

모바일 PULSE : 모바일 싱크 노드를 가진 센서 네트워크에서의 경로 복구 시간과 전력 소모량을 고려한 라우팅 프로토콜

준회원 이 치 영*, 정회원 이 신 형*, 종신회원 유 혁*

Mobile PULSE : A Routing Protocol Considering the Power and the Route Recovery Time in Sensor Networks with A Mobile Sink Node

Chi-Young Lee* *Associate Member*, Shin-Hyoung Lee* *Regular Member*,
Chuck Yoo* *Lifelong Member*

요 약

PULSE 프로토콜은 센서 노드의 슬립을 통해 소모 전력량을 크게 줄인다. 그러나 이 프로토콜은 싱크 노드의 이동을 고려하지 않는 문제가 있다. 모바일 싱크 노드를 가진 센서 네트워크를 위한 라우팅 프로토콜은 빠른 경로 복구와 전력 소모 최소화를 모두 달성해야 한다. 본 논문은 두 목적을 달성하기 위해 모바일 PULSE 프로토콜을 제안한다. 또한, 실험을 통해 제안한 프로토콜의 경로 복구 능력과 전력 소모량을 확인하였다. 이 프로토콜은 기존의 PULSE 프로토콜에 비해 약 40% 정도 경로 복구 시간을 단축하였고, 드롭되는 패킷의 양도 약 1/2 수준으로 줄였다. 전력 소모량 면에서는 최대 0.8%의 증가만이 나타남으로써, 기존의 PULSE 프로토콜의 전력 효율적인 특징을 그대로 유지함을 보였다. 본 논문은 모바일 PULSE 프로토콜의 경로 복구 능력과 전력 소모량을 확인함으로써 모바일 센서 네트워크에 적합한 프로토콜로서의 가능성을 보여준다.

Key Words : Mobile Sink, Routing Protocol, Sensor Networks, Power Saving, Route Recovery

ABSTRACT

The PULSE protocol can greatly reduce power consumption using a node's sleep state. But this protocol does not consider movement of a sink node in a sensor network. In the mobile sensor network, a routing protocol must recover path error by movement of a sink node as quickly as possible. Therefore we have to achieve fast path recovery and power saving to support movement of a sink node in a sensor network. This paper proposes the Mobile PULSE protocol which is a improved routing protocol for a mobile sink node. And we evaluate Mobile PULSE and show that the Mobile PULSE reduces the recovery time about 40% compared with original PULSE protocol. Mobile PULSE increases energy consumption than PULSE as a maximum of 0.8%, which means Mobile PULSE is similar to PULSE in energy consumption. This paper shows mobile PULSE's capability in the mobile sensor network through evaluation of path recovery time and power consumption.

※이 연구에 참여한 연구자는 '2단계 BK21사업'의 지원을 받아 연구를 수행함.

* 고려대학교 컴퓨터학과 운영체제연구실 ({cylee, shlee, hxy}@os.korea.ac.kr)

논문번호 : KICS2008-11-500, 접수일자 : 2008년 11월 11일, 최종논문접수일자 : 2009년 2월 10일

I. 서 론

센서 네트워크를 구성하는 각 센서 노드는 일반 네트워크에서의 노드들과 달리 매우 제한된 전력을 갖는다. 그러므로 PULSE^[1], SMAC^[2], STEM^[2], PAMAS^[3] 등과 같은 전력 소모를 최소화하는 프로토콜에 대한 많은 연구가 있어 왔다^[4]. 특히 PULSE 프로토콜은 일부 노드를 절전 모드로 전환함으로써 전력 소모를 크게 줄일 수 있다. 그러나 이 프로토콜은 모든 노드가 고정된 위치에 있고, 노드의 전력 부족으로 인한 토폴로지 변화만을 고려한다. 따라서 노드가 이동하는 네트워크에는 적합하지 않다. 센서 네트워크에서의 노드 이동은 다양한 서비스의 제공으로 이어진다. 다음은 모바일 싱크 노드의 이용 예이다.

