

무선 센서 네트워크 상에서 network coding을 이용한 효율적인 코드 전파 기법

차정우, 김일휴, 김창훈, 권영직

대구대학교 대학원 컴퓨터정보공학과
경북 경산시 진량읍 내리리 15번지 대구대학교
Tel: 053-850-6578, E-mail: cju@dsp.daegu.ac.kr

An Efficient Dissemination Protocol Using Network Coding in Wireless Sensor Network

Cha Jeong Woo, Il Hyu, Kim, Chang Hoon Kim, and Young Jik Kwon
Dept. of Computer and Information Engineering, Daegu University, Korea
Jinryang, Gyeongsan, Gyeongbuk, 712-714, Korea
Tel: 053-850-6578, E-mail: cju@dsp.daegu.ac.kr

요약

소프트웨어 업데이트를 위한 업데이트 코드 전파 기법은 매우 중요한 기술 중 하나이다. 본 논문에서는 네트워크 코딩 기법을 이용한 새로운 업데이트 코드 전파 기법을 제안한다. 제안된 코드전파 기법은 기존의 파이프라이닝 방식에 비해 데이터 송수신 횟수에 있어 약 49%의 성능 향상을 보인다. 따라서 본 논문에서 제안한 코드 업데이트 기법을 사용할 경우 속도, 에너지, 네트워크 혼잡도 측면에서 효율적인 소프트웨어 업데이트를 수행할 수 있다. 뿐만 아니라 본 논문에서 제안한 방식은 네트워크 코딩의 overhearing 문제점인 원본 데이터의 분실이나 데이터의 미 수신시 발생하는 디코딩문제를 미리 정의된 메시지를 이용, 방지함으로써 신뢰성있는 데이터 전송을 가능하게 한다.

Keywords

Dissemination Protocol, Network Coding, 무선 센서네트워크

1 서론

무선 센서 네트워크는 제한적인 하드웨어를 가진 수많은 센서 노드들이 특정한 환경에 배치되어 네트워크를 구성한 후 다양한 정보 제공 매체로 활용된다. 센서 노드는 한번 배치되면 사람의 간섭 없이 오랜 기간 동안 동작하지만 수행 중에

오작동이나 성능 개선이 필요한 경우 인위적인 소프트웨어 업데이트가 필요하다[1-6]. 센서 노드를 프로그래밍하기 위한 방법으로 노드에 직접 연결해서 프로그램을 다운로드하는 ISP(In-System Programming)방식이 있으나 센서 노드가 광범위하고 접근이 어려운 곳에 배치될 수 있기 때문에 무선 통신을 이용한 코드 전파 기법이 요구된다[1].

센서 네트워크에서 코드전파 방법에 관한 대표적인 연구결과로는 XNP[2], MOAP[3], Deluge[4], MNP[5] 등이 있다. XNP[2]는 싱글 홉(hop) 내에서 센서 노드의 TinyOS 커널과 응용 코드를 업데이트 하기 위해 개발되었으며, Code capsules로 전체 파일을 전송한다. MOAP[3]는 리플 전달 프로토콜을 사용하여 멀티 홉으로 코드 업데이트 데이터를 전달하는 기법이다. 데이터 전달에 있어서 플로딩(flooding)보다 향상된 기능을 제공하고 XNP의 1홉의 제한적인 전달 방식을 보완하였다. Deluge[4]는 멀티홉을 지원하고, 효율적인 데이터 전송을 위해 고정 크기의 페이지 단위 파이프라이닝 방식과 3단계 handshaking 프로토콜 사용한다. MNP [5]는 멀티 홉을 지원하고 데이터는 고정 크기 페이지 단위의 spatial multiplexing을 이용하여 전송한다. 또한 3단계의 handshaking 프로토콜을 사용하고 전송자를 선택할 수 있는 알고리즘을 사용함으로써 코드 전파의 불필요한 부분을 제거하였다.

