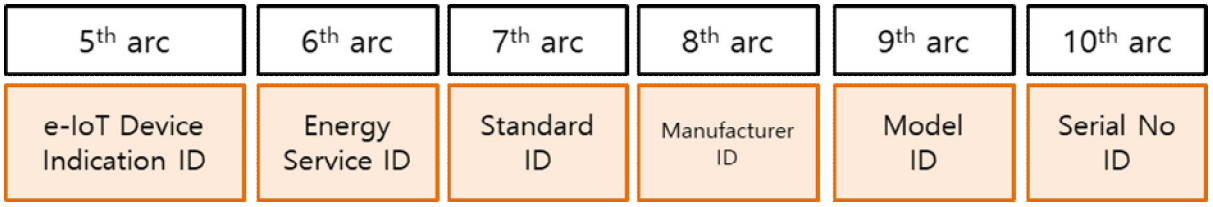
6.3.1 개요

e-IoT 장치 식별자는 최상의 Arc에 e-IoT Device Indication ID로 하위는 순서대로 Service ID, Standard ID, Manufacturer ID, Model ID, Serial No ID로 구성된다.



(그림 6-3) 전력회사 기관 e-IoT 장치식별자 체계 예시

6.3.2 계층 구조



(그림 6-4) 전력회사의 장치 식별자 구조

a) 5th Arc-e-IoT Device

전력회사 e-IoT 장치(e-IoT 게이트웨이와 e-IoT 디바이스)를 의미하는 번호.

종합적으로 전력회사가 한국전력이라고 가정할 때 IoT Indication ID는 '1.2.410.200073.1'이 된다.

<표 6-2> Device Indication ID 예시

Indication ID	설명
1	e-IoT 장치

b) 6th Arc- Service

e-IoT 디바이스 ID 체계에서 Service ID는 전력회사에서 제공하는 서비스 영역을 의미한다. 예를 들어, 배전, 송전, 발전, 변전, 통신, 공공서비스 등을 의미한다.

<표 6-3> Service ID 예시

Service ID	설명
0	공통
1	발전
2	송전
3	변전
4	배전
5	판매
6	분산에너지자원
7	통신
8	공공서비스
101-199	실험서비스용

c) 7th Arc - Standard

e-IoT 디바이스 ID 체계에서 Standard ID는 e-IoT 표준으로서 채택되는 oneM2M, LWM2M, OCF를 구분하는 ID이다.

<표 6-4> Standard ID 목록 예시

Standard ID	설명
0	oneM2M
1	LWM2M
2	OCF
3	AllJoyn
4	OIC

d) 8th Arc - Manufacturer

제조사 ID는 e-IoT에 납품하는 회사의 식별자로 전력회사에서 부여하며, 전력회사 내에서 유일성이 보장되도록 관리되는 번호이다. 별도의 체계나 DB를 이용하여 관리하여야 한다.

<표 6-5> Manufacturer ID 목록 예시

Manufacturer ID	설명
0	제조사1
1	제조사2

e) 마. 9th~10th Arc-Model, Serial No

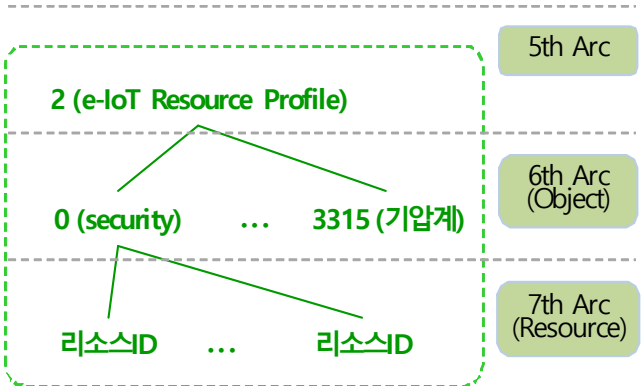
모델 ID와 시리얼 ID는 제조사에서 부여하는 번호이다.

예를 들어, e-IoT 플랫폼의 경우 Service ID는 '0'으로 전력회사 범용에 해당하며, 기본적으로 oneM2M 표준을 체계를 따르고(Standard ID: 0), 제조사 ID는 '0'으로, 모델 ID, 시리얼 ID를 각각 '1', '1'로 정의할 수 있다. 그래서 e-IoT 플랫폼의 OID의 예시는 '1.2.410.200073.1.0.0.0.1.1'로 표현할 수 있다.

6.4 Resource 프로파일 식별자

6.4.1 개요

본 표준에서는 Resource의 특징을 기술할 수 있는 ID를 OID 체계로 기술하여 oneM2M 기반의 Resource 모델에서 포함하도록 한다.
예를 들어 <container> Resource의 실제 담고 있는 데이터의 의미를 ontologyRef attribute를 이용해서 표시하는데 이 정보에 resource profile ID를 표시한다. 보고 주기에 대한 설정 정보는 container-period <container>로 정의하고 여기의 ontologyRef에 '1.2.410.200073.2.3.100' 라는 e-IoT Resource Profile OID를 할당하여 이 Resource가 보고 주기 설정 정보임을 표시한다. 이 OID에서 '3.100'은 3번 Device Object에 새롭게 정의된 보고 주기 resource이다.



(그림 6-5) Resource 프로파일 식별자 체계

6.4.2 계층 구조

5th arc	6th arc	7th arc
e-IoT Resource Profile Indication ID	e-IoT Object ID	e-IoT Resource ID

(그림 6-6) Resource 프로파일 식별자 계층 구조

a) 5th Arc-e-IoT Resource Profile Indication ID
이 번호는 기관에서 Resource Profile를 의미하는 번호를 자체 할당할 수 있다. 예를 들어 IoT 서비스에서는 IoT Resource Profile ID를 의미하는 번호로 2번을 할당한다면, 종합적으로 IoT Resource Profile ID는 '1.2.410.200073.2'이다.

<표 6-6> IoT Profile Indication ID

Indication ID	설명

2	IoT Resource Profile

b) 6th Arc-Object

6th arc 와 7th arc는 번호 체계는 기관과 관계없이 다음의 체계를 따를 것을 권고한다. e-IoT Object ID 범위는 아래 표와 같다. 표준의 통일성을 위해 OMA Object 번호를 관리하고 있는 OMNA의 체계를 승계한다.

<표 6-7> e-IoT Object 범위

분류	IoT Object ID 범위	설명
Object 종속적	0-1023	OMA에서 정의한 Object
예약됨(reserved)	1024-2047	미래 사용을 위한 범위
외부표준기관 사용(ext-label)	2048-10240	제3의 표준 단체에서 정의할 경우
사설사용(x-label)	10241-32768	개인 및 업체에서 정의한 Object

c) 7th Arc - Resource

e-IoT Resource ID 범위는 다음 표와 같다. 표준의 통일성을 OMNA의 체계를 승계한다.

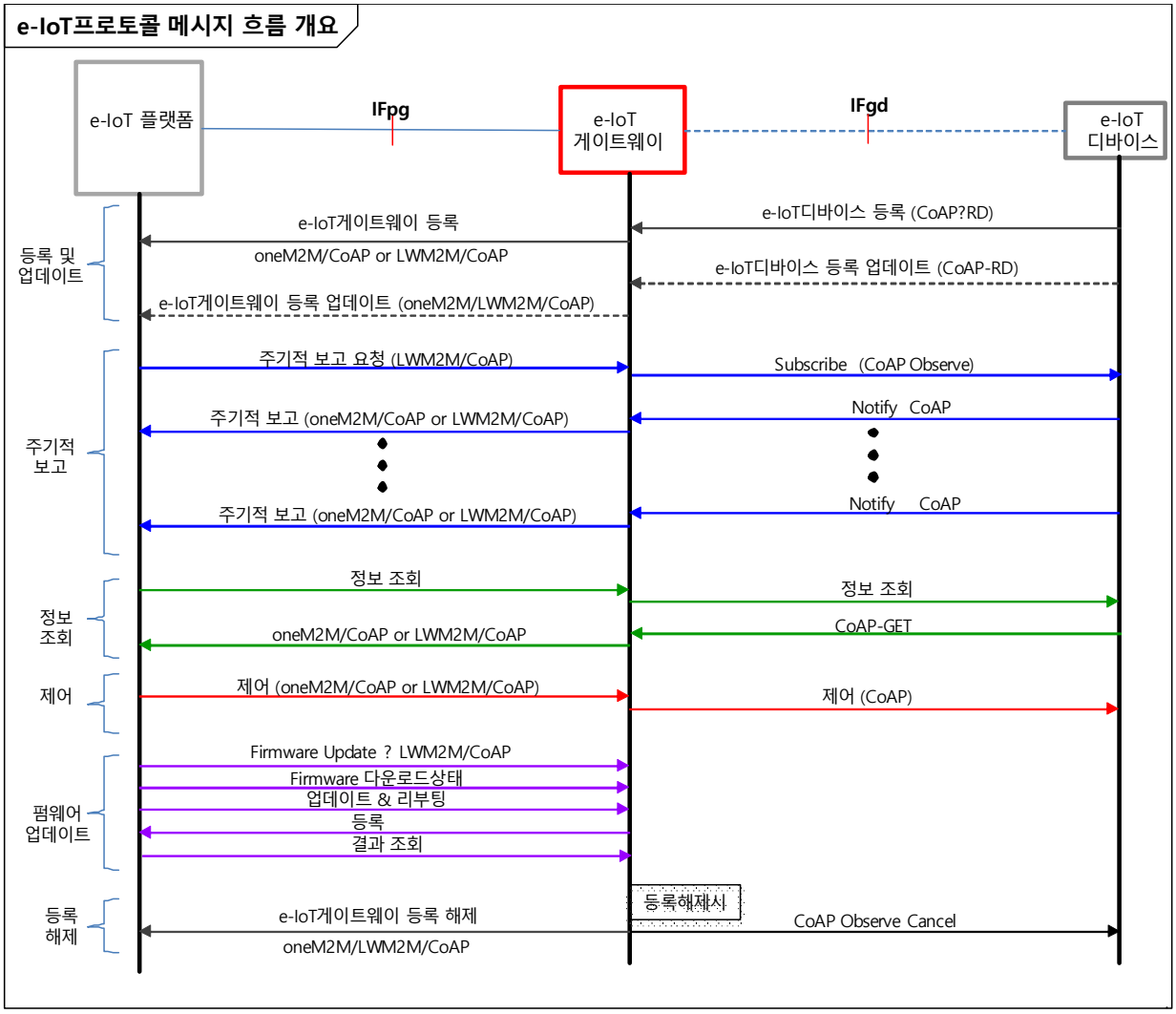
<표 6-8> 한전 IoT Resource ID 범위

분류	IoT Resource ID 범위	설명
Object 종속적	0-2047	해당 Object별로 정의된다.
재사용 가능 Resource ID	2048-32768	표준 통일성을 위해서 OMNA에서 할당한 번호를 승계한다. 모든 Object에서 사용할 수 있다.
예약된 Resource ID	32769 -	미래 사용을 위해 예약된 범위

7 e-IoT 서비스 절차

7.1 개요

e-IoT 서비스는 프로토콜의 동작에 따라서 다음과 같은 '등록 및 업데이트'와 '주기적 보고', '정보조회', '제어', '펌웨어업데이트' 동작들로 구성된다.



(그림 7-1) e-IoT 서비스에 따른 프로토콜 메시지 절차

a) CoAP Resource Directory 표준을 이용한 e-IoT 디바이스를 e-IoT 게이트웨이에 등록한다. 이때 e-IoT 게이트웨이는 Resource Directory 역할을 수행한다. 등록된 이후에는 주기적인 업데이트를 통해서 e-IoT 디바이스 정보를 유지한다.

b) e-IoT 디바이스 정보를 등록시킨 e-IoT 게이트웨이는 e-IoT 플랫폼에 등록시킨다. 이 때 표준은 oneM2M 혹은 LWM2M으로 가능하다. 등록된 이후에는 주기적으로 업데이트를 수행하여 e-IoT 게이트웨이 정보를 유지시킨다. 업데이트는 LWM2M에서만 정의되어 있다.

c) 주기적인 정보 보고를 위해서 우선 e-IoT 게이트웨이는 e-IoT 디바이스 정보에 구독(subscribe)을 수행한다. 등록된 subscribe 정보에 의거하여 주기적인 Notification 메시지를 e-IoT 게이트웨이에 보고한다.

d) e-IoT 디바이스로부터 주기적으로 보고받은 정보를 e-IoT 게이트웨이는 e-IoT 플랫폼에 주기적으로 정보를 보고한다. 이때 LWM2M과 oneM2M 표준으로 가능하다.

e) 주기적인 정보 보고 이외에도 직접 원하는 정보를 요청할 수 있다. e-IoT 플랫폼으로부터 정보 조회 요청을 받은 e-IoT 게이트웨이는 해당 e-IoT 디바이스에 재요청하여 정보를 획득하여 응답할 수 있다. 또는 e-IoT 디바이스에 바로 요청할 수 없는 경우는 이미 캐시되어 있는 정보로 응답한다.

f) 정보 조회와 유사하게 특정 동작을 요청하거나, 혹은 정보를 수정할 수 있는 제어 동작이 있다.

g) 사. e-IoT 게이트웨이 펌웨어 업데이트는 LWM2M 기반으로 수행한다.

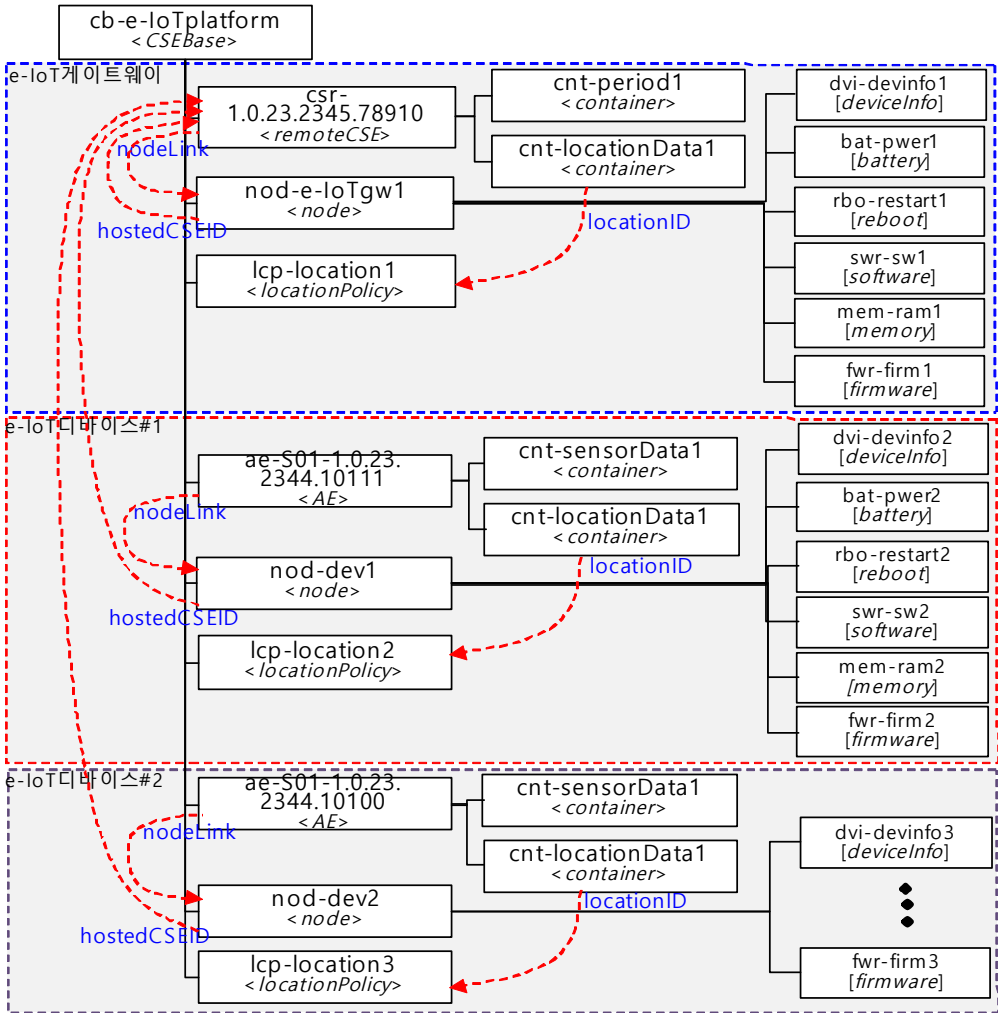
h) e-IoT 게이트웨이 및 e-IoT 디바이스 등록 해제를 할 수 있다.

7.2 IFpg - oneM2M

7.2.1 장치 등록

7.2.1.1 게이트웨이 등록

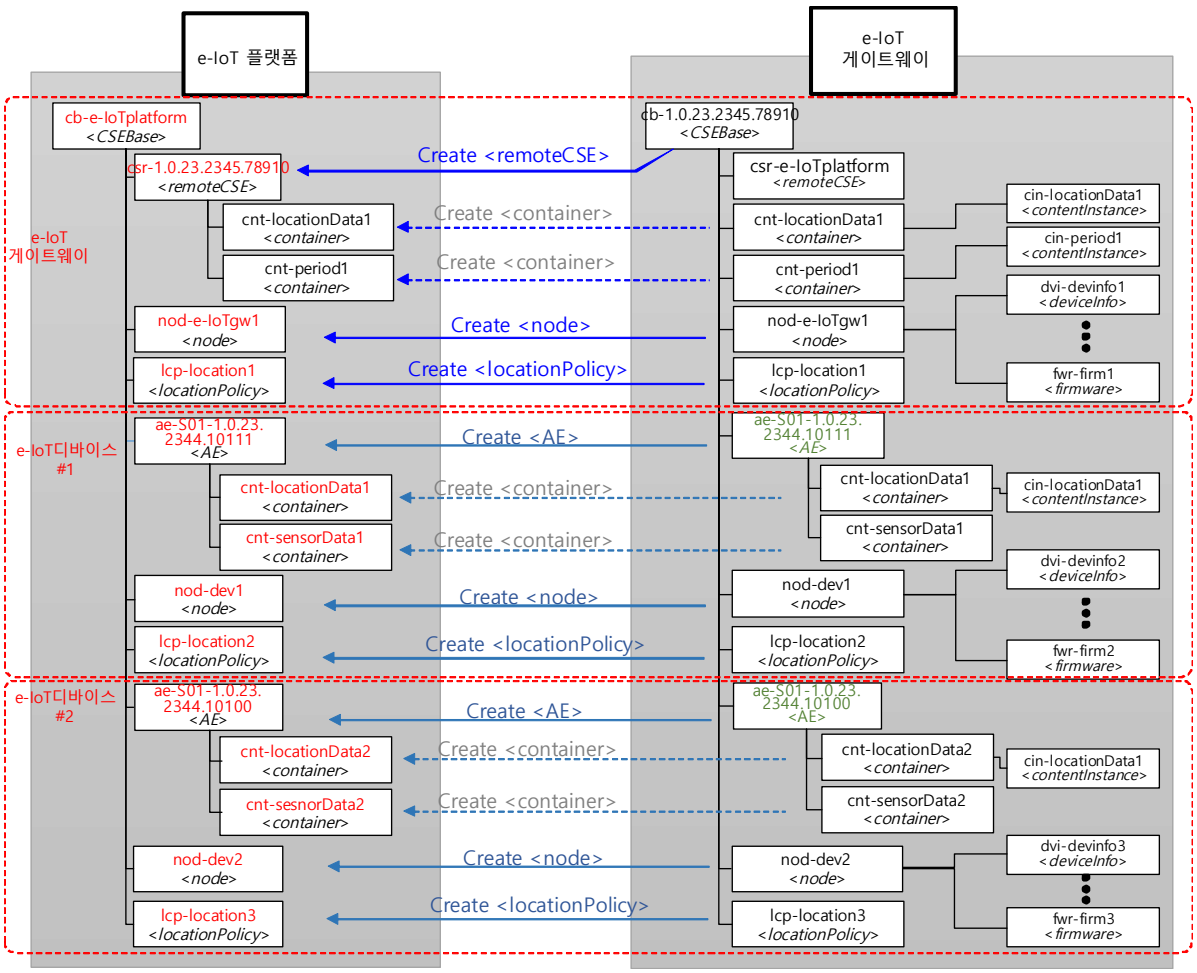
oneM2M 기반 e-IoT 게이트웨이 등록이란 e-IoT 게이트웨이 정보모델을 e-IoT 플랫폼 CSE 하부에 생성하는 것으로 정의한다. e-IoT 게이트웨이 CSEBase를 플랫폼의 CSEBase 하부에 remoteCSE Resource로 등록시키고 관련된 Resource를 등록하는 형태로 진행된다. e-IoT 게이트웨이는 <remoteCSE>로 e-IoT 디바이스는 <AE> Resource로 등록된다.



(그림 7-2) oneM2M e-IoT 게이트웨이 등록

oneM2M 기반 e-IoT 게이트웨이 등록 과정이다. 본 등록 과정은 먼저 e-IoT 게이트웨이 관련 Resource들을 등록하고 그 이후에 하부에 연결되어 있는 e-IoT 디바이스들의 Resource를 등록하는 순서로 진행된다. e-IoT 게이트웨이 등록 과정은 <remoteCSE> Resource를 먼저 생성하고 <node> Resource, <locationPolicy> Resource 순으로, e-IoT 디바이스는 <AE> Resource를 먼저 생성하고 <node> Resource, <locationPolicy> Resource 순으로 등록 과정이 진행된다. <node> Resource와 <locationPolicy> Resource의 순서는 변경되어도 무관하다. <remoteCSE>와 <AE> Resource를 <node>, <locationPolicy> Resource 보다 먼저 생성하는 이유는 <node>, <locationPolicy> Resource를 생성할 때 From 필드 정보를 통해서 기 생성된 <remoteCSE>와 <AE> Resource와 관계성을 유추할 수 있게 하기 위함이다. <remoteCSE>와 <AE> Resource는 하부에 위치한 자식 Resource를 함께 등록할 수 있다. 위 (그림 7-2)에서 <container> Resource가 <remoteCSE>의 자식 Resource에 해당한다. 자식 Resource가 포함되지 않을 경우 별도의 생성 과정을 통해서 등록하여야 한다. 자식 Resource와 포함해서 생성하는 경우와, 자식 Resource를 별도를 생성하는 것을 선택적으로 지원할 수 있다.

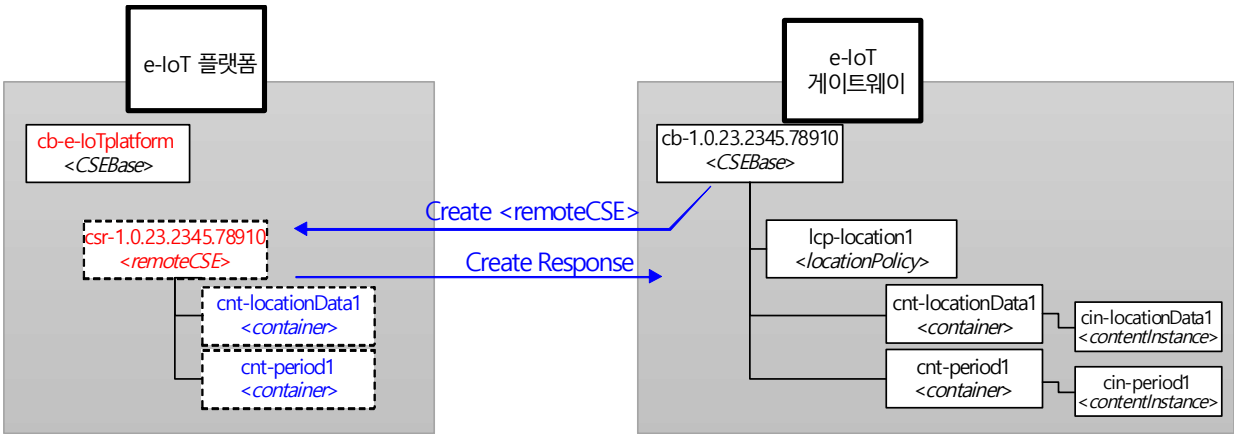
e-IoT 게이트웨이 <CBEBase> Resource를 등록하고, 위치 정보와 주기 정보를 포함할 Resource를 생성·등록한다. 그 다음 e-IoT 게이트웨이 하부에 연결되어 있는 e-IoT 디바이스 Resource인 <AE> Resource와 센서 정보 관련 Resource를 등록하는 과정으로 수행된다. 연결되어 있는 e-IoT 디바이스의 수만큼 <AE> Resource를 각각 등록시킨다.



(그림 7-3) oneM2M e-IoT 게이트웨이 등록 과정

7.2.1.1.1 CSE 등록

CREATE 메소드를 사용해서 e-IoT 플랫폼에 e-IoT 게이트웨이 <remoteCSE> Resource 생성을 요청하는 과정이다. <remoteCSE> Resource 하부에 자식 Resource로 생성될 수 있는 <container> Resource를 포함해서 같이 생성할 수 있다.



(그림 7-4) oneM2M e-IoT 게이트웨이 CSE 등록 예시

CREATE 메소드는 CoAP POST 메소드에 매핑되고 CREATE 요청 메시지의 파라미터로는 다음과 같다.

① Uri-Path 옵션: 대상 Resource인 “cb-e-IoTplatform”를 가리킨다.
② 요청자가 생성한 Message ID와 Token을 포함한다.
③ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 CSE-ID
④ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑤ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <remoteCSE>에 해당하는 16 값을 가진다.
⑥ Payload: <remoteCSE> Resource에 정의된 속성값을 포함한다. 하부에 Resource를 포함할 경우에는 반드시 resourceName 속성을 포함한다. 자식 Resource를 포함 여부는 선택적이다. 본 과정에서 자식 Resource를 포함하지 않을 경우 별도의 생성 절차를 통해서 생성해야한다.

- <remoteCSE> Resource의 속성 및 자식 Resource: rn(resourceName), cst(cseType), csi(CSE-ID), poa(pointOfAccess), nl(nodeLink), cb(CSEBase), cnt<container>

⑦ <remoteCSE> 생성 요청 메시지를 수신한 플랫폼은 생성한 <remoteCSE> Resource의 nodeLink 속성값은 추수에 생성될 <node> Resource ID로 나중에 설정한다. 또한 해당 <node> Resource의 hostedCSEID 속성값을 여기서 생성한 <remoteCSE> Resource의 ID로 설정한다.

다음은 CREATE 요청 메시지 예시이다. 본 예시 메시지의 파란색으로 표현된 부분은 자식 Resource에 해당하는 <container> Resource를 포함한 경우이다.

```
CON 0.02 POST [5f9f]
Token: a269096a
Content-format: 50
Uri-Path: cb-e-IoTplatform
oneM2M-FR: cb-1.0.23.2345.78910
oneM2M-RQI: 1779001762
oneM2M-TY: 16
Payload:
{ “csr”:{
        “rn":”cb- cnt-locationData1”,”cst”:2, “csi”:“cb-1.0.23.2345.78910”,
        “poa”:“coap://{e-IoT 게이트웨이IP 주소}:{Port번호}”,
        “cb”:”coap://{e-IoT 게이트웨이IP 주소}:{Port번호}/cb-1.0.23.2345.78910”,
        “cnt“:{ “rn”:”cnt-locationData1”, “or”:”1.2.410.200073.2.6.7” , “mni”:1,
                “cni”:1, “cbs”:10, “cin”:{ “rn”:“cin-locationData1”, “cs”:4, “con”:45366 }
        }
        “cnt“:{ “rn”: “cnt-period1”, “or”:”1.2.410.200073.2.3.100”, “mni”:1,
                “cni”:1, “cbs”:10, “cin”:{ “cs”:4, “con”:3600 }
        }
      }
}
```

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <remoteCSE> Resource의 다음 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime

② <remoteCSE> 생성요청 메시지를 수신한 e-IoT 플랫폼은 생성한 <remoteCSE> Resource의 nodeLink 속성 값을 <node> Resource ID로 설정한다. 또한 해당 <node> Resource의 hostedCSEID 속성값을 여기서 생성한 <remoteCSE> Resource의 ID로 설정한다.

③ e-IoT 플랫폼에서 성공적으로 <remoteCSE> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

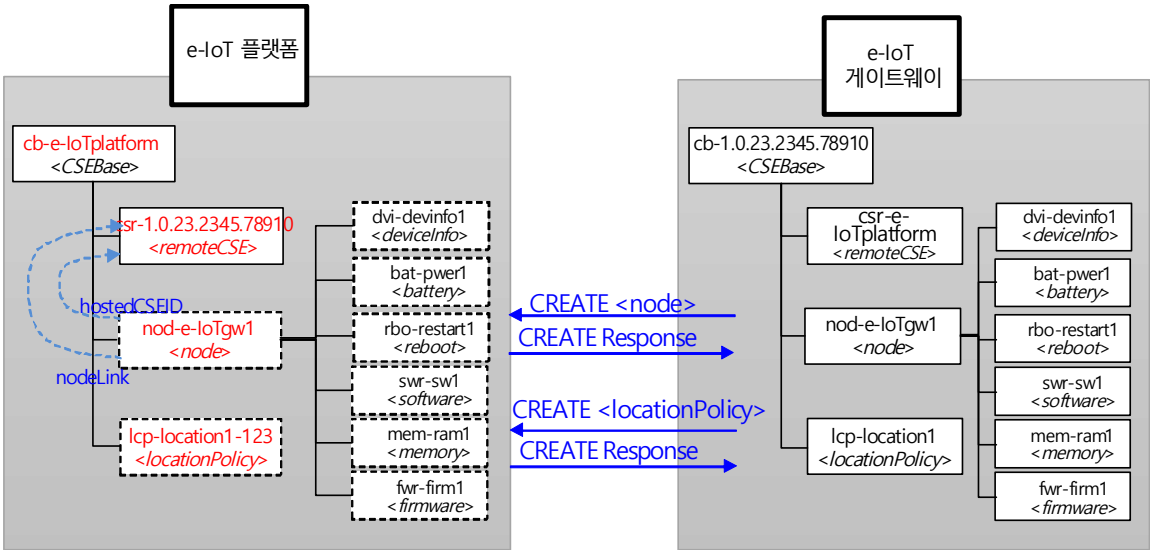
- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값
- location-Path 옵션: 마지막으로 <remoteCSE> Resource가 생성된 위치 정보를 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당. 선택적으로 포함할 수 있다.

다음은 성공적으로 <remoteCSE> Resource가 생성된 응답 메시지이다.

CON 2.01 Created [5f9f]
Token: a269096a
oneM2M-RSC: 2001
oneM2M-RQI: 1779001762
location-Path: csr-1.0.23.2345.78910

7.2.1.1.2 노드, locationPolicy 등록

CREATE 메소드를 사용해서 e-IoT 플랫폼에 e-IoT 게이트웨이 <node> Resource와 <locationPolicy> Resource 생성을 요청하는 과정이다.



(그림 7-5) oneM2M e-IoT 게이트웨이의 node 및 locationPolicy 등록 예시

<node> Resource 생성 요청 메시지 처리 절차는 다음과 같다.

CREATE 메소드는 CoAP POST 메소드에 매핑된다. CREATE 요청 메시지의 파라미터로는 다음과 같다.

① Uri-Path 옵션: 대상 Resource인 “cb-e-IoTplatform”를 가리킨다.
② 요청자가 생성한 Message ID와 Token을 포함한다.
③ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 CSE-ID로 설정한다.
④ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑤ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <node>에 해당하는 14로 설정한다.
⑥ Payload: <node> Resource에 정의된 속성값를 각 생성 메시지에 포함하다. 하부에 Resource를 포함할 경우에는 반드시 resourceName 속성을 포함한다. <node> Resource 하부에서는 장치 관리 관련 자식 Resource 및 속성값들을 포함할 수 있다. 이 자식

Resource들이 포함하지 않을 경우 별도의 생성 과정을 통해서 생성하여야 한다.

