효율적인 USN 관리를 위한 CoAP 프로토콜 분석 및 적용방안

민경주* ·유상근** ·김용운** ·김형준** ·정회경***

*충남대학교 ·**한국전자통신연구원 ·***배재대학교

CoAP Protocol Applying Method for Efficiency USN Management

Kyoung-Ju Min* ·Sang-Geun Yoo** ·Young-Woon Kim** ·Hyung-Jun Kim**

Hoe-Kyung Jung***

*ChungNam Nat'l University ·**ETRI ·***PaiChai University

E-mail : eureka@cnu.ac.kr* ·{lobbi, qkim, khj}@etri.re.kr** ·hkjung@pcu.ac.kr***

요 약

제한된 자원을 갖는 센서 노드들을 이용하여 온도, 습도 등의 센싱 된 정보를 효율적으로 관리하기 위하여 IETF 워킹그룹에서 CoAP 프로토콜을 제안하고, 인터넷 드래프트 상태에 있다. 센서 노드나 네트워크 장비들에 응용 프로토콜이 탑재되면, 프로토콜의 규격에 맞추어 효율적으로 센서 정보의 수집 및 활용이 가능하다. 하지만, 제한된 자원의 한계로 인해, 이를 센서 노드들에 적용하기 위해서는 CoAP-Lite 형태로 탑재되어야 하는데, 이러한 모델에서의 문제점을 분석하고, 이를 적용하기 위한 방법을 제안한다.

ABSTRACT

To manage a sensing information such as temperature, humidity efficiently, it's highly recommended suitable application protocol. In this reasons, IETF WG proposed a CoAP protocol, and it's on Internet draft. If it is possible to work on a specific protocol, sensor end nodes and network devices will be managed efficiently. However, end nodes have constrained resources, it is hard to apply a CoAP protocol. In this paper, we analyse a CoAP protocol for USN, and propose a applying method for CoAP over USN nodes.

키워드

CoAP, USN, Sensor, CoAP-Lite, Mote

Ⅰ. 서 론

제한된 성능을 갖는 소형 저 전력 센서 네트워크가 홈 네트워크, 빌딩 관리, 원거리 온도, 습도 검출 등 여러 분야에서 사용되면서, 각각의 응용 서비스가 요구하는 프로토콜 개발의 필요성이 대두되고 있다. 기존의 센서 네트워크에서 정보의 수집 및 전송을 위해 하드코딩 형태의 비효율적인 개발은 새로운 형태의 프로토콜이 개발되어야 개선이 가능하기 때문에, IETF (Internet Engineering Task Force)의 워킹그룹(WG: working group)에서 2010년 초에 CoAP (Constrained Application Protocol)을 개발하고, 몇 차례의 보완을 거쳐 2010년 7월 현재의 최종 인터넷 드래프트가 발표되고, 2011년을 목표로 국제 표준화 작업이 진행 중이다. 이에 본 논문에서는 현재 추진 중에 있는 인터넷 드래프트를 프로그래밍하기 위해 소프트웨어를 설계하는 과정에서 생긴 프로토콜 및 문제점을 분석하고, 이를 적용하기 위한 안을 제시한다. 본 논문의 구성은 다음과 같다. 2장에서는 CoAP 관련 분야 연구를 다룬다. 3장에서는 프로토콜의 주요 특징 및 문제점을 분석한다. 결론 및 향후 연구 과제는 4장에서 다룬다.

Ⅱ. 관련 연구

2.1. CoAP 프로토콜

CoAP 프로토콜은 센서를 이용하여, 온도, 습도, 조도, 진동 등의 정보를 효율적으로 관리하기 위한 USN(Ubiquitous Sensor Network)을 타겟으로 하는 프로토콜로, IETF에서 국제표준으로 추진하기 위해 현재 드래프트 상태에 있는 프로토콜로, 2010년 7월 8일의 draft-ietf-core-coap-01이 최신 문서이다. 다른 일반적인 PC나 모바일 단말 등과 달리 자원이 상대적으로 매우 빈약하기 때문에, 네트워크 통신을 위해 최소화된 메시지의 송수신만으로 노드 정보 간 통신을 가능하게 하기 위한 프로토콜로, 작게는 4바이트에서 수백 바이트(또는 수천 바이트)까지의 작은 데이터들로 통신이 가능하고, 현재까지는 드래프트 상태로 개발이 진행되고 있기 때문에, 다소 미흡한 부분이 존재하지만, 이와 같은 형태의 프로토콜은 센서 노드들의 관리에 효율적일 것이라고 판단된다.

2.2. 6LoWPAN

6LowPAN은 IETF 워킹 그룹 중 하나이며, IEEE 802.15.4로 대표되는 저 전력 무선 사설 네트워크(Low-power Wireless Personal Area Network), 센서 네트워크 위에서 인터넷 프로토콜을 사용하기 위한 아키텍쳐 등을 표준화하고 있는 단체이며, 이와 관련된 프로토콜을 의미한다 [1]. 이와 관련하여 여러 개의 RFC 및 드래프트가 존재하는데 이에 대해 간략히 살펴보면 다음과 같다.