- 1) 병원에서 센서 네트워크와 PDA를 이용하여 언제 어디서나 환자의 상태를 체크할 수 있다.
- 2) 빌딩 내에서 화재가 발생했을 때 건물 관리자의 위치와 관계없이 모바일 장비를 통해 이를 빠른 시간 내에 인식할 수 있다.
- 3) 전쟁터에서 부대는 항상 전선을 따라 이동하는 중에 적의 위치 등의 정보를 파악할 수 있다.

위와 같은 상황들은 PDA와 같은 모바일 장비를 이용하여 이동 중에 센싱 데이터를 수집함으로써 가능하다. 싱크 노드의 이동은 라우팅 경로의 오류를 발생시키고 이로 인해 데이터 손실이 나타난다. 그러므로 이런 손실을 최소화하기 위해 경로 복구 시간을 줄이는 방안이 요구된다. 또한, 센서 노드는 한번 설치되면 전력 공급이 어렵기 때문에 각 센서 노드의 전력 소모를 최소화해야 한다. 그러므로 싱크 노드의 이동을 고려한 센서 네트워크는 경로 복구 시간 단축과 전력 소모량 최소화의 두 목적을 모두 달성할 수 있어야 한다. 이를 위해 전력 소모를 크게 줄인 기존의 PULSE 프로토콜을 이용한다. PULSE 프로토콜은 무선 멀티-홉인프라스트럭처 액세스 네트워크(Wireless Multi-hop Infrastructure Access Network)을 위한 라우팅 프로토콜^[5]과 무선 애드-혹 네트워크(Wireless Ad-hoc Network)을 위한 라우팅 프로토콜^[6]로 제안되었고, 이후 센서 네트워크를 위한 라우팅 프로토콜로 수정된 버전이 제안되었다^[1]. 본 논문에서는 순서대로 인프라 PULSE(infra PULSE), 애드-혹 PULSE(ad-hoc PULSE), 센서 PULSE(sensor PULSE)로 명명한다.

모든 PULSE 프로토콜은 타이밍 정보를 이용해 특정 시간이 되면 모든 노드가 정해진 행동을 함으로써, 경로를 생성하고 유지한다. 이 타이밍 정보에

의해 펄스 패킷(pulse packet)이 주기적으로 전송된다. 이 펄스 패킷은 새로운 경로를 생성하는데 사용된다. 그러므로 펄스 패킷의 전송 주기를 변화시켜 경로 생성 주기를 조절할 수 있다. 펄스 패킷의 전송 주기가 길어지면 전송 횟수가 줄어들기 때문에 펄스 패킷의 브로드캐스트로 인한 전력 소모는 줄일 수 있다. 하지만 펄스 패킷 전송 횟수 감소로 경로의 생성 횟수가 적어지므로 경로 복구에 많은 시간이 걸린다. 즉, PULSE 프로토콜에서는 전력 소모량 최소화와 경로 생성 주기 사이에는 트레이드-오프(Trade-off)관계가 성립한다. 그러므로 본 논문에서는 싱크 노드의 이동을 고려하여 기존 센서 PULSE를 개선한다. 이를 통해 경로 복구 시간 단축과 전력 소모량 최소화라는 두 목표를 모두 달성하도록 한다.

이 논문의 구성은 다음과 같다. II장에서는 기존 센서 PULSE 프로토콜에서의 소모 전력 감소 방법을 설명하고, III장에서는 싱크 노드의 이동을 고려하여 PULSE 프로토콜을 개선한 모바일 PULSE (Mobile PULSE) 프로토콜을 제안한다. IV장에서는 NS2를 이용한 실험을 통해 모바일 PULSE 프로토콜이 기존 PULSE와 비교했을 때 전력 감소와 경로 복구 시간 단축을 모두 달성했음을 보인다. V장에서는 센서 네트워크에서 노드의 이동을 고려한 연구들을 소개한다. 마지막으로 VI장에서는 결론을 기술하며 본 논문을 마무리한다.