- <node> Resource의 속성과 자식 Resource: rn(resourceName), ni(nodeID), hcl(hostedCSELink), [deviceInfo], [battery], [reboot], [software], [memory], [firmware]
- <deviceInfo> Resource의 속성: rn(resourceName), mgd(mgmtDefinition), dlb(deviceLabel), man(manufacturer), mod(model), dty(deviceType), fwv(fwVersion), hwv(hwVersion)
- [battery] Resource의 속성: rn(resourceName), mgd(mgmtDefinition), btl(batteryLevel), bts(batteryStatus), blt(batteryLowThreshold), bcp(batteryCapacity)
- [reboot] Resource의 속성: rn(resourceName), mgd(mgmtDefinition), rbo(reboot), far(factoryReset)
- [software] Resource의 속성: rn(resourceName), mgd(mgmtDefinition), vr(version), nam(name), url(URL), in(install), un(uninstall), ins(installStatus), 설치된 Resource가 없는 경우는 생략한다.
- [memory] Resource의 속성: rn(resourceName), mgd(mgmtDefinition), mma(memAvailable), mmt(memTotal)
- [firmware] Resource의 속성: rn(resourceName), mgd(mgmtDefinition), vr(version), nam(name), url(URL), ud(update), uds(updateStatus), udr(updateResult)

<locationPolicy> Resource 생성 요청 메시지 처리 절차는 다음과 같다.

CREATE 메소드는 CoAP POST 메소드에 매핑된다. CREATE 요청 메시지의 파라미터로는 다음과 같다.

① Uri-Path 옵션: 대상 Resource 값, 예를 들어 “cb-e-IoTplatform”를 포함한다.
② 요청자가 생성한 Message ID와 Token을 포함한다.
③ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 CSE-ID로 설정한다.
④ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑤ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <locationPolicy>에 해당하는 10으로 설정한다.
⑥ Payload: <locationPolicy> Resource에 정의된 속성값이 포함된다. 필수 속성은 아래와 같다.

- <locationPolicy> Resource의 속성: los(locationSource), loi(locationContainerID)
- loi 속성은 요청자의 것으로 값을 입력한다. 그리고 반드시, 위치 정보 저장용 <container> Resource를 생성할 때 resource name의 정보가 locationContainerID와 동일한 값이 전달되도록 한다.

<node> Resource가 성공적으로 생성되면 생성된 <node> Resource 식별자를 기 생성한 <remoteCSE> Resource의 nodeLink 속성 값에 설정한다. 또한 해당 <node> Resource의 hostedCSEID 속성값을 앞서 생성한 <remoteCSE> Resource의 ID로 설정한다.

<node> Resource CREATE 요청 메시지 예시는 다음과 같다.

```
CON 0.02 POST [5f9f]
Token: a269096a
Content-format: 50
Uri-Path: cb-e-IoTplatform
oneM2M-FR: cb-1.0.23.2345.78910
oneM2M-RQI: 1779001762
oneM2M-TY: 14
Payload:
{     “nod”:{
                “rn”:“nod-e-IoTgw1”, “ni”:“1.2.410.200073.1.5.0.1.1001.100001”,
                ”hcl”:“cb-1.0.23.2345.78910”,
                “dvi”:{“rn”:“dvi-devinfo1”, “mgd”:”deviceInfo”, “dlb”:“e-IoTGWTestLabel”,
                “man”:“XXCom”, “mod”:“1001”, “dty”:“e-IoTgateway”, “fwv”:“1.2.0”,
                “hwv”:“1.0”},
                “bat”:{“rn”:“bat-pwer1”, “mgd”:”battery”, “btl”:60, “bts”:0, “blt”:3,
                “bcp”:2100},
                “fwr”:{“rn”:“fwr-firm1”, “mgd”:“firmware”, “vr”:“1.2.0”, “url”:“”, “ud”:false, “uds”:0,
                “udr”:0},
                “mem”:{“rn”:“mem-ram1”, “mgd”:“memory”, “mma”:141, “mmt”:256},
                “rbo”:{“rn”:“rbo-restart1”, “mgd”:“reboot”, “rbo”:false, “far”:false},
       }
}
```

① <node> CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <node> Resource의 다음 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime

② e-IoT 플랫폼에서 성공적으로 <node> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <node> Resource가 생성된 위치 정보에 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당하며 자식 Resource의 포함 여부는 선택적으로 할 수 있다. 포함 되지 않을 경우 별도의 생성 과정을 통해서 생성하여야 한다.

다음은 성공적으로 <node> Resource가 생성된 응답 메시지이다.

```
CON 2.01 Created [5f9f]
Token: a269096a
oneM2M-RSC: 2001
oneM2M-RQI: 1779001762
location-Path: nod-e-IoTgw1
```

다음은 <locationPolicy> Resource CREATE 요청 메시지 예시이다.

```
CON 0.02 POST [5fa4]
Token: a769096a
Content-format: 50
Uri-Path: cb-e-IoTplatform
oneM2M-FR: cb-1.0.23.2345.78910
oneM2M-RQI: 1779001767
oneM2M-TY: 10
Payload:
{ "lcp":
    { {"rn":"lcp-location", "los":3, "loi":"cnt-locationData1"}
}
```

<locationPolicy> Resource 생성 응답 메시지 정보는 다음과 같다.

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <locationPolicy> Resource의 다음 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime

② e-IoT 플랫폼에서 성공적으로 <locationPolicy> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 예를 들어 2.01(Created)
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <locationPolicy> Resource가 생성된 위치 정보를 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당한다.

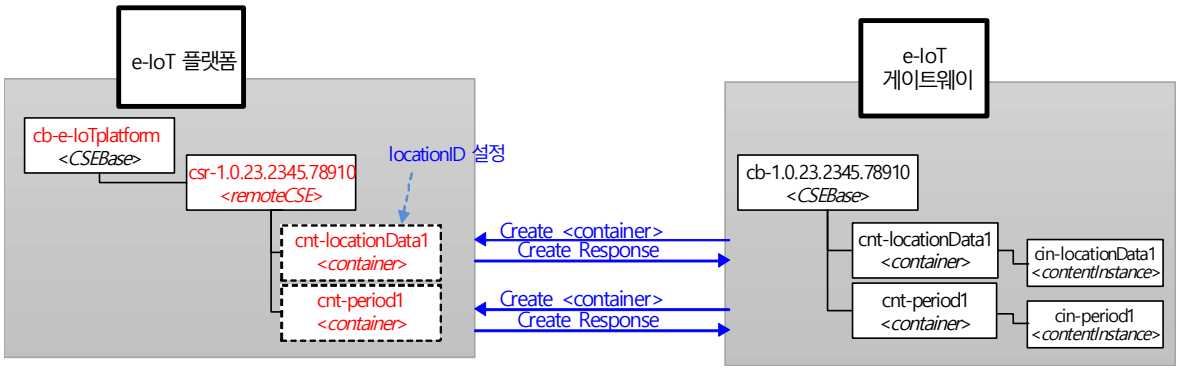
다음은 성공적으로 <locationPolicy> Resource가 생성된 응답 메시지 예시이다.

```
CON 2.01 Created [5fa4]
Token: a769096a
oneM2M-RSC: 2001
oneM2M-RQI: 1779001767
```

Location-Path: lcp-location1

7.2.1.1.3 관련 Resource 등록

e-IoT 게이트웨이의 CSE 등록 과정에서 함께 등록되지 않은 Resource를 등록하는 절차이다. 앞 절에서 정보 저장용 <container> Resource가 등록되었다면 이 과정은 생략할 수 있다. 그렇지 않은 경우 별도로 위치 정보를 저장할 수 있는 <container> Resource와 보고 주기 정보를 저장하는 <container> Resource를 생성한다. 위치정보 저장용과 보고 주기 정보 저장용 <container> Resource들은 실제 값을 저장하기 위한 <contentInstance> Resource를 포함하여 생성한다.



(그림 7-6) 게이트웨이 관련 Resource 등록 예시

위치 정보 저장용 <container> Resource 생성 요청 메시지의 정보는 다음과 같다(실제 위치값을 저장하는 <contentInstance> Resource도 포함하여 생성함).

① CREATE 메소드는 CoAP POST 메소드에 매핑된다.
② CREATE 요청 메시지의 파라미터로는 다음과 같다.
③ Uri-Path 옵션: 대상 Resource값, 예를 들어 “cb-e-IoTplatform”와 csr-1.0.23.2345.78910 각각 포함한다.
④ 요청자가 생성한 Message ID와 Token을 포함한다.
⑤ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 CSE-ID로 설정한다.
⑥ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑦ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <container>에 해당하는 3으로 설정한다.
⑧ Payload: <container> Resource에 정의된 속성값과 실제 위치값을 갖는 <contentInstance> Resource를 자식 Resource로 포함한다. 필수 속성은 아래와 같다. 하부에 Resource가 포함될 때에는 resourceName 속성을 반드시 포함한다.
㉮ 위치 정보 저장용 <container> Resource의 속성과 자식 Resource: or(ontologRef),

mni(maxNrOfInstance), cni(currentNrOfInstance), cbs(currentByteSize), li(locationID), <contentInstance>

- or 속성값: 본 <container> Resource가 저장하는 정보로서, 위치정보를 의미한다. 이에 해당하는 OID는 1.2.410.200073.2.6.7이다.
- mni 속성값: 1. 현 위치값 하나만 저장하기 때문이다. cni 속성값도 동일하게 1을 입력한다.

㉳ 위치 정보용 <contentInstance> Resource의 속성: rn(resourceName), cs(contentSize), con(content)

다음은 위치 정보 저장용 <container> Resource 생성 요청 메시지 예시이다.

```
CON 0.02 POST [5fa9]
Token: a769756a
Uri-Path: cb-e-IoTplatform
Uri-Path: csr-1.0.23.2345.78910
oneM2M-FR: cb-1.0.23.2345.78910
oneM2M-RQI: 1779001768
oneM2M-TY: 3
Payload:
{    "cnt":
     { "rn":"cnt-locationData1, "or":"1.2.410.200073.2.6.7" ,
          "mni":1, "cni":1, "cbs":10,
          "cin":{ "rn":"cin-locationData1", "cs":4, "con":45366 }
     }
}
```

위치 정보 저장용 <container> Resource 생성 응답 메시지 정보는 다음과 같다.

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <container> Resource의 다음 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime
- 위치정보 획득 방법에 관한 정보를 담고 있는 locationID 속성값을 기 생성한 <locationPolicy> Resource ID로 설정한다.

② e-IoT 플랫폼에서 성공적으로 <container> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <container> Resource가 생성된 위치 정보를 포함한다.

- Payload: oneM2M의 Content 파라미터에 해당한다.

다음은 성공적으로 위치 정보 저장용 <container> Resource 생성 요청에 대한 응답 메시지 예시이다.

CON 2.01 Created [5fa9]
Token: a769756a
oneM2M-RSC: 2001
oneM2M-RQI: 1779001768
Location-Path: cnt-locationData1

주기 정보 저장용 <container> Resource 생성 요청 메시지 정보는 다음과 같다(실제 주기값을 저장하는<contentInstance> Resource도 포함하여 생성함).

① CREATE 메소드는 CoAP POST 메소드에 매핑된다.
② CREATE 요청 메시지의 파라미터로는 다음과 같다.
③ Uri-Path 옵션: 대상 Resource 값, 예를 들어 “cb-e-IoTplatform”와 csr-1.0.23.2345.78910 각각 포함한다.
④ 요청자가 생성한 Message ID와 Token을 포함한다.
⑤ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 CSE-ID로 설정한다.
⑥ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑦ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <container>에 해당하는 3으로 설정한다.
⑧ Payload: <container> Resource에 정의된 속성값과 실제 주기값을 갖는 <contentInstance> Resource를 자식 Resource로 포함한다. 필수 속성은 아래와 같다.
㉮ 주기 정보 저장용 <container> Resource의 속성과 자식 Resource: or(ontologRef), mni(maxNrOfInstance), cni(currentNrOfInstance), cbs(currentByteSize), li(locationID), <contentInstance>

- or 속성값: 본 <container> Resource가 저장하는 정보로서, 주기정보를 의미한다. 이에 해당하는 OID는 1.2.410.200073.2.3.100이다.
- mni 속성값: 1. 주기값 하나만 저장하기 때문이며 cni 속성값도 동일하게 1을 입력한다.

㉯ 위치 정보용 <contentInstance> Resource의 속성: cs(contentSize), con(content)

다음은 주기 정보 저장용 <container> Resource 생성 요청 메시지 예시이다.

```
CON 0.02 POST [5eb0]
Token: a7697123
Content-format: 50
Uri-Path: cb-e-IoTplatform
Uri-Path: csr-1.0.23.2345.78910
oneM2M-FR: cb-1.0.23.2345.78910
oneM2M-RQI: 1779001009
oneM2M-TY: 3Payload:
{    “cnt":
     {    “rn”:”cnt-period1, “or”:”1.2.410.200073.2.3.100” , “mni”:1,
          “cni”:1, “cbs”:10, “cin”:{ “cs”:4, “con”:3600 }
     }
}
```

보고주기정보 저장용 <container> Resource 생성 응답 메시지 정보는 다음과 같다.

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <container> Resource의 다음 속성들에 값을 설정해야 한다.

㉮ parentID, creationTime, expirationTime, lastModifiedTime

㉯ e-IoT 플랫폼에서 성공적으로 <container> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 예를 들어 2.01(Created)
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <container> Resource가 생성된 위치 정보를 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당한다.

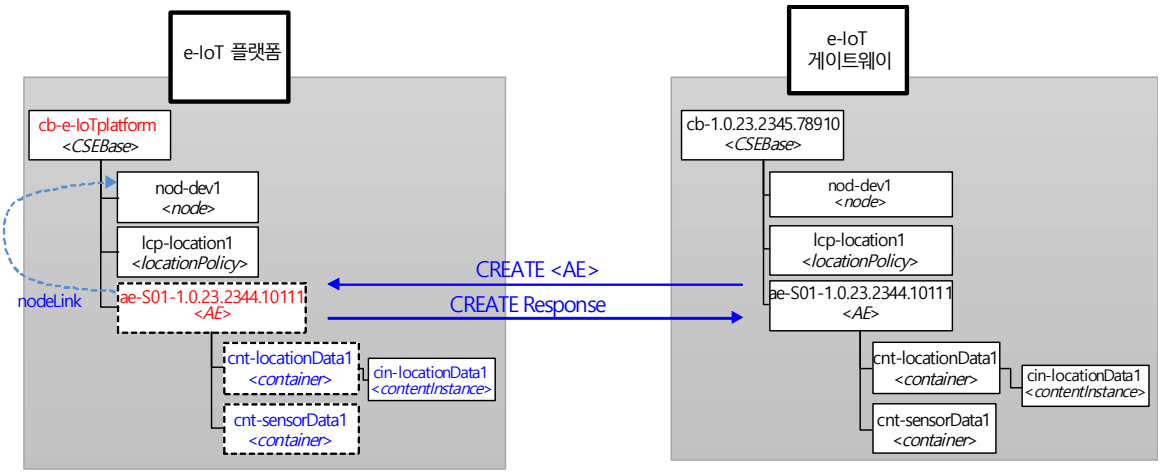
다음은 성공적으로 보고 주기 정보 저장용 <container> Resource 생성 요청에 대한 응답 메시지 예시이다.

```
CON 2.01 Created [5eb0]
Token: a7697123
oneM2M-RSC: 2001
oneM2M-RQI: 1779001009
Location-Path: cnt-period1
```

7.2.1.2 디바이스 등록

7.2.1.2.1 AE 등록

CREATE 메소드를 사용해서 e-IoT 플랫폼에 e-IoT 디바이스의 <AE> Resource 생성을 요청하는 과정이다. <AE> Resource 하부에 자식 Resource로 생성될 수 있는 <container> Resource를 포함해서 같이 생성할 수 있다. <container> 자식 Resource를 포함하지 않을 경우 별도의 생성 과정을 통해서 생성한다.



(그림 7-7) e-IoT 디바이스 AE 등록 예시

CREATE 메소드를 사용해서 e-IoT 플랫폼에 e-IoT 디바이스의 <remoteCSE> Resource 생성을 요청하는 과정이다.

① CREATE 메소드는 CoAP POST 메소드에 매핑된다. CREATE 요청 메시지의 파라미터로는 다음과 같다.
② Uri-Path 옵션: 대상 Resource인 “cb-e-IoTplatform”를 가리킨다.
③ 요청자가 생성한 Message ID와 Token을 포함한다.
④ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 AE-ID로 설정한다.
⑤ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑥ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <AE>에 해당하는 2로 설정한다.
⑦ Payload: <AE> Resource에 정의된 속성값

- <AE> Resource의 속성 및 자식 Resource: rn(resourceName), aei(AE-ID), or(ontologyRef), cnt<container>
- or 값은 <AE>에 해당한 센서나 디바이스의 종류를 Resource Profile ID로 표현한다.

다음은 <AE> Resource CREATE 요청 메시지 예시이다.

```
CON 0.02 POST [5fee]
Token: a269096f
Content-format: 50
Uri-Path: cb-e-IoTplatform
oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111
```

```
oneM2M-RQI: 1889001766
oneM2M-TY: 2
Payload:
{   “ae”:
    {   “rn”:“ae-S01-1.0.23.2344.10111”, “aei”:“ae-e-IoT-dev1”
        “cnt”:
        {  “rn”:cnt-locationData1”, “or”:”1.2.410.200073.2.6.7” , “mni”:1,
            “cni”:1, “cbs”:10,
            “cin”:{ “rn”:”cin-locationData1”, “cs”:4, “con”:45366 }
        }
        “cnt":
        {  “rn”:cnt-sensorData1, “or”:”1.2.410.200073.2.3301.5700” , “mni”:10,
           “mbs”:50, “cni”:0, “cbs”:0
    },
    }
}
```

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <AE> Resource의 다음 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime,

② e-IoT 플랫폼에서 성공적으로 <remoteCSE> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <AE> Resource가 생성된 위치 정보를 에 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당한다.

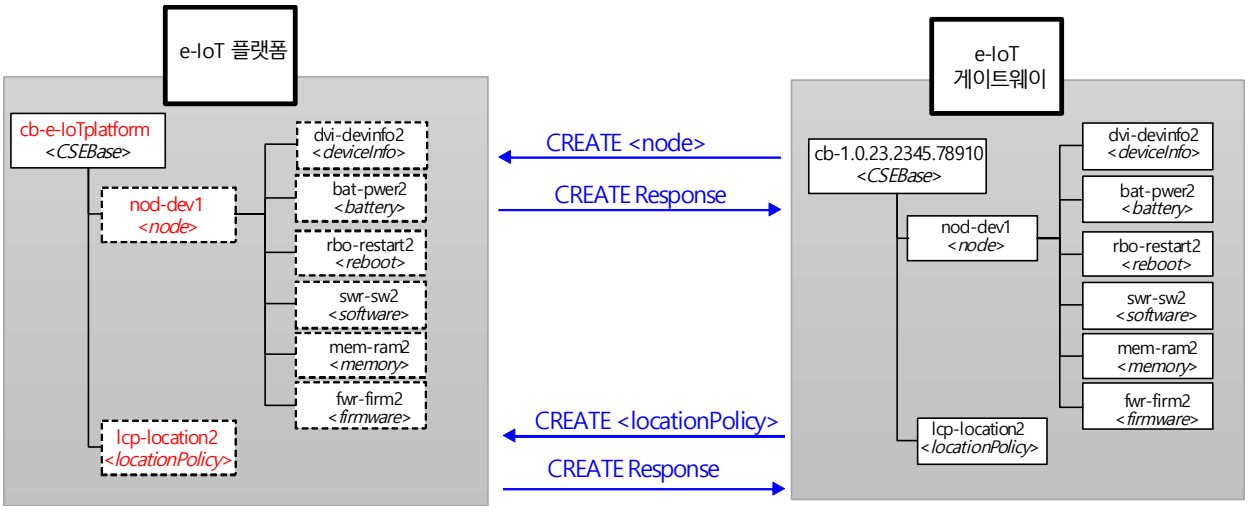
다음은 성공적으로 <remoteCSE> Resource가 생성된 응답 메시지이다.

```
CON 2.01 Created [5fee]
Token: a269096f
oneM2M-RSC: 2001
oneM2M-RQI: 1889001766
Location-Path:ae-S01-1.0.23.2344.10111
```

7.2.1.2.2 노드, locationPolicy 등록

CREATE 메소드를 사용해서 e-IoT 플랫폼에 e-IoT 디바이스의 <node> Resource 생성

을 요청하는 과정이다.



(그림 7-8) e-IoT 디바이스의 node 및 locationPolicy Resource 등록 과정

① CREATE 메소드는 CoAP POST 메소드에 매핑된다. CREATE 요청 메시지의 파라미터로는 다음과 같다.
② Uri-Path 옵션: 대상 Resource인 “cb-e-IoTplatform”를 가리킨다.
③ 요청자가 생성한 Message ID와 Token을 포함한다.
④ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 AE-ID로 설정한다.
⑤ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑥ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <node>에 해당하는 14로 설정한다.
⑦ Payload: <node> Resource 하부에서는 장치 관리 관련 Resource 및 속성값 들이 포함된다.

- <node> Resource의 자식 Resource 및 속성: ni(nodeID), hcl(hostedCSELink), [deviceInfo], [battery], [reboot], [software], [memory], [firmware]
- [deviceInfo] Resource(dvi) 의 속성: mgd(mgmtDefinition), dlb(deviceLabel), man(manufacturer), mod(model), dty(deviceType), fwv(fwVersion), hwv(hwVersion)
- [battery] Resource(bat)의 속성: mgd(mgmtDefinition), btl(batteryLevel), bts(batteryStatus), blt(batteryLowThreshold), bcp(batteryCapacity)
- [reboot] Resource(rbo)의 속성: mgd(mgmtDefinition), rbo(reboot), far(factoryReset)
- [software] Resource(swr)의 속성: mgd(mgmtDefinition), vr(version), nam(name), url(URL), in(install), un(uninstall), ins(installStatus), 설치된 Resource 없는 경우는 생략한다.
- [memory] Resource(mem)의 속성: mgd(mgmtDefinition), mma(memAvailable), mmt(memTotal)
- [firmware] Resource(fwr)의 속성: mgd(mgmtDefinition), vr(version), nam(name),

url(URL), ud(update), uds(updateStatus), udr(updateResult)

다음은 <node> Resource CREATE 요청 메시지 예시이다.

```
CON 0.02 POST [5fee]
Token: a269096f
Content-format: 50
Uri-Path: cb-e-IoTplatform
oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111
oneM2M-RQI: 1889001766
oneM2M-TY: 14
Payload:
{
        “nod”:
        {
            “rn”:“nod-dev1”, “ni”:“1.2.410.200073.10.0.1.1002.100011”,
            “hcl”:“cb-1.0.23.2345.78910”,
            “dvi”:{“mgd”:”deviceInfo”, “dlb”:“e-IoTDevTestLabel”, “man”:“XXCom”,
            “mod”:“1002”, “dty”:“e-IoTdevice”, “fwv”:“1.0.1”, “hwv”:“1.1”},
            “bat”:{“mgd”:”battery”, “btl”:50, “bts”:0, “blt”:3, “bcp”:1500},
            “fwr”:{“mgd”:“firmware”, “vr”:“1.0.1”, “url”:null, “ud”:false, “uds”:0, “udr”:0},
            “mem”:{“mgd”:“memory”, “mma”:41, “mmt”:128},
            “rbo”:{“mgd”:“reboot”, “rbo”:false, “far”:false},
        }
}
```

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <node> Resource의 다음 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime

② e-IoT 플랫폼에서 성공적으로 <node> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <node> Resource가 생성된 위치 정보에 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당한다.

③ <node> Resource가 성공적으로 생성되면 생성된 <node> Resource 식별자를 기 생성한 <AE> Resource의 nodeLink 속성 값에 설정한다. 또한 해당 <node> Resource의 hostedCSEID 속성값을 앞서 생성한 <remoteCSE> Resource의 ID로 설정한다.

다음은 성공적으로 <node> Resource가 생성된 응답 메시지이다.

```
CON 2.01 Created [5fee]
Token: a269096f
oneM2M-RSC: 2001
oneM2M-RQI: 1889001766
Location-Path: nod-dev1
```

<locationPolicy> Resource 생성 요청 메시지 처리 절차는 다음과 같다.

① CREATE 메소드는 CoAP POST 메소드에 매핑된다. CREATE 요청 메시지의 파라미터로는 다음과 같다.
② Uri-Path 옵션: 대상 Resource 값, 예를 들어 “cb-e-IoTplatform”를 포함한다.
③ 요청자가 생성한 Message ID와 Token을 포함한다.
④ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 AE-ID으로 설정한다.
⑤ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑥ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <locationPolicy>에 해당하는 10으로 설정한다.
⑦ Payload: <locationPolicy> Resource에 정의된 속성값 포함된다. 필수 속성은 아래와 같다.
  ㉮ <locationPolicy> Resource의 속성: los(locationSource), loi(locationContainerID)
    • locaitonContainerID 값과 동일한 값이 전달되도록 한다.

다음은 CREATE 요청 메시지 예시이다.

```
CON 0.02 POST [5ffe]
Token: a769123b
Uri-Path: cb-e-IoTplatform
oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111
oneM2M-RQI: 1779001700
oneM2M-TY: 10
Payload:
{
      “lcp”:{“rn”:“lcp-location1”, “los”:3, “loi”:“cnt-locationData1”}
}
```

<locationPolicy> Resource 생성 응답 메시지 정보는 다음과 같다.

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <locationPolicy> Resource의 다음 속성들에 값을 설정해야 한다.
• parentID, creationTime, expirationTime, lastModifiedTime

② e-IoT 플랫폼에서 성공적으로 <locationPolicy> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

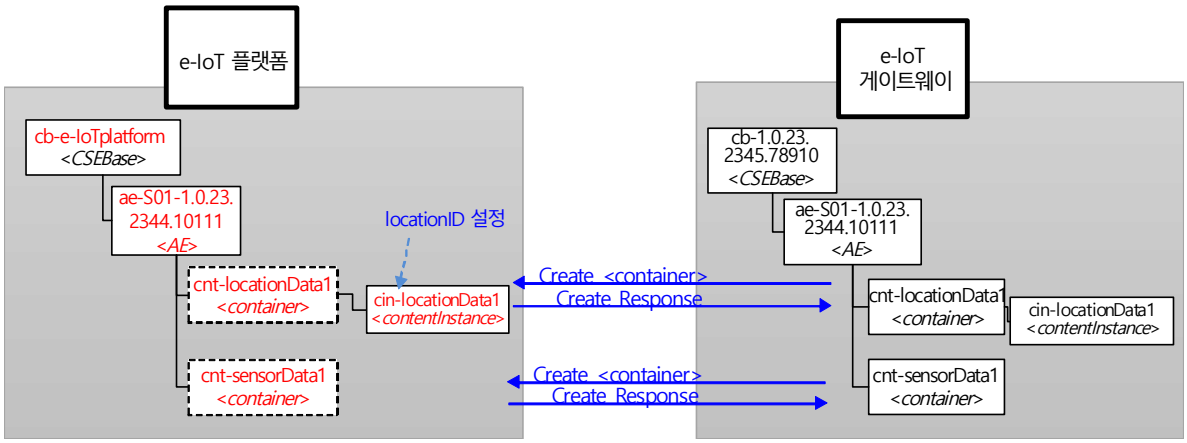
- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <locationPolicy> Resource가 생성된 위치 정보를 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당한다.

다음은 성공적으로 <locationPolicy> Resource가 생성된 응답 메시지 예시이다.

```
CON 2.01 Created [5ffe]
Token: a769123b
oneM2M-RSC: 2001
oneM2M-RQI: 1779001700
Location-Path: lcp-location1
```

7.2.1.2.3 관련 리소스 등록

e-IoT 디바이스의 <AE>를 등록하는 과정에서 등록되지 않은 Resource 정보를 등록하는 단계이다. 위치 정보를 저장할 수 있는 <container> Resource와 실제 센서 정보를 저장하는 <container> Resource를 생성한다. 위치 정보 저장용 <container> Resource들은 실제 값을 저장하기 위한 <contentInstance> Resource를 포함하여 생성하는 반면에 실제 센서 측정값을 저장하는 <container> Resource는 별도로 <contentInstance> Resource를 포함하지 않는다. 센서 측정 값은 추후에 주기적으로 센서 정보를 보고할 때 <contentInstance>를 생성하기 때문이다.



(그림 7-9) e-IoT 디바이스 관련 Resource 예시

위치 정보 저장용 <container> Resource 생성 요청 메시지(실제 위치값을 저장하는 <contentInstance> Resource도 포함하여 생성함)

① CREATE 메소드는 CoAP POST 메소드에 매핑된다.
② CREATE 요청 메시지의 파라미터로는 다음과 같다.
③ Uri-Path 옵션: 대상 Resource 값, 예를 들어 “cb-e-IoTplatform”와 ae-S01-1.0.23.2344.10111 각각 포함한다.
④ 요청자가 생성한 Message ID와 Token을 포함한다.
⑤ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 AE-ID으로 설정한다.
⑥ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑦ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <container>에 해당하는 3으로 설정한다.
⑧ Payload: <container> Resource에 정의된 속성값 과 실제 위치값을 갖는 <contentInstance> Resource를 자식 Resource로 포함한다. 필수 속성은 아래와 같다.
  ㉮ 위치 정보 저장용 <container> Resource의 속성과 자식 Resource: or(ontologRef), mni(maxNrOfInstance), cni(currentNrOfInstance), cbs(currentByteSize), <contentInstance>

- or 속성값: 본 <container> Resource가 저장하는 정보로서, 위치정보를 의미한다. 해당하는 OID는 1.2.410.200073.2.6.7이다.
- mni 속성값: 1, 현 위치값 하나만 저장하기 때문이며 cni 속성값도 동일하게 1을 입력한다

㉳ 위치 정보용 <contentInstance> Resource의 속성: rn(resourceName), cs(contentSize), con(content)

다음은 위치 정보 저장용 <container> Resource 생성 요청 메시지 예시이다.

```
CON 0.02 POST [5fa9]
Token: a769756b
Uri-Path: cb-e-IoTplatform
Uri-Path: ae-S01-1.0.23.2344.10111
oneM2M-FR: cb-1.0.23.2345.78910/ ae-S01-1.0.23.2344.10111
oneM2M-RQI: 1779001769
oneM2M-TY: 3
Payload:
{ “cnt”:
   { “rn”:”cnt-locationData1”, “or”:”1.2.410.200073.2.6.7” , “mni”:1, “cni”:1, “cbs”:10,
      “cin”:{ “rn”:”cin-locationData1”, “cs”:4, “con”:45366 }
   }
}
```

위치 정보 저장용 <container> Resource 생성 응답 메시지 정보는 다음과 같다.

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <container> Resource의 다음의 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime,
- 위치 정보 획득 방법에 관한 정보를 담고 있는 locationID 속성값을 기 생성한 <locationPolicy> Resource ID로 설정한다.

② e-IoT 플랫폼에서 성공적으로 <container> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <container> Resource가 생성된 위치 정보를 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당한다.

