

6LoWPAN상에서 네트워크 코딩 기법을 적용한 코드전파 기법†

김일휴*, 차정우*, 김창훈*, 김동휘*, 남인길*, 이채욱**

*대구대학교 대학원 컴퓨터정보공학과

**대구대학교 정보통신공학부

e-mail:dewwhite@nate.com

An Efficient Dissemination Protocol Using Network Coding Scheme in 6LoWPAN Sensor Network

Il-Hyu Kim*, Jung-Woo Cha*, Chang-Hoon Kim*, Dong-Whee Kim*, In-Gil Nam*, Chae-Wook Lee**

*Dept. of Computer and Information Engineering, Daegu University, Korea

**Dept. of Computer Communication Engineering, Daegu University, Korea

요 약

IP기반 무선 센서네트워크 기술은 센서노드를 인터넷망과 연동하여 다양한 정보를 수집/관리할 수 있음에도 불구하고 주로 제한된 환경에 설치되기 때문에 노드의 오작동이나 성능 개선이 필요한 경우 인위적인 소프트웨어 업데이트 및 업데이트된 노드의 상태를 파악하고 관리할 필요성이 있다. 소프트웨어 업데이트를 위해 필요한 코드 전파 기법은 매우 중요한 기술 중 하나로서, 본 논문에서는 6LoWPAN 프로토콜에서 제공하는 단편 패킷 전송 기법에 네트워크 코딩 기법을 적용한 새로운 업데이트 코드 전파 기법을 제안한다. 제안된 기법은 기존 단편 패킷 전송방식과 비교하여 패킷 전송의 횟수가 감소되었을 뿐만 아니라, 업데이트 코드 전파를 위한 신뢰성 보장, 전체 네트워크에서 전송에 따른 오버헤드 감소 효과를 보인다. 따라서 본 논문에서 제안한 6LoWPAN 프로토콜상에서 코드업데이트 전파 기법을 적용할 경우 속도, 에너지, 네트워크 혼잡도 측면에서 효율적인 업데이트를 수행할 수 있다. 뿐만 아니라 네트워크 코딩의 overhearding 문제점인 원본 데이터의 분실이나 데이터 미 수신시 발생하는 디코딩문제를 미리 정의된 메시지를 이용, 방지함으로써 신뢰성 있는 데이터 전송이 가능하다.

1. 서론

무선 센서네트워크는 제한적인 하드웨어를 가진 수많은 센서 노드들이 특정한 환경에 배치되어 네트워크를 구성한 후 다양한 정보 제공 매체로 활용된다. 최근에는 무선 센서네트워크를 배경으로 한 기존의 인터넷 망과 호환성이 뛰어난 IP기반 USN에 대한 다양한 연구가 이루어지고 있다. IP-USN 기술 중 대표적인 기술로서 6LoWPAN이 있다. IEEE 802.15.4를 기반으로 저전력, 저비용으로 구현한 IP 기반 센서네트워크 기술인 6LoWPAN은 센서 노드에 IP 주소 및 경량화된 IP프로토콜을 올리려는 기술로서 한정된 전력과 데이터 처리량이 많지 않고 IP 기반의 무선 환경 어플리케이션에 적합한 기술이다[1]. 센서네트워크를 인터넷과 상호 연동하여 다양한 정보를 수집/관리할 수 있음에도 불구하고 주로 제한된 환경에 설치되는 센서 노드는 수행 중에 오작동이 발생하거나 성능 개선이 필요한 경우 이를 대처하는데 많은 어려움이 따르게 된다. 센서 노드가 동작 중 문제가 발생했을 때 이를 해결하거나 성능 개선이 요구되는 경우 인위적인 소프트웨어 업데이트 및 업데이트된 노드의 상태를 파악하고 관리할 필요가

있다.