센서 네트워크에서 코드 전파 기법으로는 리플 전달 프로토콜과 파이프라이닝 방식이 있다[6]. 리플 전달 프로토콜은 소스 노드에서 전송된 프로그램 코드를 동일한 홉의 노드가 전체 코드를 수신하면 수신한 코드를 다음 홉으로 전송하는 방식이다. 파이프라인 방식은 소스 노드에서

프로그램 코드의 페이지 하나를 전송하면 동일한 홉의 노드가 페이지를 수신하자마자 다음 홉으로 수신한 페이지를 전송한다. 이러한 동작을 반복적으로 함으로써 프로그램 코드 전파를 완료한다.

본 논문에서는 파이프라이닝 코드 전파 기법에 네트워크 코딩[7]을 적용하여 보다 효율적인 코드 전파 기법을 제안한다. 제안된 코드전파 기법은 기존의 파이프라이닝 방식에 비해 데이터 송수신 횟수에 있어 약 49%의 성능 향상을 보인다. 따라서 본 논문에서 제안한 코드 업데이트 기법을 사용할 경우 속도, 에너지, 네트워크 혼잡도 측면에서 효율적인 소프트웨어 업데이트를 수행할 수 있다. 뿐만 아니라 본 논문에서 제안한 방식은 네트워크 코딩의 overhearing 문제점인 원본 데이터의 분실이나 데이터의 미 수신시 발생하는 디코딩문제를 미리 정의된 메시지를 이용, 방지함으로써 신뢰성있는 데이터 전송을 가능하게 한다.

2 관련연구

무선 통신을 이용한 소프트웨어의 업데이트시 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 Mechanism이 있다. 또한 설계목적에 따라 Message Suppression, Sender Selection, 전달방식 등이 추가될 수 있다[8].

Dissemination protocol은 아래의 그림 1과 같이 무선통신을 이용해서 기존에 배치된 무선 센서 노드의 소프트웨어 업데이트를 위한 코드 전파 프로토콜이다.

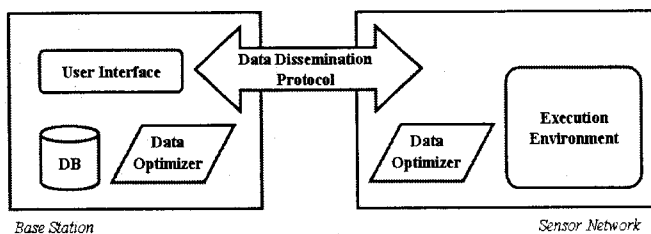


그림 1 - Dissemination Protocol[1]

효율적인 코드 전파를 위해 다양한 조건을 고려해야 하며, 대표 적인 고려사항으로는 신뢰성(Reliability), 일치성(Consistency), 작은 메모리(Small Memory), 에너지 효율성(Energy Efficient), 혼란 최소화(Minimize Disruptions), 결함 허용(Fault Tolerance)등이 있다[8].

일반적인 Dissemination 프로토콜은 아래 그림 2와 같이 새로운 코드의 advertisement, 소스 노드 선택, 대상노드로의 데이터 전송과 같이 3단계로 이루어진다[9].

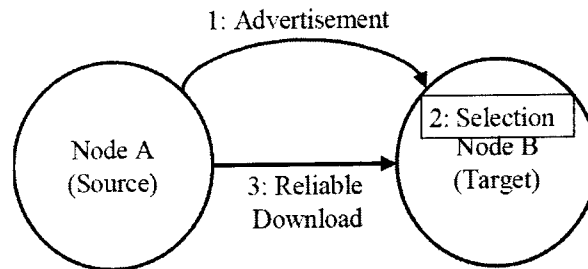


그림 2 - Dissemination Protocol[6]

2.1 XNP[2]

XNP는 싱글 홉(hop) 내에서 센서노드의 TinyOS 커널과 응용 프로그램을 업데이트 하기 위해 개발되었다. 전송 데이터 코드의 양을 줄이기 위해 코드를 캡슐형태로 분할하여 전체 파일을 전송한다. 전체적으로는 네트워크 프로그래밍 모듈과 부트로더로 구성되어있으며, 동작순서는 코드 전송 준비, 코드 전송, 검증, Reprogramming 과정으로 이루어진다.