다음은 성공적으로 위치 정보 저장용 <container> Resource 생성 요청에 대한 응답 메

시지 예시이다.

```
CON 2.01 Created [5fa9]
Token: a769756b
oneM2M-RSC: 2001
oneM2M-RQI: 1779001769
Location-Path: cnt-locationData1
```

센서 측정 정보 저장용 <container> Resource 생성 요청 메시지 처리 절차는 다음과 같다.

① CREATE 메소드는 CoAP POST 메소드에 매핑된다.
② CREATE 요청 메시지의 파라미터로는 다음과 같다.
③ Uri-Path 옵션: 대상 Resource 값, 예를 들어 “cb-e-IoTplatform”와 ae-S01-1.0.23.2344.10111 각각 포함한다.
④ 요청자가 생성한 Message ID와 Token을 포함한다.
⑤ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 AE-ID으로 설정한다.
⑥ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑦ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <container>에 해당하는 3으로 설정한다.
⑧ Payload: <container> Resource에 정의된 속성값을 포함한다. 필수 속성은 아래와 같다.

㉮ 센서 측정 정보 저장용 <container> Resource의 속성: or(ontologRef), mni(maxNrOfInstance), cni(currentNrOfInstance), cbs(currentByteSize), li(locationID)

- or 속성값: 본 <container> Resource가 저장하는 정보. 예를 들어OID 1.2.410.200073.2.3301.5700라 하면 조도 센서(3301)의 측정값(5700)을 의미한다.
- mni 속성값: <contentInstance>의 최대 개수.
- cni 속성값: 0, 현재 <contentInstance>의 수.

다음은 센서 측정 정보 저장용 <container> Resource 생성 요청 메시지 예시이다.

```
CON 0.02 POST [5eb0]
Token: a7697129
Uri-Path: cb-e-IoTplatform
Uri-Path: ae-S01-1.0.23.2344.10111
oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111
oneM2M-RQI: 1779001119
oneM2M-TY: 3
payload:
{ “cnt":
```

```
        { “rn”:“cnt-sensorData1”, “or”:”1.2.410.200073.2.3301.5700”,
            “mni”:10, “mbs”:50, “cni”:0, “cbs”:0
        }
}
```

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <container> Resource의 다음 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime

② e-IoT 플랫폼에서 성공적으로 <container> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

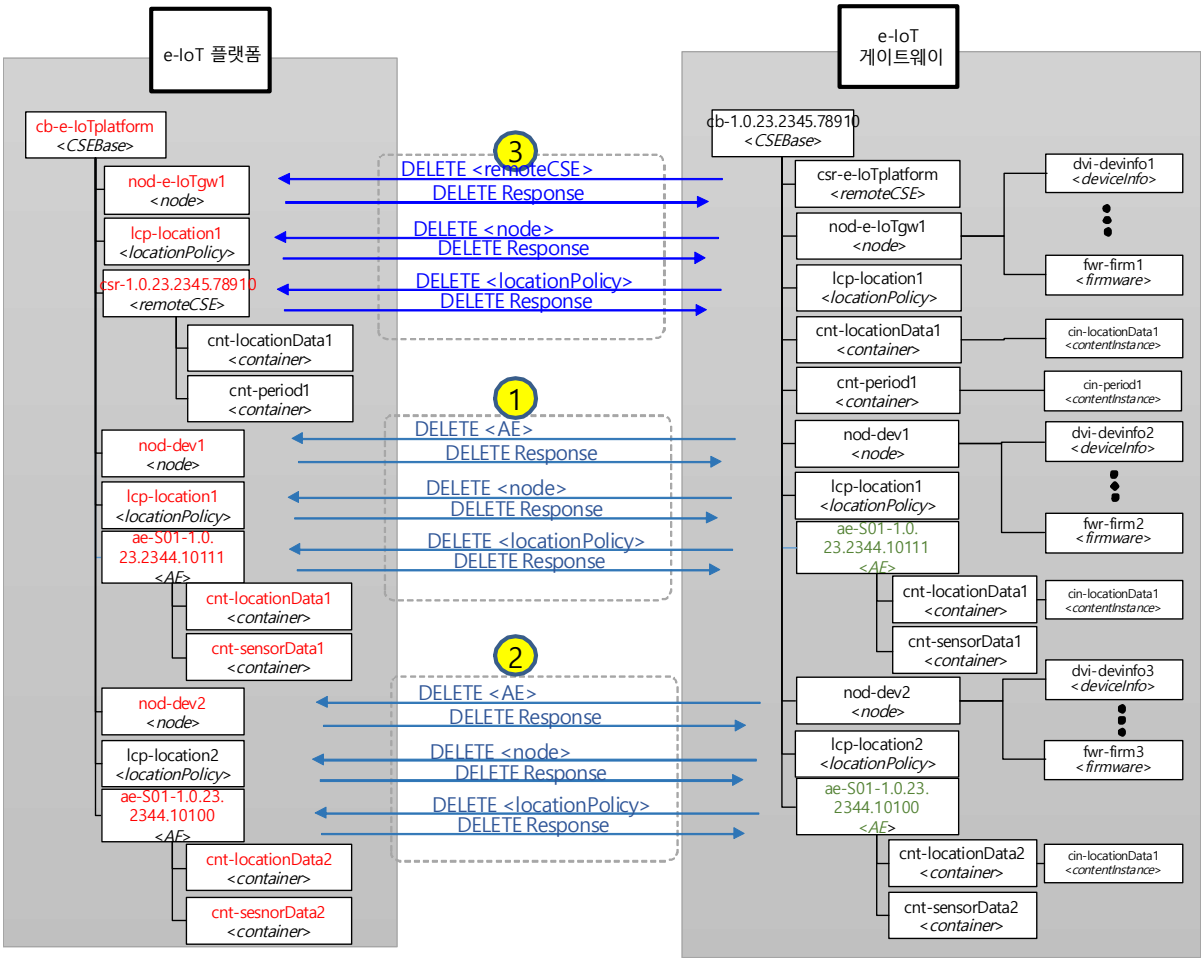
- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <container> Resource가 생성된 위치 정보를 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당한다.

다음은 센서 측정 정보 저장용 <container> Resource 생성 요청에 성공적으로 처리한 응답 메시지 예시이다.

```
CON 2.01 Created [5eb0]
Token: a7697129
oneM2M-RSC: 2001
oneM2M-RQI: 1779001119
Location-Path: cnt-sensorData1
```

7.2.1.3 게이트웨이 등록 해제

DELETE 메소드를 사용해서 e-IoT 플랫폼에 e-IoT 게이트웨이가 등록했던 <remoteCSE> Resource와 <AE> Resource, <node> Resource, <locationPolicy> Resource 삭제를 요청하는 과정이다. 특히, e-IoT 게이트웨이 하부에 연결되어 있는 e-IoT 디바이스의 Resource까지 삭제해야 한다. 삭제 순서는 아래 그림과 같이, e-IoT 디바이스의 것을 먼저 하고 맨 마지막에 e-IoT 게이트웨이의 것을 삭제한다. 아래 예시에서 두개의 e-IoT 디바이스를 먼저 등록 해제하고 마지막으로 e-IoT 게이트웨이를 등록 해제시키는 과정을 보여주고 있다. <remoteCSE>, <AE>, <node>, <locationPolicy>의 순서는 관련이 없으나, 되도록 <remoteCSE> 와 <AE> Resource를 먼저 삭제 요청하는 것을 권고한다. 삭제 관련 메시지 형식은 다음과 같이 동일하다.



(그림 7-10) oneM2M 기반 e-IoT 게이트웨이 등록해제 과정

① DELETE 메소드는 CoAP DELETE 메소드에 매핑된다. DELETE 요청 메시지의 파라미터로는 다음과 같다.

② Uri-Path 옵션: 대상 Resource이다. <remoteCSE> Resource인 경우 예를 들어 “cb-e-IoTplatform/csr-1.0.23.2345.78910”를 가리킨다.

③ 요청자가 생성한 Message ID와 Token을 포함한다.

④ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 AE ID 혹은 CSE ID로 설정한다.

⑤ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.

⑥ Payload: 없음.

등록 해제(DELETE) 요청 메시지를 수신한 e-IoT 플랫폼은 삭제하고자 하는 대상 Resource가 존재하는 지 여부를 판단하고 존재한 경우 해당 <remoteCSE> 와 하부에 있는 자식 Resource를 삭제하고 응답 메시지를 생성하여 전달한다.

① e-IoT 플랫폼에서 성공적으로 <remoteCSE> Resource를 삭제한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 2.02(Deleted).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.

- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- Payload: 선택적으로 포함할 수 있음, 포함할 경우 삭제된 Resource 정보를 포함한다.

먼저, e-IoT 디바이스의 <AE> Resource 등록 해제 요청 DELETE 메시지와 응답 메시지는 다음과 같다.

```
CON 0.04 DELETE [5125]
Token: fe34796a
Uri-Path: cb-e-IoTplatform
Uri-Path: ae-S01-1.0.23.2344.10111
oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111
oneM2M-RQI: 1779001764
```

```
CON 2.02 Deleted [5125]
Token: fe34796a
oneM2M-RSC: 2002
oneM2M-RQI: 1779001764
```

e-IoT 디바이스의 <node>와 <locatoinPolicy> Resource 등록해제 요청 DELETE 메시지와 응답 메시지는 다음과 같다.

```
CON 0.04 DELETE [5123]
Token: fed9096a
Uri-Path: cb-e-IoTplatform
Uri-Path: nod-e-IoTgw1
oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111
oneM2M-RQI: 1779001762
```

```
CON 2.02 Deleted [5123]
Token: fed9096a
oneM2M-RSC: 2002
oneM2M-RQI: 1779001762
```

```
CON 0.04 DELETE [5124]
Token: fed90123
Uri-Path: cb-e-IoTplatform
Uri-Path: lcp-location1
oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111
```

oneM2M-RQI: 1779001763

CON 2.02 Deleted [5124] Token: fed90123 oneM2M-RSC: 2002 oneM2M-RQI: 1779001763

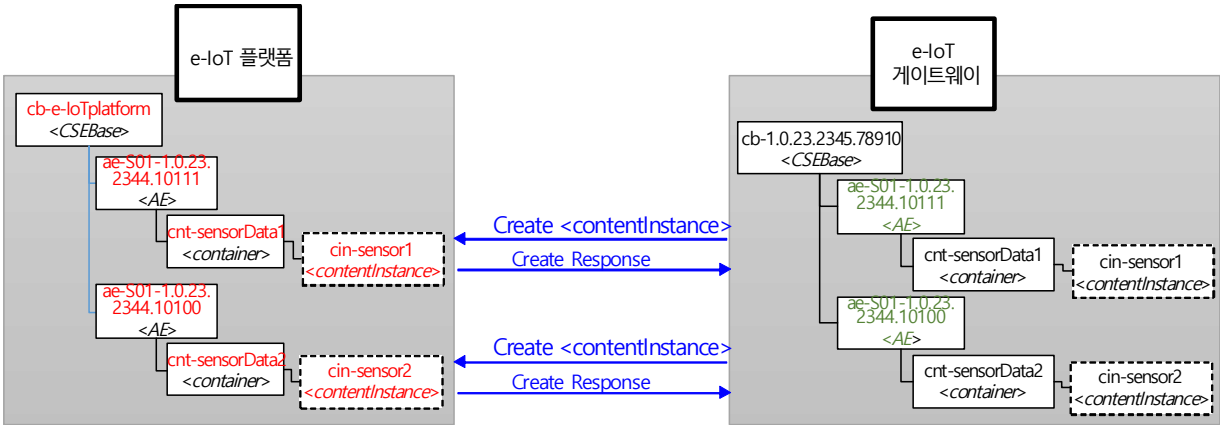
다른 e-IoT 디바이스의 <AE>와 e-IoT 게이트웨이의 <remoteCSE> Resource 등록 해제 요청 DELETE 메시지와 응답 메시지는 다음과 같다. <node>, <locationPolicy> Resource 등록 해제 요청 DELETE 메시지와 응답 메시지는 위의 e-IoT 디바이스의 Resource 등록 해제 요청 메시지와 유사하여 생략한다.

CON 0.04 DELETE [5128] Token: fed9077a Uri-Path: cb-e-IoTplatform Uri-Path: ae-S01-1.0.23.2344.10100 oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10100 oneM2M-RQI: 1779001765
CON 0.04 DELETE [5130] Token: fed1236a Uri-Path: cb-e-IoTplatform Uri-Path: csr-1.0.23.2345.78910 oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10100 oneM2M-RQI: 1779001766
CON 2.02 Deleted [5128] Token: fed9077a oneM2M-RSC: 2002 oneM2M-RQI: 1779001765
CON 2.02 Deleted [5130] Token: fed1236a oneM2M-RSC: 2002 oneM2M-RQI: 1779001766

7.2.2 주기적 보고

e-IoT 플랫폼에 데이터를 주기적으로 전달하는 것을 데이터 주기적 보고라 한다. e-IoT

플랫폼에 데이터 보고는 해당 <container>에 자식 Resource로 <contentInstance> Resource를 생성하는 과정이다. 더불어, 데이터 보고 주기는 기본적으로 e-IoT 게이트웨이에 설정되어 있어야 하며, e-IoT 플랫폼에서 변경 가능하다. 본 예시에서는 두 개의 e-IoT 디바이스가 센서 정보를 주기적으로 보고하는 경우다.



(그림 7-11) oneM2M 기반 e-IoT 게이트웨이의 주기적 보고

데이터 전달을 위한 <contentInstance> Resource 생성 요청 메시지 처리 절차는 다음과 같다.

① CREATE 메소드는 CoAP POST 메소드에 매핑된다.
② CREATE 요청 메시지의 파라미터로는 다음과 같다.
③ Uri-Path 옵션: 대상 Resource 값, 예를 들어 “cb-e-IoTplatform/ae-e-IoTdev1/cnt-sensorData1” 인 경우 “cb-e-IoTplatform” 와 “ae-e-IoTdev1”, “cnt-sensorData1” 를 각각 포함한다.
④ 요청자가 생성한 Message ID와 Token을 포함한다.
⑤ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 AE-ID로 설정한다.
⑥ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑦ oneM2M-TY 옵션: 생성될 Resource Type 값으로 <contentInstance>에 해당하는 4로 설정한다.
⑧ Payload: <contentInstance> Resource에 정의된 속성값을 포함한다. 필수 속성은 아래와 같다.

- <contentInstance> Resource의 속성: rn(resource name), cr(creator), cs(contentSize), con(content)
- or 속성값: 이미 <container> 부모 Resource에 포함되어 있기 때문에 or 속성값을 포함하지 않는다.

다음은 <contentInstance> Resource 생성 요청 메시지 예시이다.

```
CON 0.02 POST [1b55]
Token: a5ec187f
Content-format: 50
Uri-Path: cb-e-IoTplatform
Uri-Path: ae-S01-1.0.23.2344.10111
Uri-Path: cnt-sensorData1
oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111
oneM2M-RQI: 2132339877
oneM2M-TY: 4
Payload:
{ “cin”:
{ “rn”:“cin-sensor1”, “cr”:“ae-S01-1.0.23.2344.10111”, “cs”:2, ”con”:1234 }
}
```

① CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <contentInstance> Resource의 다음 속성들에 값을 설정해야 한다.

- parentID, creationTime, expirationTime, lastModifiedTime

② e-IoT 플랫폼에서 성공적으로 <contentInstance> Resource를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.

- CoAP 응답 코드: 예를 들어 2.01(Created).
- Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
- oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.
- oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
- location-Path 옵션: 마지막으로 <contentInstance> Resource가 생성된 위치 정보를 포함한다.
- Payload: oneM2M의 Content 파라미터에 해당. 성공한 경우와 실패한 경우 모두 포함하지 않는다.

다음은 <contentInstance> Resource 생성 요청에 성공적으로 처리한 응답 메시지 예시이다.

```
CON 2.01 Created [1b55]
Token: a5ec187f
oneM2M-RSC: 2000
oneM2M-RQI: 2132339877
Location-Path: cin-sensor1
```

다음은 e-IoT 게이트웨이에 연결되어 있는 다른 온도센서의 측정 값을 주기적으로 전달하는 요청 메시지와 그에 대한 응답 메시지를 예시이다.

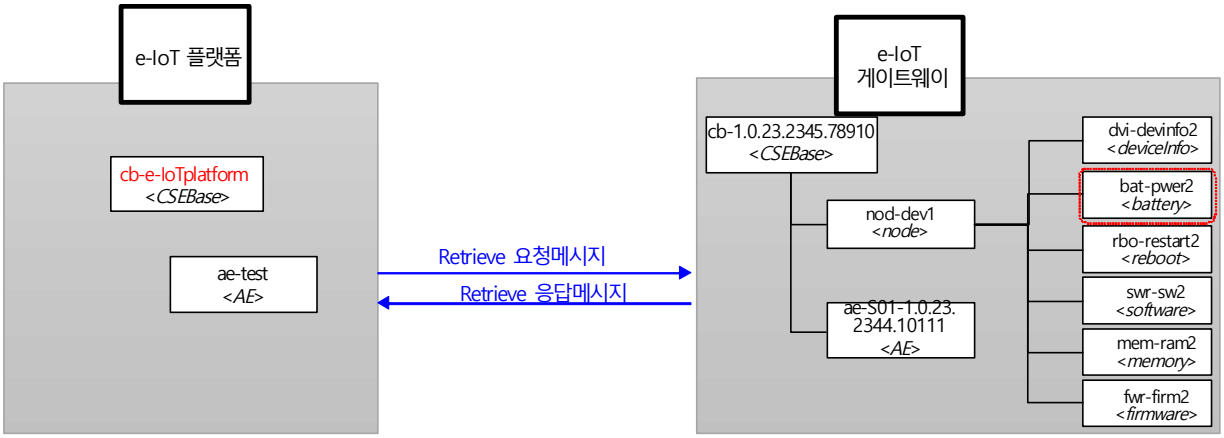
CON 0.02 POST [2b55] Token: a5fe887f Content-format: 50 Uri-Path: cb-e-IoTplatform Uri-Path: ae-S01-1.0.23.2344.10111 Uri-Path: cnt-sensorData2 oneM2M-FR: cb-1.0.23.2345.78910/ae-S01-1.0.23.2344.10111 oneM2M-RQI: 2132339876 oneM2M-TY: 4 Payload: { “cin”: { “rn”:”cin-sensor2”, “cr”:“ae-S01-1.0.23.2344.10100”, “cs”:2, ”con”:17 } }
CON 2.01 Created [2b55] Token: a5fe887f oneM2M-RSC: 2000 oneM2M-RQI: 2132339876 Location-Path: cin-sensor2

7.2.3 정보 조회

oneM2M 기반의 정보 조회 과정은 응답자의 대상 Resource의 속성에 저장된 정보를 조회하는 과정이다. 대상 Resource의 이름을 요청 메시지의 Content 파라미터에 포함하여 전달한다. 요청자는 e-IoT 플랫폼의 CSE가 되며, 응답자는 e-IoT 게이트웨이의 CSE가 된다.

요청자는 oneM2M 표준에서 정의한 일반적인 요청 응답 과정을 준수하여 요청 메시지를 생성하며, 특히 정보 조회를 하고자 하는 대상 Resource의 모든 속성을 조회하거나, 특정 속성만을 조회할 수 있다. 대상 Resource를 CoAP 메시지의 Payload에 명시한 경우 모든 속성을 조회할 수 있고, 특정 속성만을 조회할 경우는 Payload에 그 속성들을 명시하면 된다.

응답자는 요청 메시지의 요청한 속성값들로 응답하거나, 혹은 에러가 발생할 경우 해당 에러 코드를 전달한다.



(그림 7-12) oneM2M 기반 e-IoT 게이트웨이의 데이터 조회 절차 예시

속성의 정보 조회를 위한 요청 메시지 처리 절차는 다음과 같다.

① RETRIEVE 메소드는 CoAP GET 메소드에 매핑된다.
② RETRIEVE 요청 메시지의 파라미터로는 다음과 같다.
③ 요청자가 생성한 Message ID와 Token을 포함한다.
④ Content-Format 옵션: 대상 Resource의 일부 속성들 리스트를 Payload에 포함할 경우 '50'(JSON)을 표시한다.
⑤ Uri-Path 옵션: 대상 Resource 값, 예를 들어 "cb-1.0.23.2345.78910"와 "nod-dev1", "bat-pwer2"를 각각 포함한다.
⑥ oneM2M-FR 옵션: 요청 메시지를 생성한 e-IoT 플랫폼의 AE-ID로 설정한다.
⑦ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 플랫폼이 생성한다.
⑧ Payload: 대상 Resource의 일부 속성들의 이름을 전달할 때 "atrl"(attributeList) 데이터 Type을 이용하여 속성들 이름의 리스트를 전달함 이때 속성의 값은 null로 표시한다.

e-IoT 플랫폼에서 성공적으로 정보 조회 요청 메시지를 처리한 후, 다음 CoAP 응답 메시지를 생성하여 전달한다.

① ACK Transaction 설정한다.
② CoAP 응답 코드: 예를 들어 2.05(Content).
③ Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
④ Content-Format: 50(JSON)
⑤ oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다. 성공의 경우 2000으로 설정한다.
⑥ oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
⑦ Payload: 요청 메시지의 대상 Resource에 해당하는 속성들의 정보를 포함한다.

다음은 대상 Resource의 모든 속성들의 정보 조회 요청 메시지 예시이다.

```
CON 0.01 GET [9faf]
Token: a776723b
Uri-Path: cb-1.0.23.2345.78910
Uri-Path: nod-dev1
Uri-Path: bat-pwer2
oneM2M-FR: cb-e-IoTplatform/ae-test
oneM2M-RQI: 1779001785
Payload:
```

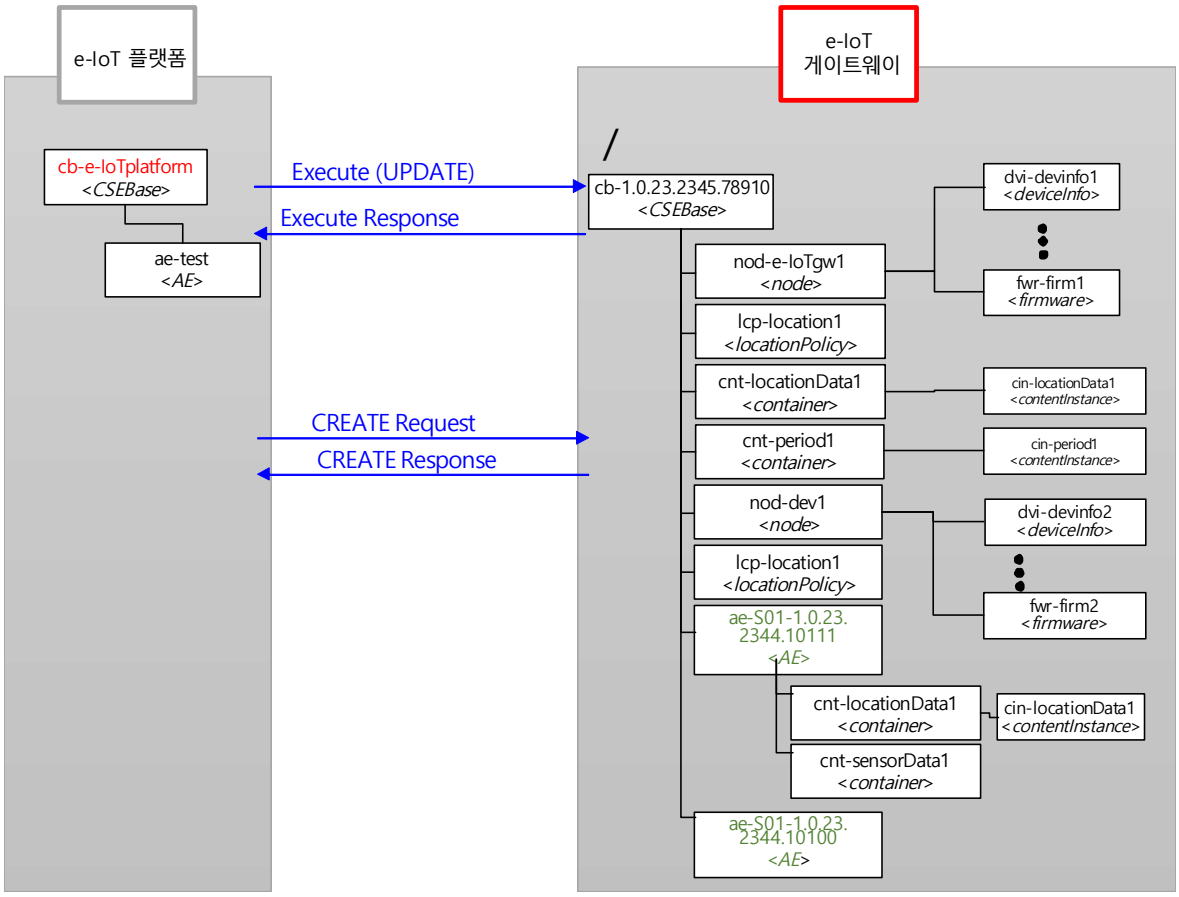
※ 대상 Resource의 모든 속성값을 조회한 경우이므로 Payload가 없음.

다음은 정보 조회 요청 메시지 예시이다.

```
ACK 2.05 Content [9faf]
Token: a776723b
Content-format: 50
oneM2M-RSC: 2000
oneM2M-RQI: 1779001785
Payload:
{ “bat”:
{“mgd”:“battery”, “btl”:86, “bts”:0, “blt”:3, “bcp”:1500}
}
```

7.2.4 제어

제어 과정은 일반적으로 oneM2M의 UPDATE 절차에 해당한다. 그런데, <container> 리소스의 경우에는 CREATE 절차를 통해서 제어를 한다. 대상 리소스의 속성에 저장되어 있는 값을갱신하는 과정으로 갱신하고자 하는 속성의 이름과 변경하고자 하는 값을 Content(Payload)에 포함하여 요청 메시지를 전달한다.



(그림 7-13) oneM2M 기반 e-IoT 게이트웨이 제어 절차 예시

7.2.4.1 일반 리소스 대상 제어

<container>리소스를 제외한 리소스를 대상으로 업데이트할 경우
Execute(UPDATE) 요청 메시지 처리 절차는 다음과 같다.

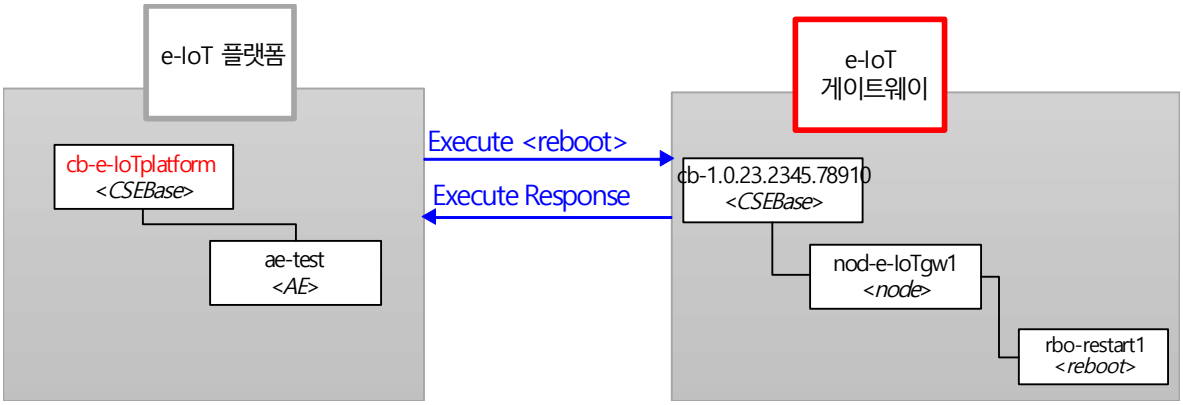
① UPDATE 메소드는 CoAP PUT 메소드에 매핑된다. UPDATE 요청 메시지의 파라미터는 다음과 같다.
② URI-Path 옵션: 대상 리소스 ID로서, 예를 들어 e-IoT 게이트웨이의 배터리 낮은 전압 기준을 변경하고자 할 때, “cb-1.0.23.2345.78910”와 nod-e-IoTgw1, bat-pwer1 각각 포함한다.
③ 요청자가 생성한 Message ID와 Token를 포함한다.
④ Content-format: 50(Json)
⑤ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자의 AE-ID로 설정한다.
⑥ oneM2M-RQI 옵션: Request Identifier, 요청 메시지 생성자인 e-IoT 게이트웨이가 생성한다.
⑦ Payload: 해당 리소스의 업데이트하고자 하는 속성 이름과 변경하고자 하는 값이다.

Execute(UPDATE) 응답 메시지 정보는 다음과 같다.

UPDATE 요청 메시지를 수신한 e-IoT 플랫폼은 요청한 해당 리소스가 존재하는지를 검사한 이후 존재한 경우에 해당 속성값을 갱신하고 그 결과를 다음 CoAP 응답 메시지 반영하여 생성하여 전달한다. 만약 속성값이 존재하지 않는 경우는 속성값을 생성하고 요청된 값으로 반영하고 CoAP 응답 메시지를 생성 전달한다.
만약 UPDATE 요청되는 속성의 값을 NULL로 되어 있는 경우, 해당 속성을 삭제한다.

① ACK Transaction 설정한다.
② CoAP 응답 코드: 예를 들어 성공적으로 갱신된 경우 2.04(Changed).
③ Message ID와 Token을 포함한다 요청 메시지와 동일한 값이다.
④ oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다. 성공한 경우 2004 입력한다.
⑤ oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.
⑥ Payload: 선택적으로 포함할 수 있음. 포함할 경우 갱신된 정보를 포함한다.

다음은 제어의 대표적인 예인 재부팅 과정의 메시지 예시이다.



(그림 7-14) oneM2M 기반 e-IoT 게이트웨이 재부팅 절차

다음은 재부팅 Execute(UPDATE) 요청 메시지 예시이다.

```
CON 0.02 PUT [5f9f]
Token: a26909bb
Content-format: 50
URI-Path: cb-1.0.23.2345.78910
URI-Path: nod-e-IoTgw1
URI-Path: rbo-restart1
oneM2M-FR: cb-e-IoTplatform
oneM2M-RQI: 1779993456
Payload:
{
```

```
        "reb": true
}
```

다음은 재부팅 Execute(UPDATE) 응답 메시지 예시이다.

```
ACK 2.04 Changed [5f9f]
Token: a26909bb
oneM2M-RSC: 2004
oneM2M-RQI: 1779993456
```

7.2.4.2 <container> 리소스 대상 제어

<container>리소스를 대상으로 업데이트할 경우에 해당한다.
Execute(CREATE) 요청 메시지 처리 절차는 다음과 같다.

① CREATE 메소드는 CoAP POST 메소드에 매핑된다. CREATE 요청 메시지의 파라미터로는 다음과 같다.
② 요청자가 생성한 Message ID와 Token을 포함한다.
③ Content-format: 50(Json)
④ URI-Path 옵션: 대상 리소스 ID로서, 해당 <container>의 ID이다. 예를 들어 주기정보를 저장한 리소스를 제어할 경우, "cb-1.0.23.2345.78910" 와 cnt-period1 각각 포함한다.
⑤ oneM2M-FR 옵션: 요청 메시지를 생성한 요청자(e-IoT플랫폼)의 CSE-ID로 설정한다.
⑥ oneM2M-RQI 옵션: Request Identifier, 요청메시지 생성자인 e-IoT플랫폼이 생성한다.
⑦ oneM2M-TY 옵션: 리소스 타입 정보로서, <contentInstance>에 해당하는 4로 설정한다.
⑧ Payload: 해당 리소스의 업데이트하고자 하는 속성 이름과 변경하고자 하는 값이다.

