

논문 2014-09-30

클러스터링을 이용한 SW 업데이트 방법

(A Software Update Method Using Clustering WSNs)

정혜영, 안병철*

(Hyeyeong Jeong, Byoungchul Ahn)

Abstract : Wireless Sensor Networks(WSNs) are applied to many monitoring applications. Present sensor nodes can perform many functions at the same time and contain complex software. During the lifetime of sensor nodes, they are required to reprogram their software because of their new functions, software, software bug fixes. The nodes are inaccessible physically or it is very difficult to upgrade their software by one by one. To upgrade the software of sensor nodes in WSNs remotely, this paper presents an energy efficient method by selecting an optimal relay node. The CHR(Cluster Head Relay) method is compared with SPIN and RANDOM method. Three methods are simulated in NS-2 with the same environmental parameters. Simulation results show that CHR shows faster update time and less power consumption compared with other two methods.

Keywords : Firmware upgrade, Clustering, SPIN, Relay node, WSN

I. 서론

WSN(Wireless Sensor Networks) 기술은 최근 유비쿼터스 컴퓨팅을 실현하기 위해 가장 중요한 장치로 자리 잡아가고 있다. 무선센서 네트워크 분야의 응용이 빠른 속도로 보급되고 반도체 기술의 발전으로 무선센서 네트워크에 사용되는 노드에 많은 기능들이 하나의 칩으로 구현되고 있다. 이러한 센서 노드들은 사람이 직접 감시하기 힘든 곳에 설치하여 환경감시, 이동 탐지, 재난 관리 등을 위해 사용되고 있다.

센서노드는 사람이 접근하기 힘든 지역에 설치되어 운영됨에 따라 한번 설치된 센서노드들은 그 배터리수명이 다하여도 회수되거나 교체가 이루어지지 않는다. 따라서 각 센서 네트워크는 에너지 소모 효율을 매우 중요시 한다. 일반적인 센서노드들의 에너지 소모율이 가장 높은 부분은 무선전송 부분이다. 무선 전송 부분을 짧게 함으로써 에너지 소모를 줄일 수 있다[1, 2]. 무선 전송을 줄이기 위해서는 최적의 릴레이 노드를 선택해야 한다[3]. 따라

*Corresponding Author(b.ahn@yu.ac.kr)

Received: 2 Jan. 2014, Revised: 11 Feb. 2014,

Accepted: 26 Feb. 2014.

H. Jeong, B. Ahn: Yeungnam University

© IEMEK J. Embed. Sys. Appl. 2014 Aug.: 9(4) 245-251

ISSN : 1975-5066

http://dx.doi.org/10.14372/IEMEK.2014.9.4.245

서 본 논문에서는 원격 WSNs에서 센서노드의 소프트웨어를 업데이트하기 위한 최적의 릴레이 노드를 선택하여 에너지 효율을 높이는 방법을 제공하는 모델과 방법을 제시하고자 한다. 본 논문은 모두 6개의 장으로 구성되며, 2장에서는 펌웨어 업데이트 관련 연구에 관련하여 기술한다. 3장에서는 본 연구의 대상인 무선 센서 네트워크의 기본 구조를 제시한다. 4장에서는 재전송노드의 선택 방법과 CHR 알고리즘, 5장에서는 4장에 제시된 알고리즘을 기반으로 한 모의실험 결과를 제시한다. 6장에서 결론을 기술한다.

II. 펌웨어 업데이트 관련 연구들

Infuse는 TDMA(Time division multiple access)방식의 프로토콜로 암시적 응답(implicit acknowledgement)과 배압(back pressure) 메커니즘을 사용함으로써 전송의 신뢰성을 제공하는 Go-Back-N의 선택적 재전송 방식을 사용한다[4]. 에너지 절약을 위해 TDMA방식의 청취(listening)방법을 사용하고 있다. 그 특징은 전임센서노드가 그 다음 센서노드로 연속적으로 데이터를 전달한다.

MOAP(Multihop Over-the-Air Programming)는 CSMA/CA(Carrier Sense Multiple Access /

Collision Avoidance)방식의 프로토콜로 리플 전달 프로토콜을 사용하여 멀티 홉으로 업데이트 코드를 전송하고, 슬라이딩 윈도우 방식을 기반으로 손실된 패킷의 재전송 방법을 사용한다[5]. 네트워크 부하 및 중복된 데이터 전송을 줄일 수 있는 장점이 있는 반면 업데이트와 관련되지 않은 모든 노드들은 불필요한 에너지 소비를 발생시킨다.