2.2 MOAP[3]

MOAP는 리플 전달 프로토콜을 사용하여 멀티 홉으로 업데이트 코드를 전달하는 기법이다. 데이터를 전달하는데 있어 플로딩 방식 보다 네트워크 부하 및 중복된 데이터 전송을 줄인다. 또한 싱글 홉에 제한적인 XNP의 단점을 보완하였으며, 업데이트가 완료되기 전까지 업데이트할 코드는 안정된 저장공간에 보관된다. 슬라이딩 윈도우(Sliding Window) 방식에 기반한 손실된 패킷의 재전송 기법을 사용함으로써 전송 시 발생하는 데이터의 손실을 보증한다. 그러나 이 기법은 업데이트 코드 데이터 전송이 이루어지는 동안 싱글 홉 내 다른 모든 노드들은 수신 대기(idle listening) 상태에 있다. 따라서 업데이트에 관여하지 않는 모든 노드들은 불필요한 에너지를 소비하게 된다.

2.3 Deluge[4]

Deluge[4]는 멀티홉을 지원하고, 효율적인 데이터 전송을 위해 고정 크기의 페이지 단위 파이프라이닝 방식과 3단계 handshaking 프로토콜 사용한다. 그림 3은 Deluge의 기본적인 동작과정을 나타내며, 세부적인 절차는 다음과 같다. (a) 소스노드 A가 새로운 버전 2인 프로그램의 advertisement를 브로드캐스트하면 이를 수신한 노드들은 노드 A에게 0번째 페이지를 요청하고 (b) 노드 A는 수신자가 요청한 페이지를 패킷단위로 전송(노드 C, D는 새로운 코드 수신완료), (c) 노드 B는 손실된 코드의

재전송 요청, (c) 노드 C, D는 전송자 노드가 되어서 advertisement 브로드캐스트를 수행한다.

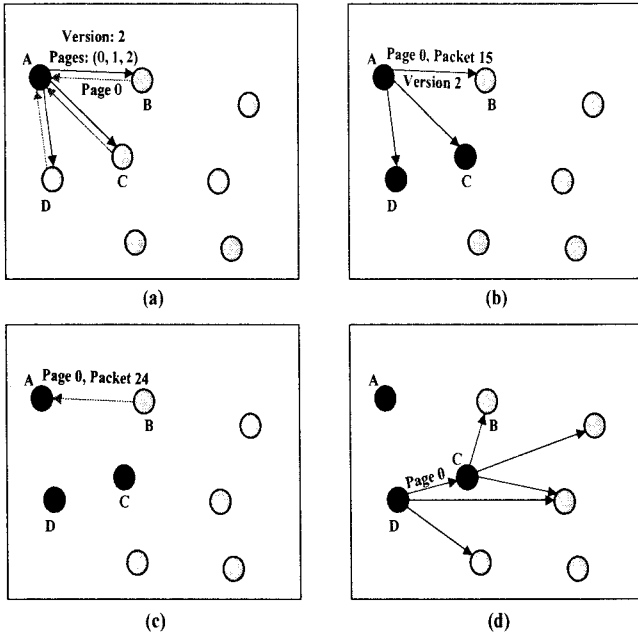


그림 3 - Deluge의 수행 예제

2.4. MNP[5]

MNP는 멀티 홉을 지원하고 데이터는 고정 크기 페이지 단위의 spatial multiplexing을 이용하여 전송한다. 또한 3단계의 handshaking 프로토콜을 사용하고 전송자를 선택할 수 있는 알고리즘을 사용함으로써 코드 전파의 불필요한 부분을 제거하였다. 손실된 코드를 복구하기 위해 비트 벡터 테이블을 사용하며, 5개의 기본적인 기능 상태(IDLE, DOWNLOAD, ADVERTISE, FORWARD, SLEEP)로 구성된 상태머신을 통하여 코드전파와 신뢰성 메커니즘을 제공한다.