Execute(CREATE) 응답 메시지 정보는 다음과 같다.
<contentInstance> 리소스 CREATE 요청 메시지를 수신한 e-IoT 플랫폼은 <contentInstance> 리소스의 다음의 속성들에 값을 설정해야 한다. 부모 리소스 <container>에서 허용하는 <contentInstance> 수가 1인 경우 현재 존재한 <contentInstance> 리소스를 삭제하고 새로운 것을 생성한다.
parentID, creationTime, expirationTime, lastModifiedTime

e-IoT 플랫폼에서 성공적으로 <contentInstance> 리소스를 생성한 다음 CoAP 응답 메시지를 생성하여 전달한다.
① ACK Transaction 설정한다.

② CoAP 응답 코드: 예를 들어 성공한 경우 2.01(Created).

③ Message ID와 Token를 포함한다 요청 메시지와 동일한 값이다.

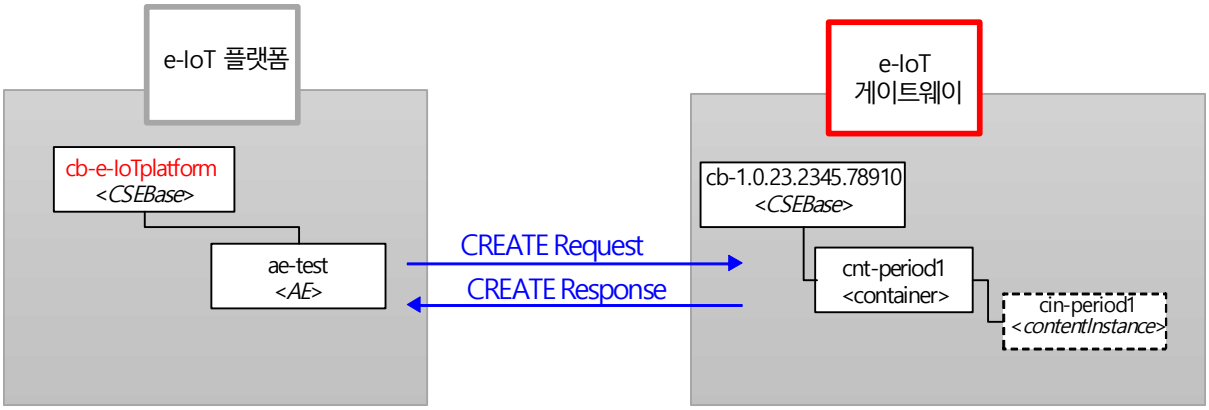
④ oneM2M-RSC 옵션: oneM2M의 응답 상태 코드를 포함한다.

⑤ oneM2M-RQI 옵션: 요청 메시지의 것과 같은 값이다.

⑥ location-Path 옵션: 마지막으로 <contentInstance>리소스가 생성된 위치 정보를 포함한다.

⑦ Payload: oneM2M의 Content 파라미터에 해당한다. 선택적으로 포함할 수 있다. 포함할 경우 갱신된 정보를 포함한다.

다음은 제어의 정보 보고 주기를 변경하는 제어 메시지 예시이다.



(그림 7-15) oneM2M <container> 리소스 대상 제어 절차 예시

다음은 정보 보고 주기 정보 <contentInstance> 리소스 생성 요청 메시지이다.

```
CON 0.02 POST [243b]
Token: ad2a4c7c
Content-format: 50
URI-Path: cb-1.0.23.2345.78910
URI-Path: cnt-period1
oneM2M-FR: cb-e-IoTplatform
oneM2M-RQI: 2085366445
oneM2M-TY: 4
Payload:
{
“cin”: { “cs”:4, “con”: 5400 }
}
```

다음은 정보 보고 주기 정보 <contentInstance> 리소스 생성 응답 메시지이다.

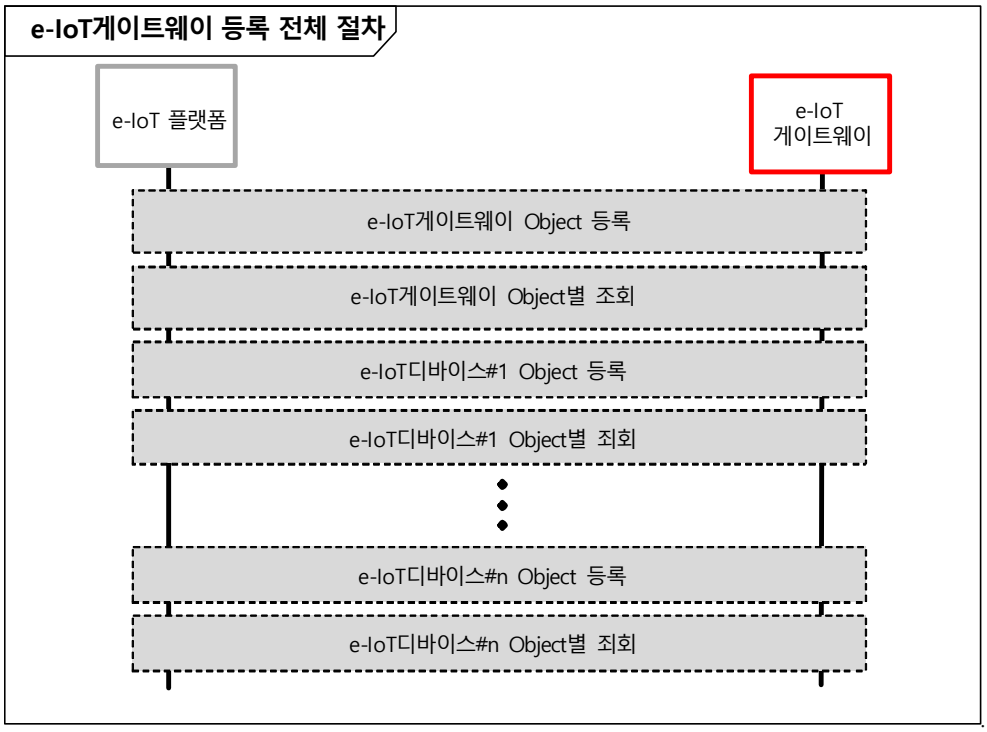
ACK 2.01 Created [243b]
Token: ad2a4c7c
oneM2M-RSC: 2001
oneM2M-RQI: 2085366445
Location-Path: cnt-periodx

7.3 IFpg-LWM2M

e-IoT 플랫폼과 e-IoT 게이트웨이 사이의 인터페이스로서, 게이트웨이 등록, 정보 조회, 제어, 펌웨어 업데이트 기능, 슬립(sleep) 노드 제어 표준을 기술한다. 특히 등록 과정은 '장치 기본등록' 기능과 '장치 단순등록' 기능으로 나뉜다.

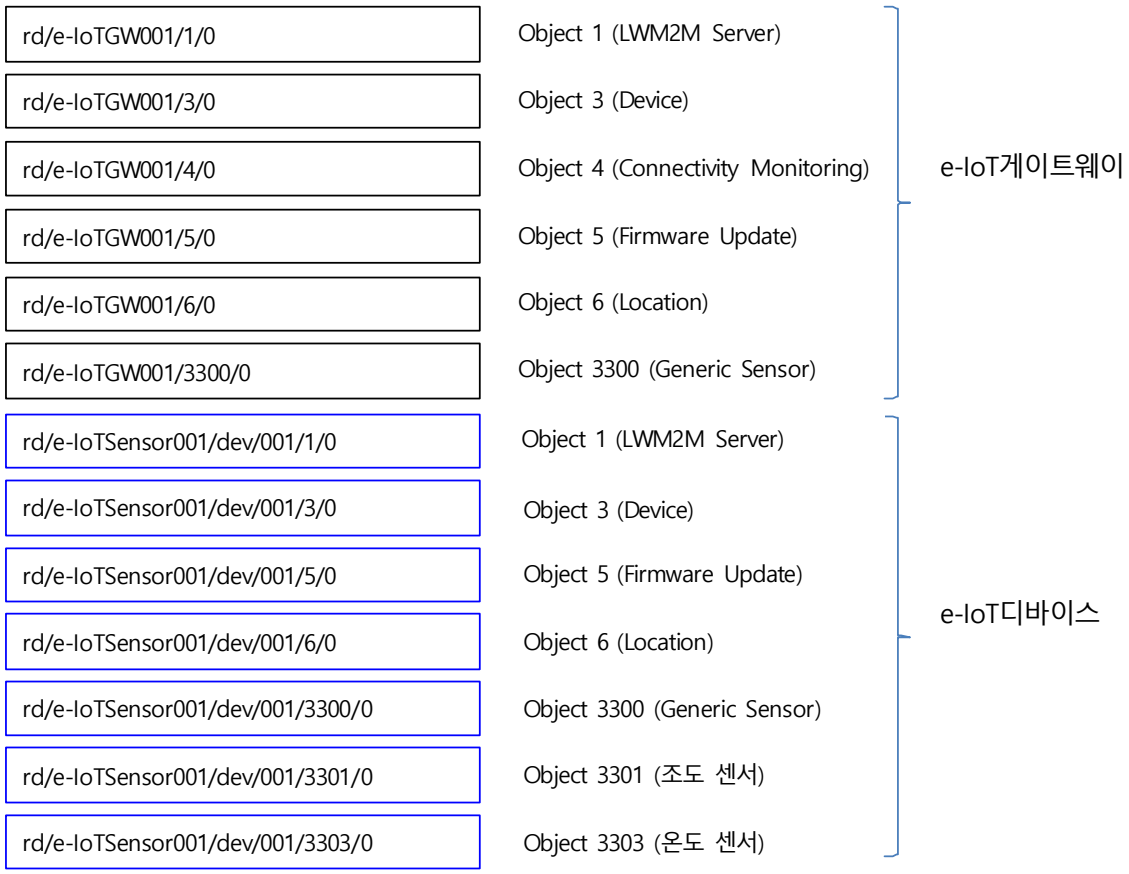
7.3.1 IFpg: 장치 등록

e-IoT 게이트웨이 등록은 게이트웨이 Resource 등록과 e-IoT 디바이스 Resource 등록으로 각각 나누어서 수행된다. 즉 e-IoT 게이트웨이는 e-IoT 디바이스를 대신하는 프록시 역할을 수행한다. e-IoT 게이트웨이와 e-IoT 디바이스의 등록하는 과정은 Object를 등록하는 과정과 각 Object별로 Resource를 조회하는 과정으로 되어 있다.
게이트웨이와 디바이스 등록함에 있어 동일한 IP 주소와 Port 번호를 통해서 등록이 수행되며, end point 이름만 차별화 하여서 등록을 수행하다. 이를 처리하는 e-IoT 플랫폼에서는 이를 다른 Object의 등록 과정으로 인식하고 처리하여야 한다.
본 절에서는 e-IoT 게이트웨이와 e-IoT 디바이스의 Object를 link-format으로 등록하는 과정을 기술하고 각 Object별 Resource를 조회하고 디바이스 Object 검색은 개념을 설명하며 좀 더 상세한 설명은 조회 절차를 참조하면 된다.



(그림 7-16) LWM2M 기반 e-IoT 게이트웨이의 등록 전체 절차도

e-IoT 게이트웨이와 e-IoT 디바이스의 정보 모델의 등록 과정을 수행한 결과 e-IoT 플랫폼에 등록된LWM2M 정보 모델의 형상은 아래 그림과 같다. e-IoT 게이트웨이 등록 시 전달되는 위치 경로(location Path) 정보와 e-IoT 디바이스 등록 시 전달되는 location Path 정보가 각각 생성되고 그 하부에 Object와 Resource의 정보가 저장된다.
e-IoT 플랫폼 내부에서는 동일한 IP 주소와 Port 번호로 등록된 장치들에 대해서는 계층적인 관계를 인식할 수 있어야 한다.



(그림 7-17) e-IoT 플랫폼에 등록될 LWM2M 기반 e-IoT 게이트웨이 정보 모델 예시

7.3.1.1 IFpg: 게이트웨이 등록

e-IoT 게이트웨이 Resource등록은 LWM2M의 등록 인터페이스를 이용한다. LWM2M 프로토콜 측면에서 e-IoT 게이트웨이가 클라이언트가 되며, e-IoT 플랫폼은 서버가 된다.

등록(Registration) 인터페이스는 다음과 같이 정의한다.

a) Interaction: e-IoT 게이트웨이(클라이언트) -> e-IoT 플랫폼(서버)
b) Confirmable Transaction으로 설정한다.
c) 메소드: POST
d) URI Template: /{+rd}{?ep, lt, lwm2m, b}
e) URI Template Variables:
① rd:= RD Function Set Path(필수). Uri-Path 옵션으로 표현한다.
② ep:= Endpoint name(필수). 반드시 해당 영역에서는 유일성이 보장되는 식별자 이어야 한다. 본 표준에서는 해당 노드의 OID이다. 최대 허용 길이는 63 bytes이다. Uri-Query 옵션으로 표현한다.
③ lt:= Lifetime(선택), 등록된 정보의 수명, 단위는 초이며 값의 범위는 60 - 42949672950이다. 이 값이 포함되지 않았을 경우 기본값은 86400이다(24시). Uri-Query 옵션으로 표현한다.

④ lwm2m:= LWM2M Version.(선택) 기본값은 1.0. Uri-Query 옵션으로 표현한다.

⑤ b:= Binding Mode. 트랜스포트 방법을 의미, “U” 는 UDP이다. Uri-Query 옵션으로 표현한다.

f) Content-Format: 40(application/link-format)

g) Payload: e-IoT 게이트웨이에 정의된 Object(Object Instance 포함)들을 포함한다.

h) 응답 코드는 다음과 같다.

① Success: 2.01 “Created”, 등록된 Resource 엔트리 정보를 포함하는 location 헤더(Location-Path 옵션)를 반드시 포함해야 한다.

- 이 위치 정보는 e-IoT 플랫폼에서 발행하는 값으로 유일성이 보장되어야 한다.

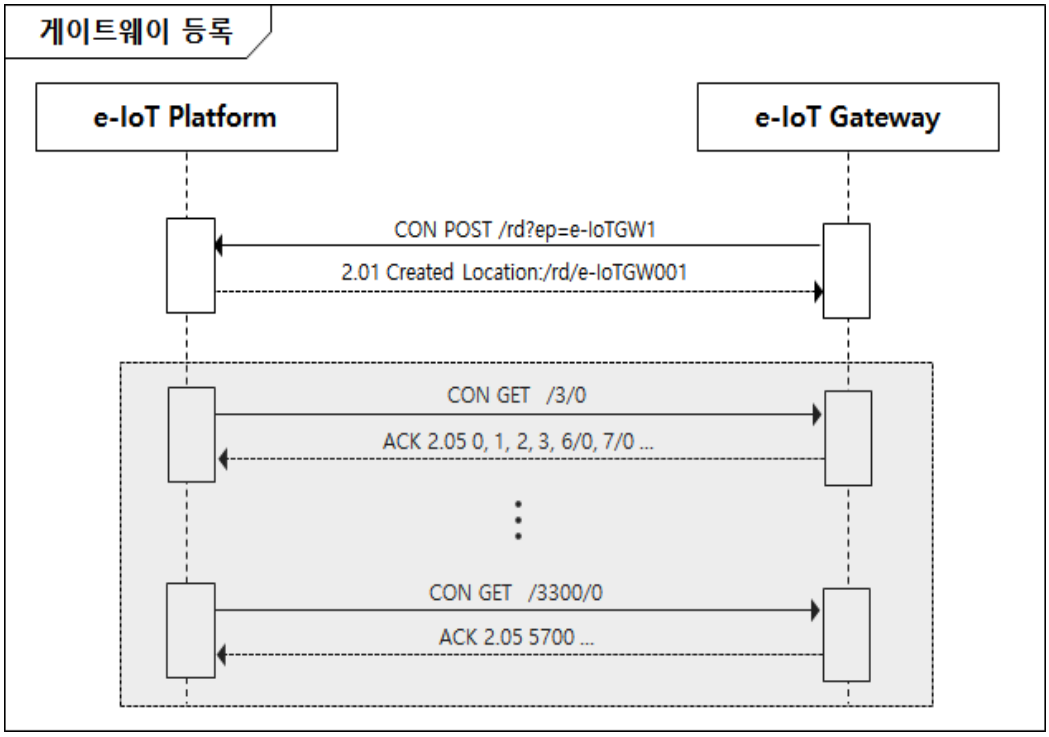
② Failure: 4.00 “Bad Request”, 잘못 정의된 Unicast 요청 메시지를 수신할 때.

③ Failure: 4.03 “Forbidden”.

등록 실패 메시지 4.00(Bad Request), 4.03(Forbidden)을 수신한 경우 e-IoT 게이트웨이는 60초 주기로 등록 요청을 시도해야 하며, 등록 실패 로그를 파일로 저장하여 한다. 더불어 LWM2M Device(3) Object의 Error Code(11) Resource에 ‘8’ 또는 ‘9’ 코드 값을 입력한다.

Object Instance의 하부에 존재하는 Resource에 대한 정보는 [10.3.다]에서 정의한 정보 조회 방법을 이용하여 획득한다.

다음의 그림은 e-IoT 게이트웨이 Resource 모델을 등록하는 과정이다. 이후 Resource 정보를 조회하는 과정을 확인할 수 있다.



(그림 7-18) LWM2M 기반 e-IoT 게이트웨이 등록 절차 예시(게이트웨이)

다음은 LWM2M 기반 e-IoT 게이트웨이 등록 요청 메시지 예시이다.

CON POST [a129] Token: 1111 Uri-Path: rd Content-Format: 40 Uri-Query: ep={e-IoT 게이트웨이OID} Uri-Query: lt=60 Uri-Query: lwm2m=1.0 Uri-Query: b=U Payload: </1/0>,</3/0>,<4/0>,<5/0>,</6/0>,</3300/0>

다음은 LWM2M 기반 e-IoT 게이트웨이 등록에 대한 응답 메시지 예시이다.

ACK 2.01 Created [a129] Token: 1111 Location-Path: rd Location-Path: e-IoTGW001

7.3.1.2 IFpg: 디바이스 등록

e-IoT 게이트웨이에 e-IoT 디바이스의 Resource가 생성되면, e-IoT 게이트웨이는 e-IoT 디바이스를 대신하여 e-IoT 디바이스 기본등록 과정을 e-IoT 플랫폼과 수행한다. LWM2M 프로토콜 측면에서 e-IoT 게이트웨이가 클라이언트가 되며, e-IoT 플랫폼은 서버가 된다.

등록(Registration) 인터페이스는 다음과 같이 정의한다.

a) Interaction: e-IoT 게이트웨이(클라이언트) -> e-IoT 플랫폼(서버)
b) Confirmable Transaction으로 설정한다.
c) 메소드: POST
d) URI Template: /{+rd}{?ep, lt, lwm2m, b}
e) URI Template Variables:
① rd:= RD Function Set Path(필수). Uri-Path 옵션으로 표현한다.
② ep:= Endpoint name(필수). 반드시 해당 영역에서는 유일성이 보장되는 식별자 이어야 한다. 본 표준에서는 해당 e-IoT 디바이스의 OID이다. 최대 허용 길이는 63 bytes이다. Uri-Query 옵션으로 표현한다.
③ lt:= Lifetime(선택), 등록된 정보의 수명, 단위는 초이며 값의 범위는 60 - 42949672950이다. 이 값이 포함되지 않았을 경우 기본값은 86400이다(24시). Uri-

Query 옵션으로 표현한다.

④ lwm2m:= LWM2M Version.(선택) 기본값은 1.0. Uri-Query 옵션으로 표현한다.

⑤ b:= Binding Mode. 트랜스포트 방법을 의미, “U”는 UDP. Uri-Query 옵션으로 표현한다.

f) Content-Format: 40(application/link-format)

① Core Link Format(RFC 6690) 형식으로 e-IoT 디바이스의 Object 리스트 만 전달한다. “/dev/###” Directory 위치 정보를 포함하고 그 하부에 Object 리스트를 위치하여서 전달하여야 한다. 여기서 “###” IoT 게이트웨이가 설정하는 숫자이다.

g) Payload: e-IoT 게이트웨이에 정의된 디바이스의 Object(Object Instance 포함) 들을 포함한다. 이때 디바이스의 Security Object는 포함되어 있지 않다.

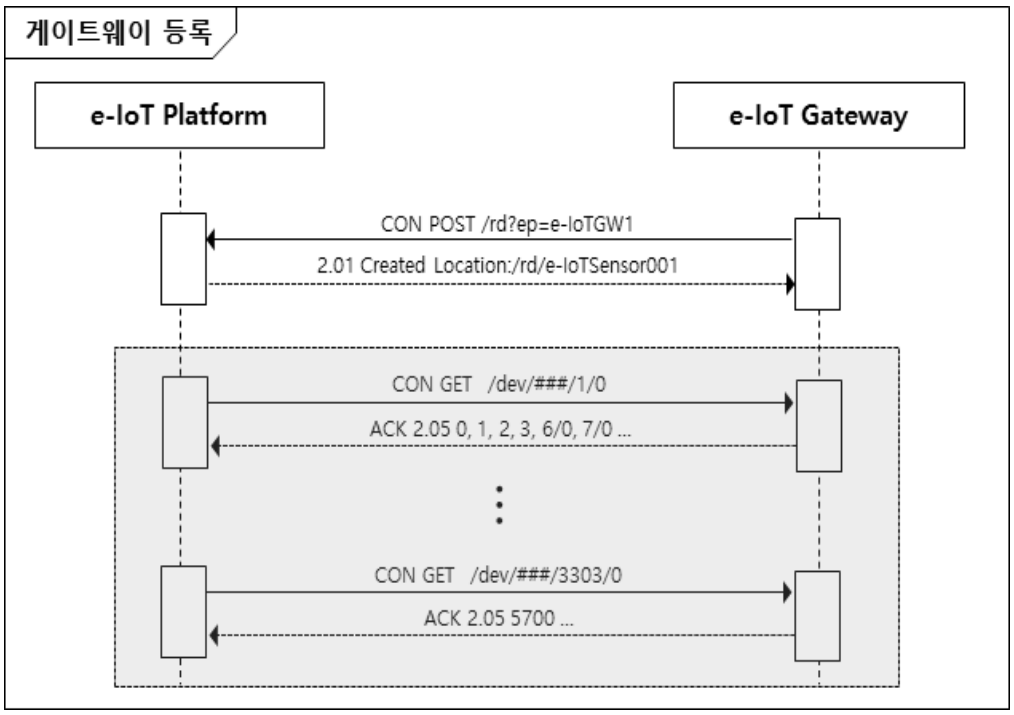
h) 응답 코드는 다음과 같다.

- Success: 2.01 “Created”, 등록된 Resource 엔트리 정보를 포함하는 location 헤더(Location-Path 옵션)를 반드시 포함해야 한다.
  - 이 위치 정보는 e-IoT 플랫폼에서 발행하는 값으로 유일성이 보장되어야 한다.
- Failure: 4.00 “Bad Request”, 잘못 정의된 Unicast 요청 메시지를 수신할 때.
- Failure: 4.03 “Forbidden”.

디바이스 등록이 실패한 경우, 즉 4.00(Bad Request), 4.03(Forbidden) 응답 메시지를 수신한 경우, 60초 주기로 등록 요청을 시도하여야 한다. 동시에 실패 로그를 파일로 저장해야 한다. 더불어 “dev/###” 하부에 위치한 Device(3) Object의 Error Code(11) Resource에 ‘8’ 또는 ‘9’ 코드 값을 입력한다.

Object Instance의 하부에 존재하는 Resource에 대한 정보는 정보 조회 방법을 이용하여 획득한다.

e-IoT 게이트웨이 Resource 모델을 등록하는 과정은 다음 그림과 같다. 이후 Resource 정보를 조회하는 과정을 확인할 수 있다.



(그림 7-19) LWM2M기반 e-IoT 게이트웨이의 디바이스 기본 등록 절차 예시

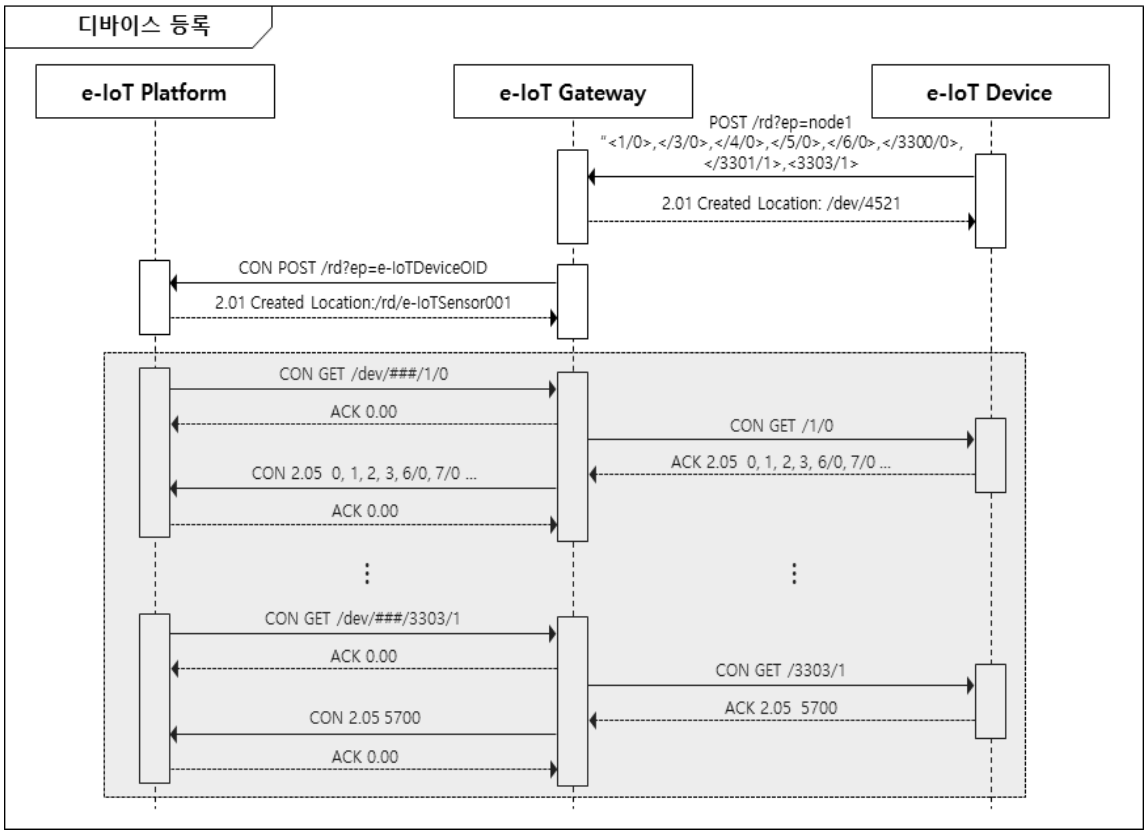
다음은 e-IoT 게이트웨이를 통한 LWM2M 기반 e-IoT 디바이스 등록 요청 메시지 예시이다.

```
CON POST [a129]
Token: 1111
Uri-Path: rd
Content-Format: 40
Uri-Query: ep={e-IoTSensorOID}
Uri-Query: lt=60
Uri-Query: lwm2m=1.0
Uri-Query: b=U
Payload:
</dev/###/1/0>,</dev/###/3/0>,</dev/###5/0>,</dev/###/6/0>,</dev/###/3301/0>,
</dev/###/3303/0>
```

다음은 e-IoT 게이트웨이를 통한 LWM2M 기반 e-IoT 디바이스 등록 요청에 대한 응답 메시지 예시이다.

```
ACK 2.01 Created [a129]
Token: 1111
Location-Path: rd
Location-Path: e-IoTSensor001
```

다음은 e-IoT 디바이스가 e-IoT 게이트웨이에 기본등록하고 이어서 e-IoT 플랫폼까지 기본 등록하는 과정이다. e-IoT 디바이스가 e-IoT 게이트웨이에 기본 등록하는 구체적인 과정은 다음 절에 기술된다.



(그림 7-20) LWM2M기반 e-IoT 게이트웨이의 디바이스 기본 등록 절차 예시

7.3.1.3 IFpg: 게이트웨이 등록 업데이트

e-IoT 게이트웨이 Resource 등록 업데이트는 LWM2M의 등록 인터페이스를 이용한다. LWM2M 프로토콜 측면에서 e-IoT 게이트웨이가 클라이언트가 되며, e-IoT 플랫폼은 서버가 된다.

a) Interaction: e-IoT 게이트웨이(클라이언트) -> e-IoT 플랫폼(서버)
b) Confirmable Transaction 으로 설정한다.
c) 메소드: POST
d) URI Template: /{+location}{?lt, b}
e) URI Template Variables:
   ① Location:= 앞서 수행한 등록과정에서 e-IoT 게이트웨이가 응답으로 수신한 경로 정보이다. Uri-Path 옵션으로 표현한다.
   ② lt:= Lifetime, 등록된 정보의 수명, 최근 등록된 Lifetime값과 비교하여 변경되었을 때 제공한다. 단위는 초이며 값의 범위는 60 - 42949672950이다. 이 값이 포함되지 않았을 경우 기본값은 86400이다(24시). Uri-Query 옵션으로 표현한다.

③ b:= Binding Mode. 트랜스포트 방법을 의미하며 최근 등록된 Binding Mode와 비교하여 변경되었을 때 제공한다 “U” 는 UDP. Uri-Query 옵션으로 표현한다.

f) 6) Content-Format: 40(application/link-format) (선택), Payload가 있는 경우 포함한다.

g) 7) Payload: 수정된 Resource 데이터가 있는 경우 포함한다. 수정된 Resource 데이터뿐만 아니라 모든 Resource를 포함한다.

h) 8) 응답 코드는 다음과 같다.

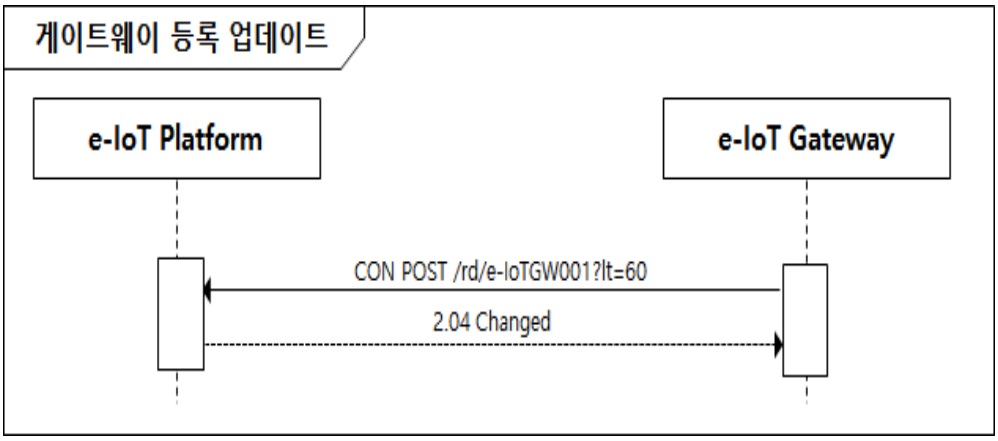
① Success: 2.04 “Changed”

② Failure: 4.00 “Bad Request”, 잘못 정의된 Unicast 요청 메시지를 수신할 때.

③ Failure: 4.04 “Not Found”.

실패 메시지 4.00(Bad Request), 4.04(Not Found) 메시지를 수신한 경우 등록 모드로 전환하여 게이트웨이 등록 과정을 다시 수행한다.