소프트웨어 업데이트를 위해 센서 노드를 프로그래밍하는 방법으로 노드에 직접 연결해서 프로그램을 다운로드 하는 ISP(In-System Programming) 방식이 있으나 센서 노드가 광범위하고 접근이 어려운 지역에 설치될 수 있기 때문에 무선 통신을 이용한 코드 전파 기법이 요구된다[2]. 현재 센서네트워크에서 코드 전파기법으로 주로 사용되는 파이프라인 방식은 소스 노드에서 프로그램 코드의 페이지 하나를 전송하면 동일한 홉의 노드가 페이지를 수신한 즉시 다음 홉으로 페이지를 전송하는 방법이다. 이러한 동작을 반복적으로 수행함으로써 프로그램 코드 전파를 완료한다.

본 논문에서는 6LoWPAN을 사용하는 무선 센서네트워크 환경에서 파이프라이닝 코드 전파 기법을 적용하여 보다 효율적인 코드 전파 기법을 제안한다. 제안된 기법은 6LoWPAN 프로토콜을 사용하는 무선 센서네트워크 환경에서 센서 노드의 상태나 성능을 개선하기 위해 필요한 소프트웨어 업데이트 기법 중, 기존의 업데이트 시스템에서 사용된 코드 전파 기법에 비해 약 2배 정도의 코드 전파 능력이 향상된다. 따라서 본 논문에서 제안한 시스템을 사용할 경우 속도, 에너지, 네트워크 혼잡도 측면에서 효율적인 소프트웨어 업데이트를 수행할 수 있다.

† “본 연구는 지식경제부 및 정보통신산업진흥원 의 대학 IT연구센터 지원사업의 연구결과로 수행되었음”

2. 6LoWPAN[3]

인터넷망(IPv6)에서 전송 가능한 IPv6 데이터(1280Byte)를 무선 센서네트워크상에서 전송하기 위해 6LoWPAN에서 제공하는 기술적인 특징들은 다음과 같다. 단편과 재조합, 자동주소할당 기법, LOAD 라우팅 프로토콜 등 IP기반 센서네트워크의 요구사항을 만족시키기에 적합한 여러 가지 기술적인 특징들을 가지고 있다. 6LoWPAN에서 정의된 기술들을 아래에 요약하였다.

단편은 여러 개의 IEEE 802.15.4. 프레임이 나뉘어져 도착하는 것을 말하는데 IEEE 802.15.4에서 전송 가능한 프로토콜 데이터 단위(PDU)는 81바이트 정도로 작은 크기이지만, IPv6의 최대 전송 크기(MTU)는 1280바이트로 큰 차이를 보이기 때문에 패킷 단편화를 지원하지 않는 IPv6 프로토콜의 특성 상 두 프로토콜 사이에서 전송이 가능하도록 하기 위해 단편과 재조합을 담당하는 Adaptation Layer가 반드시 존재하여야 하며, IP계층 아래에서 제공되어야 한다. 단편화 기능과 함께 IPv6 헤더 압축도 중요한 기능 중의 하나이다.

IPv6 헤더 압축 기능은 기존 IPv6 헤더를 압축하여 크기를 줄이는 기술로 IEEE 802.15.4 PDU의 가용 Payload 공간을 늘리는데 목적이 있다. 헤더를 압축하지 않고 데이터를 전송하게 되면 데이터 크기가 수십 바이트에 불과 하더라도 과도한 단편과 재조합이 일어나게 되고 이는 센서 노드의 에너지나 처리량에 있어서 상당한 자원의 낭비를 불러오게 된다. 따라서 IPv6 헤더 압축은 반드시 제공되어야 한다.

주소 자동 할당 기능은 센서노드가 스스로 IP주소를 생성하고 검증을 통해 중복되지 않는 유일한 주소를 부여하는 방법이다. 주소 생성 방법은 노드가 동작을 시작하게 되면 MAC계층의 주소를 얻어오게 되고 얻어온 주소는 64비트 주소와 16비트 주소 두 가지로 구분할 수 있다. 호스트에 대해서 작은 생성 오버헤드를 지니고 있기 때문에 비상태형 주소생성 방식을 사용하여 이는 LoWPAN에 보다 적합한 특징을 가진다. 생성된 주소는 IPv6주소를 사용하기 때문에 외부 네트워크와 연결 시 하나의 장치를 식별하기 위한 추가적인 기능을 필요로 하지 않는다.