II. 센서 네트워크를 위한 PULSE 프로토콜 : 전력 소모량의 감소

센서 PULSE 프로토콜^[1]은 전력 효율적인 라우팅 프로토콜로, 라우팅 경로에 포함되지 않는 노드를 슬립(sleep) 상태로 전환함으로써 전력 소모를 줄인다. 센서 PULSE 프로토콜은 고정된 펄스 간격(Pulse Interval)으로 펄스 패킷을 전송한다. 이 펄스 패킷은 싱크 노드에서 생성되고 네트워크 전체로 브로드캐스트된다. 이 주기적인 펄스 패킷의 전송은 라우팅 정보의 갱신과 전체 네트워크의 시간 동기화를 제공한다. 네트워크 내의 각 노드는 수신한 펄스 패킷을 기억함으로써 싱크 노드를 향한 경로를 생성한다. 중복된 펄스 패킷이 수신되었을 경우, 가장 작은 홉 수(hop count)를 가진 펄스 패킷을 기억하여 최선의 경로를 선택하게 된다. 만약 펄스 패킷을 수신한 노드가 전송하고자 하는 패킷이 있다면, 수신한 펄스 패킷에 대한 응답으로 싱크 노드를 향해 예약 패킷(reservation packet)을 전송한다. 예약 패킷은 자신을

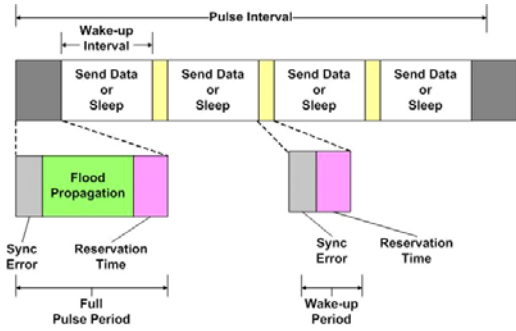


그림 1. 센서 PULSE 프로토콜의 타이밍^[1]
Fig. 1. The Timing of the sensor PULSE protocol^[1]

생성한 노드의 주소와 자신이 거쳐 간 노드의 주소를 포함한다. 이 정보를 이용해 싱크 노드와 소스(source) 노드 사이의 역 경로(reverse route)가 생성한다. 이것은 AODV^[7]의 RREP 메시지를 통한 응답 메커니즘과 유사하다^{[1][5][6]}. 예약 패킷의 전송은 매 펄스 패킷의 수신에 있을 때마다 이루어져야 하고, 더 이상 보낼 데이터가 없다면 펄스 패킷에 대해 응답하지 않는다.

센서 PULSE 프로토콜은 그림 1과 같은 고정된 시간 동기화를 이용한다. 네트워크 내의 모든 노드는 펄스 주기(Pulse Period) 동안 활동(Wake-up) 상태로 존재한다. 이 펄스 주기는 펄스 패킷의 송수신을 위한 펄스 흐름 전송(Pulse Flood Propagation)과 예약 패킷의 송수신을 위한 예약 시간(Reservation Time)으로 구성된다. 이 시간 동안에는 데이터 패킷의 전송이 중지된다. 각각의 펄스 패킷과 예약 패킷은 정해진 펄스 흐름 전송 시간 또는 예약 시간 내에 수신된 패킷만 유효하다. 예약 패킷을 전송하지 않은 노드는 다음 펄스 주기가 시작될 때까지 슬립 상태가 된다. 이를 통해 전체 네트워크의 전력 소모를 줄이는 효과를 얻게 된다. 이 동작 과정은 다음의 알고리즘 1에 기술된 PULSE 프로토콜의 타이머 핸들러 코드에 의해 구현된다.

