IoT 환경에서 sleepy node 지원을 위한 경량화 전송 프로토콜 연구

윤주상*, 홍용근**
*동의대학교 멀티미디어공학과
**한국전자통신연구원 표준연구센터
e-mail:jsyoun@deu.ac.kr

A Study on Lightweight Transport Protocol to support Sleepy node in IoT

JooSang Youn*, Yong-Geun Hong**
*Department of Multimedia Engineering, Dong-Eui UniversityI
** Standards Research Center, ETRI

요 약

본 논문에서는 IoT 전송 계층 요구 사항 및 문제점을 기술하고 이를 해결하기 위한 전송 계층 프로토콜을 제안한다. 우선, IoT 환경을 구성하는 디바이스의 경우 물리적으로 많은 제한 조건을 가진 constrained node로 정의된다. 여기서 constrained node는 메모리 크기 및 컴퓨팅 파워가 부족한 노드로 정의된다. 따라서 기존 전송 계층 프로토콜을 그대로 탑재할 수 없다. 더불어 IoT 서비스 특징 중 데이터 전송과 관련된 내용을 보면 데이터 사이즈는 작고, 자주 데이터를 발생 시키지 않는다. 따라서 IoT 디바이스에서 발생하는 데이터는 신뢰적 전송 서비스가 요구된다. 따라서 본 논문에서는 오버헤드가 적은 전송 서비스를 제공하는 전송 프로토콜을 제안하며 더불어 이 프로토콜은 sleepy node를 인지하고 이를 지원하는 IoT 전송 프로토콜이다.

키워드
IoT, Transport protocol, Network,

I. 서 론

본 논문에서는 IoT 전송 계층 요구 사항 및 문제점을 기술하고 이를 해결하기 위한 전송 계층 프로토콜을 제안한다. 우선, IoT 를 구성하는 디바이스의 경우 많은 제한 조건을 가진 constrained node로 정의된다. 현재 IETF lwig WG 내 "Terminology for Constrained Node Networks", draft-ietf-lwig-terminology-00" 문서[1]를 통해 IoT 환경에서 사용될 용어를 정의하고 있다. 이 문서에서는 constrained node를 3개로 구분하고 있다. 이중 class 0, 1로 분류된 노드는 메모리 크기 및 컴퓨팅 파워가 부족한 노드로 정의하고 있다. 따라서 기존 전송 계층 프로토콜을 그대로 탑재할 수 없다. 이런 이유로 기존 전송 계층 프로토콜인 UDP, TCP 프로토콜 중에서 코드 사이즈가 작으며 기능과 동작 과정에 오버헤드가 적은 UDP 사용을 권장하고 있다. 이에 따라 CoAP[2]는 전송 계층 프로토콜로 UDP를 사용하고 있다. 하지만 IoT 서비스 관점에서 보면 UDP는 신뢰적 전송 서비스를 제공하기 못한다. IoT/M2M 서비스 특징 중 데이터 전송과 관련된 내용을 보면 데이터 사이즈는 작고, 자주 데이터를 발생 시키지 않는다. 따라서 IoT 디바이스에서 발생하는 데이터는 신뢰적 전송 서비스가 요구된다.

기존 TCP 프로토콜 기능들을 보면 크게 두 가지로 분류 할 수 있다. 하나는 신뢰적 전송 서비스 기법이며 다른 하나는 높은 성능을 제공받기 위한 High-Throughput 기법이다. 우선, 신뢰적 전송 서비스 기법은 flow control, connection-oriented service 방법 등이 있다. 높은 성능을 제공 받기 위한 기법은 sliding window 및 congestion control 방법 등이다. IoT 응용의 경우 높은 성능 보다는 신뢰적 전송 서비스가 중요한 서비스 요구사항이다. 따라서 TCP 기능 중 sliding window 및 congestion control 방법은 IoT 전송 계층 기능에서 구현이 필요 없는 기능이다. 더불어 위 두 기능은 동작 과정에서도 IoT 디바이스의 에너지 사용 측면에서도 디바이스에 부담을 초래한다.