LOAD 라우팅 프로토콜은 멀티 홉 메시 네트워크에 적합한 라우팅 기술로 AODV에 기반한 요구기반 라우팅 프로토콜이다. 6LoWPAN에서 제공하는 LOAD 라우팅은 IP Layer가 아닌 그 밑의 Adaptation Layer에서 수행된다. Adaptation Layer에서 라우팅을 수행하게 되면 효율적인 IPv6헤더 압축을 제공할 뿐만 아니라 64비트 주소 또는 16비트 주소 사용이 가능하여 라우팅 테이블의 크기도 줄일 수 있다.

3. 소프트웨어 업데이트 기법

소프트웨어 업데이트 기법이란 기존의 ISP(In-System Programming)과 같이 유선을 이용해 직접 센서 노드의 프로그램을 업데이트 하는 것과 달리 무선통신을 이용해 특정 환경에 배치된 수십~수천개 노드의 프로그램을 모두 업데이트하는 방법이다[4].

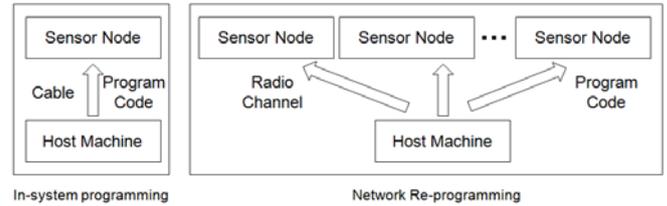


그림 1. In-system 프로그래밍 VS. 네트워크 Re-programming

무선 통신을 이용한 소프트웨어의 업데이트 시 필요한 구성요소는 1) Preparation Mechanism(패킷의 길이보다 데이터가 큰 경우 네트워크 진입 전 데이터를 분산시킴), 2) Dissemination Mechanism, 3) 손실된 패킷과 데이터의 모든 부분을 받기 위한 Reliability Mechanism(업데이트는 데이터가 완전하게 받지 않은 경우에는 사용할 수가 없으므로 손실된 패킷과 완전한 데이터를 받기 위함)이 있다. 또한 설계목적에 따라 Message Suppression(다양한 프로토콜의 불필요한 메시지를 제어), Sender Selection(이웃 노드들 간의 활성화된 sender들의 숫자를 제한), 전달방식 등이 추가될 수 있다[5].

효율적인 코드 전파를 위해 다양한 조건을 고려해야 하며, 대표 적인 고려사항으로는 신뢰성(Reliability, 센서노드는 프로그램 이미지 수신에 완전할 때 사용이 가능), 일치성(Consistency, 네트워크 내의 모든 노드가 전체 프로그램 이미지를 수신하고 동일한 버전으로 업데이트가 되어야 함), 작은 메모리(Small Memory, 업데이트 프로토콜은 작은 메모리 footprint를 가져야함), 에너지 효율성(Energy Efficient, 업데이트 프로토콜은 에너지 사용을 최소화), 혼란 최소화(Minimize Disruptions, 업데이트 프로토콜은 주된 어플리케이션이 아니기 때문에 주된 어플리케이션의 기능에 영향을 주면 안됨), 결함 허용(Fault Tolerance, 노드가 고장 나거나 추가되는 것을 허용)등이 있다[6].

코드를 전파를 위한 방법으로는 리플(Ripple) 전달 방식과 파이프라인(Pipeline) 전달 방식이 있다. 리플 전달 방식은 노드가 주기적인 공개 메시지를 브로드캐스트하여 자신의 현재 코드 버전을 이웃하는 모든 노드들에게 알리고 이웃 노드들은 공개 메시지를 수신하여 상위 코드 버전 노드에게 신청 메시지를 요청 후 코드 업데이트 데이터를 전달 받게 된다. 하위 코드 버전 노드가 이웃하는 상

위 코드 버전 노드에게 코드 업데이트를 모두 전달 받게 되면 자신의 코드를 새로운 버전으로 업데이트를 하고 다시 이웃하는 다른 하위 코드 버전 노드들에게 전달하기 위해 주기적으로 공개 메시지를 브로드캐스트 한다. 이러한 흐름을 계속적으로 반복하여 코드 업데이트 데이터를 전파한다.