센서 PULSE는 인프라 PULSE, 애드-혹 PULSE 프로토콜과 달리 다음 펄스 주기가 시작되기 전에 중간 활성 주기(Intermediate Wake-up Period)를 둔다^[1]. 이 기간에는 펄스 패킷을 전송하지 않고 예약 패킷만을 전송하여 경로를 갱신한다. 이것은 펄스 패킷의 브로드캐스트 횟수를 줄이기 때문에 브로드캐스트에 의한 전력 소모를 줄일 수 있다. 또한, 센서 PULSE는 모든 PULSE 프로토콜 중 가장 긴 펄스 주기를 갖기 때문에 펄스 패킷의 전송 횟수가 더욱 줄어든다. 이런 이유로 센서 PULSE는 PULSE 프로토콜들

알고리즘 1 PULSE 프로토콜의 타이머 핸들러
Algorithm 1 The Timer handler of PULSE protocol

```

/* 시간 간격 정의 */
Time_sync      : Sync Error 시간 간격.
Time_flood     : Flood Propagation 시간 간격.
Time_reserv    : Reservation 시간 간격.
Time_sleep     : Sleep or Send Data 시간 간격.
Time_pulse_interval : 타이머 주기 값을 의미.

/* 현재 시간 모드 정의 */
SYNC_TIME     : Sync Error 시간에 속함
FLOOD_PROPA  : Flood Propagation 시간에 속함
RESERVATION   : Reservation 시간에 속함
SLEEP_TIME    : Sleep or Send Data 시간에 속함

/* 변수 정의 및 초기화 */
integer timer ← SYNC_TIME // 타이머 상태
integer role ← NODE // 노드의 역할
integer onePeriod ← 0 // 타이머의 한 주기

/* 타이머 인터럽트에 의해 주기적으로 실행 */
PULSETimer_Handle ()
if (timer = SYNC_TIME)
    if (onePeriod > Time_pulse_interval)
        onePeriod ← 0
        timer ← FLOOD_PROPA
        if (role = SINK) send_pulse_pkt()
        set_timer_interval (Time_flood)
    else
        onePeriod ← onePeriod + Time_sync
        timer ← RESERVATION
        set_timer_interval (Time_reserv)

else if (timer = FLOOD_PROPA)
    onePeriod ← onePeriod + Time_flood
    if (role = NODE && EmptyQueue() = false)
        send_reservation_packet ()
    set_timer_interval (Time_reserv)

else if (timer = RESERVATION)
    onePeriod ← onePeriod + Time_reserv
    if (role = NODE && EmptyQueue() = false)
        send_data_packet()
    else if (role = NODE) go_to_sleep()
    set_timer_interval (Time_sleep)

else if (timer = SLEEP_TIME)
    onePeriod ← onePeriod + Time_sleep
    timer ← SYNC_TIME
    wake_up_node ()
    set_timer_interval (Time_sync)
END MPULSETimer_Handle ()
    
```

중 가장 전력 소모가 적은 특징을 갖는다.

그러나 센서 PULSE 프로토콜은 센서 네트워크가 이동이 거의 없는 특징을 갖는 고정 네트워크이고, 배터리의 소모와 환경 변화에 따른 변화만이 존재한다^[1]고 가정한다. 즉, 싱크 노드의 이동을 고려하지 않기 때문에 노드의 이동으로 인한 불규칙적인 경로 오류에 취약하다. 이는 경로 오류를 복구하기 위해서는 다음 펄스 주기에 펄스 패킷이 새로 전송될 때까지 기다려야 하기 때문이다. 본 논문은 경로 복구 시간을 단축하기 위한 기법을 추가하여 취약점을 극복

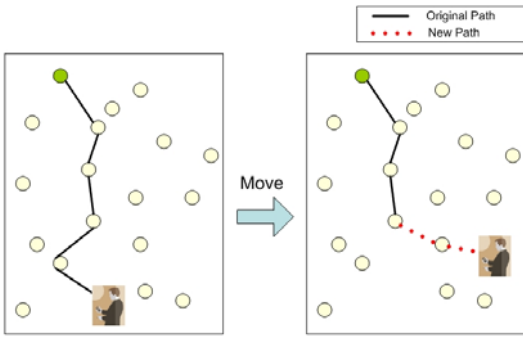


그림 2. 싱크 노드의 이동과 경로 복구
 Fig. 2. The route recovery in the sensor network with a mobile sink node

하고, 센서 PULSE의 적은 전력 소모 특징을 그대로 유지할 수 있도록 한다.