2.5. Infuse[8]

Infuse는 다른 코드전파 기법과 다르게 TDMA(Time Division Multiple Access)방식을 사용한 프로토콜이다. Implicit acknowledgement와 back pressure 메커니즘을 사용함으로써 전송의 신뢰성을 제공한다. 즉, 실패된 센서노드를 처리하기 위해, 센서노드가 정해진 딜레이 후에 implicit acknowledgement를 얻지 못하면 센서노드는 재전송을 한다. 또한 에너지 절약을 위해 TDMA방식의 Listening 기법을 사용하고 있다.

2.6. Sprinkler[10]

Sprinkler는 Deluge, Infuse, MNP, PSFQ에 비해 에너지 소비를 줄였다. 낮은 복잡성의 CDS 알고리즘을 제공하고 있으며, Infuse와 같이 TDMA방식을 이용하여 데이터를 전송한다.

3. Network coding을 이용한 코드 전파 기법

본 절에서는 무선 네트워크상에서 일반적인 데이터 전송에 사용되는 네트워크 코딩 기법을 소개하고 네트워크 코딩을 적용한 무선 업데이트 코드 전파 기법에 대해 설명한다.

3.1. 네트워크 코딩

네트워크 코딩이란 네트워크 계층에서 이루어지는 코딩 기법으로써 통신 자원을 효율적으로 사용하여 통신 네트워크의 전송 속도를 향상시키기 위해 개발되었다[7]. 네트워크 코딩의 주된 목적은 정보의 압축이며, 디코딩 절차는 통신 네트워크의 여러 경로로부터 수신된 데이터의 정보를 이용한다. 네트워크 코딩은 최초 유선상에서 효율적인 멀티캐스트를 위해 고안되었으나, 최근 무선 상황에서도 구현의 가능성이 증명되었다.

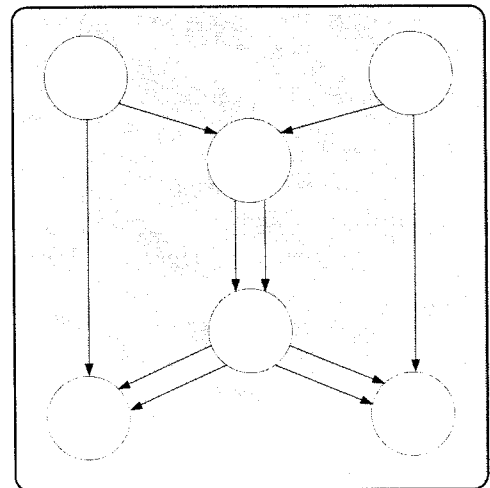


그림 4 - 기존의 butterfly 네트워크

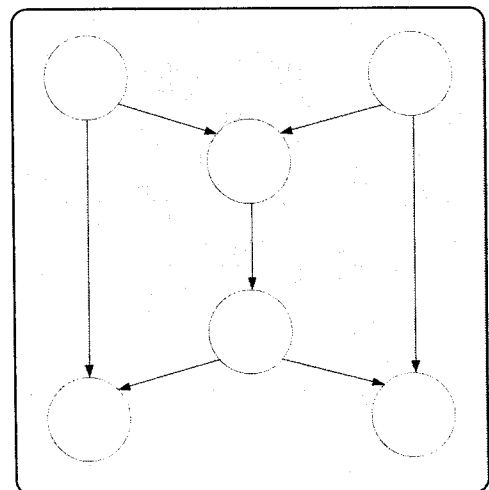


그림 5 - Butterfly 네트워크(네트워크 코딩 적용)