파이프라인 전달 방식은 소스 노드에서 프로그램 코드를 일정한 크기의 페이지로 분할한다. 그 분할된 페이지 하나를 전송하면 동일한 홉의 노드가 페이지를 수신하자마자 다음 홉으로 수신한 페이지를 전송한다. 이러한 동작을 반복적으로 수행하게 되고 모든 페이지 수신이 완료되면 프로그램 코드 전파는 완료된다. 파이프라인 방식은 리플 전달 방식에 비해 코드 전파 시간뿐만 아니라 각각의 단계에서 idle listening 시간을 단축함으로써 에너지 소모도 줄일 수 있다.

4. 6LoWPAN상에서 코드 전파 기법

6LoWPAN 프로토콜을 사용하는 무선 센서네트워크 환경에서 네트워크 혼잡도와 에너지 절약 측면에서 효율적인 코드전파 기법을 제안한다. 제안된 기법은 Dissemination protocol 중 코드 전파 기법인 파이프라인 전달 방식에 네트워크 코딩 방법을 적용하여 코드 전파시 발생하는 네트워크 과부하를 감소시킨다. 또한 기존에 단편 패킷을 전달하기 위해서는 단편 정보를 단편헤더에 표시 하였지만, 단편 정보를 수신하는 노드가 테이블을 두고 관리함으로써 데이터의 단편 횟수를 감소시킨다. 뿐만 아니라, 네트워크 코드 전파 시 신뢰성을 높이기 위해 사용되는 Ack 메시지 전송을 따로 수행하지 않고 데이터 전달 시 함께 수행할 수 있게 된다. 이는 센서 노드의 데이터 전송 횟수를 감소시킬 뿐만 아니라, 에너지 절약 측면에서도 효율성을 가진다.

업데이트를 위해 전달되는 단편 패킷을 브로드캐스트로 전달하고, 신뢰성 보장을 위한 Ack 메시지를 동시에 송/수신할 수 있는 헤더가 필요하기 때문에 새로운 단편 헤더를 제안하였고, 그림 2에 나타내었다.

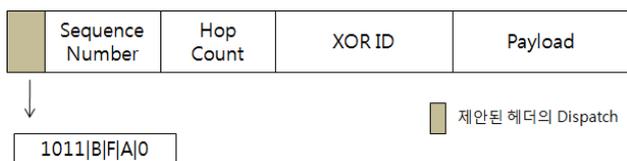


그림 2. 제안한 헤더 포맷

Sequence Number는 단편된 데이터의 순서를 나타낸다. 이 필드의 본래 기능은 데이터를 송/수신할 때 데이터의 순서를 보장하는 것이지만, 추가로 Ack 정보가 포함된 데

이터를 수신할 경우 몇 번째 단편 패킷에 대한 Ack 인지 판단하기 위해 사용된다. Hop Count는 한 패킷이 무제한 전달되는 것을 방지하는 본래의 역할에, 각 Sector를 구분 짓는 기능을 위해 추가하였다. Sector는 동일한 단편 패킷을 전달 받은 홉 수에 따라 나뉘게 되며, 이는 전달된 패킷이 역 방향으로 전달되는 것을 방지한다. XOR ID는 originator 주소와 센서 노드 자신의 MAC 주소를 XOR한 값으로 Ack 데이터를 보낸 노드를 식별하는 기능과 Originator 주소를 식별하기 위해 추가하였다. 그림 3은 XOR field가 포함된 헤더를 송/수신하는 과정을 나타낸 것이다.