MNP(Multihop Network Reprogramming)는 CSMA/CA프로토콜 방식으로 멀티 홉을 지원하고 데이터는 고정 크기 페이지 단위의 공간 다중화(spatial multiplexing)를 이용하여 전송한다[6]. 3단계의 핸드 셰이킹(handshaking) 프로토콜과 전송자를 선택할 수 있는 알고리즘을 사용한다. 코드 전송의 불필요한 부분을 제거할 수 있으며 손실된 코드를 복구하기 위해 비트 벡터 테이블을 사용하는 특징을 가지고 있다.

Deluge는 CSMA/CA 프로토콜 방식으로 파이프라이닝 방식과 3단계 핸드 셰이킹(handshaking) 프로토콜을 사용하는 방식이다[7]. 이전 버전으로부터 효율적인 증가, 업데이트 활성화의 장점을 볼 수 있으며 페이지 단위 데이터 관리가 그 특징으로 볼 수 있다.

PSFQ (Pump Slowly, Fetch Quickly) 메커니즘은 hop-by-hop 방식의 전송 프로토콜이다[8]. PSFQ는 느린 전송과 빠른 복구의 두 단계 특징이 있다. PSFQ는 데이터 배포단계에서 중복을 억제하기 위해 데이터 캐시를 사용 MOPA와 같이 NACK 신호를 사용하여 손실 패킷을 요구하는 방식이다. 모든 패킷이 도착할 때까지 받은 데이터를 보관하므로 복잡성 측면에서 상당한 오버헤드(overhead)를 유발 할 수 있다.

SRM(Scalable Reliable Multicast protocol)은 멀티캐스팅의 최소 필요조건만 부과하는 방식이다[9]. SRM방식은 그룹 구성원에 대한 모든 데이터 전송은 보장되지만 명령은 그렇지 않다. SRM역시 NACK기반이며 그것은 제어 패킷(control packet) 또는 수신요구 차단(repair request implosion) 무효화를 방지하기 위해 댐핑 방법(damping mechanisms)을 사용한다. 그러나 최악의 경우 복구 요청 시 그룹에 있는 모든 노드들을 검색해야 할 필요가 있다.

III. 제안 모델의 기본 구조

본 논문에서 소프트웨어 업데이트 모델을 위해

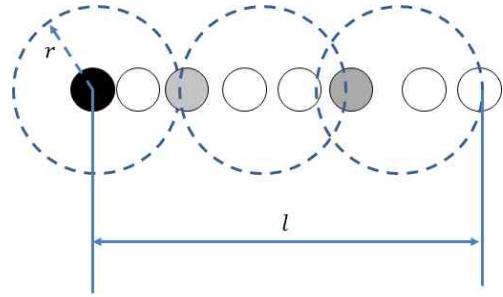


그림 1. 1차원 노드 배치 및 데이터 전달 방법
Fig. 1 One dimensional node allocation and data relay method

아래와 같이 몇 가지 가정을 한다.

- ① 무선 센서 네트워크는 CSMA/CA 프로토콜을 기반으로 하는 매쉬 구조의 ad-hoc 네트워크를 사용한다.
- ② 전체 네트워크에서 베이스 노드는 하나만 존재한다.
- ③ 네트워크상의 모든 노드들은 같은 펌웨어를 사용한다.
- ④ 각 노드의 위치는 고정한다.

1. 펌웨어 소비 에너지

본 절에서는 소프트웨어 업데이트를 위한 에너지 소모 모델에 대해 기술한다. 소프트웨어 데이터 전송 노드가 소모하는 에너지는 소프트웨어 전송을 위해 소모한 에너지와 제어용 데이터를 송수신한 에너지, 그리고 전송 오류로 인한 데이터 재전송을 위해 사용한 에너지의 합으로 나타낼 수 있다.