RFC 4944는 최소 MTU (Maximum Transmission Unit)가 1,280 옥텟(Octet)인 IPv6(Internet Protocol Version 6) 패킷을 페이로드에 최대 127 옥텟을 담을 수 있는 IEEE 802.15.4 프레임에 전송하기 위한 헤더 압축, 링크 단계 단편화/재조립 매커니즘 등을 수행하는 어플리케이션 계층을 정의하고 있다[2].

draft-ietf-6lowpan-nd는 일반적인 신뢰성 있는 링크(가령 이더넷 등)를 가정하고 동작하는 IPv6 Neighbor Discovery 프로토콜은 멀티 캐스트를 기반으로 하고 있기 때문에, LoWPAN에는 잘 맞지 않는다. 이 드래프트는 멀티캐스트의 사용을 지양하고, LoWPAN 영역에 대해서는 그 경계에 있는 경계 라우터(Edge Router)에 Neighbor Discovery에 대한 기능을 집중시켰다.

Ⅲ. CoAP 주요 특징 및 문제점 분석

3.1. 메시지 포맷

CoAP 프로토콜은 PC 환경등과 비교하였을 때, 매우 빈약한 자원을 위한 프로토콜이기 때문에, 메시지 포맷이나, 길이 등을 최소화하는 것을 주목적으로 개발되고 있는 프로토콜이다. 하지만, 메시지 길이를 최소화하는 경우 상대적으로 계산 처리 및 자원 낭비를 초래할 수 있다. 최근 수개월에 걸쳐 3~4회의 드래프트가 갱신되었는데 다음의 그림1은 이전 드래프트에서의 메시지 포맷를 도식화한 것이고, 그림2는 최근 드래프트에서 단순화된 모습을 보이고 있다.

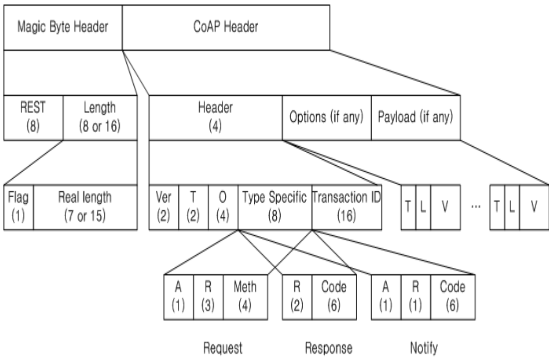

그림 1. 이전 드래프트에서의 메시지 포맷

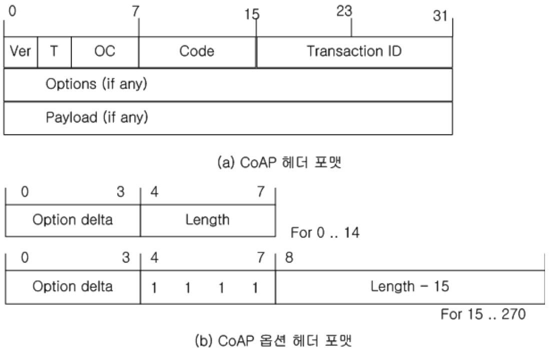

그림 2. 최근 드래프트에서의 메시지 포맷

위 그림1, 그림2에서 보는 바와 같이 매우 단순화되고, 고정된 형태의 메시지 포맷으로 변경되었음을 알 수 있다. 하지만, 그림 2에서 옵션 처리를 위해 기존의 네트워크에서 사용하는 TLV (Type-Length-Value) 형태의 처리를 위한 메시지 포맷을 사용하는데, 옵션의 길이가 짧은 경우 (<=14)와 긴 경우(>=15)에 대해 옵션 헤더의 길이가 가변적으로 변하는 것을 알 수 있다. 이 방법은 1바이트의 메시지 길이를 절약하기 위해 많은 단점을 포함하고 있다. 이는 C언어로 구현을 하는 경우엔 형 변환(Type Casting)을 자유롭게 사용하지 못하게 하는 상황이 발생하는데, 특히 컴퓨팅 능력이 떨어지는 노드들에 직접 적용을 할 때 부하를 발생 시킨다.

3.2. 페이로드에서 실제값 사용

CoAP 프로토콜에서 센싱된 결과를 사용자들이 사용하는 실제 값으로 사용한다. 가령 온도가 23.4도 인 경우, 페이로드에는 "23.4 oC"와 같이 결과를 반환하도록 제시되었다. 하지만 이 방법은