아래 그림의 예시는 앞에서 등록한 e-IoT 게이트웨이 Resource를 기본 등록 업데이트하는 과정이다.



(그림 7-21) LWM2M 기반 e-IoT 게이트웨이 등록 업데이트 절차 예시(게이트웨이)

CON POST [349a] Token:2222 Uri-Path: rd Uri-Path: e-IoTGW001
ACK 2.04 Changed Token:2222

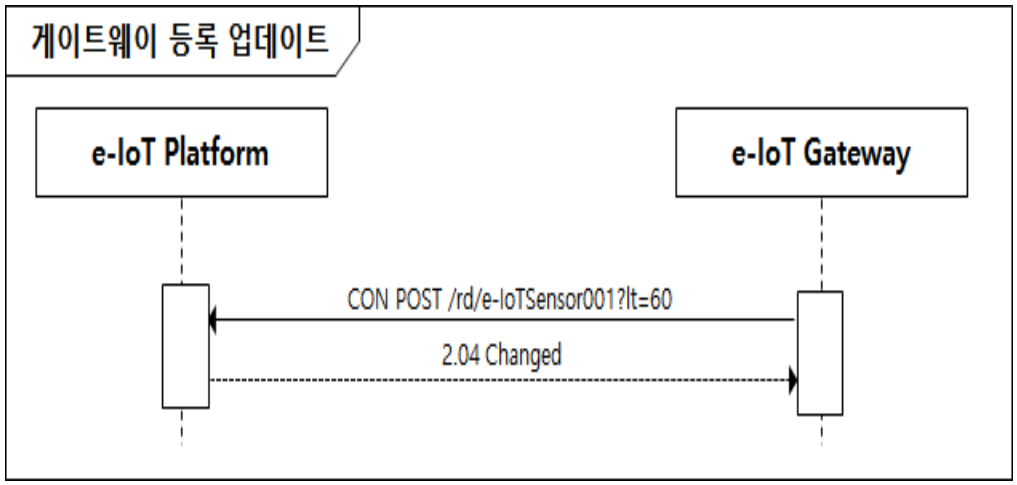
7.3.1.4 IFpg: 디바이스 등록 업데이트

e-IoT 디바이스 등록업데이트는 e-IoT 게이트웨이가 LWM2M의 등록 인터페이스를 이용해서 대신 수행한다. IFgd 인터페이스를 통해서 수행되는 e-IoT 디바이스 등록 업데이트 과정에 의해서 트리거 될 수도 있으며, 혹은 e-IoT 게이트웨이 단독으로 LWM2M 프로토

콜 측면에서 e-IoT 게이트웨이가 클라이언트가 되며, e-IoT 플랫폼은 서버가 된다.

a) Interaction: e-IoT 게이트웨이(클라이언트) -> e-IoT 플랫폼(서버)
b) Confirmable Transaction 으로 설정한다.
c) 메소드: POST
d) URI Template: /{+location}{?lt, b}
e) URI Template Variables:
   ① Location:= 앞서 수행한 등록과정에서 e-IoT 게이트웨이가 응답으로 수신한 경로 정보이다. Uri-Path 옵션으로 표현한다.
   ② lt:= Lifetime, 등록된 정보의 수명, 최근 등록된 Lifetime값과 비교하여 변경 되었을 때 제공한다. 단위는 초이며 값의 범위는 60 - 42949672950이다. 이 값이 포함되지 않았을 경우 기본값은 86400이다(24시). Uri-Query 옵션으로 표현한다.
   ③ b:= Binding Mode. 트랜스포트 방법을 의미, 최근 등록된 Binding Mode와 비교 하여 변경되었을 때 제공한다. "U" 는 UDP. Uri-Query 옵션으로 표현한다.
f) Content-Format: 40(application/link-format) (선택), Payload가 있는 경우 포함한다.
g) Payload: 수정된 Resource 데이터가 있는 경우 포함한다. 수정된 Resource 데이터 뿐만 아니라 모든 Resource를 포함한다.
h) 응답 코드는 다음과 같다.
   ① Success: 2.04 "Changed"
   ② Failure: 4.00 "Bad Request", 잘못 정의된 Unicast 요청 메시지를 수신할 때.
   ③ Failure: 4.04 "Not Found".

실패 메시지 4.00(Bad Request), 4.04(Not Found)를 수신한 경우 등록 모드로 전환하여 디바이스 등록 과정을 다시 수행한다.
다음의 그림은 앞에서 등록한 e-IoT 디바이스 Resource를 기본 등록 업데이트하는 과정이다.



(그림 7-22) LWM2M 기반 e-IoT 게이트웨이 등록 업데이트 절차 예시(디바이스)

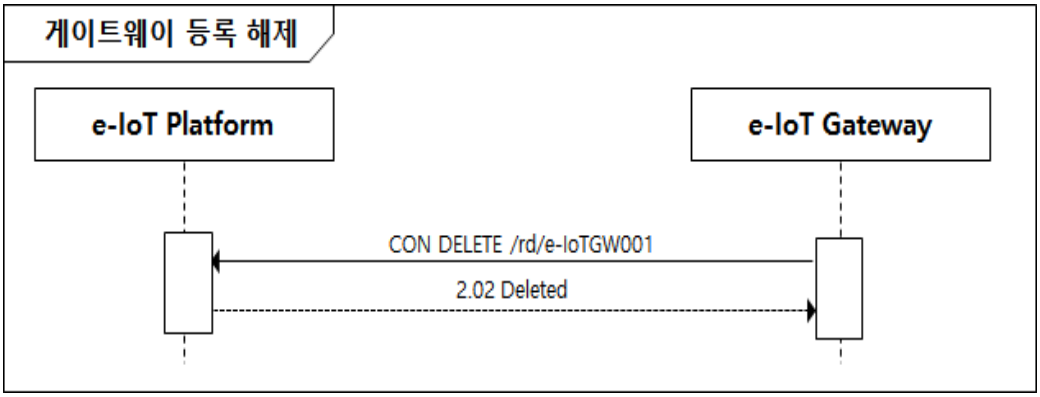
CON POST [349a] Token:2222 Uri-Path: rd Uri-Path: e-IoTsensor001
ACK 2.04 Changed Token:2222

7.3.1.5 IFpg: 게이트웨이 등록 해제

e-IoT 게이트웨이 등록 해제는 LWM2M의 등록 인터페이스를 이용한다. LWM2M 프로토콜 측면에서 e-IoT 게이트웨이가 클라이언트가 되며, e-IoT 플랫폼은 서버가 된다. 등록(Registration) 해제 과정은 다음과 같이 정의한다.

a) Interaction: e-IoT 게이트웨이(클라이언트) -> e-IoT 플랫폼(서버)
b) 메소드: DELETE
c) URI Template: /{location} e-IoT 게이트웨이 등록 시 응답 메시지로 등록된 위치
d) 응답 코드는 다음과 같다.
   ① Success: 2.02 "Deleted"
   ② Failure: 4.04 "Not Found", 등록 해제 요청한 위치

다음의 그림은 LWM2M 기반 등록 해제 과정의 예시이다.



(그림 7-23) LWM2M 기반 e-IoT 게이트웨이 등록 해제 절차 예시(게이트웨이)

다음은 LWM2M 기반 등록 해제 요청 메시지 예시이다.

CON DELETE [12fe] Token: 1111 Uri-Path: rd Uri-Path: e-IoTGW001

다음은 LWM2M 기반 등록 해제 응답 메시지 예시이다.

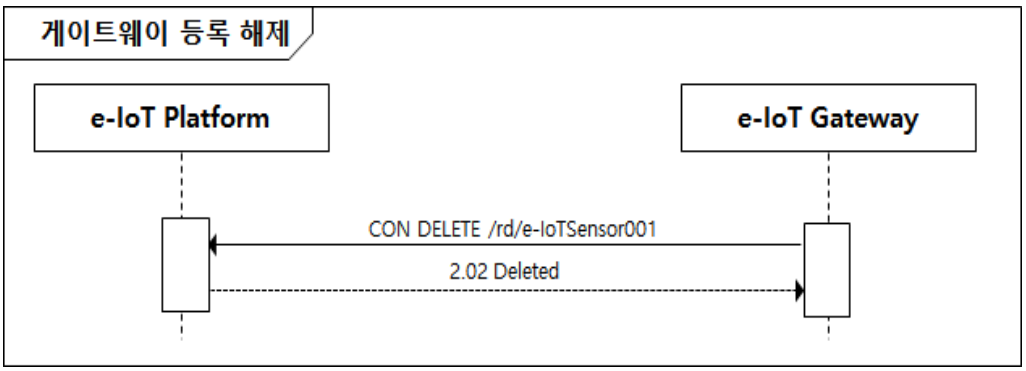
CON 2.02 Deleted [12fe] Token: 1111

7.3.1.6 IFpg: 디바이스 등록 해제

e-IoT 디바이스 등록 해제는 LWM2M의 등록 인터페이스를 이용해서 e-IoT 게이트웨이가 대신 수행한다. e-IoT 게이트웨이에 등록된 e-IoT 디바이스의 등록 정보가 삭제된 경우에 디바이스 등록 해제 기능이 수행된다. LWM2M 프로토콜 측면에서 e-IoT 게이트웨이가 클라이언트가 되며, e-IoT 플랫폼은 서버가 된다.
등록(Registration) 해제 과정은 다음과 같이 정의한다.

a) Interaction: e-IoT 게이트웨이(클라이언트) -> e-IoT 플랫폼(서버)
b) 메소드: DELETE
c) URI Template: /{location} e-IoT 게이트웨이 등록 시 응답 메시지로 등록된 위치
d) 응답 코드는 다음과 같다.
   ① Success: 2.02 “Deleted”
   ② Failure: 4.04 “Not Found”, 등록 해제 요청한 위치

다음의 그림은 LWM2M 기반 등록 해제 과정의 예시이다.



(그림 7-24) LWM2M 기반 e-IoT 게이트웨이 등록 해제 절차 예시(디바이스)

다음은 LWM2M 기반 등록 해제 요청 메시지 예시이다.

CON DELETE [12fe] Token: 1111 Uri-Path: rd Uri-Path: e-IoTSensor001

다음은 LWM2M 기반 등록 해제 응답 메시지 예시이다.

CON 2.02 Deleted [12fe] Token: 1111

7.3.2 IFpg: 주기적 보고

7.3.2.1 보고 주기 및 관련 속성값 설정

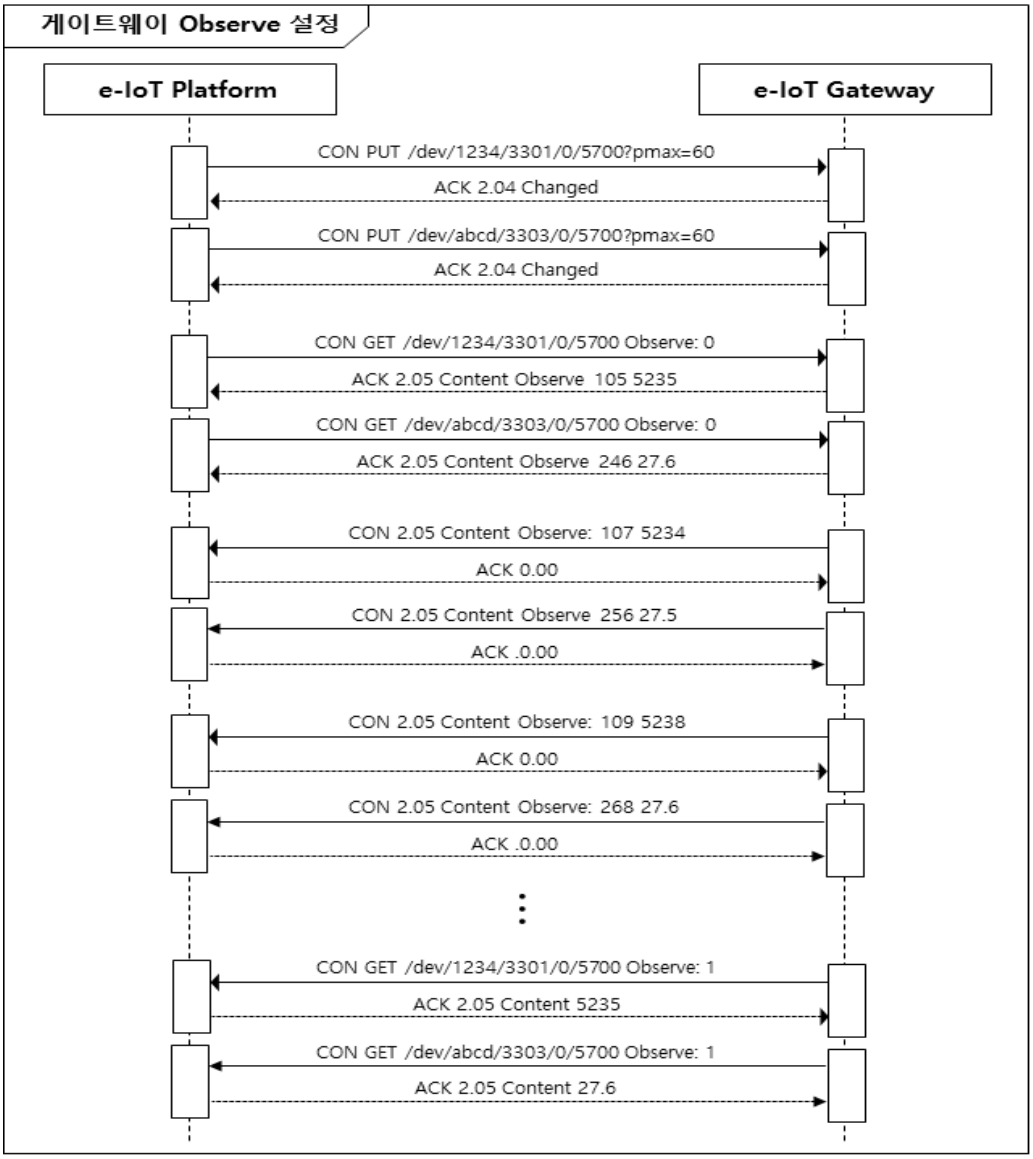
LWM2M의 Write Attribute 기술을 이용하여 보고 주기 값에 해당하는 대상 Resource의 pmax 속성과 보고 관련 “pmin, gt, lt, ntype” 속성을 설정하는 절차이다. 설정 요청 메시지는 다음과 같이 구성한다. 보고 주기가 필요한 모든 정보 모델에 이 과정을 수행해야 한다. 예를 들어 e-IoT 게이트웨이에 연결되어 있는 센서의 측정값이 보고되어야 하기 때문에 모든 /dev/####로 연결되어 있는 센서들의 측정 Resource에 pmax 속성을 설정한다.

a) CoAP PUT 메소드로 매핑한다.
b) 생성한 Message ID와 Token을 포함한다.
c) 대상 Resource를 Uri-Path 옵션으로 표현한다.
d) Uri-Query 옵션을 이용해서 설정하고자 하는 속성값을 표현한다.(pmax={주기값})

보고 주기 값 설정 요청 메시지를 수신하고 해당 속성을 변경한 후, 다음과 같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
    1) 성공: 2.04 Changed
    2) 실패: 4.00 Bad Request, 4.04 Not Found.
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값

플랫폼으로부터 수신한 “pmax”, “pmin”, “gt”, “lt”, “ntype” 설정 메시지를 수신하면 해당 Object, 혹은 Resource의 “pmax“, “pmin”, “gt”, “lt”, “ntype” 값을 변경한다. 플랫폼의 설정 변경 요청이 없는 경우 Server Object(1)의 최대 주기 Resource(3) 값과 최소 주기 Resource(2) 값을 참조한다. Server Object(1)의 최대 주기 Resource(3) 값 즉, 1/0/3 의 기본 값을 1시간(3600)으로 하고, 최소 주기 Resource(2) 값 즉, 1/0/2의 기본 값을 1분(60)으로 한다. “ntype”의 설정 변경 요청이 없는 경우 “true”(Confirmable)를 기본값으로 한다. 동일한 설정 메시지의 타겟 URL이 “/dev/###“를 포함한 디바이스의 Resource라면 먼저 게이트웨이에 저장되어 있는 디바이스의 정보모델들의 속성값을 변경하고 디바이스의 정보 모델 속성을 변경하는 절차를 수행한다.



(그림 7-25) LWM2M 기반 e-IoT 게이트웨이 Observe 설정 및 주기적 보고 예시

다음은 보고 주기 설정 요청 메시지의 예시이다. 연결되어 있는 조도 센서의 측정값을 60 초 주기로 보고하도록 설정하는 예시이다.

```
CON PUT [251ee]
Token: 2191
Uri-Path: dev
Uri-Path: 1234
Uri-Path: 3301
Uri-Path: 0
Uri-Path: 5700
Uri-Query: pmax=60
```

다음은 보고 주기 설정 요청에 대한 응답 메시지 예시이다.

ACK 2.04 Changed [251ee] Token: 2191

7.3.2.2 CoAP Observe 설정 및 주기 보고

대상 데이터 Object Instance와 Resource에 Observe를 요청하고 요청이 수락되면 주기마다 정보를 보고하는 절차이다. Observe를 하는 대상이 Resource인 경우는 주기적 보고가 필요한 모든 Resource에 이 과정을 수행해야 한다. 각 Object별로 “lt”, “gt” 값이 있는 경우는 센싱된 값이 “lt” 값보다 작을 경우와 “gt” 값보다 큰 경우에 보고 CoAP 메시지를 생성해서 보고한다. 이때 최소 “pmin” 값 시간만큼 지난 후에 전달한다.
e-IoT 플랫폼에서는 CoAP Observe 설정 요청 메시지는 다음과 같이 구성한다.

a) Conformable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.
c) 생성한 Message ID와 Token을 포함한다
d) 대상 Resource를 Uri-Path 옵션으로 표현한다.
e) Observe 옵션: 0
f) Accept 옵션:
  ① Observe 대상이 Resource인 경우: Plain-text(0) 또는 JSON(50)
  ② Observe 대상이 Object Instance인 경우: JSON(50)

e-IoT 게이트웨이는 Observe 요청 메시지를 수신하고 Observe를 설정하고 다음과 같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
  1) 성공: 2.05 Content
  2) 실패: 4.00 Bad Request 또는 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값.
d) Observe 옵션: 0보다 큰 값
e) Content-format: 요청 메시지의 Accept 옵션의 값이다.
f) Payload: 해당 데이터 Resource의 값을 포함한다.

다음은 e-IoT 게이트웨이가 주기적 보고 메시지를 다음과 같이 구성하고 전달한다.

a) CON Transaction 설정한다.
b) CoAP 코드: 2.05 Content
c) Message ID: 임의로 설정한다.
d) Token 값은 Observe 설정 요청 메시지의 것과 동일한 값으로 설정한다.

e) Observe 옵션: 이전 보다 증가한 값
f) Content-format: Observe 요청 메시지의 Accept 옵션의 값
g) Payload: 해당 데이터 Resource의 값을 포함한다.

e-IoT 플랫폼은 주기적 보고 메시지를 수신하고 다음과 같이 응답 메시지를 구성하고 전달한다.
a) ACK Transaction 설정한다.
b) CoAP 응답 코드
    1) 성공: 0.00 empty message
    2) 실패: 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.

다음은 연결되어 있는 e-IoT 디바이스의 조도 센서 측정값(dev/1234/3301/0/5700)에 Observe를 설정하는 요청 메시지의 예시이다.

```
CON GET [2acbf]
Token: 2339
Uri-Path: dev
Uri-Path: 1234
Uri-Path: 3301
Uri-Path: 0
Uri-Path: 5700
Observe: 0
Accept: 0
```

다음은 Observe 설정 요청에 대한 응답 메시지 예시이다.

```
ACK 2.05 Content [2acbf]
Token: 2339
Observe: 105
Content-format: 0
Payload: 5235
```

다음은 Observe가 설정된 조도 센서의 측정값(3301/0/5701) Resource의 주기적인 보고 메시지 예시이다. 아래 예시에서 Token 값은 앞의 메시지와 동일함을 확인할 수 있다.

```
CON 2.05 Content [24f13]
Token: 2339
Observe: 107
Content-format: 0
```

```
Payload:
5234
```

다음은 Observe가 설정된 조도 센서의 측정값(3301/0/5701) Resource의 주기적인 보고 메시지에 대한 응답 메시지 예시이다. 아래 예시에서 Token 값은 앞의 메시지와 동일함을 확인할 수 있다.

```
ACK 0.00 empty [24f13]
Token: 2339
Payload:
```

7.3.2.3 보고 주기 해제

CoAP Observe기반 주기적 보고의 해제는 CoAP의 Observe 옵션을 이용한다.
e-IoT 플랫폼에서는 CoAP Observe 설정 요청 메시지는 다음과 같이 구성한다.

a) Conformable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.
c) 생성한 Message ID와 Token을 포함한다
d) 대상 Resource를 Uri-Path 옵션으로 표현한다.
e) Observe 옵션: 1
f) Accept 옵션:
  ① Observe 대상이 Resource인 경우: Plain-text(0) 또는 JSON(50)
  ② Observe 대상이 Object Instance인 경우: JSON(50)

다음은 연결되어 있는 e-IoT 디바이스의 조도 센서 측정값(dev/1234/3301/0/5700)에 Observe를 해제하는 요청 메시지의 예시이다.

```
CON GET [2acbf]
Token: 2339
Uri-Path: dev
Uri-Path: 1234
Uri-Path: 3301
Uri-Path: 0
Uri-Path: 5700
Observe: 1
Accept: 0
```

e-IoT 게이트웨이는 Observe 해제 요청 메시지를 수신하고 Observe를 해제하고 다음과

같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
   1) 성공: 2.05 Content
   2) 실패: 4.00 Bad Request 또는 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값.
d) Observe 옵션: 반드시 제외하여야 한다.
e) Content-format: 요청 메시지의 Accept 옵션의 값
f) Payload: 해당 데이터 Resource의 값을 포함한다.

다음은 Observe 해제 요청에 대한 응답 메시지 예시이다.

ACK 2.05 Content [2acbf] Token: 2339 Content-format: 0 Payload: 5235

7.3.3 IFpg: 정보 조회

LWM2M 기반 e-IoT 게이트웨이의 정보를 조회하는 기능은 LWM2M의 Device Management & Service Enablement Interface를 이용한다. 정보 조회에는 해당 Object와 Resource의 값을 읽어 오는 Read 동작과, 해당 Object와 Resource에 속해 있는 속성값을 읽어오거나 해당 Object에 구현되어 있는 Resource를 조회하는 Discover 동작으로 구분된다.
Read 동작은 정의한 정보 모델 형식을 조회한다. 즉, {Object ID}/{Object Instance ID}/{Resource ID}/{Resource Instance ID} 경로로 정의되어 있는 정보 모델을 CoAP GET 메소드를 이용하여 조회한다. 이때 응답 메시지의 Payload 형식은 JSON을 따른다.
Discover 동작은 Read 동작과 달리 link-format Accept 옵션을 포함한 CoAP GET 메소드를 이용하여 요청한다. 응답 메시지의 Payload 형식은 link-format 형식을 따른다.

7.3.3.1 정보 조회 - Read

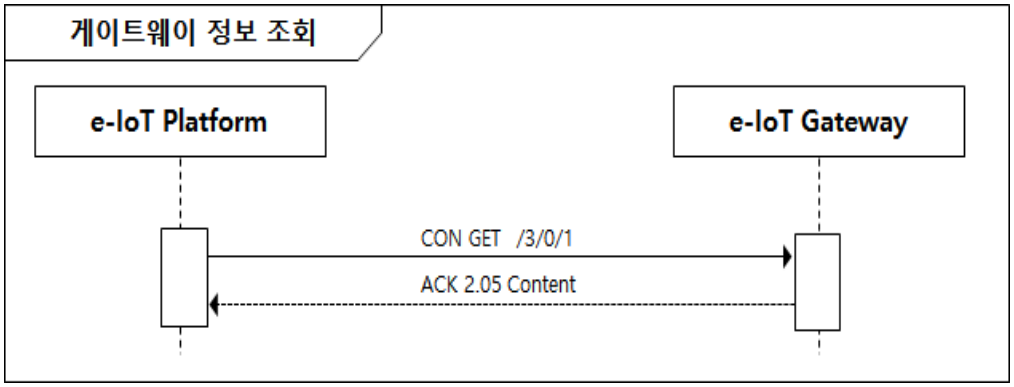
e-IoT 게이트웨이의 원하는 정보 모델을 조회하는 요청 메시지는 다음과 같이 구성한다.

a) Confirmable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 조회하고자 하는 대상 Resource를 Uri-Path 옵션으로 표현한다. 대상 Resource는 LWM2M의 정보모델이다.
e) Accept 옵션을 통해서 응답 받을 정보의 형식을 JSON으로 설정한다.

정보 조회를 요청한 해당 Resource의 존재 유무를 판단하고 존재하는 경우 해당 값을 포함하여 다음과 같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정
b) CoAP 응답 코드
   1) 성공: 2.05 Content
   2) 실패: 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
d) Content-format: LWM2M JSON을 의미하는 11543
e) Payload: JSON 형식으로 해당 Resource의 값을 포함한다.

다음은 e-IoT 게이트웨이의 모델 번호를 조회하는 과정이다.



(그림 7-26) LWM2M 기반 e-IoT 게이트웨이 정보 조회 과정

다음은 e-IoT 게이트웨이의 모델 번호를 조회하는 과정의 GET 메시지 예시이다.

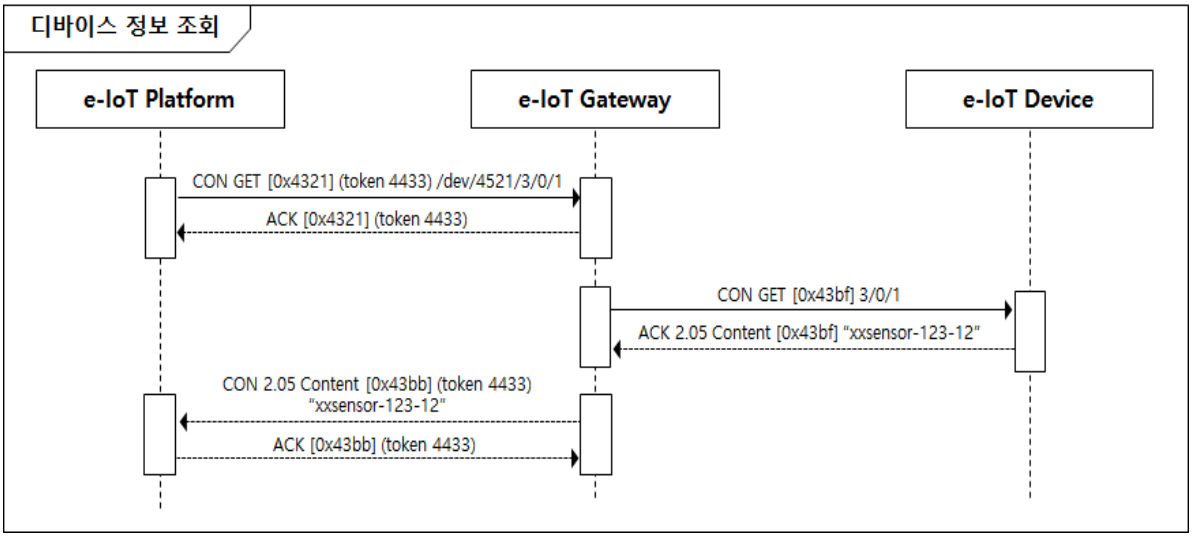
```
CON GET [24322]
Token: 4444
Uri-Path: 3
Uri-Path: 0
Uri-Path: 1
Accept: 50
```

LWM2M 기반으로 e-IoT 게이트웨이 모델 번호를 조회하는 요청에 대한 응답 메시지 예시이다.

```
ACK 2.05 Content [24322]
Token: 4444
Content-format: 11543
Payload
```

```
{ “e”:[
      {“n”:“”, “sv”:“xxcom-1234-12”} ]
}
```

다음은 e-IoT 플랫폼에서 e-IoT 디바이스 정보를 조회하는 경우이다. 위와 같이 e-IoT 플랫폼이 e-IoT 게이트웨이의 정보를 조회 하는 경우와 동일하나, 정보 모델의 경로에 차이가 있다. Object와 Resource 경로에 앞에 e-IoT 디바이스를 의미하는“dev/4521/” 경로가 추가되어 있어, e-IoT 게이트웨이는 e-IoT 디바이스의 정보임을 인식하고, e-IoT 디바이스에 조회하여 응답한다. 이때 Separated ACK 기술을 이용해서 먼저 Empty ACK로 응답한 다음에 e-IoT 디바이스로부터 정보 조회를 수행한 결과를 Content 메시지를 이용해서 다시 e-IoT 플랫폼에 전달한다.



(그림 7-27) LWM2M 기반 e-IoT 플랫폼의 e-IoT 디바이스 정보 조회 과정

IFpg 인터페이스 구간에서 먼저 e-IoT 플랫폼이 e-IoT 디바이스의 모델 번호 정보를 조회하는 요청 메시지를 다음과 같이 생성한다.

a) Confirmable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 조회하고자 하는 대상 Resource를 Uri-Path 옵션으로 표현한다. 대상 Resource는 LWM2M의 정보 모델이다.
f) Accept 옵션을 통해서 응답 받을 정보의 형식을 JSON으로 설정한다.

다음 메시지 예시는 e-IoT 플랫폼에서 생성한 e-IoT 디바이스의 모델 번호를 조회하기 위한 요청 메시지이다.

```
CON GET [4321]
Token: 4433
Uri-Path: dev
Uri-Path: 4521
Uri-Path: 3
Uri-Path: 0
Uri-Path: 1
Accept: 50
```

e-IoT 게이트웨이는 정보 조회를 요청한 해당 Resource가 하부에 연결되어 있는 e-IoT 디바이스의 것임을 판단하고 바로 응답할 수 없기 때문에 Confirmable 요청 메시지에 대한 응답 메시지를 생성하여 e-IoT 플랫폼에 응답한다. 이 단순 ACK 응답 메시지에는 실제 값은 없다. 단순 ACK 응답 메시지는 다음과 같이 생성한다.

a) ACK Transaction 설정한다.
b) CoAP 코드는 0.00 으로 설정한다.
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값으로 설정한다.

다음은 e-IoT 디바이스의 Resource 정보 조회 요청에 대한 Confirmable 요청 메시지에 대한 단순 ACK 응답 메시지의 예시이다.

```
ACK 0.00 [4321]
Token: 4433
```

e-IoT 게이트웨이는 단순 ACK 메시지를 e-IoT 플랫폼에 응답함과 동시에 e-IoT 디바이스에 해당 Resource의 정보를 조회하는 요청 메시지를 다음과 같이 생성해서 전송한다.

a) Transaction 설정한다.
b) GET 메소드로 매핑한다.
c) Message ID와 Token을 각각 e-IoT 게이트웨이가 생성하여 포함한다.
d) 조회하고자 하는 대상 Resource를 Uri-Path 옵션으로 표현한다. LWM2M Object 이후의 경로만을 포함한다.

e-IoT 게이트웨이가 e-IoT 디바이스의 해당 Resource의 정보를 조회하기 위한 요청 메시지의 예시이다.

```
CON GET [43bf]
Token: 2789
Uri-Path: 3
Uri-Path: 0
```

Uri-Path: 1

e-IoT 디바이스는 정보 조회를 요청한 해당 Resource의 존재 유무를 판단하고 존재하는 경우 해당 값을 포함하여 다음과 같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
   1) 성공: 2.05 Content
   2) 실패: 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
d) Content-format: 요청 메시지 Accept 옵션 값, Plain-text(0)를 기본으로 한다.
e) Payload: 해당 Resource의 값을 포함한다.

e-IoT 디바이스의 모델 번호를 조회에 대한 응답 메시지의 예시

ACK 2.05 Content [43bf] Token: 2789 Content-format: 0 Payload: xxsensor-123-12

e-IoT 게이트웨이는 원하는 정보를 e-IoT 디바이스로부터 수신함으로써 해당 정보를 추출하고 다시 요청하는 형태로 변환하여 Payload를 구성하고 나머지 CoAP 메시지를 아래와 같이 구성하여 e-IoT 플랫폼에 전달한다.

a) CON Transaction 설정한다.
b) CoAP 응답 코드: 수신한 응답 코드와 동일하게
c) Message ID를 새롭게 생성하여 포함시킨다.
d) Token 값은 원래 e-IoT 플랫폼의 조회 요청 메시지의 것과 동일하게 설정한다.
e) Content-format: e-IoT 플랫폼의 조회 요청 메시지의 Accept 옵션 값
f) Payload: Resource의 값을 포함한다.