또한 IoT 네트워크 관점에서 보면 네트워크를 구성하는 IoT 디바이스의 수는 무수히 많을 것으로 예상한다. 따라서 IoT 디바이스에서 발생시키는 작은 크기의 메시지는 접속 네트워크 및 코아 네트워크에 혼잡을 발생시킬 수 있다. 따라서 IoT 디바이스 내에 구현될 전송 계층의 경우 간단한 혼잡 제어 기법이 필요하다. 또한

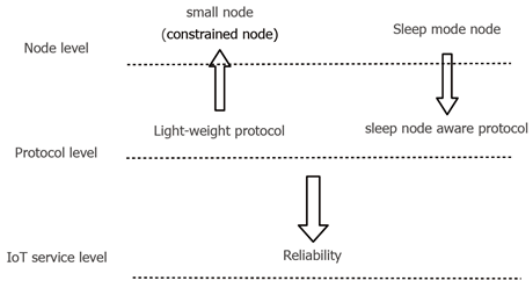

그림 2 IoT 전송 계층 요구 사항 및 서비스

sleepy node에 대한 고려가 필요하다. 위에서 언급했던 것처럼 IoT 디바이스는 다수가 sleepy node일 것으로 예상하며 이런 sleepy node로 인해 전송 계층의 단대단 전송 서비스 수행 시 잘못된 재전송과 같은 불필요한 전송이 이루어진다. 더불어 기존 전송 계층 프로토콜들은 sleepy node를 지원하는 기능을 가지고 있지 않다. 따라서 IoT 전송 계층은 그림 1에 도시된 것처럼 경량화된 전송 프로토콜 및 sleepy node 인지형 프로토콜로 개발되어야 한다.

본 논문에서는 그림 1에서처럼 IoT 디바이스는 small node이면 sleep mode node로 가정한다. 제안할 프로토콜은 경량화된 전송 프로토콜 및 sleepy node 인지형 프로토콜이며 이 프로토콜은 IoT 응용에 신뢰적 전송 서비스를 제공하여야 한다.

II. IoT 디바이스를 위한 전송 프로토콜 디자인

IoT 환경에서 신뢰적 전송 서비스 제공 시 데이터 손실이 발생하는 경우는 그림 2에 도시하고 있다. 손실 지점을 기준으로 구분하며 다음과 같다.

- Congestion loss between CN and Gateway, loss in wireless channel
- Congestion loss in network
- loss in sleepy node

손실지점 중 sleepy node에서 발생하는 손실은 IoT 환경에서 새롭게 추가된 것이다. 따라서 IoT 전송 계층에서는 위에 기술한 손실 지점을 기준으로 손실 데이터에 대한 재전송 방법을 수행하여야 한다.

본 논문에서 제안하는 전송계층은 UDP처럼 프로토콜 오버헤드가 작으며 신뢰적 전송 서비스를 제공한다. 따라서 신뢰적 서비스 제공을 위한 connection setup 과정을 최소화 하고 데이터 전송 과정은 stop and wait 기법을 사용하여 IoT 디바이스 내 메모리 사용을 최소화 한다. 또한 신뢰적 전송과 비신뢰적 전송 서비스를 모두 제공하기 위해 두 서비스를 수행한다. 비신뢰적 전송 서비스의 경우 UDP처럼 데이터 전송 후 신뢰성 제공을 위한 별도의 과정을 수행하지 않는다.