그림 1은 1열로 배치되었을 경우 데이터 전송을 나타내고 있다. 그림 1에서 점선안의 “ r ”은 노드의 무선 영역의 반경을 나타내며 “ l ”은 베이스 노드에서 마지막 센서 노드까지의 거리를 나타낸다. 검은 색 노드는 베이스 노드, 회색 노드는 소프트웨어를 전송하는 릴레이 노드이며, 흰색 노드는 단지 데이터를 수신 하는 노드이다. 그림 1에서 베이스 노드에서 마지막 노드까지 소프트웨어를 전달하기 위한 최소 릴레이 노드의 수(N)은 식(1)과 같다.

$$N = \left\lceil \frac{l}{r} \right\rceil \quad (1)$$

r = 무선 반경

l = 첫 번째 노드에서 마지막 노드까지의 거리

본 논문에서는 노드들은 2차원적으로 배치되며 노드들의 위치는 고정되었고, 노드간의 거리는 같다고 가정한다. 하나의 송신노드로부터 수신하는 평균 노드의 수(N_a)는 식(2)와 같다.

$$N_a = \left[\frac{\pi \cdot r^2}{A} \cdot N_t \right] \quad (2)$$

r = 노드의 무선 반경
 A = 센서 필드의 면적
 N_t = 전체 노드의 수

소프트웨어 송수신을 위해 소모한 전력은 전송 노드가 소모한 전력(E_s)과 수신 노드가 소모한 전력(E_r)으로 표현 된다. 소프트웨어 데이터 수신 노드의 에너지 소모를 식(3)과 같이 나타낼 수 있고, 송신 노드의 에너지 소모를 식(4)와 같이 나타낼 수 있다.

$$E_s = (D_f \cdot e_s + D_{cr} \cdot e_r) \quad (3)$$

$$E_r = (D_f \cdot e_r + D_{cr} \cdot e_s) \quad (4)$$

D_f = 펌웨어 데이터를 위한 패킷 수
 e_s = 단위 데이터 당 전송 에너지
 D_{cr} = 오류복구를 위한 제어 데이터양
 e_r = 수신 시 사용되는 에너지

릴레이노드 선택에 따라 에너지와 소프트 업데이트 시간에는 많은 차이를 볼 수 있으므로 적당한 릴레이노드를 선택하는 것이 중요하다. 식(5)는 센서 필드에 있는 전체 노드들을 업데이트하기 위해 소모되는 에너지(E)를 나타내고 있다.

$$E = \left(E_r \cdot \left(\frac{\pi \cdot r^2}{A} \cdot N_t \right) + E_s \right) \cdot N \quad (5)$$

r = 노드의 무선 반경
 A = 센서 필드의 면적
 N_t = 전체 노드의 수
 N = 데이터 재전송 노드의 수

2. 펌웨어 업데이트 시간

각 노드의 펌웨어를 업데이트 하는 시간은 데이터 전송 시간과 오류 복구 시간 그리고 노드를 재프로그램 하는 시간의 합으로 표시된다. 데이터 전송을 위해 필요한 시간(T_{data})는 다음 식(6)으로 나타낸다.

$$T_{data} = \left(\frac{P_d + P_h}{bit_rate} + P_0 \right) \cdot C_f \quad (6)$$

P_d = 데이터 패킷의 크기
 P_h = 패킷 헤더 크기
 P_0 = 무선 채널 접근 시간
 C_f = 소프트 업그레이드를 위한 패킷 수

전체적인 오류 복구 시간(T_{err})은 식(7)과 같다.

$$T_{err} = \left(\frac{P_d + P_h}{bit_rate} + P_0 \right) \cdot C_f \cdot P_{err} + \left(\frac{P_c + P_h}{bit_rate} + P_0 \right) \cdot C_f \cdot P_{err} \quad (7)$$

P_c = 데이터 패킷크기

단일 홉에서 노드를 업데이트 하는 시간(T_{step})은 식(8)과 같다.

$$T_{step} = T_{data} + T_{err} + T_{up} \quad (8)$$

T_{up} = 데이터 수신 후 노드 업데이트 시간

센서 필드의 모든 노드를 업데이트하기 위한 전체 시간(T)은 각 단계의 업데이트 시간에 전체 재전송 단계의 배수로 나타낼 수 있다. 식(9)는 전체 노드를 업데이트하기 위한 시간을 나타낸 것이다.

$$T = T_{step} \cdot \frac{l}{r} \quad (9)$$

l = 베이스 노드에서 가장 먼 노드까지의 거리

IV. 재전송노드의 선택방법과 CHR 알고리즘

1. 재전송 노드의 선택방법

본 논문에서는 펌웨어 재전송 노드를 선택하기 위해 아래와 같이 3가지 방법을 사용하여 비교하였다.