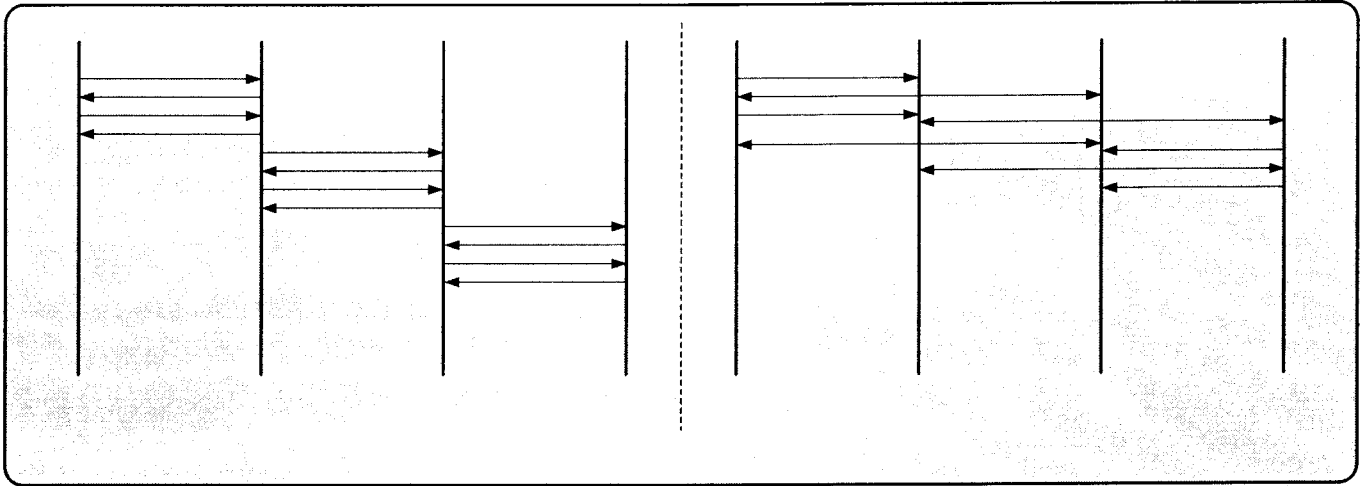


그림 6- 네트워크 코딩을 적용한 파이프라인 방식

[7]에서 제안된 네트워크 코딩의 기본적인 절차는 그림 4, 5와 같다. 그림 4, 5에 기술된 바와 같이 두 개의 소스노드 A, B가 각각 a, b 정보를 C, D에 전달하기 위해 C와 D는 a, b 모두 전달받아야 한다. 그림 4를 보면 기존의 네트워크에서는 a와 b를 받은 E에서 a를 전송 후 b를 전송해야 한다. 그러나 그림 5를 보면 E에서 a와 b를 XOR해서 전송 함으로써 두 번 전송해야 할 것을 한 번의 전송으로 해결한다. 또한 그림 5의 F에서도 C와 D에 데이터 a, b를 전송한다면 $a \oplus b$ 결과를 보냄으로써 코드 전송 횟수를 단축 시킬 수 있다. C와 D는 첫 번째 슬롯에서 전송된 a와 b를 보관하고 있다. F에 의해서 전달된 $a \oplus b$ 에 각각 a와 b를 XOR함으로써 보관하고 있지 않았던 새로운 정보 b와 a를 각각 얻게 된다.

이는 멀티캐스트 네트워크에서 선형 코드를 기반으로 한 네트워크 코딩을 적용하면, 이론적으로 최적의 네트워크 효율을 얻을 수 있다는 결과를 보여주고 있다.

네트워크 코딩은 수신 노드에서, 전달받은 코딩된 정보를 얼마나 성공적으로 디코딩할 수 있는지에 따라 그 효율성이 좌우된다. 성공적인 디코딩을 위해서는 각 노드마다 디코딩을 위한 정보를 필수적으로 사전에 가지고 있어야 가능하다.