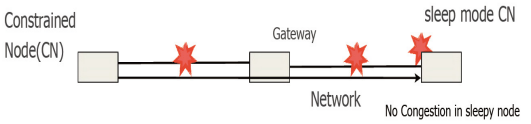

그림 2 IoT 환경에서 손실 발생 예

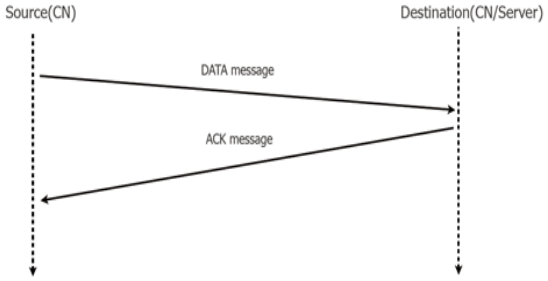

그림 3 IoT 전송 프로토콜 기본 동작

그림 3는 제안하는 전송 프로토콜의 기본 동작이다. 소스 노드 데이터 전송 후 목적지 노드는 데이터 전달에 대한 ACK 메시지를 소스 노드에 전송하는 과정을 기본 동작으로 수행한다. 이 과정에서 소스 노드 응용이 비신뢰적 전송 서비스를 요청할 경우 소스 노드는 데이터 전송을 수행하고 목적지 노드는 받은 메시지에 대한 ACK 메시지를 전달하지 않는다. 만약 소스 노드가 여러 메시지를 전송하며 목적지 노드가 non-sleepy node이며 손실이 없는 경우는 그림 4과 같은 동작이 수행된다.

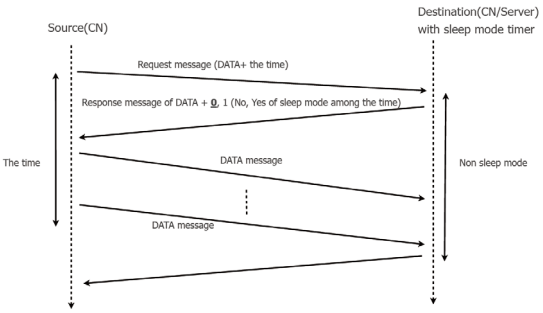

그림 4 Multiple message transmission, no loss, non-sleep mode server의 동작

제안하는 전송 프로토콜은 메시지 번호를 부여하지 않는다. 메시지 번호를 부여할 경우 IoT 디바이스에 부담으로 작용한다. 또한 제안하는 전송 프로토콜은 stop and wait 전송 방식을 사용하기 때문에 메시지 번호가 필요 없다. 그림 4에 도시된 것처럼 여러 메시지를 전송할 경우 소스 노드는 목적지 노드에 자신이 전송할 데이