III. Mobile PULSE 프로토콜 : 경로 복구 시간 단축

3.1 센서 네트워크 내의 노드 이동

일반적으로 센서 네트워크는 이동이 거의 없는 네트워크로 여겨진다. 이는 특정 영역에 많은 수의 센서 노드를 설치하므로 센서 노드의 이동이 불필요하기 때문이다. 그러므로 대부분의 센서 네트워크 연구에서 이동성은 고려하지 않는다. 센서 PULSE^[1] 역시 노드의 이동은 고려하지 않고 있다.

하지만 센서 노드들로부터 데이터를 수집하고 이를 분석해야 하는 싱크 노드의 경우는 다르다. 싱크 노드는 수많은 센서 노드로부터 수집된 데이터를 모두 수용할 수 있어야 하고, 이 데이터를 분석할 수 있는 능력을 가져야 한다. 그러므로 싱크 노드는 센서 노드에 비해 고성능이어야 한다. 이것은 많은 수의 싱크 노드를 설치하는 것은 비용문제로 인해 어려움이 많다는 것을 의미한다. 하지만 싱크 노드가 이동을 한다면, 적은 비용으로 여러 개의 싱크 노드를 설치한 것과 같은 효과를 얻을 수 있다.

또한, 싱크 노드가 분석한 데이터를 보고 행동을 취해야 하는 것은 언제나 사람이다. 다시 말해, 센서로부터 언제 어떤 정보를 담고 있는 데이터가 전달될지 모르기 때문에 항상 담당자가 싱크 노드 앞에 대기하고 있어야 한다. 반면, PDA, 랩톱 컴퓨터 등의 휴대용 단말기를 이용한다면, 담당자는 단말기를 가지고 이동하면서 어디서든 센싱 데이터에 대한 확인이 가능하다. 이런 이유로 센서 네트워크에서 싱크 노드의 이동을 고려해야 한다. 싱크 노드가 이동할 수 있는 가능성에

대해서는 앞서 설명한 두 가지 이유와 서론의 3가지 시나리오 예로 설명이 가능하다.

서론에서 언급한 시나리오에서는 그림 2와 같이 싱크 노드는 센서 네트워크 내의 근거리를 이동하게 된다. 즉, 한번의 이동으로 수km ~ 수십km만큼의 장거리를 이동하지 않는다. 이런 근거리 이동은 싱크 노드의 이동으로 인해 영향을 받게 되는 센서 노드들을 싱크 노드의 주변 노드들로 한정할 수 있게 한다. 즉, 싱크 노드의 이동으로 인한 경로 오류는 싱크 노드와 가까운 주변 노드들과의 경로에만 한정된다^[8]. 그러므로 그림 2와 같이 전체 경로가 아닌, 싱크 노드와 주변 노드 사이의 경로에 대한 복구가 필요하다.

여기서 우리는 2가지의 가정을 정의할 수 있다. 첫째, 센서 네트워크 내에서 싱크 노드는 이동할 수 있다. 둘째, 싱크 노드의 이동에 영향을 받는 경로는 싱크 노드 주변 2-3홉 거리 내에 있는 경로이다.

3.2 싱크 노드의 이동을 고려한 PULSE 프로토콜의 개선

센서 PULSE 프로토콜에서 각 센서 노드는 펄스 패킷을 받게 되면 경로를 새로 생성한다. 즉, 싱크 노드의 이동이 발생했을 때, 펄스 주기에 따라 경로 복구 시간이 결정된다. 펄스 주기가 짧아지면 빠른 복구가 가능하지만, 펄스 패킷의 잦은 전송으로 인해 노드의 전력 소모가 많아진다. 본 논문에서는 이 단점을 극복하고 빠른 복구가 가능하도록 센서 PULSE 프로토콜을 개선하였다.