다음은 e-IoT 게이트웨이가 e-IoT 디바이스의 해당 Resource 정보를 담은 CoAP 메시지 예시이다.

CON 2.05 Content [43bb] Token: 4433 Content-format: 11543 Payload: { “e”: [

```
        {“n”: “”, “sv”: “xxsensor-1234-12”} ]
}
```

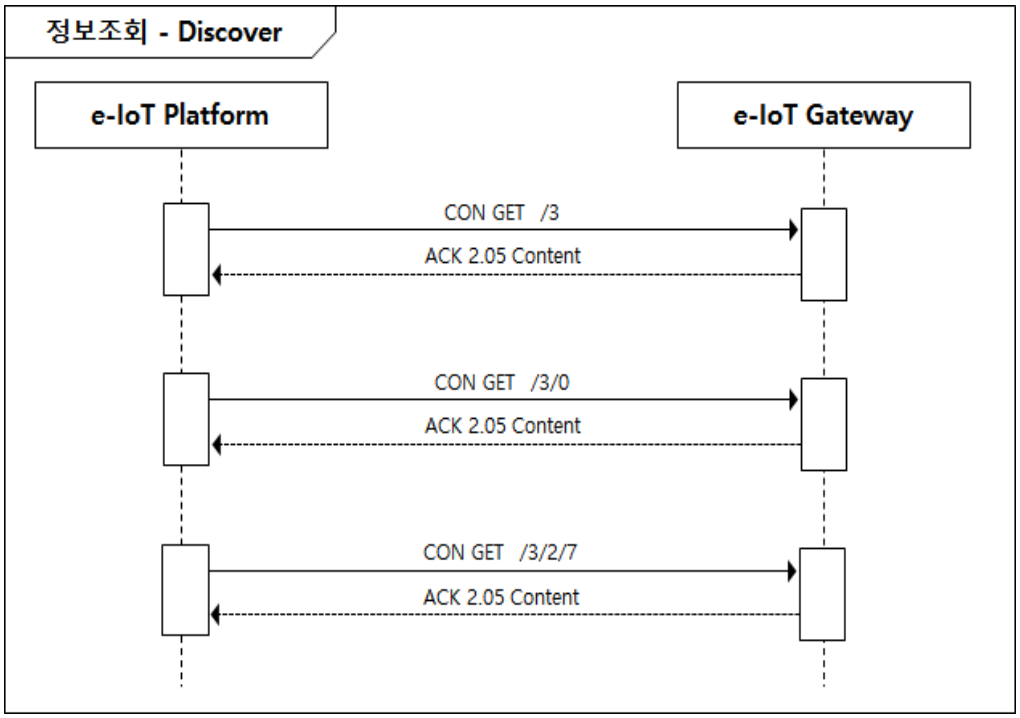
e-IoT 플랫폼은 조회 요청한 정보를 수신하고 이에 대한 단순 ACK 메시지를 아래와 같이 생성하여 전달한다.

```
ACK 0.00 [43bb]
Token: 4433
```

a) ACK Transaction 설정한다.
b) CoAP 코드는 0.00 으로 설정한다.
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값으로 설정한다.

7.3.3.2 정보 조회 - Discover

타겟 정보가 Object인 경우는 해당 Object에 구현되어 있는 Object Instance 리스트와 각 Object Instance에 구현되어 있는 Resource 리스트를 반환하며, 각 Object, Object Instance, Resource에 정의되어 있는 속성값을 반환한다.
타겟 정보가 Object Instance인 경우 해당 Object Instance에 구현되어 있는 Resource 리스트와 함께 각각에 정의되어 있는 속성값을 함께 반환한다.
타겟 정보가 Resource인 경우 해당 Resource에 정의되어 있는 속성값을 반환한다.
다음은 정보 조회-Discover하는 과정의 예시이다.



(그림 7-28) 정보 조회-Discover 예시

Discover할 대상이 3번 Object인 경우 요청 메시지이다. 해당 경로 정보를 Uri-Path 옵션으로 표시하고, Accept 옵션에 link-format(40)를 표시한다.

```
CON GET [24322]
Token: 4444
Uri-Path: 3
Accept: 40
```

3번 Object에 대한 Discover 응답 메시지이다. 3 번 Object에 정의된 속성값, 3번 Object에 구현된 Object Instance 리스트와 각각에 정의된 속성값이 포함된다. 다음으로 구현된Resource 들의 리스트와 각 속성값을 포함한다. Content-format 옵션에 link-format(40)을 설정하고 Payload에 다음과 같이 link-format으로 정보를 포함한다.

```
ACK 2.05 Content [24322]
Token: 4444
Content-format: 40
Payload
{
    </3>;pmin=10, </3/0>;pmax=60, </3/0/1>, <3/0/2>, </3/0/3>, </3/0/4>,
    <3/0/6>;dim=8, <3/0/7>;dim=8; gt=50;lt=42.2,<3/0/8>;dim=8,
    <3/0/11>,<3/0/16>
}
```

Discover할 대상이 3번 Object와 0번 Object Instance인 경우 요청 메시지이다. 해당 경로 정보를 Uri-Path 옵션으로 표시하고, Accept 옵션에 link-format(40)를 표시한다.

```
CON GET [24323]
Token: 4445
Uri-Path: 3
Uri-Path: 0
Accept: 40
```

3/0 Object Instance에 대한 Discover 응답 메시지이다. 3/0 Object Instance에 정의된 속성값, 구현된 Resource 들의 리스트와 각 속성값을 포함한다. Content-format 옵션에 link-format(40)을 설정하고 Payload에 다음과 같이 link-format으로 정보를 포함한다.

```
ACK 2.05 Content [24323]
Token: 4445
Content-format: 40
Payload
{
    </3/0>;pmax=60, </3/0/1>, <3/0/2>, </3/0/3>, </3/0/4>, <3/0/6>;dim=8,
    <3/0/7>;dim=8; gt=50;lt=42.2, <3/0/8>;dim=8,<3/0/11>,<3/0/16>
}
```

Discover할 대상이 3번 Object와 2번 Object Instance, 7번 Resource인 경우 요청 메시지이다. 해당 경로 정보를 Uri-Path 옵션으로 표시하고, Accept 옵션에 link-format(40)를 표시한다.

```
CON GET [24325]
Token: 4446
Uri-Path: 3
Uri-Path: 2
Uri-Path: 7
Accept: 40
```

3/2/7 Resource에 대한 Discover 응답 메시지이다. 3/2/7 Resource에 정의된 속성값을 포함한다. Content-format 옵션에 link-format(40)을 설정하고 Payload에 다음과 같이 link-format으로 정보를 포함한다.

```
ACK 2.05 Content [24325]
Token: 4446
Content-format: 40
Payload
{
```

```
    </3/2/7>;dim=8;pmin=10;pmax=60;gt=50;lt=42.2
}
```

7.3.4 IFpg: 제어

LWM2M 기반 e-IoT 게이트웨이 기본 제어 기능은 LWM2M의 Device Management & Service Enablement Interface를 이용한다. e-IoT 게이트웨이는 4.4.6에서 정의한 정보 모델 형식의 Resource에 값을 설정하는 기능이다. 즉, 제어 목적의 정보 모델이 {Object ID}/{Object Instance ID}/{Resource ID}/{Resource Instance ID} 경로로 정의되어 있어 해당 모델에 CoAP PUT 또는 POST 메소드를 이용하여 값을 설정하여 제어한다.

제어하고자 하는 대상 Resource가 쓰기 동작을 허용하는 경우는 PUT 메소드를 사용하고 실행 동작을 허용하면 POST 메소드를 사용한다.

e-IoT 게이트웨이의 원하는 데이터 Resource를 제어하는 요청 메시지는 다음과 같이 구성한다.

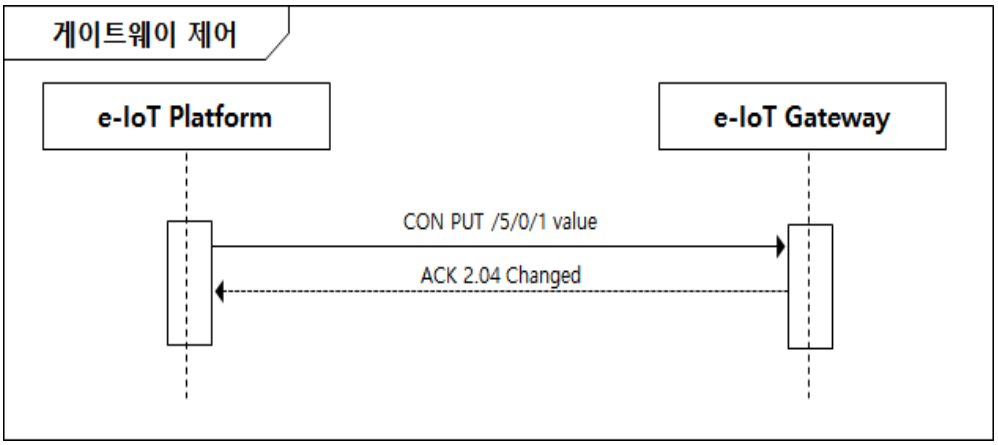
a) Confirmable Transaction 설정한다.
b) 쓰기 동작인 경우는 CoAP PUT 메소드로, 실행 동작을 요하는 경우는 POST로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 제어하고자 하는 대상 Resource를 Uri-Path 옵션으로 표현한다.
e) Content-format 옵션은 Plain-text(0) 또는 LWM2M JSON(11543)으로 설정한다.
f) Payload에 설정하고자 하는 값을 설정한다.

요청 메시지의 대상 Resource의 존재 유무를 판단하고 존재하는 경우 해당 Resource의 값을 변경하거나 실행한다. PUT 요청 메시지는 값을 변경하고, POST 요청 메시지는 정의되어 있는 동작을 실행한다. 그 다음 응답 메시지는 다음과 같이 구성하여 응답한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
    1) 성공인 경우 2.04 Changed
    2) 실패인 경우 4.00 Bad Request, 4.04 Not Found,
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.

재부팅 제어는 '/3/0/4'를 대상으로 하며, 재부팅의 최신 결과는 '/3/0/204' Resource에 저장 관리한다. 재등록(Re-registration) 제어는 '/1/0/8'를 대상으로 수행한다. 공장 초기화 제어는 '/3/0/5'를 대상으로 하며, 하드웨어 리셋 명령을 수행한 이후 재부팅하고 그 결과를 '3/0/205' Resource에 저장 관리한다.

다음은 e-IoT 게이트웨이의 펌웨어 다운로드 주소를 변경하는 과정의 예시이다.



(그림 7-29) e-IoT 게이트웨이 제어 절차 예시

LWM2M 기반으로 e-IoT 게이트웨이의 펌웨어 다운로드 주소를 변경하는 요청 메시지 예시이다.

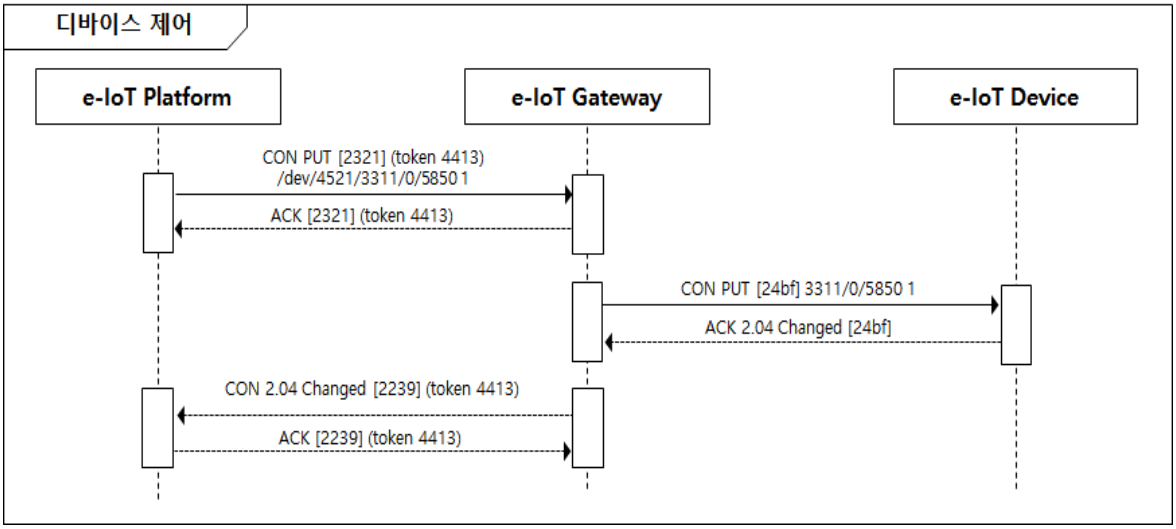
```
CON PUT [abcd]
Token: 4444
Uri-Path: 5
Uri-Path: 0
Uri-Path: 1
Content-Format: 11543
Payload:
{ “e”: [
    {"n":”“, "sv":”http://firmawareServer:Port/firm”} ]
}
```

LWM2M 기반으로 e-IoT 게이트웨이 펌웨어 다운로드 주소 변경 요청에 대한 응답 메시지 예시이다.

```
ACK 2.04 Changed [abcd]
Token: 4444
```

다음은 e-IoT 플랫폼에서 e-IoT 디바이스(전등 제어기)의 전원을 켜는 경우이다. 위와 같이 e-IoT 플랫폼이 e-IoT 게이트웨이를 제어하는 경우와 동일하나, Resource의 경로에 차이가 있다. Object와 Resource 경로에 앞에 e-IoT 디바이스를 의미하는 “dev/4521/” 경로가 추가되어 있어, e-IoT 게이트웨이는 e-IoT 디바이스의 정보임을 인식하고, e-IoT 디바이스를 제어한다. e-IoT 게이트웨이가 제어 요청 메시지를 수신하면,

ACK 메시지로 응답한 이후에 디바이스용 제어 요청메시지를 생성하여 해당 디바이스에 전송한다. 디바이스로부터 수신한 응답 메시지를 Confirmable CoAP 메시지로 다시 플랫폼에 전달한다.



(그림 7-30) e-IoT 디바이스 제어 절차 예시

IFpg 인터페이스 구간에서 먼저 e-IoT 플랫폼이 e-IoT 디바이스의 전원을 켜기 위한 요청 메시지를 다음과 같이 생성한다.

a) Confirmable Transaction 설정한다.
b) CoAP 메소드는 쓰기 속성인 Resource인 경우는 PUT으로 실행 속성의 Resource인 경우는 POST로 매핑한다. 위 예시에서는 3311/0/5850의 속성이 쓰기 속성이기 때문에 PUT으로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 제어하고자 하는 대상 Resource를 Uri-Path 옵션으로 표현한다. 대상 Resource는
e) LWM2M의 정보 모델이다.
f) Content-format은 Plain-text 또는 LWM2M JSON으로 설정할 수 있다. 본 예제에서는 JSON으로 설정한다.

다음 메시지 예시는 e-IoT 플랫폼에서 생성한 e-IoT 디바이스의 전원을 켜기 위한 요청 메시지이다.

```
CON PUT [2321]
Token: 4413
Uri-Path: dev
Uri-Path: 4521
Uri-Path:3311
Uri-Path: 0
Uri-Path: 5850
Content-Format: 11543
Payload:
{ “e”: [
        {“n”: “”, “bv”: 1} ]
}
```

e-IoT 게이트웨이는 제어 요청한 해당 Resource가 하부에 연결되어 있는 e-IoT 디바이스의 것임을 판단하고 바로 응답할 수 없음으로 Confirmable 요청 메시지에 대한 응답 메시지를 생성하여 e-IoT 플랫폼에 응답한다. 이 단순 ACK 응답 메시지에는 실제 값은 없다. 단순 ACK 응답 메시지는 다음과 같이 생성한다.

a) ACK Transaction 설정한다.
b) CoAP 코드는 0.00으로 설정한다.
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값으로 설정한다.

다음은 e-IoT 디바이스의 Resource 정보 조회 요청에 대한 Confirmable 요청 메시지에 대한 단순 ACK 응답 메시지의 예시이다.

```
ACK 0.00 [2321]
Token: 4413
```

e-IoT 게이트웨이는 단순 ACK 메시지를 e-IoT 플랫폼에 응답함과 동시에 e-IoT 디바이스에 해당 Resource를 제어하는 요청 메시지를 다음과 같이 생성해서 전송한다.

a) Confirmable Transaction 설정한다.
b) CoAP PUT 메소드로 매핑한다. 쓰기 속성인 Resource인 경우는 PUT으로 실행 속성의 Resource인 경우는 POST로 매핑한다.
c) Message ID와 Token을 각 e-IoT 게이트웨이가 생성하여 포함한다.
d) 제어하고자 하는 대상 Resource를 Uri-Path 옵션으로 표현한다. LWM2M Object 이후의 경로만을 포함한다.
e) Content-format으로는 Plain-text 또는 LWM2M JSON으로 설정할 수 있다. 본 예제에 서는 plain-text인 0으로 설정한다.

e-IoT 게이트웨이가 e-IoT 디바이스의 해당 Resource를 제어하기 위한 요청 메시지의

예시이다.

```
CON PUT [24bf]
Token: 2489
Uri-Path: 3311
Uri-Path: 0
Uri-Path: 5850
Content-format: 0
Payload:
1
```

e-IoT 디바이스는 제어 요청한 해당 Resource의 존재 유무를 판단하고 존재하는 경우 해당 Resource에 제어값을 설정한다. PUT인 경우는 단순 쓰기 동작을 POST의 경우는 정의되어 있는 동작을 수행한다. 쓰기인 경우에도 적절한 동작이 요하는 경우는 정의되어 있는 동작을 수행한다. 그 다음으로 다음과 같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
    1) 성공: 2.04 Changed
    2) 실패: 4.00 Bad Request, 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.

e-IoT 디바이스를 제어한 결과를 포함한 응답 메시지의 예시이다.

```
ACK 2.04 Changed [24bf]
Token: 2489
```

e-IoT 게이트웨이는 제어 결과를 e-IoT 디바이스로부터 수신함으로써 다음의 CoAP 메시지를 구성하여 e-IoT 플랫폼에 전달한다.

a) CON Transaction 설정한다.
b) CoAP 응답 코드: 수신한 응답 코드와 동일하게 설정한다.
c) Message ID를 새롭게 생성하여 포함시킨다.
d) Token 값은 원래e-IoT 플랫폼의 조회 요청 메시지의 것과 동일하게 설정한다.

다음은 e-IoT 게이트웨이가 e-IoT 디바이스의 제어 결과를 포함한 CoAP 메시지 예시이다.

```
CON 2.04 Changed [2111]
Token: 4413
```

e-IoT 플랫폼은 제어 결과를 수신하고 이에 대한 단순 ACK 메시지를 아래와 같이 생성하여 전달한다.

ACK 0.00 [2111] Token: 4413

a) ACK Transaction 설정한다.
b) CoAP 코드는 0.00 으로 설정한다.
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값으로 설정한다.

7.3.5 IFpg: 펌웨어 업데이트

LWM2M 기반 게이트웨이 펌웨어 업데이트 절차는 다음과 같다.

a) 먼저 펌웨어Object의 Package Resource(5/0/0)에 업데이트할 펌웨어 파일을 전달 저장한다.
b) e-IoT 플랫폼은 e-IoT 게이트웨이가 펌웨어를 정상적으로 다운로드 했는지 상태를 파악한다.
c) 성공적으로 다운로드가 완료되었음을 확인한 이후 펌웨어 업데이트 및 재부팅을
d) e-IoT 게이트웨이는 펌웨어 업데이트를 수행한 이후에 재부팅하고 LWM2M 등록 절차를 수행한다.
e) e-IoT 플랫폼은 펌웨어 업데이트 결과 상태를 조회하여 완료 여부를 판단함으로써 펌웨어 업데이트 과정을 완료한다.

7.3.5.1 e-IoT 게이트웨이 펌웨어 파일 전달

e-IoT 게이트웨이의 펌웨어 패키지 Resource(5/0/0)에 펌웨어 파일을 전달하는 요청 메시지는 다음과 같이 구성한다.

a) Confirmable Transaction 설정한다.
b) CoAP PUT 메소드 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 펌웨어 패키지 Resource인 5/0/0 를 Uri-Path 옵션으로 표현한다.
e) Content-format 옵션을 octet-stream(42)으로 설정한다.
f) 전달하고자 하는 파일의 사이즈가 MTU 보다 큰 경우 Block1 옵션을 사용하여 분할 전송 한다. Block1 옵션의 NUM은 0번으로 M(More bit)는 1로, SZX는 Payload에 전달될 데이터의 크기로써, 사이즈는 2(SZX+4)로 계산된다.

그 다음 응답 메시지는 다음과 같이 구성하여 응답한다.

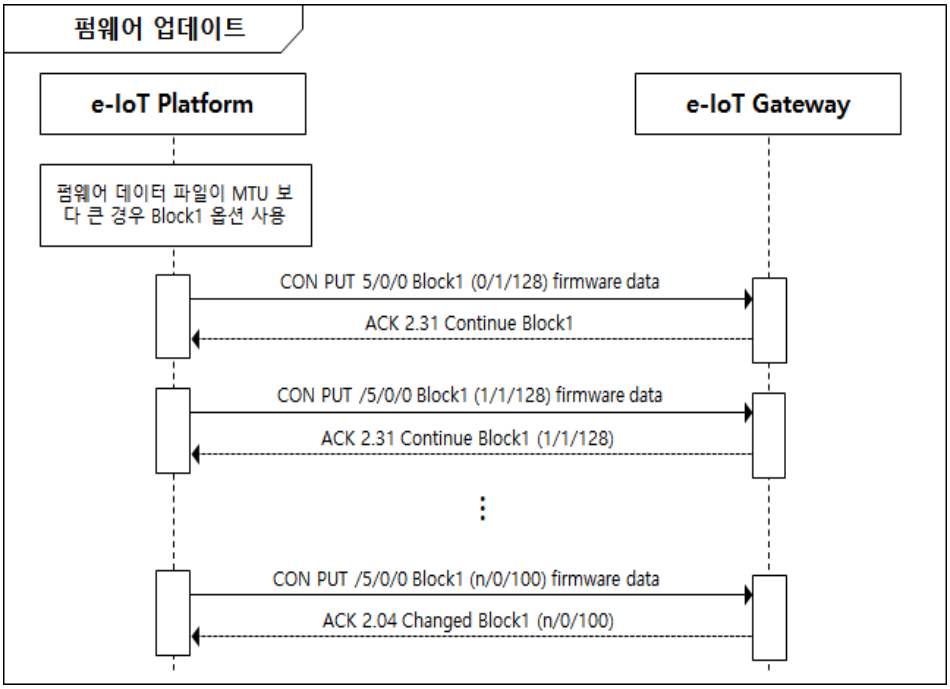
a) ACK Transaction 설정한다.

b) CoAP 응답 코드: 2.31 Continue

c) Message ID는 요청 메시지의 것과 동일하게 설정한다.

d) Token는 Block Transfer 과정이 끝날 때까지 첫 요청 메시지와 동일한 값으로 설정한다.

e) Block1 옵션을 요청 메시지의 것과 동일한 것으로 설정한다.



(그림 7-31) Block1 옵션을 이용한 펌웨어 파일 전달 과정

펌웨어 파일 전달 메시지 예시로서 Block1 옵션을 사용하여 펌웨어파일을 분할하여 전달한다.

```
CON PUT [123d]
Token: 4564
Uri-Path: 5
Uri-Path: 0
Uri-Path: 0
Content-Format: 42
Block1: 0/1/128
Payload:
#firmware file data#
```

펌웨어 파일 분할 전달의 첫 메시지에 대한 응답 메시지 예시로서 Block1 옵션이 포함되었으며, 동일한 Message ID와 토큰 값이 설정되어 있다.

```
ACK 2.31 Continue [123d]
Token: 4564
Block1: 0/1/128
```

펌웨어 파일 분할 전달의 마지막 (n+1번째) 메시지 예시로서 Block1 옵션의 More bit가 0으로 설정되어 있다.

```
CON PUT [1261]
Token: 4564
Uri-Path: 5
Uri-Path: 0
Uri-Path: 0
Content-Format: 42
Block1: n/0/64
Payload:
#firmware file data#
```

펌웨어 파일 분할 전달의 마지막 (n+1번째) 메시지에 대한 응답 메시지 예시로서 동일한 토큰 값이 유지되어 있음을 확인할 수 있다.

```
ACK 2.04 Changed [1261]
Token: 4564
Block1: n/0/64
```

7.3.5.2 펌웨어 파일 다운로드 상태 확인

펌웨어 파일 전달 과정이 완료되면, e-IoT 플랫폼은 e-IoT 게이트웨이의 펌웨어 다운로드 상태 정보를 최종 조회한다. 펌웨어 다운로드 상태 Resource(5/0/3)의 정보를 조회하는 요청 메시지는 다음과 같이 구성한다.

a) Confirmable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 5/0/3 Resource를 Uri-Path 옵션으로 표현한다. Object Instance 인 0번은 임의의 값으로서 변경될 수 있다.

펌웨어 파일 다운로드 상태 조회 요청에 적절한 상태 값을 Payload에 포함한 응답 메시지를 구성하고 다음과 같이 응답 메시지 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드

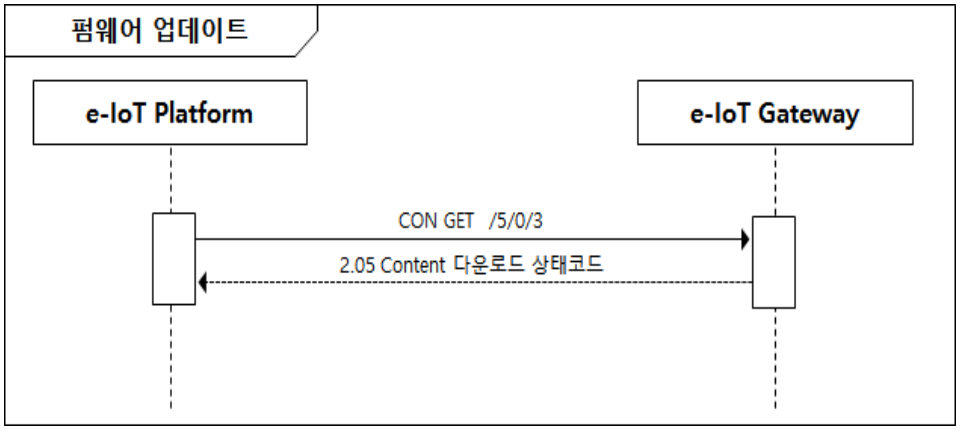
1) 성공 : 2.05 Content
2) 실패: 4.04 Not Found

c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
d) Content-format: Plain-text를 의미하는 0이다.
e) Payload: 아래 다운로드 상태값을 참조하여 알맞은 값으로 설정한다. 정상적으로 다운로드가 완료된 상태인 경우 2번으로 설정한다.

펌웨어 파일 다운로드 상태 값은 다음과 같다.

<표 7-1> Firmware file download 상태

다운로드 상태값(state)	의미
0	Idle(클라이언트가 설정, 다운로드 이전 혹은 성공적으로 업데이트 한 이후에 설정)
1	Downloading
2	Downloaded
3	Updating



(그림 7-32) 펌웨어 파일 다운로드 상태 확인 절차

펌웨어 파일 다운로드 상태 조회 요청 메시지 예시

```
CON GET[1467]
Token: 5674
Uri-Path: 5
Uri-Path: 0
Uri-Path: 3
```

펌웨어 파일 다운로드 상태 조회에 따른 응답 메시지 예시

ACK 2.05 Content [1467] Token: 5674 Content-format: 0 Payload: 2

7.3.5.3 펌웨어 업데이트 수행

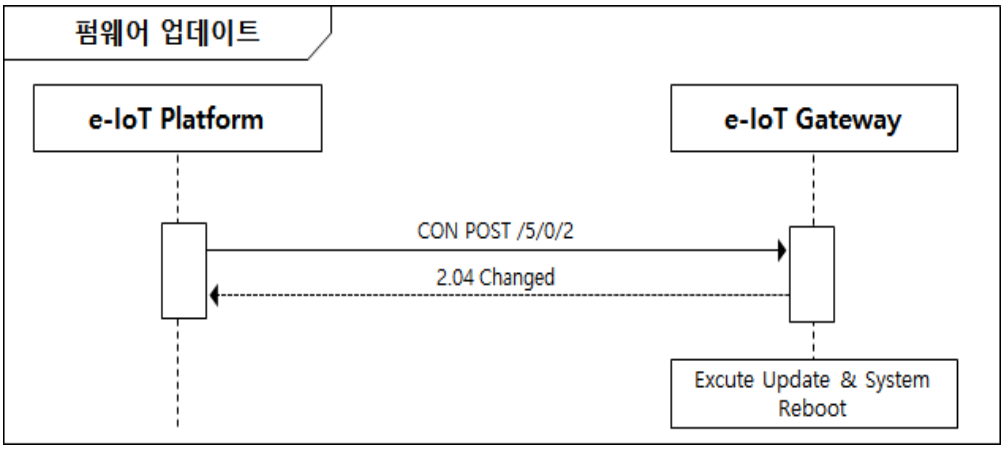
펌웨어 파일 다운로드가 성공적으로 완료되면 e-IoT 플랫폼은 펌웨어 업데이트를 지시한다. 펌웨어 업데이트 요청 메시지는 다음과 같이 구성한다.

a) Confirmable Transaction 설정한다.
b) 실행 동작을 요하는 POST로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 제어하고자 하는 대상인 Firmware Update Object(5), Update Resource(2)를 Uri-Path 옵션으로 표현한다. 예시의 경우 Object Instance를 0으로 설정한다.

펌웨어 업데이트 요청메시지를 수신한 다음 응답 메시지는 다음과 같이 구성하여 응답한 후 펌웨어 업데이트를 수행하고 재부팅을 시도한다.

a) ACK Transaction 설정
b) CoAP 응답 코드
   1) 성공인 경우 2.04 Changed
   2) 실패인 경우 4.00 Bad Request, 4.04 Not Found,
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.

다음은 e-IoT 게이트웨이의 펌웨어 업데이트 과정의 예시이다.



(그림 7-33) 펌웨어 업데이트 수행

펌웨어 업데이트 요청 메시지의 예시이다.