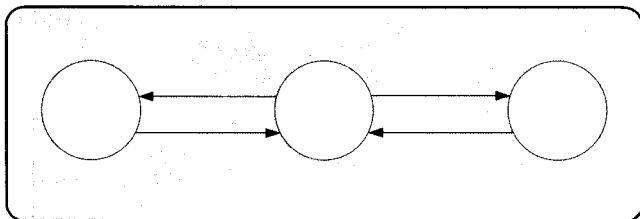


그림 7 - 양방향 선형 네트워크

그림 7과 같이 선형으로 구성된 네트워크에서 양쪽 끝에 존재하는 두 개의 노드가 상호 정보 교환을 중간의 중계 노드를 활용하여 한다면,

네트워크 코딩의 적용이 비교적 용이하게 이루어질 수 있다. 이 경우에는 일종의 overhearding 문제는 존재하지 않게 된다.

3.2. 네트워크 코딩을 적용한 파이프라인 방식

업데이트 코드의 데이터 전달방식에는 크게 리플 전달 방식과 파이프라인 전달 기법이 있다. 파이프라인 전달 기법은 리플 전달 방식에 비해 코드 전파 시간이 단축되며 Idle listennig 시간을 줄임으로써 에너지를 절약할 수 있다.

그림 6의 (a)에 기존 Deluge[4]의 파이프라인 전달 방식을 도식화 하였다. 파이프라인 전달 방식은 소스 노드에서 프로그램 코드의 페이지 하나를 전송하면 동일한 홉의 노드가 페이지를 수신하자마자 다음 홉으로 수신한 페이지를 전송한다. 이러한 동작을 반복적으로 수행함으로써 프로그램 코드 전파를 완료하게 된다.

리플 전달 프로토콜 방식보다 효율적이지만 한 페이지를 송신하기 위해서는 노드사이에 4번의 통신이 필요하다. 따라서 노드수의 수가 증가하면 할수록 에너지 소모량은 기하급수적으로 늘어나게 된다.

본 논문에서는 Dissemination protocol 중 코드전파 기법인 파이프라인 전달 방식에 네트워크 코딩을 적용하여 보다 효율적인 전파 기법을 제안한다. 3.1절에서 설명하였듯이 네트워크 코딩은 한 번의 전송으로 2개의 정보를 보낼 수 있는 기법이다. 즉, 기존의 파이프라인 전달 방식에 동시에 보낼 수 있는 정보를 보냄으로써 노드수가 증가함에 따라 전송에 필요한 정보는 감소하게 된다. 따라서 에너지뿐만 아니라 전송속도, 네트워크 혼잡을 줄일 수 있다.

아래 그림 8와 같이 네트워크 전송과정에서 Adv와 Ack, Req는 미리 정의 되어있다고 가정한다.