센서 PULSE 프로토콜은 시간 동기화에 의해 동작한다. 기존의 센서 PULSE에서는 펄스 패킷의 전송 횟수를 줄이기 위해서 펄스 주기를 길게 하고 중간에 중간 활성 주기를 두었다. 중간 활성 주기에는 예약 패킷만 송수신하기 때문에 펄스 패킷 전송으로 인한 전력 소모를 줄이면서도 경로를 갱신할 수 있다. 그러나 중간 활성 주기는 기존의 경로를 갱신하는 역할만 하기 때문에 노드의 이동에 대처하기 어렵다. 이동으로 인해 발생한 경로 오류를 복구하려면 새로운 경로를 생성하기 위해 긴 펄스 주기 동안 기다려야 한다. 이를 해결하기 위해 우리는 싱크 노드의 이동으로 인한 경로 오류의 복구 시간을 단축할 수 있도록 센서 PULSE의 타이밍(timing)을 수정하였다. 그림 3은 수정된 타이밍^[9]을 나타낸다.

수정된 타이밍은 기본적으로 센서 PULSE의 타이밍과 유사하다. 다만, 중간 활성 주기 외에 중간 펄스 주기(Intermediate Pulse Period)를 추가한 것이 다르다. 중간 펄스 주기는 앞서 언급한 두 가정을 기반

알고리즘 2 모바일 PULSE 프로토콜의 타이머 핸들러
Algorithm 2 The Timer Handler of Mobile PULSE protocol

```

/* 시간 간격 정의 */
.....
Timeinter_flood : Intermediate Flood Propagation 시간 간격

/* 현재 시간 모드 정의 */
.....
INTER_FLOOD_PROPA
: Intermediate Flood Propagation 시간에 속함

/* 변수 정의 및 초기화 */
.....

/* 타이머 인터럽트에 의해 주기적으로 실행 */
MobilePULSETimer_Handle ()
if (timer = SYNC_TIME)
    if (onePeriod > Timepulse_interval)
        onePeriod ← 0
        timer ← FLOOD_PROPA
        if (role = SINK) send_pulse_pkt()
        set_timer_interval (Timeflood)
    else if (onePeriod > Timepulse_interval / 2)
        onePeriod ← onePeriod + Timesync
        if (role = SINK) send_intermediate_pulse_pkt()
        set_timer_interval (Timeinter_flood)
    else
        onePeriod ← onePeriod + Timesync
        timer ← RESERVATION
        set_timer_interval (Timereserv)

else if (timer = FLOOD_PROPA OR
timer = INTER_FLOOD_PROPA)
    onePeriod ← onePeriod + Timeflood
    if (role = NODE && EmptyQueue() = false)
        send_reservation_packet ()
        set_timer_interval (Timereserv)
.....
END MobilePULSETimer_Handle ()
    
```

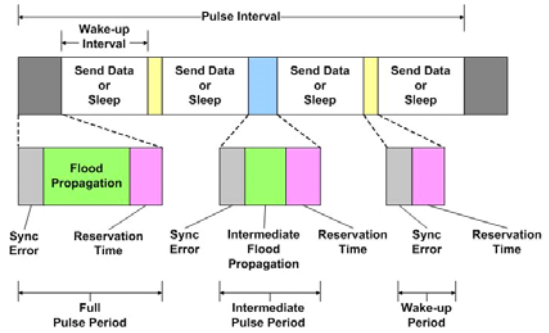


그림 3. Mobile PULSE 프로토콜의 타이밍^[8]
Fig. 3. The timing of the Mobile PULSE protocol^[8]

추가되는 부분은 중간 펄스 흐름 전송 시간뿐이다. 이 시간은 중간 펄스 패킷이 적은 홉 거리만 전송되기 때문에 매우 짧다. 그러므로 이 기간으로 인해 센서 노드들이 추가로 활성 상태를 유지해야 하는 시간도 짧다. 또한, 싱크 노드에서 2~3홉 거리에 있는 노드들만 중간 펄스 패킷의 전송을 하기 때문에 이 전송을 수행하는 노드의 수도 적다. 그러므로 기존의 센서 PULSE에 비해 전력 소모 면에서도 부담이 크지 않다. 즉, 싱크 노드로부터 2~3홉의 근거리에서 있는 주변 노드들에게 추가 펄스 패킷을 보내는 간단한 작업을 통해 싱크 노드의 이동으로 인한 데이터 패킷의 드롭(drop)을 줄일 수 있다. 반면, 전력 소모량은 기존 센서 PULSE 프로토콜과 비슷한 수준으로 유지할 수 있어 전력 소모를 줄이는 효과도 유지할 수 있다.