- SPIN(Sensor protocols for information via negotiation) : 메타데이터를 이용하여 에너지 손실과 데이터 중복을 반으로 줄이는 라우팅 방식이다[10]. 그림 2는 SPIN 프로토콜에서 SPIN의 동작 단계를 볼 수 있다.

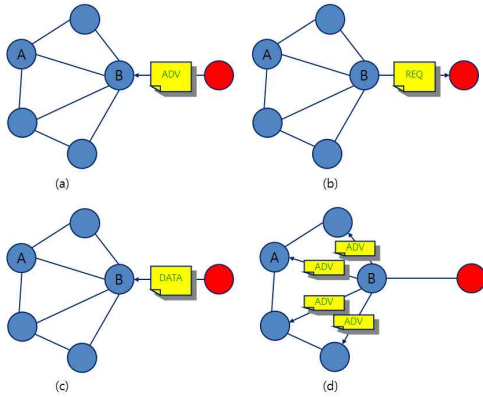


그림 2. SPIN 프로토콜
Fig. 2 SPIN protocol

그림 2의 (a)는 소스 노드가 전송해야 할 데이터가 있으면 ADV신호를 방송하여 전송 데이터가 있음을 알린다. (b)는 데이터를 원하는 노드는 소스 노드에게 REQ신호로 응답한다. (c)에서 소스 노드는 데이터를 원하는 노드에 데이터를 전송한다.

- RANDOM : 데이터를 수신한 노드 중 가장 먼저 재 프로그램을 완료한 노드가 다음 릴레이 노드를 임의로 선택하여 데이터를 전송한다.
- CHR(Cluster Head Relay) : 소프트웨어 업데이트 시작 전, 노드들은 수집된 데이터를 주기적으로 전송하는 전송 모드로 동작한다. 이때 센서네트워크는 클러스터로 구성되며, 클러스터 구성 노드들은 클러스터 헤드에게 데이터를 전송 하고 클러스터 헤드는 이러한 데이터를 수집하여 베이스 스테이션으로 수집한 데이터를 전송한다. 이때 구성되는 클러스터 헤드 방식은 LEACH방식을 따른다[11]. 베이스 스테이션이 수신한 데이터 자료 중 클러스터 헤드 노드 정보를 검색하여 클러스터 헤드 노드를 재전송 노드로 선택한다. 만약 클러스터 헤드노드와 통신이 되지 않으면 통신이 되지 않은 클러스터에 속하는 일반 노드중 하나를 재전송 노드로 선택하여 데이터를 전송한다. 그러나 클러스터 헤드노드는 에너지와 신호가 가장 강한 노드이므로 클러스터헤드와의 연결이 끈 어지는 경우가 많지 않다.

2. 클러스터 헤드 릴레이 노드(CHR) 알고리즘

LEACH방식을 따라 구성된 클러스터 헤드 노드는 수집한 정보를 베이스 스테이션으로 전송한다.

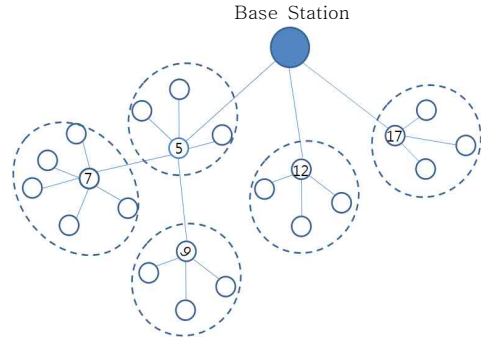


그림 3. 클러스터 헤드 릴레이 노드
Fig. 3 Relay nodes of cluster head

```

Data send node
Broadcast status information periodically
If (receive "relay-start" && exist proper neighbor node) {
    Send "data-start" to neighbor nodes.
    Send data
}
While(No. data receive nodes > 0) {
    If(receive "reprogram-done") {
        Select one node.
        Send "relay-start" to selected node.
        Decrease No. of data receive node.
    }
}
Go to 1.1

Data receive node
Broadcast status information periodically
If (receive "data-start") {
    If (receive firmware ver. > stored firmware ver.) {
        Receive data and store it to memory
        Request missing or lost packet
        Reprogramming it-self.
        Send "reprogram-don" to data send node.
    }
}
If (receive "relay-start")
    Go to 1.1
Else
    Go to 2.1
    
```