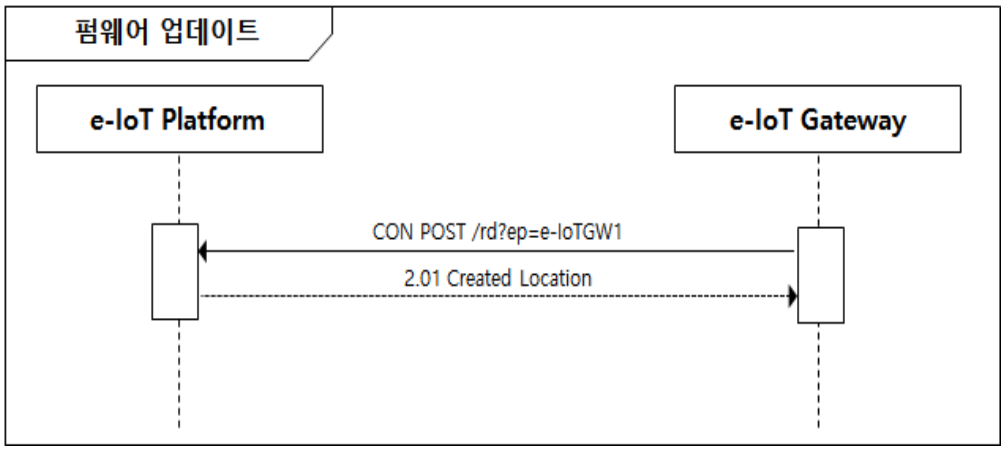
```
CON POST [14ee]
Token: 5124
Uri-Path: 5
Uri-Path: 0
Uri-Path: 2
```

펌웨어 업데이트 요청에 대한 응답 메시지 예시이다.

```
ACK 2.04 Changed [14ee]
Token: 5124
```

7.3.5.4 e-IoT 게이트웨이 등록

e-IoT 게이트웨이는 성공적으로 펌웨어를 업데이트하고 재부팅이 완료되면 아래 그림과 같이 곧바로 LWM2M 등록 과정을 수행한다. 등록 요청 메시지와 응답 메시지는 앞 절에서 소개한 등록 과정과 형식은 동일하다.



(그림 7-34) 재부팅 이후 e-IoT 게이트웨이 등록 과정

7.3.5.5 펌웨어 업데이트 결과 확인

e-IoT 플랫폼은 e-IoT 게이트웨이의 등록 예상 시간(기본 30초) 이후, 펌웨어 업데이트 결과 상태 Resource(5/0/5)의 정보를 조회하여 펌웨어 업데이트 과정을 마무리한다. 요청 메시지는 다음과 같이 구성한다.

a) Confirmable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.

c) Message ID와 Token을 각각 생성하여 포함한다.
d) 5/0/5 Resource를 Uri-Path 옵션으로 표현한다. Object Instance인 0번은 임의의 값이며, 변경될 수 있다.

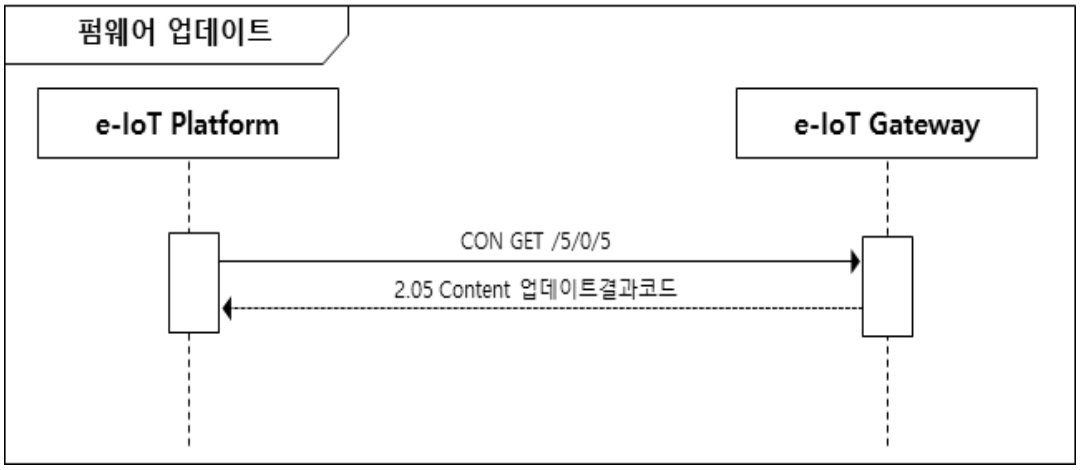
펌웨어 업데이트 결과 상태 조회를 요청에 적절한 상태 값을 Payload에 포함한 응답 메시지를 다음과 같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정
b) CoAP 응답 코드
   1) 성공 : 2.05 Content
   2) 실패: 4.04 Not Found
c) Message ID와 Token을 포함한다 요청 메시지와 동일한 값으로 설정한다.
d) Content-format: Plain-text를 의미하는 0으로 설정한다.
e) Payload: 아래 펌웨어 업데이트 결과 상태값을 참조하여 알맞은 값으로 설정한다. 성공적으로 펌웨어 업데이트가 된 경우 1번으로 설정한다.

펌웨어 업데이트 결과 코드는 다음과 같다.

<표 7-2> Firmware Update 결과 코드

Update 결과 값	의미
0	초기값
1	성공
2	새 펌웨어 패키지 저장공간 부족
3	다운로드 중 저장공간 부족
4	다운로드 중 네트워크 연결 실패
5	새 펌웨어 패키지 CRC 실패
6	미지원 펌웨어 패키지 타입
7	잘못된 URI
8	펌웨어 업데이트 실패



(그림 7-35) 펌웨어 업데이트 결과 확인 절차

펌웨어 업데이트 결과 확인 요청 메시지 예시이다.

```
CON GET[1f67]
Token: 1274
Uri-Path: 5
Uri-Path: 0
Uri-Path: 5
```

펌웨어 업데이트 결과 확인 요청에 대한 응답 메시지 예시이다.

```
ACK 2.05 Content [1f67]
Token: 1274
Content-format: 0
Payload:
1
```

7.4 IFgd-LWM2M

게이트웨이와 디바이스 사이의 인터페이스로서, 디바이스 등록, 주기적 보고, 정보 조회 표준을 기술한다.

7.4.1 IFgd: 디바이스 등록

e-IoT 디바이스를 e-IoT 게이트웨이에 등록시키는 과정이다. CoAP 프로토콜을 사용하면 CoAP Resource Directory 표준을 기반으로 정의한다. 등록 과정은 먼저 e-IoT 게이트웨이의 Resource Directory(RD)를 검색하고 e-IoT 디바이스가 검색된 RD에 등록하는 과정으로 구분된다. 등록된 이후에도 주기적으로 등록 업데이트를 통해서 e-IoT 게이트웨이에 등록 정보를 유지시켜 준다. e-IoT 디바이스의 물리적인 제약사항 때문에 등록에 관

련된 기능 중에서 아래 표와 같이 선택과 필수 기능으로 구분하였다.

<표 7-3> Device 기본 등록 관련 기능

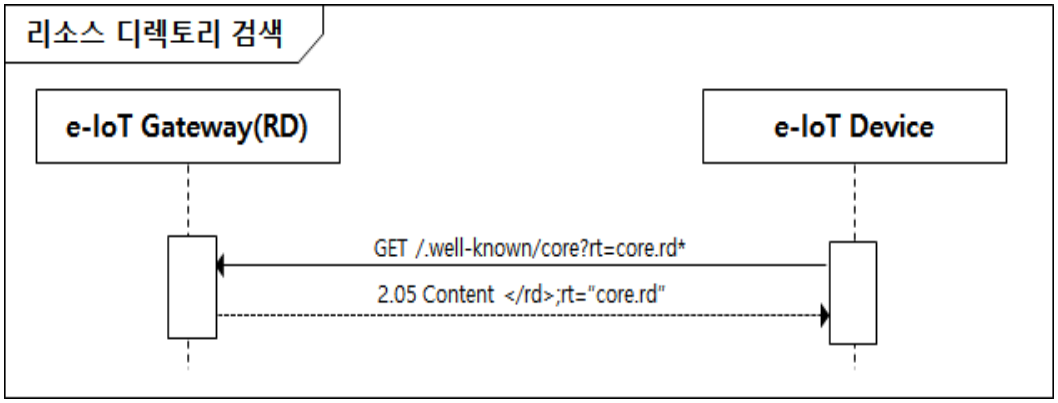
	등록 관련 기능	필수/선택
1	Resource Directory 검색	선택
2	e-IoT 디바이스 등록	필수
3	e-IoT 디바이스 업데이트	필수

7.4.1.1 IFgd: Resource Directory 검색

e-IoT 디바이스가 RD를 이용하기 위해서는 먼저 RD를 지원하는 장치(e-IoT 게이트웨이)의 IP 주소, port 번호, RD function set를 지원하는 경로(Path) 정보를 알아야 한다. 이 기능을 선택적으로 구현되지 않았을 경우 e-IoT 게이트웨이의 주소와 Resource Directory 정보를 미리 설정해야 한다. Resource Directory는 “/rd”로 설정한다.

이 정보를 획득하기 위한 동작이 Resource Directory 검색이다. 검색 요청 인터페이스는 다음과 같이 정의한다.

a) Interaction: e-IoT 디바이스(EP) -> e-IoT 게이트웨이(RD)
b) Confirmable Transaction으로 설정
c) 메소드: GET
d) URI Template: /.well-known/core{?rt}
e) URI Template Variables:
    1) rt := Resource Type(선택). 다음과 같은 값을 포함할 수 있다.
        (a) “core.rd”, 또는 “core.rd=lookup”, “core.rd-group”, “core.rd*”
f) Contents-Type: 40(application/link-format)
g) 응답 코드는 다음과 같다.
    1) Success: 2.05 “Content”, Payload에는 application/link-format Content-Format으로 RD resource를 담는다.
    2) Failure: 4.04 “Not Found”, unicast로 요청되는 것에 매칭되는 엔트리가 없을 때
    3) Failure: 4.00 “Bad Request”, 잘못 정의된 Unicast 요청 메시지를 수신할 때



(그림 7-36) Resource Directory 검색 과정

Resource Directory 검색 요청 메시지 예시는 다음과 같다.

```
CON GET [2479]
Token: ef8769cd2
Uri-Host: {Multicast or Unicast 주소}
Uri-Path: .well-known
Uri-Path: core
Uri-Query: rt=core.rd*
```

위 요청 메시지에서 Broadcast 주소를 사용하여 RD를 지원하는 e-IoT 게이트웨이가 응답하게 할 수 있다. 혹은 e-IoT 게이트웨이 주소를 직접 입력할 수도 있다. Resource Directory 검색 응답 메시지 예시는 다음과 같다.

```
ACK 2.05 Content [2479]
Token: ef8769cd2
Content-Format: 40
Payload:
</rd>; rt="core.rd",
</rd-lookup>;rt="core.rd-lookup",
</rd-group>;rt="core.rd-group"
```

7.4.1.2 IFgd: 등록

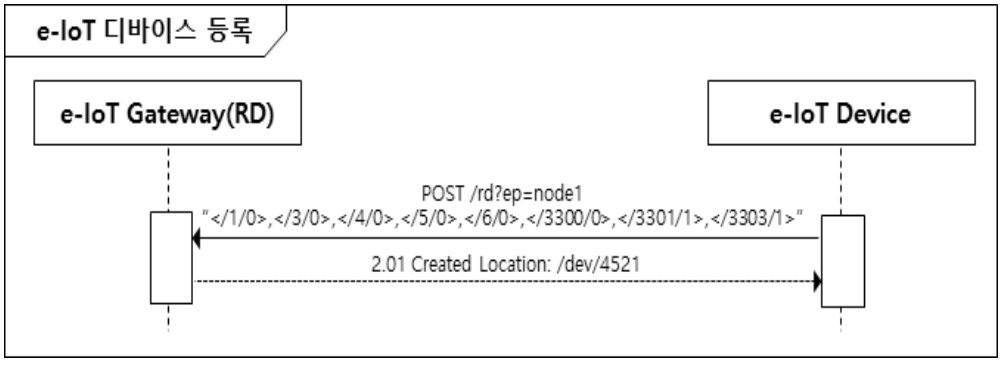
위 검색 과정이 끝난 이후에 e-IoT 디바이스는 자신의 Resource를 e-IoT 게이트웨이에 등록시킨다. Link Format(RFC 6690) 형식의 Resource 리스트를 POST 메소드를 이용해서 등록한다. 이때 자신의 end point 이름 쿼리에 반드시 넣어서 전달해야 한다. End point 이름 이외의 파라미터는 선택적으로 넣을 수 있다. 등록 POST 메시지를 수신한 RD(e-IoT 게이트웨이)는 요청된 Resource를 Directory에 생성하거나 업데이트하고 그

위치 정보를 응답 메시지에 전달한다.

등록(Registration) 인터페이스는 다음과 같이 정의한다.

a) Interaction: e-IoT 디바이스(EP) -> e-IoT 게이트웨이(RD)
b) Confirmable Transaction으로 설정
c) 메소드: POST
d) URI Template: /{+rd}{?ep,lt}
e) URI Template Variables:
   1) rd:= RD Function Set Path(필수). Uri-Path 옵션으로 표현한다.
   2) ep:= Endpoint name(필수). 반드시 해당 영역에서는 유일성이 보장되는 식별자이어야 한다. 본 표준에서는 해당 노드의 OID이다. 최대 허용 길이는 63 bytes이다. Uri-Query 옵션으로 표현한다.
   3) lt:= Lifetime(선택), 등록된 정보의 수명, 단위는 초이며 값의 범위는 60 -야 한다. 42949672951이다 이 값이 포함되지 않았을 경우 기본값은 86400(24시). Uri-Query 옵 션으로 표현한다.
f) Content-Format: 40(application/link-format)
g) Payload: e-IoT 디바이스에 정의된 Object와 Object Instance 정보들을 포함시킨다.
h) 응답 코드는 다음과 같다.
   1) Ack Transaction 설정한다.
   2) Success: 2.01 “Created " , 등록된 Resource 엔트리 정보를 포함하는 Location-Path 헤더를 반드시 포함해야 한다.
      (a) Location-Path 정보는 RD(e-IoT 게이트웨이)에서 발행하는 값으로 e-IoT 게이트웨이 내에서 유일성이 보장되어야 한다. 본 표준에서는 /dev/ Directory에 임의의 숫자로 구성한다.
   3) Failure: 4.00 “Bad Request”, 잘못 정의된 Unicast 요청 메시지를 수신할 때.
   4) Failure: 5.03 “Service Unavailable”. 본 동작을 잘 수행하지 못했을 경우.

다음의 그림은 “node1” 이름을 갖는 e-IoT 디바이스(endpoint)가 자신의 Resource들을 RD에 등록하는 과정이다. 그에 대한 응답으로 생성된 위치는 ‘/dev/4521’이다. 이때 4521은 e-IoT 게이트웨이 RD에서 생성한 임의 숫자이다.



(그림 7-37) e-IoT 디바이스 등록 절차 예시

e-IoT 디바이스 등록 요청 메시지의 예시는 다음과 같다.

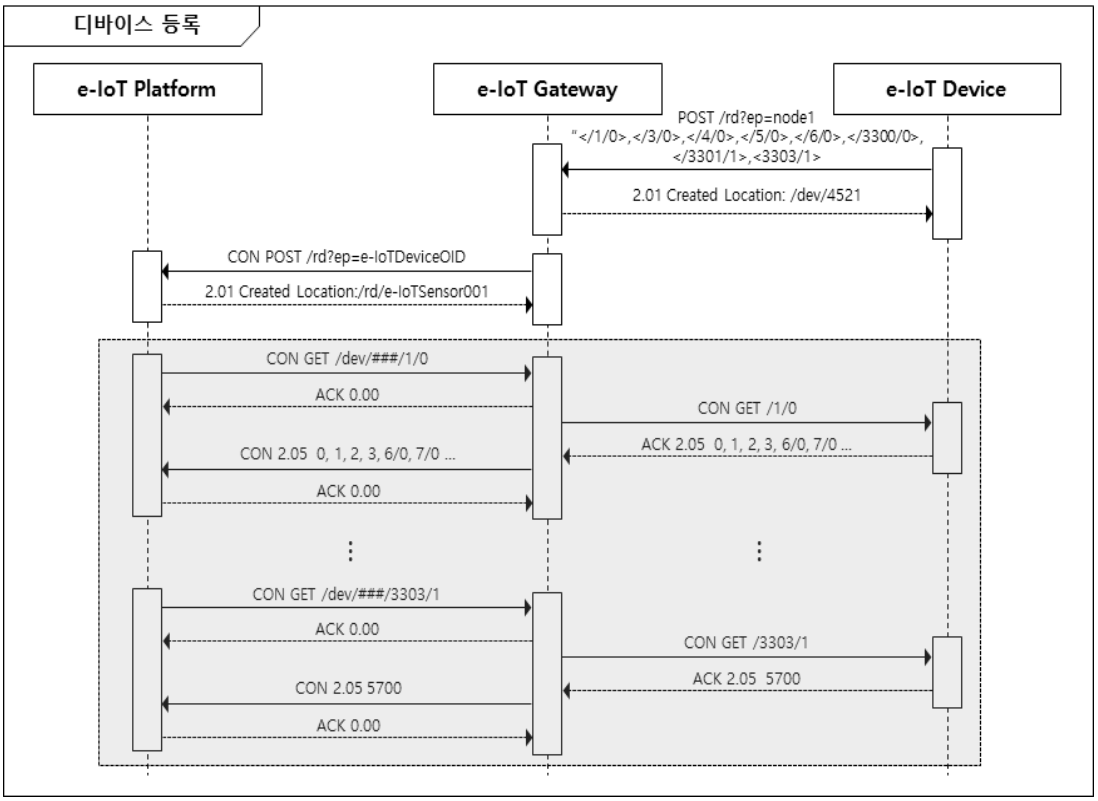
```
CON POST [4793]
Token: ef87abcd2
Uri-path: rd
Uri-query: ep=node1
Content-Format: 40
Payload:
</1/0>,</3/0>,</4/0>,<5/0>,<6/0>,<3300/0>,<3301/1>,<3303/1>
```

e-IoT 디바이스 등록에 대한 응답 메시지의 예시는 다음과 같다.

```
ACK 2.01 Created [4739]
Token: ef87abcd2
Location-Path: dev
Location-Path: 4521
```

e-IoT 디바이스의 Object와 Object Instance 하부에 존재하는 Resource에 대한 정보는 정보 조회 기능을 이용하여 획득할 수 있다.

다음은 e-IoT 게이트웨이에 e-IoT 디바이스가 기본 등록된 이후에 바로 e-IoT 플랫폼에 기본 등록하는 전체 과정의 메시지 흐름이다. e-IoT 디바이스가 e-IoT 게이트웨이 등록된 이후에 e-IoT 게이트웨이는 e-IoT 디바이스를 e-IoT 플랫폼에 등록하는 과정을 수행한다. IFpg 인터페이스의 기본 등록 과정은 IFpg-LWM2M 절에 소개되어 있다.
그 이후 등록된 Object별로 조회하는 과정을 통해 e-IoT 플랫폼과 e-IoT 게이트웨이는 e-IoT 디바이스의 Object와 Resource 정보를 획득할 수 있게 된다.



(그림 7-38) e-IoT 디바이스 기본 등록 전체 절차 예시

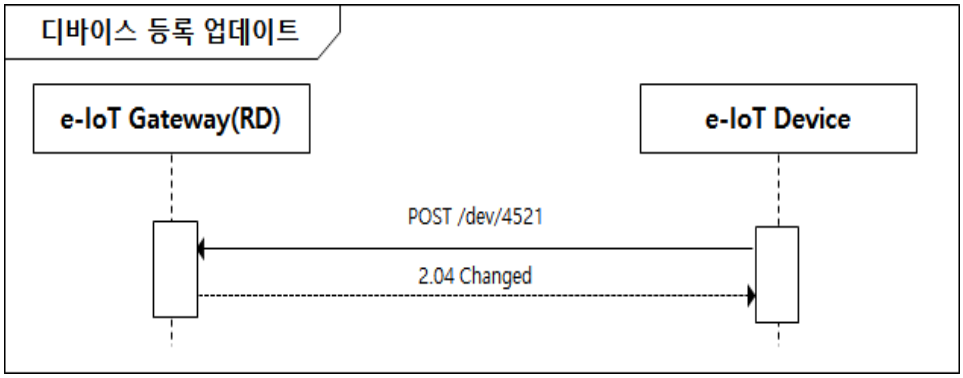
7.4.1.3 IFgd: 등록 업데이트

앞서 e-IoT 디바이스가 e-IoT 게이트웨이에 등록한 정보를 기 설정된 Lifetime이 종료되기 전에 정보를 업데이트하여 e-IoT 게이트웨이의 e-IoT 디바이스 정보를 유지시켜준다. POST 메소드를 사용한다. Lifetime 정보를 업데이트한다.

a) Interaction: e-IoT 디바이스(EP) -> e-IoT 게이트웨이(RD)
b) 메소드: POST
c) URI Template: /{+location}{?lt}
d) URI Template Variables:
   1) location:= 앞서 수행한 등록과정에서 e-IoT 디바이스가 응답으로 수신한 Location -Path다. Uri-path 옵션으로 표현한다.
   2) lt := Lifetime(선택), 등록된 정보의 수명, 단위는 초이며 값의 범위는 60 - 42949672950이다. 이 값이 포함되지 않았을 경우 기본값은 86400(24시). Uri-Query 옵션으로 표현한다.
e) Content-Format: application/link-format(선택적으로, Payload에서 수정된 정보가 포함되어 있을 경우)
f) Payload: 수정된 정보 모델이 있을 경우만 Payload에 포함시킨다. 수정되지 않은 정보까지도 함께 포함되어야 한다.
g) 응답 코드는 다음과 같다.
   1) Success: 2.04 “Changed”

2) Failure: 4.00 “Bad Request”, 잘못 정의된 Unicast 요청 메시지를 수신할 때.
3) Failure: 4.04 “Not Found”, 만약 e-IoT 게이트웨이가 재부팅되어서 등록 정보가 없을 경우.
4) Failure: 5.03 “Service Unavailable”. 본 동작을 잘 수행하지 못했을 경우.

다음의 그림은 앞서 RD에 Resource를 등록한 이후에 생성된 “/dev/4521” 위치에 등록 업데이트를 수행하는 과정이다.



(그림 7-39) e-IoT 디바이스 등록 업데이트 예시

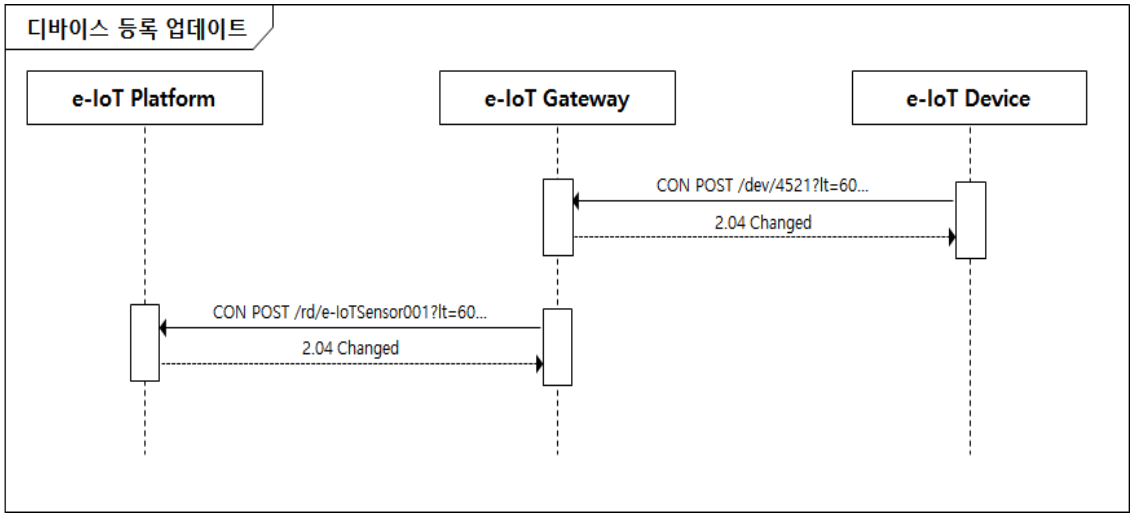
등록 업데이트 요청 메시지 예시이다.

```
CON POST /dev/4521 [ab34]
Uri-Path: dev
Uri-Path: 4521
Token: 1234bdef
```

등록업데이트에 대한 응답 메시지 예시이다.

```
ACK 2.04 Changed [ab34]
Token: 1234bdef
```

성공 응답 메시지를 수신하지 못하고 실패 응답 메시지를 수신한 경우는 e-IoT 디바이스 등록 절차를 다시 수행한다.



(그림 7-40) e-IoT 디바이스 기본 등록 업데이트 전체 과정

IFgd 인터페이스를 통한 e-IoT 디바이스 기본 등록 업데이트와 IFpg 인터페이스를 통한 e-IoT 디바이스 기본 등록 업데이트는 위 그림과 같이 독립적으로 동작한다. 즉, e-IoT 디바이스 등록 업데이트(IFgd)로 인해 e-IoT 게이트웨이가 e-IoT 디바이스 기본 등록 업데이트(IFpg) 과정을 이어서 수행할 필요가 없다.

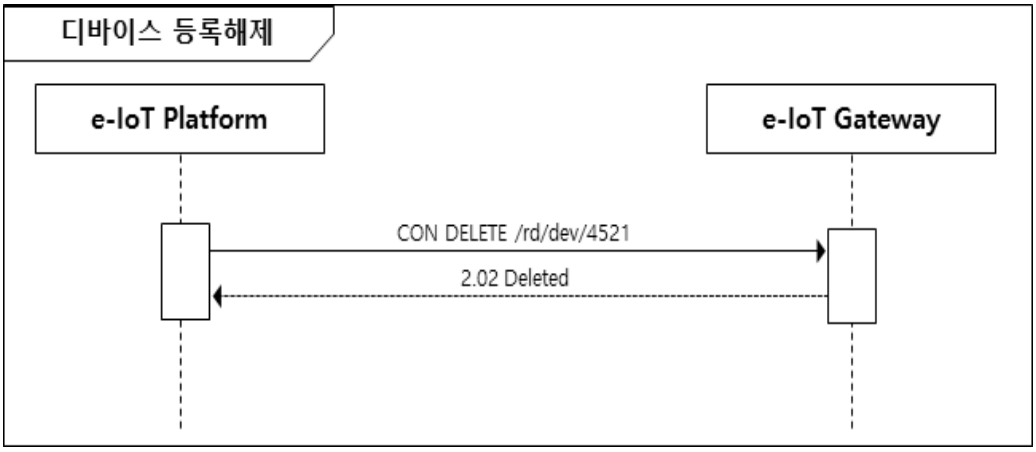
7.4.1.4 IFgd: 등록 해제

e-IoT 디바이스 Resource 등록 해제는 LWM2M의 등록 인터페이스를 이용한다. LWM2M 프로토콜 측면에서 e-IoT 게이트웨이가 서버가 되며, e-IoT 디바이스는 클라이언트가 된다.

등록(Registration) 해제 과정은 다음과 같이 정의한다.

a) Interaction: e-IoT 디바이스(클라이언트) -> e-IoT 게이트웨이(서버)
b) 메소드: DELETE
c) URI Template: /{location} e-IoT 디바이스 등록 시 응답 메시지로 등록된 위치
d) 응답 코드는 다음과 같다.
    1) Success: 2.02 “Deleted”
    2) Failure: 4.04 “Not Found”, 등록 해제 요청한 위치

다음의 그림은 LWM2M 기반 등록 해제 과정의 예시이다.



(그림 7-41) LWM2M 기반 e-IoT 디바이스 등록 해제 절차 예시

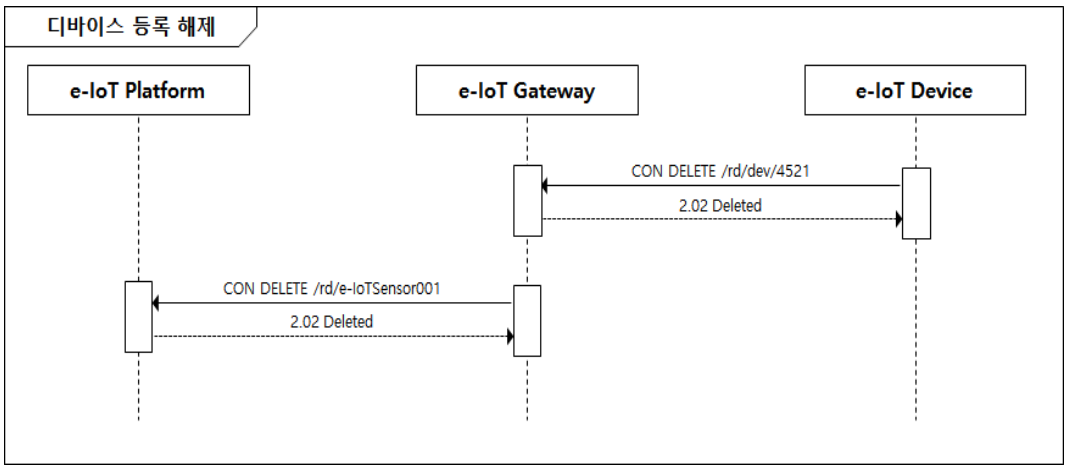
다음은 LWM2M 기반 등록 해제 요청 메시지 예시이다.

```
CON DELETE [123a]
Token:2fff
Uri-Path: rd
Uri-Path: dev
Uri-Path: 4521
```

다음은 LWM2M 기반 등록 해제 응답 메시지 예시이다.

```
ACK 2.02 Deleted [123a]
Token:2fff
```

다음은 e-IoT 디바이스 등록 해제의 결과로 e-IoT 게이트웨이에 존재하는 e-IoT 디바이스 정보가 삭제되었기 때문에 아래 그림과 같이 e-IoT 게이트웨이는 IFpg 인터페이스를 통해 e-IoT 디바이스 등록 해제를 수행해야 한다.