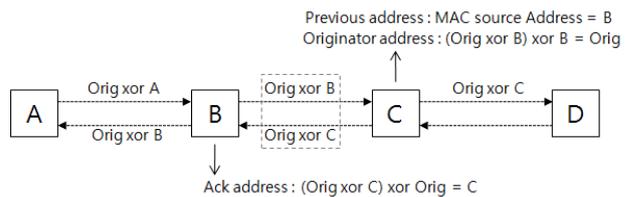


그림 3. XOR 헤더 송/수신 과정

두 번째 이후 단편 패킷을 Originator(A)는 브로드캐스트로 전달을 시작한다. 이 패킷을 전달 받은 노드중 하나(B)는 패킷의 MAC 주소를 통해 패킷을 보낸 노드의 주소를 식별하고, XOR 헤더를 통해 단편 테이블을 검색한다. 검색된 테이블의 Sequence number 값을 패킷의 Sequence number와 비교하여 만약 기존에 받지 않은 데이터이면 재조합 과정을 수행한다. 또한 다음 노드로 데이터 전달과 이전 노드로 Ack를 보내기 위해 Originator 주소와 자신의 주소를 xor 해서 XOR 헤더에 넣고, Ack bit를 set한 후 재전송을 한다. 다음 홉의 노드(C)가 이 패킷을 수신하면, B와 동일한 동작을 수행한다. 이때 B도 C에서 보낸 패킷을 수신하게 된다. B는 Ack bit이 set 되었는지 판단한 후 만약 set 되어있으면 Hop count 정보를 살펴본다. 이 Hop count 정보가 현재 자신이 보유한 테이블의 정보보다 큰 경우 다음 Sector에서 전달된 패킷임으로 인식하게 된다. B는 XOR 헤더의 정보를 다시 Originator 주소로 xor함으로써 C가 보낸 Ack 라는 것을 알 수 있다. 또한 Sequence Number field를 통해 몇 번째 단편 데이터의 Ack를 수신한지 알게 된다. Ack 정보는 업데이트에 관여한 모든 노드가 테이블을 두고 관리하며 노드가 모든 업데이트 코드를 전달 받았는지 체크하는데 사용한다. 코드 전송이 완료된 노드는 표 1에 나타난 업데이트 업데이트 테이블 정보를 판단하여 모든 노드가 데이터를 전달 받았는지를 스스로 검사하고 만약 일부 손실된 데이터가 존재할 경우 재전송을 함으로써 재전송에 따른 오버헤드를 감소할 수 있다.

두 번째 이후 단편 패킷을 전달하기 위해 헤더에서 생

표 1. 업데이트 테이블

Field	내용
Neighbor Address	이웃 노드의 주소
Fragment Number	단편된 패킷의 개수
Check bit	수신한 Ack를 bit 별로 표시

략된 정보는 첫 번째 단편 패킷을 수신한 모든 노드들이 단편 테이블을 생성하고 헤더 정보를 기록하여 이 후 생략된 헤더의 단편 패킷을 수신할 경우 헤더를 판별하는데 사용된다.

그림 4는 기존에 6LoWPAN에서 사용되는 헤더 포맷으로 첫 번째 단편 패킷을 브로드캐스트 하기 위해 필요한 헤더로 구성되어 있다. 첫 번째 단편 패킷의 헤더는 그대로 사용하며, 이 정보는 수신하는 노드가 다음 헤더 정보를 식별하기 위해 업데이트 테이블에 기록하고 사용한다.