그림 4. CHR 알고리즘
Fig. 4 CHR algorithm

베이스 스테이션으로 데이터가 전송 될 때 클러스터 헤드 노드정보도 같이 전송되게 된다. 그림 3의 예에서 베이스 스테이션까지 수집된 데이터가

표 1. 모의실험 파라미터
Table 1. Simulation parameters

Parameters	Value
MAC 프로토콜	CSMA/CA Back-off Window 2~26 Slot Time=0.384ms IFS=1.664ms
RF 전송 반경	60m
무선 대역폭	250Kbps
노드의 수	100
송수신 소모 전력	75.9mW(TX)/62.7mW(RX)
노드간 거리	20m~50m
데이터 패킷의 크기	64바이트 ~ 256바이트
데이터 헤더 크기	8바이트
프로토콜 오버헤드 (IP+ MAC)	16byte(8바이트+ 8바이트)
펌웨어 데이터 크기	128K바이트
데이터 전송율	무선 대역폭 70%

수신받기 위해 지나온 클러스터 헤드 노드의 번호는 5, 7, 9, 12, 17번 노드에 해당된다.

베이스 스테이션 노드에서 수신한 클러스터 헤드 노드를 재전송 노드로 선택하여 전송하는 방법이다. 일반 노드들은 펌웨어 업데이트 전에는 일반적인 데이터 수집 및 전송 모드로 동작한다. 그림 4와 같이 주기적으로 자신의 상태를 이웃 노드에 전송한다. 데이터 전송은 "Relay-Start" 메시지를 받음과 함께 펌웨어 데이터 전송 프로시저를 시작한다.

데이터 전송 노드가 "Data-Start"메시지와 새로운 펌웨어 버전 정보를 데이터 수신 노드에게 전송하면 펌웨어 데이터 전송이 시작됨을 알린다. 데이터 수신 노드가 소프트웨어 업데이트가 끝나면 "Reprogram-Done" 메시지를 데이터 전송 노드로 전송하고 펌웨어 업그레이드가 종료한다.

V. 모의실험 및 결과 분석

1. 모의실험 환경

모의실험은 "NS-2" 네트워크 시뮬레이터를 사용하였다. 모의실험은 1개의 베이스 노드와 99개의 일반 노드로 구성되며, 각 노드들은 일정한 간격으로 배치된다. 무선 프로토콜은 802.11과 802.15.4를 기반으로 한 CSMA/CA를 사용한다. 표 1은 모

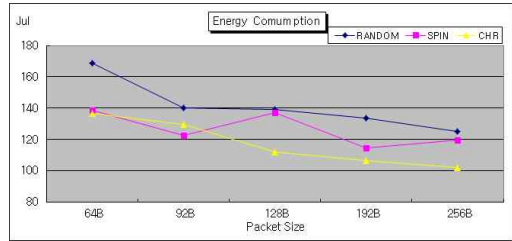


그림 5. 에너지 소모
Fig. 5 Energy consumption

의실험 파라미터를 요약한 것이다.

2. 성능 평가 항목

모의실험에서 각 파라미터를 변경하며 모의실험을 진행 하고, 최적의 값을 구하기 위해 다음의 성능 평가 항목을 구성하였다.

- ① 에너지 소모
- ② 전체 노드의 업데이트 시간
- ③ 재전송 노드의 수
- ④ 손실 및 오류 데이터

3. 모의실험 결과 분석

실험은 노드간 거리를 20m, 40m, 50m로 배치 밀도를 변경하여 각각 데이터 패킷의 크기에 따른 전체 데이터 재전송 노드의 수, 전체 업데이트 시간과 에너지 소모 그리고 오류 데이터 량을 측정하는 실험을 하였다. 실험결과 노드간 거리가 40m인 경우, 이론적인 예상 값과 거의 같은 결과를 보인다. 따라서 본 논문에서는 노드간 거리가 40m인 경우로 하였을 때의 실험 결과만 표기한다.

