

Resource Directory를 이용한 CoAP 그룹 통신 방법

김병준, 조성현*

한양대학교

kbjoz87@gmail.com, chopro@hanyang.ac.kr

CoAP Group Communication Using Resource Directory

Byungjun Kim, Sunghyun Cho*

Hanyang Univ.

요약

본 논문은 IoT 환경에서 주로 쓰이는 경량 프로토콜인 CoAP 프로토콜을 이용한 그룹 통신 방법을 제시 한다. CoAP 프로토콜의 메시지 포맷, 메시지 전송 방식을 설명하고 CoAP 프로토콜 그룹통신 방법 중 하나인 Resource Directory를 이용한 그룹별 멀티 캐스팅 방법을 제안한다. 멀티 캐스팅을 이용한 제안하는 방법은 유니 캐스트 방식에 비해 패킷 전송율과 전송 시간을 단축시킬 수 있으며 다수의 센서가 있는 환경에서 효율성이 더욱 증대된다.

I. 서론

최근 무선인터넷 기술과 센서 네트워크의 발달로 IoT[1](Internet of Things)라는 개념이 대두 되고 있다. IoT의 개념은 용어 그대로 사물 대 사물간의 통신을 통해 모든 객체간의 정보를 공유하고 이를 기반으로 이 용자에게 정보 제공 및 서비스를 제공하기 위한 기술이다. IoT 환경을 구축하기 위한 무선 센서 활용 범위가 확대 되면서 인터넷 표준 단체인 IETF(Internet Engineering Task Force) WG(Working Group)에서는 LLN(Low power Lossy Network)환경에 적합한 프로토콜인 CoAP(Constrained Application Protocol)[2] 표준화 작업이 진행 중이다. 본 논문에서는 CoAP 프로토콜을 이용한 그룹 통신 방법을 기술하고 각 그룹 별 노드 마 다 선별적으로 QoS를 적용하여 복잡한 IoT 환경에서 신뢰성 있는 네트워크를 구축할 수 있는 방안을 제안한다.

II. 본론

CoAP 프로토콜은 저전력 네트워크 및 저용량 소형 노드에 사용 될 수 있는 특수한 웹 전송 프로토콜 이다. IETF내의 CoRE(Constrained RESTful Environment) 워킹그룹에서 6LoWPAN(Low-power Wireless Personal Area Network)의 상위 애플리케이션 계층 프로토콜로 2014년에 국제 표준을 획득 하였다. CoAP는 RESTful(Representational State Transfer)사상을 따르고 있어 기존의 HTTP 웹 프로토콜과도 쉽게 변환 및 연동이 될 수 있는 장점이 있다.

2.1 CoAP 프로토콜 스택

CoAP의 아키텍처로는 아래 그림과 같이 기본적으로 하위 프로토콜 스

택의 전송계층으로 UDP 프로토콜을 사용하며 유니 캐스트와 멀티캐스트를 지원한다. 비동기적으로 동작하기 때문에 신뢰성 있는 전달을 위해 Rest Message 및 타이머 관리를 옵션으로 포함 하고 있다. 그 상위에는 요청/응답 계층과 메시지 계층으로 나뉜다.

Application Layer (CoAP)	Application Request/Response Message Layer
Transport Layer	UDP

그림 1. CoAP의 추상적 Layer

2.2 CoAP 메시지 포맷

CoAP의 메시지포맷으로는 다음과 같이 고정된 메시지 헤더와 토큰, 옵션, 페이로드로 구성되어 있다. 고정된 헤더의 2비트 V는 버전, T는 메시지(확인형(CON),비확인형(NON),승인(ACK),리셋(RST))로 구성되어 있으며, TKL은 토큰필드 길이를 나타낸다. Code는 클래스(요청(0),성공적인 응답(2), 클라이언트 에러 응답(4), 서버 에러 응답(5))로 3비트로 구성되어 있으며 5비트는(GET,POST,PUT,DELETE) 메소드로 구성되어 있다. 메시지 ID는 각각의 트랜잭션을 의미하며 Token의 경우 요청/응답 메시지를 확인 하는 필드이다.

0	1	2	3
01234567890123456789012345678901	01234567890123456789012345678901	01234567890123456789012345678901	01234567890123456789012345678901
Ver	T	TKL	Code
Message ID			
Token (if any, TKL bytes) ...			
Options (if any) ...			
Payload (if any) ...			