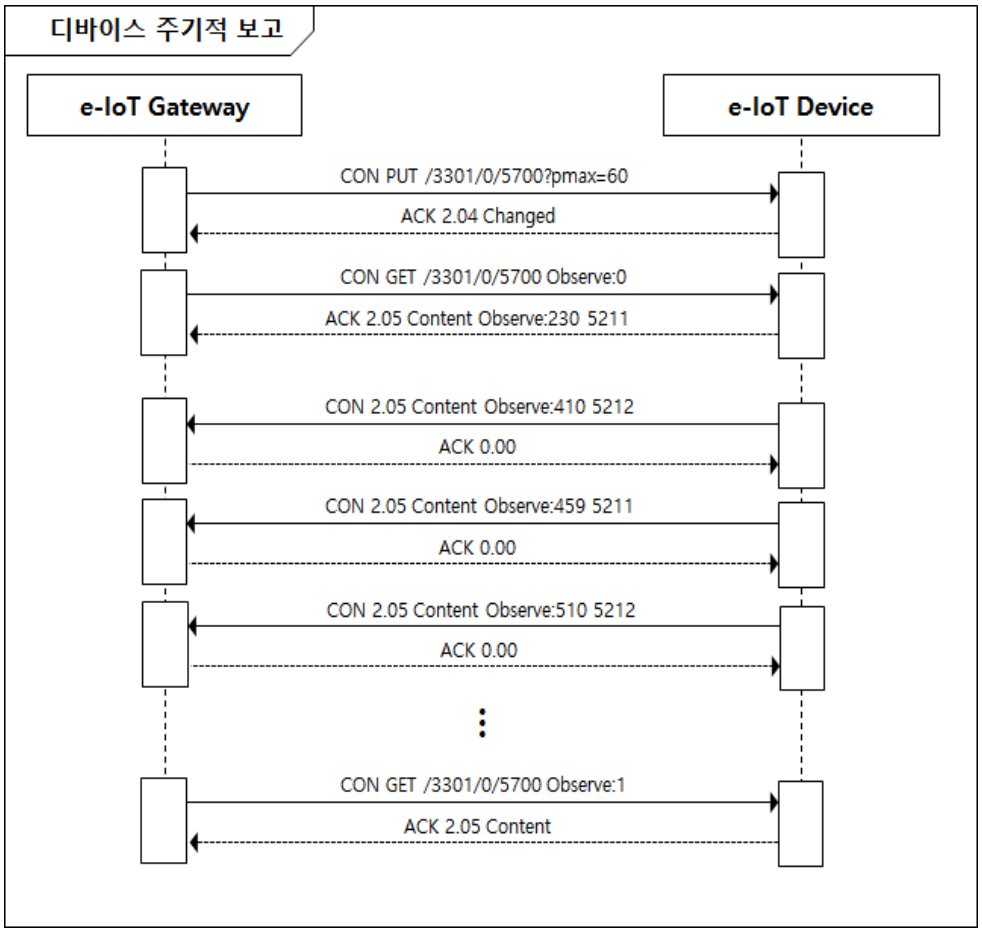
(그림 7-42) LWM2M 기반 e-IoT 디바이스 등록 해제 절차 예시

7.4.2 IFgd: 주기적 보고

e-IoT 디바이스의 주기적 보고는 두 가지 방법으로 정의한다. 첫 번째 방법은 CoAP Observation 기능을 이용한 방법이며, 두 번째 방법은 e-IoT 디바이스가 POST 메시지를 이용해서 해당 타겟 Resource URL에 값을 보고하는 방법이다. Device(3) Object의 Device Type(17) Resource의 값이 “device“인 경우 첫 번째 방법을, “constrained”인 경우 두 번째 방법을 이용해서 주기적 보고를 수행한다.

7.4.2.1 CoAP Observe 를 이용한 주기적 보고

본 절에서는 Device Type(17) Resource 값이 “device”인 경우의 주기적 보고를 정의한다.



(그림 7-43) CoAP Observe를 이용한 e-IoT 디바이스 주기적 보고 예시

7.4.2.1.1 보고 주기 값 설정

보고 주기 값은 대상 Resource의 pmax 속성을 설정하는 절차이다. 설정 요청 메시지는 다음과 같이 구성한다.

a) CoAP PUT 메소드로 매핑한다.
b) 생성한 Message ID와 Token을 포함한다.
c) 대상 Resource를 Uri-Path 옵션으로 표현한다.
d) Uri-Query 옵션을 이용해서 설정하고자 하는 속성값을 표현한다.

보고 주기 값 설정 요청 메시지를 수신하고 해당 속성을 변경한 후 다음과 같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
    1) 성공: 2.04 Changed
    2) 실패: 4.00 Bad Request, 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.

다음은 보고 주기 설정 요청 메시지의 예시이다. 조도 센서의 측정값을 60 초 주기로 보고하도록 설정하는 예시이다.

CON PUT [251ae] Token: 2123 Uri-Path: 3301 Uri-Path: 0 Uri-Path: 5700 Uri-Query: pmax=60

다음은 보고 주기 설정 요청에 대한 응답 메시지 예시이다.

ACK 2.04 Changed [251ae] Token: 2123

7.4.2.1.2 CoAP Observe 설정 및 주기 보고

대상 데이터 Resource에 Observe를 요청하고 요청이 수락된 Resource는 주기마다 정보를 보고하는 절차이다.

CoAP Observe 설정 요청 메시지는 다음과 같이 구성한다.

a) Conformable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.
c) 생성한 Message ID와 Token을 포함한다.

d) 대상 Resource를 Uri-Path 옵션으로 표현한다.
e) Observe 옵션: 0

다음은 조도 센서 측정값(3301/0/5700)에 Observe를 설정하는 요청 메시지의 예시이다.

```
CON GET [243bf]
Token: 2789
Uri-Path: 3301
Uri-Path: 0
Uri-Path: 5700
Observe: 0
Accept: 0
```

Observe 요청 메시지 수신 후 Observe를 설정하고 다음과 같이 응답 메시지를 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
    1) 성공: 2.05 Content
    2) 실패: 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값.
d) Observe 옵션: 증가한 값
e) Content-format:
    1) Observe 대상이 Resource인 경우: Plain-text(0) 또는 LWM2M JSON(11543)
    2) Observe 대상이 Object Instance인 경우: LWM2M JSON(11543)
f) Payload: 해당 데이터 Resource의 값을 포함한다.

다음은 Observe 설정 요청에 대한 응답 메시지 예시이다.

```
ACK 2.05 Content [243bf]
Token: 2789
Observe: 321
Content-format: 0
Payload:
5211
```

다음은 e-IoT 디바이스가 주기적 보고 메시지를 다음과 같이 구성하여 전달한다.

a) CON Transaction 설정한다.
b) CoAP 코드: 2.05 Content
c) Message ID: 임의로 설정한다.

d) Token 값은 Observe 설정 요청 메시지의 것과 동일한 값으로 설정한다.
e) Observe 옵션: 이전 보다 증가한 값
f) Content-format: Observe 요청 메시지의 Accept 옵션의 값
g) Payload: 해당 데이터 Resource의 값을 포함한다.

다음은 주기적 보고 메시지 예시이다.

CON 2.05 Content [243c0] Token: 2789 Observe: 410 Content-format: 0 Payload: 5212

다음은 주기적 보고 메시지에 대한 응답 메시지를 empty 메시지로 구성해서 전달한다.
a) ACK Transaction 설정한다.
b) CoAP 코드
   1) 성공: 0.00 empty message
   2) 실패: 4.04 Not Found
c) Message ID: 주기적 보고 메시지의 것으로 한다.

다음은 Observe가 설정된 조도 센서의 측정값(3301/0/5701) Resource의 주기적인 보고 메시지에 대한 응답 메시지 예시이다. 아래 예시에서 Token 값은 앞의 메시지와 동일함을 확인할 수 있다.

ACK 0.0 empty [243c0] Token: 2789 Payload:

※ Payload가 없음.

7.4.2.1.3 주기적 보고 해제

CoAP Observe기반 주기적 보고의 해제는 CoAP의 Observe 옵션을 이용하며, CoAP Observe 설정 요청 메시지는 다음과 같이 구성한다.

a) Conformable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.
c) 생성한 Message ID와 Token을 포함한다.
d) 대상 Resource를 Uri-Path 옵션으로 표현한다.
e) Observe 옵션:
f) Accept 옵션: Plain

1) Observe 대상이 Resource인 경우: Plain-text(0) 또는JSON(50)
2) Observe 대상이 Object Instance인 경우: JSON(50)

다음은 조도 센서 측정값(3301/0/5700)에 Observe를 해제하는 요청 메시지의 예시이다.

```
CON GET [24eef]
Token: 2789
Uri-Path: 3301
Uri-Path: 0
Uri-Path: 5700
Observe: 1
Accept: 0
```

Observe 해제 요청 메시지 수신 후 Observe를 해제하고 다음과 같이 응답 메시지를 전달한다.

a) ACK Transaction 설정한다.
b) CoAP 응답 코드
   1) 성공: 2.05 Content
   2) 실패: 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.
d) Content-format:
   1) Observe 대상이 Resource인 경우: Plain-text(0) 또는 LWM2M JSON(11543)
   2) Observe 대상이 Object Instance인 경우: LWM2M JSON(11543)
e) Observe 옵션: 반드시 제외하여야 한다.
f) Payload: 해당 데이터 Resource의 값을 포함한다.

다음은 Observe 설정 요청에 대한 응답 메시지 예시이다.

```
ACK 2.05 Content [24eef]
Token: 2789
Content-format: 0
Payload:
5200
```

7.4.2.2 POST 를 이용한 주기적 보고

본 절에서는 Device Type(17) Resource의 값이 “constrained“ 인 경우의 주기적 보고를 정의한다. 미리 설정된 “pmax” 값을 주기로, 해당 Resource 값을 POST 메시지를 이용해서 주기적으로 전송한다.
주기적 보고 대상이 되는 Resource의 최종 경로는 e-IoT 디바이스 등록 시 획득한

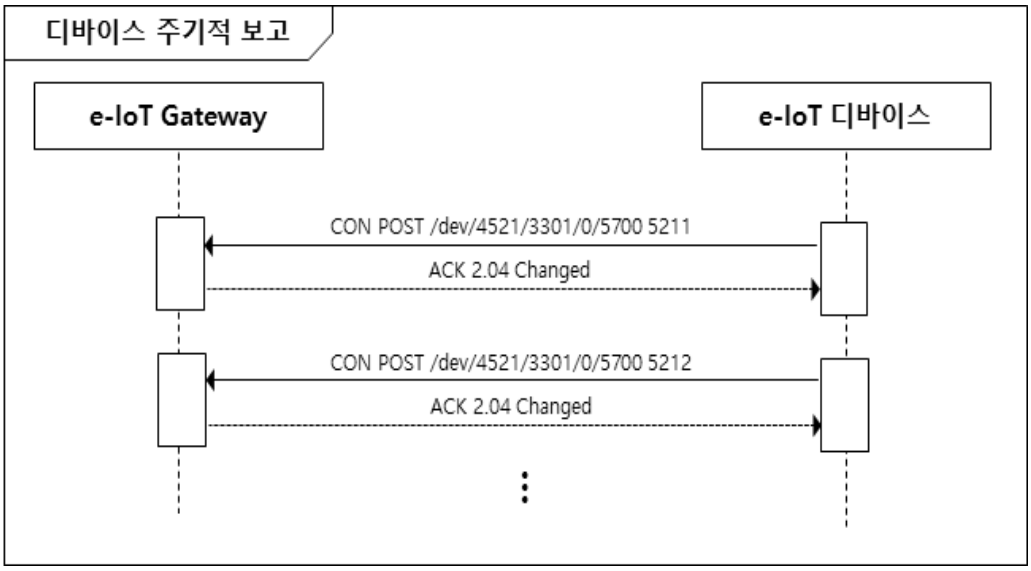
Location-Path 정보와 사전에 지정한 Resource 경로의 조합으로 결정된다. 예를 들어, 주기적 보고할 대상이 '3301/0/5700'이고, 등록 Location-Path가 '/dev/1234566'인 경우 최종 대상 경로는 '/dev/1234566/3301/0/5700'이 된다.

e-IoT 디바이스의 POST를 이용한 주기적 보고 메시지는 다음과 같이 구성한다.

a) Confirmable Transaction 설정한다.
b) CoAP POST 메소드로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 보고하고자 하는 대상 Resource를 Uri-Path 옵션으로 표현한다. 대상 Resource는 LWM2M의 정보모델이다. 대상 Resource는 사전에 디바이스에 설정되어야 한다.
e) Content-format: plain-text를 의미하는 '0'으로 설정한다.
f) Payload: 해당 Resource의 값을 포함한다.

정보 보고 메시지에 대해 다음의 응답메시지를 수신 확인한다.
a) ACK Transaction 설정한다.
b) CoAP 응답 코드
    1) 성공: 2.04 "Changed"
    2) 실패: 4.04 "Not Found", 요청되는 대상 URL에 매칭되는 엔트리가 없을 때
    3) 실패: 4.00 "Bad Request", 잘못 정의된 Unicast 요청 메시지를 수신할 때
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값이다.

다음의 그림은 e-IoT 디바이스의 POST를 이용한 주기적 보고 과정의 메시지 예시이다.



(그림 7-44) POST를 이용한 e-IoT 디바이스 주기적 보고 예시

아래 메시지는 e-IoT 디바이스가 POST를 이용해서 주기적으로 전송하는 메시지의 예시이다.

```
CON POST [251ae]
Token: 2123
Uri-Path: dev
Uri-Path: 4521
Uri-Path: 3301
Uri-Path: 0
Uri-Path: 5700
Content-format: 0
Payload:
5211
```

아래 메시지는 e-IoT 디바이스가 POST를 이용해서 주기적으로 전송하는 메시지의 응답 메시지 예시이다.

```
ACK 2.04 Changed [251ae]
Token: 2123
```

7.4.3 IFgd: 정보 조회

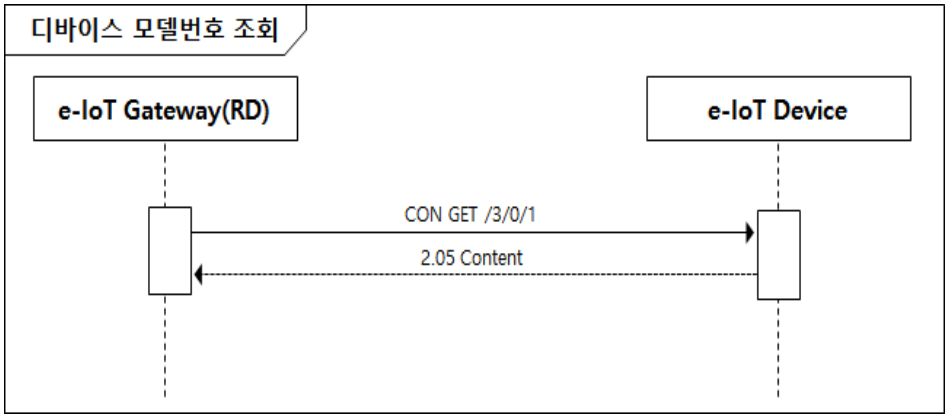
e-IoT 디바이스의 원하는 정보 모델을 조회하는 요청 메시지는 다음과 같이 구성한다.

a) Confirmable Transaction 설정한다.
b) CoAP GET 메소드로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) 조회하고자 하는 대상 Resource를 Uri-Path 옵션으로 표현한다. 대상 Resource는 LWM2M의 정보모델이다.
e) Accept 옵션을 통해서 응답 받을 정보의 형식을 Plain-text 또는 JSON으로 설정한다.

정보 조회를 요청한 해당 Resource의 존재 유무를 판단하고 존재하는 경우 해당 값을 포함하여 다음과 같이 응답 메시지를 구성하고 전달한다.

a) ACK Transaction 설정
b) CoAP 응답 코드
   1) 성공: 2.05 Content
   2) 실패: 4.04 Not Found
c) Message ID와 Token을 포함한다. 요청 메시지와 동일한 값.
d) Content-format: 요청메시지의 Accept 옵션의 값, 기본값을 Plain-text로 한다.
e) Payload: 해당 Resource의 값을 포함한다.

다음 그림은 e-IoT 디바이스의 모델 번호 조회하는 과정의 메시지 예시이다.



(그림 7-45) e-IoT 디바이스 모델 번호를 조회하는 과정

e-IoT 디바이스의 모델 번호 조회를 요청하는 메시지의 예시이다.

CON GET [243bf]
Token: 2789
Uri-Path: 3
Uri-Path: 0
Uri-Path: 1

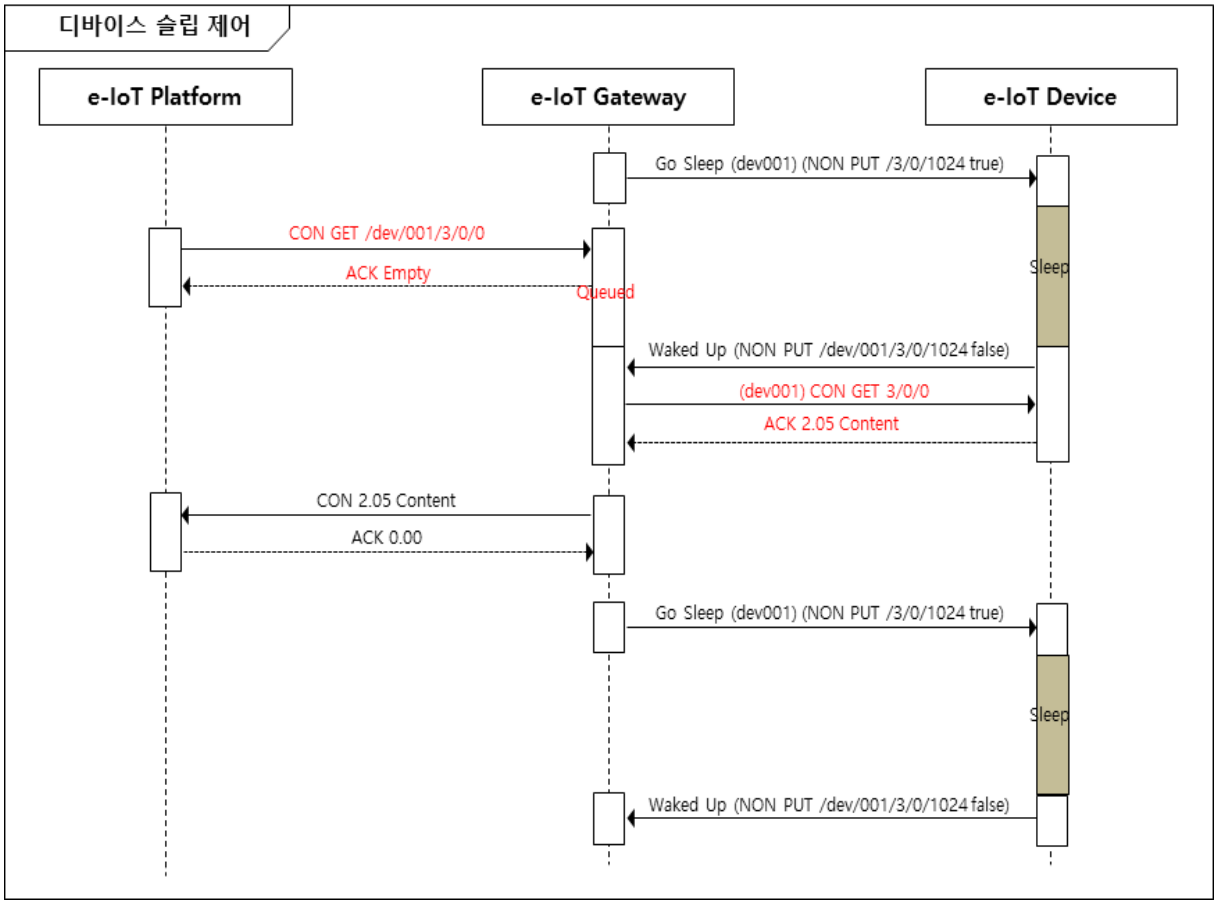
e-IoT 디바이스의 모델 번호를 조회에 대한 응답 메시지의 예시이다.

ACK 2.05 Content [243bf]
Token: 2789
Content-format: 0
Payload:
xxsensor-123-12

7.4.4 IFgd: Sleep Node Control

e-IoT 게이트웨이가 e-IoT 디바이스의 Sleep를 제어하는 기능으로 'Go Sleep'와 'Waked Up' 메시지를 이용한다. 아래 예에서 e-IoT 게이트웨이는 e-IoT 디바이스와 통신할 작업이 없음을 판단하고 'Go Sleep' 메시지를 e-IoT 디바이스에 전송한다. 이 메시지를 수신한 e-IoT 디바이스는 자체 Sleep 메커니즘에 의해 슬립한다. 'Go Sleep' 메시지를 전달한 e-IoT 게이트웨이는 해당 디바이스에 대해서 Queue 모드로 전환하여 해당 e-IoT 디바이스에게 전달된 메시지를 저장하게 된다((그림 10-50)의 빨간색 메시지)
e-IoT 디바이스가 Sleep에서 깨어나는 방법 또한 디바이스 자체 방식으로 결정된다. 예를 들어 타이머 기반으로 깨어날 수 있다. 깨어난 e-IoT 디바이스는 'Waked Up' 메시지를 e-IoT 게이트웨이에게 전송하여 자신이 통신할 준비가 되었음을 알린다. 이 메시지를

수신한 e-IoT 게이트웨이는 저장된 메시지를 해당 디바이스에 전송한다.



(그림 7-46) e-IoT 디바이스 Sleep 제어하는 과정

'Go Sleep' 메시지는 다음과 같이 구성한다.

a) Non-Confirmable Transaction 설정한다.
b) CoAP PUT 메소드로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) Sleep Resource 경로를 Uri-Path 옵션으로 표현한다. 예를 들어 "/3/0/1024"
e) Content-format: plain-text를 의미하는 '0'으로 설정한다.
f) Payload: true

아래는 e-IoT 게이트웨이가 e-IoT 디바이스에 전송한 'Go Sleep' 메시지 예시이다.

```
NON PUT [24bf]
Token: 2489
Uri-Path: 3
Uri-Path: 0
Uri-Path: 1024
Content-format: 0
```

```
Payload:
true
```

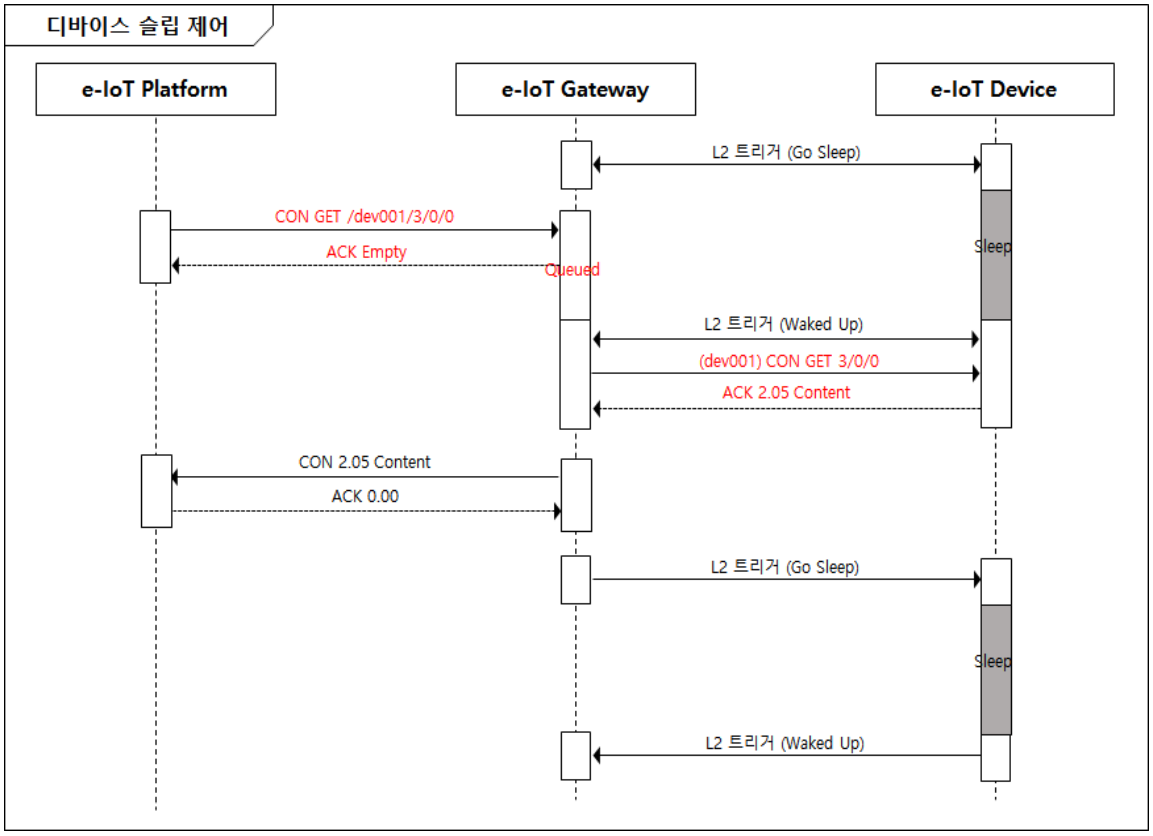
‘Waked up’ 메시지는 다음과 같이 구성한다.

a) Non-Confirmable Transaction 설정한다.
b) CoAP PUT 메소드로 매핑한다.
c) Message ID와 Token을 각각 생성하여 포함한다.
d) Sleep Resource 경로를 Uri-Path 옵션으로 표현한다.
   예) “/dev/001/3/0/1024”
e) Content-format: plain-text를 의미하는 ‘0’으로 설정한다.
f) Payload: false

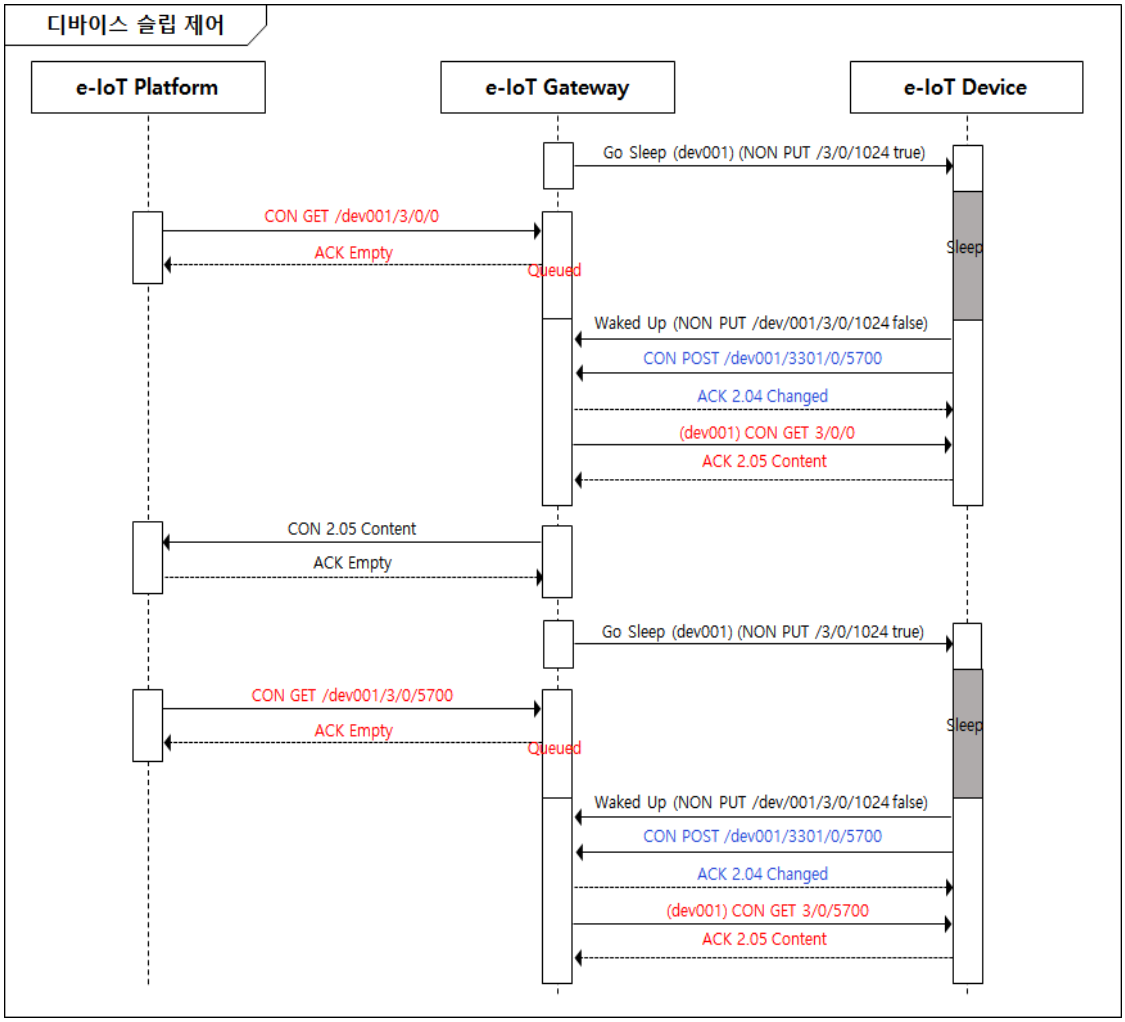
아래는 e-IoT 디바이스가 e-IoT 게이트웨이에게 전송한 Waked Up 메시지 예시이다.

```
NON PUT [24be]
Token: 2400
Uri-Path: dev
Uri-Path: 001
Uri-Path:3
Uri-Path:0
Uri-Path: 1024
Content-format: 0
Payload:
false
```

만약 e-IoT 게이트웨이와 e-IoT 디바이스 사이의 2계층 통신 기법에서 Sleep에 대한 신호를 알려주는 L2 Trigger 기술이 존재한다면 Go Sleep 메시지와 Waked Up 메시지는 생략될 수 있다. e-IoT 게이트웨이가 2계층 트리거 신호에 따라서 Device(3) Object의 Sleep(1024) Resource의 값을 변경하여 주면 된다.



(그림 7-47) L2-Trigger 기반 e-IoT 디바이스 Sleep 제어하는 과정



(그림 7-48) 주기적 보고 메시지를 전송하는 e-IoT 디바이스 Sleep 제어하는 과정

e-IoT 디바이스가 단순 주기적 보고를 하는 경우에는 위 그림과 같이 'Waked Up' 메시지를 전송한 다음에 주기적 보고 메시지를 전송하면 된다. 주기적 보고 메시지 전송 주기에 따라서 Sleep 상태에서 깨어나도록 타이머를 설정된다.

부 록 Ⅰ-1

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

지식재산권 확약서 정보

해당 사항 없음

부 록 Ⅰ-2

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

시험인증 관련 사항

해당 사항 없음

부 록 Ⅰ-3

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

본 표준의 연계(family) 표준

Ⅰ-3.1 에너지 전력 분야 사물인터넷(e-IoT)

표준번호	표준제목
TTAK.KO-10.1121-part1	에너지 전력분야 사물인터넷(e-IoT) - 제1부: 시스템 규격
TTAK.KO-10.1121-part2	에너지 전력분야 사물인터넷(e-IoT) - 제2부: 단순등록 규격
TTAK.KO-10.1121-part3	에너지 전력분야 사물인터넷(e-IoT) - 제3부: 데이터 보고 규격
TTAK.KO-10.1121-part4	에너지 전력분야 사물인터넷(e-IoT) - 제4부: 현장단말 서비스 규격
TTAK.KO-10.1121-part5	에너지 전력분야 사물인터넷(e-IoT) - 제5부: 협대역 무선통신 물리계층 규격

부 록 Ⅰ-4

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

참고 문헌

[1] OMA Lightweight Machine to Machine Technical Specification Approved version 1.0-00 Feb 2017
[2] TTAE.IF-RFC7252 (2015), “제한된 환경에서의 응용 프로토콜(CoAP)”
[3] IETF draft-ietf-core-resource-directory-05 : “CoRE Resource Directory“
[4] IETF RFC 7641 : “Observing Resources in (CoAP)“
[5] IETF RFC 6690 : “Constrained RESTful Environments(CoRE) Link Format”
[6] IETF RFC 7959 : “Block-wise-transfers in CoAP“
[7] oneM2M TS-0001-V1.13.1 : “Functional Architecture“
[8] oneM2M TS-0002-V1.0.1 : “Requirements“
[9] oneM2M TS-0004-V1.6.0 : “Service Layer Core Protocol“
[10] KS X 4105: 2004, 개방형 시스템 간 상호 접속 - 객체 식별자(OID)의 구성과 등록 절차

부 록 Ⅰ-5

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

영문표준 해설서

해당 사항 없음

부 록 Ⅰ-6

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

표준의 이력

판수	채택일	표준번호	내용	담당 위원회
제1판	2018.12.19	TTAK.KO-10.1121-Part1		사물인터넷네트워킹 프로젝트그룹 (SPG12)