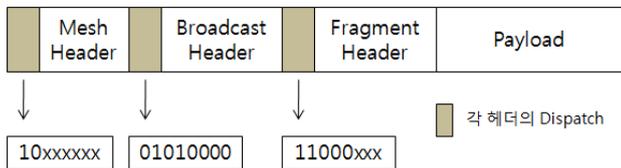


그림 4. 브로드캐스트/첫 번째 단편 헤더

5. 성능평가

본 장에서는 6LoWPAN상에서 제공되는 단편 패킷 전송 기법을 이용하여 업데이트 코드 전파 기법을 제안하였다. 기존의 파이프라인 전송 방식을 사용한 코드 전파기법과 비교하여 정량적인 계산에 의한 이론적인 성능을 분석하여 그림 6에 나타내었다. 아래 그래프는 기존의 파이프라인 방식을 통한 코드 전파 기법과 제안한 기법을 비교하여 데이터 송수신 횟수에 따른 결과를 나타낸 비교그래프이다. 기존의 파이프라인 전송 기법은 처음 데이터를 전송하기 위해서는 Advertisement, request, data, ack 메시지와 같은 4번의 전송이 이루어진 후 data, ack 메시지를 전송한다. Data, ack 메시지를 송/수신하기 위해서는 기본적으로 2번의 전송과 2번의 수신이 필요하다. 하지만, 본 논문에서 제안한 방식을 적용하면 data와 ack 메시지가 한 번의 전송으로 이루어지기 때문에 1번의 전송과 1번의 수신으로 업데이트 코드 전파가 가능하다. 또한 6LoWPAN 프로토콜에서 제공되는 단편 기법과 제안한 단편 기법을 비교한 결과를 그림 7에 나타내었다. 제안한 단편 전송 방식은 기존의 방식에 비해 약 7% 정도 전송 횟수가 감소됨을 보였다. 따라서 본 논문에서 제안한 방식은 기존의 방식과 비교하여 약 2배의 성능향상을 보인다.

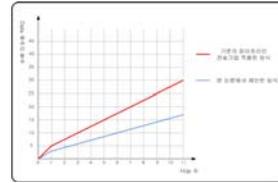


그림 6. Data 송수신 횟수에 따른 비교

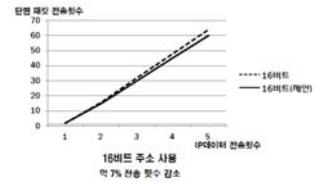


그림 7. 단편 패킷 전송 결과

6. 결론

센서 노드는 제한된 환경에 설치되기 때문에 노드의 오작동이나 성능 개선이 필요한 경우 인위적인 소프트웨어 업데이트 및 업데이트된 노드의 상태를 파악하고 관리할 필요성이 있다. 본 논문에서는 6LoWPAN 프로토콜을 사용하는 무선 센서네트워크 환경에서 소프트웨어 업데이트를 위한 코드 전파 기법을 제안하였다. 제안한 기법은 기존 6LoWPAN 프로토콜에서 제공하는 단편 패킷의 전송 횟수를 감소시키고 코드 전파 시 최소 전송 횟수로 신뢰성 보장뿐만 아니라 네트워크 혼잡도를 감소시킬 수 있게 된다. 이는 업데이트 코드 전파의 속도, 에너지, 네트워크 혼잡도 측면에서 효율적인 업데이트를 수행할 수 있을 뿐만 아니라 네트워크 코딩의 overhearding 문제점인 원본 데이터의 분실이나 데이터의 미 수신시 발생하는 디코딩 문제를 미리 정의된 메시지를 이용, 방지함으로써 신뢰성 있는 데이터 전송이 가능하다.

참고문헌

- [1] Geoff Mulligan, "The 6LoWPAN architecture," ACM, Proceedings of the 4th workshop on Embedded networked sensors, pp. 78-82, 2007.
- [2] Thanos Stathopoulos, John Heidemann, and Deborah Estrin. "A Remote Code Update Mechanism for Wireless Sensor Networks", Technical Report CENS-TR-30, University of California, Los Angeles, Center for Embedded Networked Computing, November 2003.
- [3] G. Montenegro and N. Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC4944, IETF, September 2007.
- [4] C.-C. Han, R. Kumar, R. Shea, and M. Srivastava, "Sensor Network Software Update Management: A Survey," International Journal on Network Management, Vol.15, No.4, pp.283-294, July 2005.
- [5] Patrick E. Lanigan, "Disseminating Code Updates in Sensor Networks", CMU-ISRI-05-122, October 2005.
- [6] Jonathan W. Hui and David Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale", ACM Press, pp. 81 - 84, 2004.