그림 2. CoAP Message Format

The diagram shows two sequence diagrams. The left diagram has two vertical lifelines. The top lifeline sends a message labeled 'CON[0x7d34]' to the bottom lifeline. The bottom lifeline then sends a message labeled 'ACK[0x7d34]' back to the top lifeline. The right diagram has a single vertical lifeline. It sends a message labeled 'NON[0x7d34]' to the right, towards an implied participant.

그림 4. 비신뢰성 메시지 전송

CoAP 그룹 통신을 하기 위한 그룹 멤버 설정에는 3가지 방법이 있다.[3] 첫째, 통신 하기 위한 각 노드들에 IP 멀티캐스트 주소 혹은 호스트 이름을 미리 설정 하는 방법이다. 둘째, DNS-SD(DNSbasedService Discovery) 혹은 Resource Discovery[4] 에 그룹 통신할 노드들을 미리 설정 하는 방법이다. 셋째, Commissioning Device로 설정하는 방법이다. 본 논문에서는 Resource Discovery을 이용하는 그룹 통신 방법을 제시한다. 그룹통신을 하기 위한 각각의 IoT 센서 노드들은 다른 노드들을 찾거나 연계 동작을 할 수 있도록 노드가 가지고 있는 자원이나 속성을 프록시 서버에 있는 Resource Directory에 CoAP Link Format[5]을 사용하여 등록한다. 그룹 등록의 경우 각각의 IoT 센서 노드들은 생성 메소드인 POST 메시지로 “./well-known/core” 을 통하여 리소스를 등록하고, Resource Directory에 등록된 리소스를 확인하기 위해서는 선택 메소드인 GET 메시지를 사용 하여 리소스 정보를 얻게 된다. 다음 그림은 프록시를 통한 멀티 캐스트 그룹 통신 동작 흐름도이다.

```

sequenceDiagram
    participant CoAP1 as CoAP Node1
    participant CoAP2 as CoAP Node2
    participant Proxy as Proxy (RD)
    participant Client as Client (Controller)

    CoAP1->>CoAP2: CoAP NON Mcast (POST /Well-known/core "<sensor/node1>")
    CoAP2-->>CoAP1: CoAP NON ( 2.01 Created Location : rd-group )
    CoAP2->>CoAP1: CoAP NON Mcast (POST /Well-known/core "<sensor/node2>")
    CoAP1-->>CoAP2: CoAP NON ( 2.01 Created Location : rd-group )
    CoAP1->>Proxy: REQUEST (to MULTICAST address)
    Proxy->>Client: CoAP NON (GET /rd-lookup?cap=sensor) (To UNICAST Address)
    Client-->>Proxy: (optional)RESPONSE
    Proxy->>CoAP1: Support IP Multicast Routing
    CoAP1->>CoAP2: CoAP NON (2.05 Content <node1> <node2>)
  
```

The diagram illustrates the CoAP message exchanges between four entities: CoAP Node1, CoAP Node2, Proxy (RD), and Client (Controller). The sequence of messages is as follows:

- CoAP Node1 sends a CoAP NON Mcast (POST /Well-known/core "<sensor/node1>") to CoAP Node2.
- CoAP Node2 returns a CoAP NON (2.01 Created Location : rd-group) to CoAP Node1.
- CoAP Node2 sends a CoAP NON Mcast (POST /Well-known/core "<sensor/node2>") to CoAP Node1.
- CoAP Node1 returns a CoAP NON (2.01 Created Location : rd-group) to CoAP Node2.
- CoAP Node1 sends a REQUEST (to MULTICAST address) to the Proxy (RD).
- The Proxy (RD) sends a CoAP NON (GET /rd-lookup?cap=sensor) (To UNICAST Address) to the Client (Controller).
- The Client (Controller) returns an (optional)RESPONSE to the Proxy (RD).
- The Proxy (RD) sends Support IP Multicast Routing to CoAP Node1.
- CoAP Node1 sends a CoAP NON (2.05 Content <node1> <node2>) to CoAP Node2.

Red annotations on the right side of the diagram indicate specific process steps:

- 그룹 등록 과정 (Group Registration Process) is associated with the initial CoAP Mcast exchange between CoAP Node1 and CoAP Node2.
- 그룹 제어 과정 (Group Control Process) is associated with the CoAP GET request from the Proxy (RD) to the Client (Controller).

– 118 –