그림 5는 모든 노드를 업데이트하는 총 에너지 사용량을 보여준다. 무선 센서 네트워크의 총 에너지 소비는 식(5)를 사용하여 계산하였다. 본 실험결과 3가지 방법 모두 패킷 크기가 64바이트에서 에너지 소비가 가장 높음을 알 수 있다. 그 이유는 패킷크기가 너무 작으면 송·수신 시 데이터들의 충돌이 더 많이 발생하고 충돌이 발생한 데이터를 다시 재전송해야 하는 이유로 에너지 소비와 소프트웨어 업데이트 시간이 길어진다. 본 논문에서 제안한 CHR 방법에서 패킷 크기가 256바이트인 경우 101J로 나머지 두 방법에 비해 에너지 소비가 가장 낮음을 볼 수 있고,

그림 6은 모든 노드에 소프트웨어 업데이트 시간을 보여주고 있다. 소프트웨어 업데이트 총 시간은 식(9)을 적용한다. 실험결과 그림 5와 유사한 패

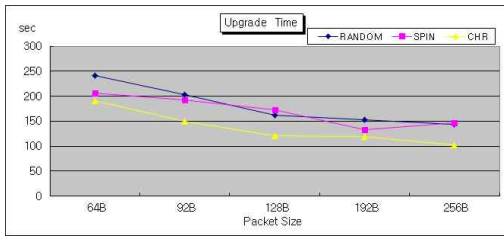


그림 6. 업데이트 시간
Fig. 6 Update time

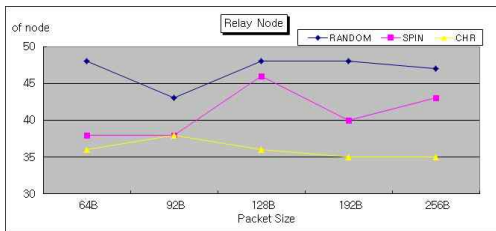


그림 7. 재전송 노드의 수
Fig. 7 Number of re-transmission nodes

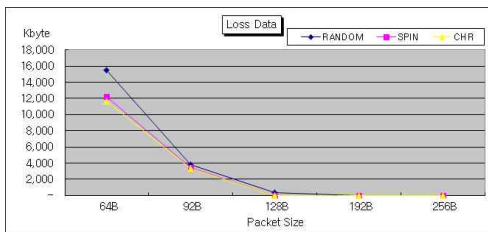


그림 8. 손실 및 오류 데이터
Fig. 8 Number of re-transmission nodes

턴을 나타내고 있다. 본 논문에서 제안한 CHR 방법에서 패킷 크기가 256바이트인 경우, 업데이트 시간은 102sec이다. 역시 업데이트 시간에서도 패킷 크기가 64바이트 일 때 업데이트 시간이 가장 긴 것을 알 수 있다.

그림 7은 모든 노드에 재전송 릴레이 노드 수를 보여주고 있다. 결과적으로 패킷 크기가 256바이트인 경우 재전송 릴레이 노드의 수가 35개로 가장 적음을 알 수 있다.

그림 8은 손실 데이터를 보여주고 있다. 결과적으로 패킷 크기가 256바이트인 경우 손실 데이터가 1.05Kbyte이다. 패킷 크기가 64바이트인 경우 송수신 사이의 시간적 간격이 적으므로 데이터의 많은 충돌이 발생하고, 손실 데이터 또한 높음을 알 수 있다. 결과적으로 패킷크기가 128바이트에서 256바

이트 사이가 안정적임을 알 수 있다.

위의 시뮬레이션 결과 CHR, SPIN, RANDOM 방법에서 성능 차이를 볼 수 있다. 이러한 성능차이의 가장 큰 이유는 CHR방법의 경우 중간노드를 선택하는 방법에 있어 클러스터 헤더를 선택하므로 중간노드 선택을 위한 에너지와 시간의 불필요한 낭비를 줄일 수 있다. SPIN은 정확한 데이터 전송을 보장할 수 없다. 예를 들어 데이터에 관심이 있는 목적노드가 소스 노드보다 멀리 떨어져 있다고 가정하자, 소스 노드와 목적 노드 사이에 전달 노드가 그 데이터에 관심이 없는 경우 전달 노드는 소스 노드에 “REQ” 신호로 응답하지 않으며, 소스 노드는 “REQ” 신호를 수신하지 않아 데이터를 보내지 않으므로 먼 거리에 있는 목적 노드에 데이터 전달이 이루어지지 않는다. 따라서 목적 노드는 다른 전달 노드를 통해 데이터를 받아야 함으로 많은 에너지 소모를 볼 수 있다. RANDOM 방식의 경우 많은 노드들이 릴레이 노드가 되며 결과적으로 많은 데이터 충돌이 발생하여 에너지 소모와 업데이트 시간이 길어진다.