PC환경에서는 아무런 문제가 되지 않지만, 센서 노드들에서는 문제가 생기는데, 가령 센싱 된 오실로스코프 값이 0xABCD와 같을 때, 그 값을 23.4라고 한다면 일반적으로 sprinf(payload, "%.1f", sensing_temperature)와 같은 형태로 PC에서는 간단하게 구현할 수 있는 문제임에도 센서 노드에서는 실수 연산을 하기 위해 스택 오버플로우가 발생하여, 원하는 결과를 얻을 수 없고, 23이라는 값과, 4라는 값을 별도로 처리하는 등의 매우 불합리한 계산을 통해서만 가능하다. 이는 그림 3에서 보는 바와 같이 센서 노드는 PC가 아닌 단순한 센서이기 때문에, 프로토콜이 그대로 적용되는 것보다는, 사용자에게 화면 출력을 하는 단계에서, 온도 등에 대한 실제 값으로 변경하는 것도 고려되어야 한다. 또, 실제 센서들에서는 온도의 실제 값을 알기 위해서는 내부 전압 등이 인자로 사용되기 때문에, 내부전압 등이 센싱 되었을 때 실제 값으로 변환이 가능하다는 문제점을 갖는다. 즉, 센싱 된 결과 값으로 데이터를 송수신하는 것이 바람직하다고 판단된다.

그림3. 센서 노드 및 정보 수집 장치 연결

3.3. 프락시

일반적인 프로토콜 구현에서 프락시는 stateful 프락시와 stateless 프락시로 구분하여 기술되고, 이에 따라 구현되는 경우가 많다. 하지만, CoAP 프로토콜 드래프트에는 이에 대한 명확한 기술이 되지 않았다. 더욱, 개발 과정에서 수차례의 전폭적인 문서 구조가 변경되는 과정에서 최종적으로 프락시 기능을 간단히 기술하고 있다.

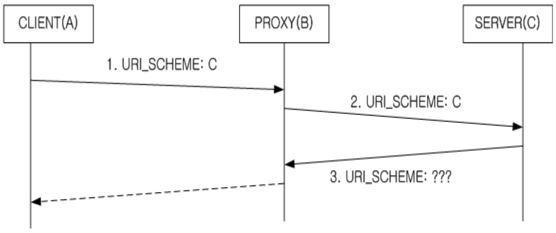

그림 4. 프락시 구조의 문제점 플로우

하지만 그림 4에서와 같은 문제점을 가지고 있는데, 이를 살펴보면 다음과 같다.

그림 4에서 보는 바와 같이 클라이언트(A)가 서버(C)에게 요청 메시지를 전송하려고 하는 경우, URI SCHEME 정보를 옵션에 추가하여 전달한다. 이를 수신한 프락시(B)는 옵션의 URI SCHEME의 주소 정보가 자신의 것이 아니기 때문에, 서버(C)에게 전달하기 위해, 수신한 URI 정보를 수신자 주소로 변경하여 전달한다. 이를 수신한 서버(C)는 결과를 얻은 후, 응답 메시지를 보내기 위해 프락시에게 전달하려고 한다. 하지만, 자신에게 전달한 프락시의 주소 정보는 알 수 있지만, 최종 도착지인 클라이언트의 위치를 알 수 있는 방법이 메시지에는 존재하지 않는다.

이러한 문제점을 해결하기 위한 하나의 방법으로 서버, 클라이언트, 프락시 각각에 대한 설정 파일을 사용하는 방법이 있지만, 이는 근본적인 해결책이 되지 못한다. 근본적인 해결을 위해서는 프락시가 상태를 관리하는 것으로 해결 할 수 있다. 이는 서버가 송수신(특히 송신)한 데이터에 대해 TID(Transaction ID) 및 주소 정보, URI SCHEME 정보 등을 별도로 저장하고, 응답으로 되돌아오는 메시지의 TID 값으로 최초의 메시지 출발 위치인 클라이언트(A)의 위치를 설정하는 것으로 해결 할 수 있지만, 드래프트에서 프락시의 동작이 명확히 기술되어 있지 않다.

Ⅳ. 결론 및 향후 연구 과제

지금까지 살펴본 바와 같이, 온도, 습도 등 각종 원격에 있는 센싱 정보 획득을 위해 소형, 저전력 단말을 효율적으로 관리하기 위해 IETF WG에서 개발 중에 있는 CoAP 프로토콜의 구현 과정에서 생기는 프로토콜의 문제점을 기술하였다. 센서 노드들의 자원이 빈약하기 때문에 메시지 포맷 중 TLV는 고정된 형태로 함으로써 1바이트의 메시지 낭비로 인해 얻을 수 있는 효과가 클 것으로 예상되고, 페이로드에 실제 값을 탑재하는 경우, 성능, 계산 등에 문제점이 있다. 뿐만 아니라, 프락시 서버에 대해 명확히 기술되지 않아, 응답 메시지가 전달이 되지 않는 현상이 발생할 수 있음을 확인하였다.

향후 연구 과제로는 개선된 프로토콜을 기반으로 하여 센서 노드(Mote)에 CoAP-Lite 형태로 탑재하여 CoAP 프로토콜의 동작과정을 확인하고, 이에 따른 문제점 분석을 통해, RFC로 확정되기 위한 개선이 필요하다.

참고문헌

[1] “6LoWPAN", http://ko.wikipedia.org/wiki/6LoWPAN/ 2010. 9

[2] "IPv6 over Low Power WPAN(6lowpan)", http://datatracker.ietf.org/wg/6lowpan/