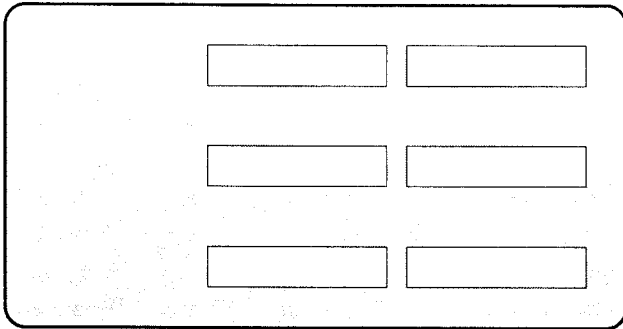


그림 8 - 메시지의 대한 정의

그림 6의 (a)에서 ① Node A에서 B로 전송하기 위해 Advertisement를 보낸다. ② Node B에서 Request를 보내게 되고 ③ Node A에서 Data를 보낸다. ④ Node B가 데이터를 예러없이 수신하게 되면 ack신호를 보내주게 되고 데이터를 완전히 받은 Node B는 Source노드가 되어 Node C로 동일한 순서로 데이터를 보내게 된다. 여기서 그림 6의 (b)와 같이 네트워크 코딩 기법을 적용하게 되면 ① Node A에서 Advertisement를 보내면 Node B에서 수신을 한다. ② Node B는 Node A에게 보낼 Request와 Node C에게 보낼 Advertisement를 XOR을 해서 broadcast를 하게 된다. ③ Node B에서 보낸 메시지가 Node A, Node C에서 받게 되고 Node A에서는 Node B에서 보낸 XOR데이터를 자신이 가지고 있는 Advertisement 메시지를 이용하여 디코딩한다. 그러면 Request 메시지가 남게 되어 B가 자신에게

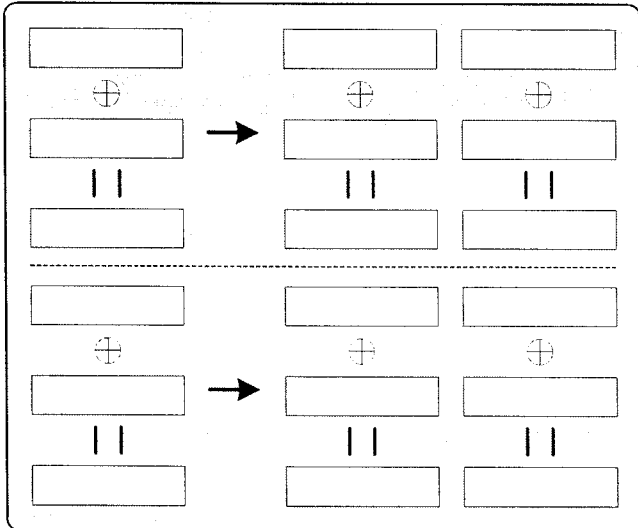


그림 9 - Packet Format의 디코딩

Request 를 보냈다는 것을 알게 되고, Node C는 XOR데이터를 자신이 가지고 있는 미리 정해놓은 Request 메시지를 이용하여 디코딩한다. 따라서 Advertisement 메시지가 남게 되어 B가 자신에게 Advertisement를 보냈다는 것을 알게 된다. ④ Node B에서는 Node A에서 보낸 Data와 Node C에서 보낸 Request 메시지를 받게 된다. ⑤ Node B에서 Node A에게 보낼 ack신호와 Node C에서 보낼 Data를

XOR해서 보내게 된다. ⑥ Node A는 Node B에서 보낸 메시지에서 Data 메시지를 디코딩하여 ack신호를 추출하고, Node C는 Node B에서 보낸 메시지에서 미리 정의된 Ack 메시지를 디코딩하여 Data 메시지를 추출한다. 그림 6의 (a)에서 Node A에서 Node D까지 데이터를 전송하기 위해서는 12번의 데이터 송수신이 필요한 것에 반해 그림 6의 (b)에서는 Node A에서 Node D까지 데이터를 전송하기 위해서는 8번의 데이터 송수신이 필요하다. 지금까지 설명한 바와 같이 네트워크 코딩 기법을 업데이트 코드 전파에 적용하면 노드수가 증가 할수록 데이터 송수신 빈도는 감소하게 된다.

5. 성능평가

본 절에서는 본 논문에서 제안한 방식과 기존의 파이프라인 전송 방식인 Deluge[4]와 데이터 송수신 횟수에 대해 이론적인 성능을 비교 분석 한다. 아래 그림 11은 그림 6에 기반한 데이터 송수신 횟수에 따른 성능 비교결과 그래프이다. 그림 11에 나타나듯이 기존의 방식은 함수의 증가에 따라 데이터 전송횟수가 3.9씩 증가하지만, 본 논문 제안한 방식은 1.9씩 증가한다. 따라서 본 논문에서 제안한 방식은 기존의 방식에 비해 약 2.1배의 성능향상을 보인다.