VI. 결론

본 논문은 세 가지 소프트웨어 업데이트 방법 결과 비교를 위해 “NS-2” 네트워크 시뮬레이션을 사용하였다. 노드간의 거리를 40m인 경우 패킷 사이즈가 128 바이트 ~ 256 바이트사이에서 좋은 성능을 보여준다. 이 경우 CHR방식이 다른 방식 보다 릴레이 노드 수가 12% ~ 15% 감소하며 손실 데이터는 0.5% ~ 1.5% 감소한다. 에너지 효율은 0.07 ~ 19%이 증가하며, 업데이트 시간이 0.22 ~ 0.25% 단축됨을 확인하였다.

그리고 시뮬레이션 결과로 패킷의 크기가 160byte를 넘을 경우 에너지 소모, 업데이트 시간 등의 성능 면에서 큰 차이가 없음을 볼 수 있다.

References

[1] W. Dong, Y. Liu, C. Wang, X. Liu, C. Chen, J. Bu, “Link Quality Aware Code Dissemination in Wireless Sensor Networks,” Proceedings of IEEE International Conference on Network Protocol, pp. 89-93, 2011.
[2] H.S. Noghabi, A.G. Askar, A. Boustani, A. Moghani, M.B. Zanjan, “Implementing a

Greedy Chain Routing Technique with Spread Spectrum on Gridbased WSNS,” International Journal of Wireless & Mobile Networks, Vol. 4, No. 4, pp. 195-202, 2012.

[3] Y. Kwon, B. Ahn, “A Firmware Upgrade Model for Wireless Sensor Networks,” KISSE, Vol. 39, No. 4, pp. 293-301, 2011 (in Korean).

[4] S. Kulkarnia, M. Arumugama, “Infuse: A TDMA Based Data Dissemination Protocol for Sensor Networks,” Technical Report MSU-CSE-04-46., Dept. of Computer Science and Engineering, Michigan State University, 2004.

[5] T. Stathopoulos, J. Heidemann, D. Estrin, “A Remote Code Update Mechanism for Wireless Sensor Networks,” Centre for Embedded Networked Sensing, UCLA, CENS Tech. Report #30, 2003.

[6] S.S. Kulkarni, “MNP: Multihop Network Reprogramming Service for Sensor Networks,” Proceedings of IEEE International Conference on Distributed Computing Systems, pp. 7-16, 2005.

[7] A. Hagedorn, D. Starobinski, A. Trachtenberg, “Rateless Deluge: Over-the-Air Programming of Wireless Sensor Networks Using Random Linear Codes,” Proceedings of IEEE International Conference on Information Processing in Sensor Networks, pp. 457-456, 2008.

[8] C.Y. Wan, A.T. Campbell. “PSFQ: A Reliable Transport Protocol For Wireless Sensor Networks,” Proceedings of ACM International Work-shop on Wireless Sensor Networks and Applications, 2002.

[9] S.K. Kasera, J. Kurose, D. Towsley, “Sneha K. Kasera, Jim Kurose and Don Towsley,” Proceedings of IEEE INFOCOM, 1998.

[10] K. Akkaya, M. Younis, “A survey on routing protocols for wireless sensor networks,” Ad Hoc Networks, Vol. 3, No. 3, pp. 325 - 349, 2005.

[11] W.R. Heinzelman, A. Chandrakasan, H. Bala-krishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” Proceedings of IEEE Annual Hawaii

International Conference on system Science, 2000.

저 자 소 개

정혜영



2013년, 영남대학교 컴퓨터 공학과 석사.

2013년, 영남대학교 컴퓨터 공학과 박사과정.

관심분야 : 무선센서네트 워크

Email: dasom_01@yu.ac.kr

안병철



1976년 영남대학교 전기 공학과 학사.

1986년 오레곤주립대 전기 및 컴퓨터공학 석사.

1989년 오레곤주립대 전기 및 컴퓨터공학 박사.

1976년~1984년 국방과학 연구소연구원.

1989년~1992년 삼성전자 수석연구원.

1992년~현재 영남대학교 교수.

관심분야: 임베디드시스템, 실시간처리, 멀티 미디어처리, 센서네트워크

Email: b.ahn@yu.ac.kr