IV. 실험

4.1 실험 환경 및 시나리오

기존의 센서 PULSE 프로토콜과 모바일 PULSE 프로토콜의 성능 비교를 위해 두 프로토콜을 2.29 버전의 NS2^[9] 네트워크 시뮬레이터를 이용해 구현하였다. 새로 추가된 중간 펄스 주기 외의 다른 영향을 배제하기 위해 타이밍 부분을 제외한 나머지 부분은 동일하게 구현하였다.

실험에서 사용한 시나리오는 휴대용 단말기를 싱크 노드로 하는 경우를 가정한다. 이를 위해, 4 km X 1 km의 공간에 100개의 센서 노드를 배치하였다. 싱크 노드는 왼쪽 끝에서 오른쪽 끝으로 1.5m/s의 속도로 이동한다. 이는 사람의 걷는 속도와 각 노드의 전송 범위를 고려한 결과이다. 노드의 이동속도가 느리기 때문에 충분한 데이터를 얻기 위해 1시간에 걸쳐 실험을 진행하였다. 또한, 802.11 MAC을 사용하

으로 한다. 즉, 싱크 노드의 이동으로 인한 영향은 싱크 노드 주변의 노드에 한정되므로, 이들 노드에게만 펄스 패킷을 전송함으로써 경로의 복구가 가능해진다. 이를 위해 중간 펄스 주기는 중간 펄스 흐름 전송(Intermediate Flood Propagation) 시간 동안 중간 펄스 패킷을 전송한다. 중간 펄스 패킷은 2~3홉 정도만 전송하기 때문에 보통의 펄스 패킷 전송에 비해 훨씬 적은 수의 펄스 패킷 전송이 발생한다. 즉, 적은 컨트롤 패킷으로도 빠른 경로 복구를 제공할 수 있다. 이것은 중간 펄스 패킷의 전송으로 인한 전력 소모가 크지 않다는 것을 의미한다. 알고리즘 2

는 PULSE 프로토콜의 타이머 핸들러 코드에 경로 복구 시간 단축을 위해 추가된 부분을 나타낸다. 이 추가된 부분에 의해 중간 펄스 패킷을 전송함으로써 다음 펄스 주기가 시작되기 전, 기존보다 더 빠른 시점에서 경로를 복구할 수 있다.

중간 펄스 주기와 중간 활성 주기를 비교했을 때,

표 1. 802.11 무선 랜 전력 소모량
Table 1. 802.11 WLAN Power Consumption

전압	Energy Mode	전류	Power Consumption
4.74 V	Transmit	280 mA	1327 mW
	Receive	204 mA	966 mW
	Idle	178 mA	843 mW
	Sleep	14 mA	66 mW

여 무선 애드-혹 네트워크를 구성하였다. 일반적으로 센서에서는 802.11의 RTS/CTS 메커니즘이 사용되지 않기 때문에 해당 기능을 제거하였다^[1]. 본 논문에서는 센서 노드가 수집한 데이터를 일정 주기마다 전송한다고 가정하고, 일정한 주기에 따라 트래픽 이벤트를 발생하도록 하였다. 하나의 트래픽 이벤트는 20ms 간격으로 전송하는 5개의 패킷으로 구성하고, 이벤트의 발생 간격은 500ms, 600ms, 700ms, 800ms, 900ms, 1s, 10s, 60s로 세분화하였다.

각 센서 노드의 전력 소모를 측정하기 위해서 표 1과 같은 802.11 무선 랜의 전압, 전류, 그리고 이를 바탕으로 계산된 전력을 이용한다^[10]. 전력 모드는 Idle, Transmit, Receive, Sleep의 네 가지 모드로 구성된다.