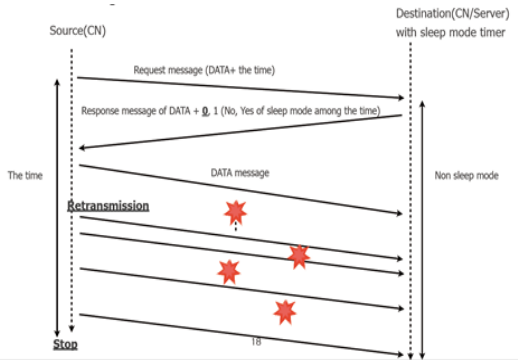

그림 5 Multiple message transmission, no loss, non-sleep mode server의 동작

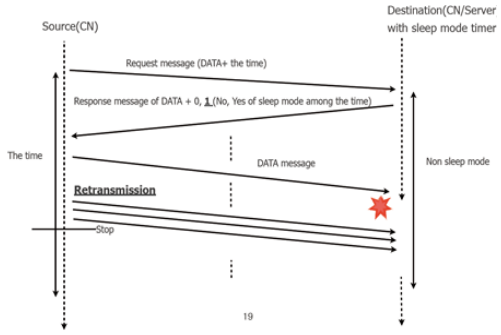

그림 6. Multiple message transmission, no loss, sleep mode server의 동작

터를 전부 전송하기 위해 필요한 시간을 계산하고 요청 메시지에 데이터와 함께 전달한다. 요청 메시지를 전달 받은 목적지 노드는 요청메시지 내에 있는 시간 동안 자신이 sleep 모드로 동작을 할 것인지 아닌지를 판단하고 응용메시지로 유무를 전달한다. 이때 응용메시지는 첫 번째 전달한 메시지의 ACK 메시지를 함께 포함해서 전달한다. 여기서 목적지 노드가 sleep 모드로 동작하지 않는다는 의미의 "0"의 값을 전달할 경우 이 메시지를 받은 소스 노드는 sleepy node에 의해서 발생할 손실이 없다고 가정하고 나머지 데이터 전달을 수행한다. 데이터 전달 과정에서 손실이 발생할 경우 그림 5과 같은 동작이 수행된다. 소스 노드는 손실이 발생하며 네트워크 내 혼잡 손실로 판단하고 기존 TCP 동작처럼 재전송과정을 수행한다. 이때 재전송을 3회 이루어지며 3회 후에도 ACK 메시지를 받지 못할 경우 심각한 혼잡으로 가정하여 더 이상 전송을 수행하지 않는다. 만약 목적지 노드가 시간 동안 sleep mode동작에 대한 가능성을 유로 판단하고 "1" 값을 전달할 경우는 그림6에 도시된 것처럼 동작이 이루어진다. 이 경우는 소스 노드가 데이터 전달 과정에서 목적지 노드의 sleep mode로 인한 손실을 경험하게 되고 sleepy node 손실로 판단할 경우 그림 6에 도시된 것처럼 짧은 재전송 과정을 수행한다. 짧은 재전송 과정을 수행하는 이유는 혼잡손실과의 비교 판단을 위해서이다. 짧은 재전송 과정 후에도 ACK 메시지를 전달 받지 못할 경우 소스는 메시지 전달 과정을 종료한다.

III. 결론

최근 IETF에서는 IoT 환경에서 사용될 인터넷 프로토콜 경량화 방법이 여러 WG에서 진행 되어 왔다. 하지만 기존 경량화가 진행 중인 인터넷 프로토콜들은 IoT 관점에서 이루어 지지 않았다. 따라서 IETF에서는 lwig WG을 통해서 경량화가 진행 중인 또는 경량화된 인터넷 프로토콜에 대한 구현 방법에 대한 분석 결과를 토대로 IoT 환경에서 사용될 IoT 디바이스의 구현 가이드라인을 개발 중이다. 본 논문에서는 IoT 서비스를 위한 전송 계층 설계에 있다. IoT 네트워크를 구성하는 디바이스는 constrained node로 분류된다. 이는 하드웨어 제한조건으로 인해 기존 프로토콜을 그래도 수용할 수 없으며 또한 기존 프로토콜 기능을 전부 구현할 수 없다. 그래도 전송 계층 프로토콜은 오버헤드가 작은 UDP가 사용되고 있다. 하지만 UDP는 데이터 전달과정에서 신뢰적 전송 서비스가 제공되지 못하며 sleepy node를 지원하지 못한다. 따라서 제안한 전송 계층 프로토콜은 낮은 오버헤드의 경량화된 프로토콜이다. 또한 sleepy node를 지원하는 기능이 제공되고 있다. 현재까지 연구된 내용으로 IoT 전송 계층에 기본 동작만을 제안하고 있으며 추후 손실 유형 판단 및 손실 유형에 따른 구체적 재전송 기법 등을 연구할 예정이다.

ACKNOWLEDGMENT

본 연구는 미래창조과학부가 지원한 2013년 정보통신·방송(ICT) 연구개발사업의 연구결과로 수행되었음.

참고문헌

[1] C. Bormann, Ed., "Guidance for Light-Weight Implementations of the Internet Protocol Suite", draft-ietf-lwig-guidance-03 (work in progress), February 25, 2013.

[2] Bormann, C. M. Ersue and Keranen, A, "Terminology for Constrained Node Networks", draft-ietf-lwig-terminology-00 (work in progress), April 23, 2013.