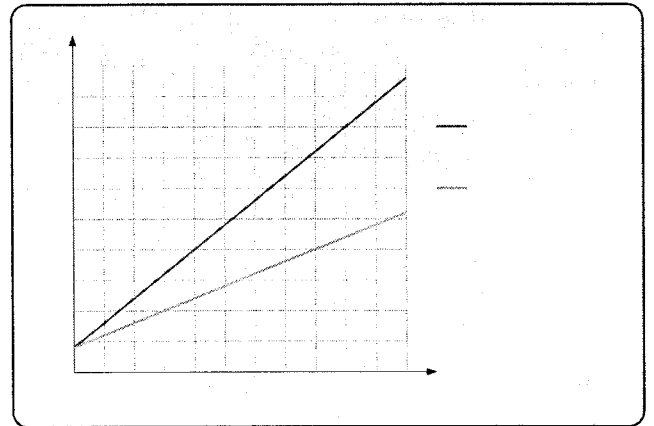


그림 11 - Data 송수신 횟수에 따른 비교

6. 결론

본 논문에서는 네트워크 코딩 기법을 이용한 새로운 업데이트 코드 전파 기법을 제안하였다. 5절에서 설명한 바와 같이 제안된 코드전파 기법은 기존의 파이프라이닝 방식에 비해 데이터 송수신 횟수에 있어 약 49%의 성능 향상을 보였다. 데이터의 송수신 횟수는 데이터 전송 속도, 에너지, 네트워크 혼잡도에 직접적인 영향을 미치는 요소이다. 따라서 본 논문에서 제안한 코드 업데이트 기법을 사용할 경우 속도, 에너지,

네트워크 혼잡도 측면에서 효율적인 소프트웨어 업데이트를 수행할 수 있다. 뿐만 아니라 미리 정의된 메시지를 이용함으로써 네트워크 코딩의 문제점인 overhearding 문제에 따른 원본 데이터의 분실이나 데이터의 미 수신에 따른 디코딩문제를 방지함으로써 무선 네트워크상에서 신뢰성있는 데이터 전송이 가능하다.

감사의 글

본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 결과임

참고문헌

- [1] C.-C. Han, R. Kumar, R. Shea, and M. Srivastava. (2005) "Sensor Network Software Update Management: A Survey," *International Journal on Network Management*, Vol. 15, No. 4, pp. 283-294, July.
- [2] Crossbow Technology Inc (2003). Mote In-Network Programming User Reference Version 20030315, <http://webs.cs.berkeley.edu/tos/tinyos-1.x/doc/Xnp.pdf>
- [3] Thanos Stathopoulos, John Heidemann, and Deborah Estrin. (2003). "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.
- [4] A. Chlipala, J. Hui, and G. Tolle. Deluge: Data dissemination for network reprogramming at scale. ClassProject, <http://www.cs.berkeley.edu/~jwhui/research/deluge/cs262/cs262a-report.pdf>, Fall 2003.
- [5] Sandeep S. Kulkarni. (2005). "MNP: Multihop Network Reprogramming Service for Sensor Networks" pages 7-16, Washington, DC, USA, June, IEEE Computer Society
- [6] Thanos Stathopoulos. (2003). "A Remote Code Update Mechanism for Wireless Sensor Networks", CENS Technical Report # 30.
- [7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard and J. Crowcroft. (2006). "XORs in the air: practical wireless network coding", Proc. *ACM SIGCOMM*.
- [8] Patrick E. Lanigan. (2005). "Disseminating Code Updates in Sensor Networks". October 2005 *CMU-ISRI-05-122*
- [9] Jonathan W. Hui and David Culler. (2004). "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale", *ACM Press*, pages 81-4.
- [10] Vinayak Naik. (2007). "Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Extreme Scale Wireless Networks of Embedded Devices", *IEEE Transactions on Mobile Computing*, VOL. 6, NO. 7.