Flood Propagation과 Reservation Time, Wake-up Period의 수, Wake-up Interval 등의 시간 파라미터는^[11]에 나타난 값을 참고로 결정하였다.

실험은 기존의 센서 PULSE와 모바일 PULSE의 비교를 중심으로 실행하였다. 먼저 모바일 PULSE와 센서 PULSE의 경로 복구에 걸리는 시간과 드롭되는 패킷의 양을 측정하였다. 또한, 경로 복구를 위해 프로토콜을 수정함으로써 나타나는 전력 변화를 측정하기 위해 두 프로토콜의 평균 전력 소모량을 측정하였다.

4.2 센서 PULSE와 모바일 PULSE 성능 비교

모바일 PULSE 프로토콜의 성능 측정을 위해 기존의 센서 PULSE 프로토콜과 비교 실험을 진행하였다. 실험 대상 프로토콜 중 1/2 PULSE는 센서 PULSE의 펄스 주기를 반으로 줄인 프로토콜을 의미한다. 이 절에서는 모바일 PULSE와 센서 PULSE의 성능 비교를 다루고, IV장의 3절에서 1/2 PULSE와의 성능 비교를 다룬다. 본 논문에서는 IV장의 1절에서 언급한 7가지 이벤트 주기마다 복구 시간을 측정하였다.

그림 4는 경로 복구에 걸리는 시간을 측정한 그래프이다. x축은 경로 복구에 걸리는 시간을 의미한다. 왼쪽 y축은 반복적인 경로 오류를 발생시키고 이를

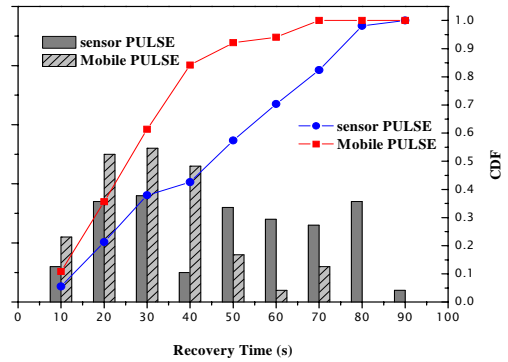


그림 4. 경로 복구 시간
Fig. 4 The route recovery time

복구하는데 걸리는 시간을 모두 측정한 후, 시간 대별로 구분하여 해당 시간 범위 안에 경로 복구를 완료한 사건의 개수(Frequency)로 나타낸다. 또한, 오른쪽의 y축은 왼쪽의 경우의 수를 모두 합한 누적 분포(CDF)를 의미한다. 경로 복구에 걸리는 시간은 데이터 패킷이 처음 드롭된 시점부터 드롭 이후 싱크에서 데이터 패킷을 수신한 시점까지의 시간으로 정의하였다. CDF 결과에 의하면, 모바일 PULSE는 복구에 걸리는 시간이 50초 이전에 약 90%가 몰려 있는 반면, 센서 PULSE는 약 60%의 경우만 존재한다. 즉 모바일 PULSE는 대부분의 경로 오류를 50초 이내에 복구할 수 있다.

이 결과는 각 복구 시간 별로 복구된 횟수를 나타낸 막대그래프에서도 나타난다. 빗금 친 모바일 PULSE의 경우 많은 수가 50초 이전에 몰려있고, 50초를 초과하는 경우는 총 101가지 경우의 수 중 8개로 나타난다. 즉, 경로 복구 시간은 모바일 PULSE가 센서 PULSE에 비해 약 1.5배 정도 빠르다. 이 실험 결과를 통해, 모바일 PULSE가 기존 센서 PULSE에 비해 경로 복구 시간이 더 짧다는 것을 알 수 있다.

경로 복구 시간의 단축은 현저히 데이터 패킷 드롭량의 감소로도 확인할 수 있다. 기존 센서 PULSE 보다 이른 시기에 경로가 복구됨으로써 이후의 데이터 패킷은 드롭되지 않게 된다. 그림 5는 네트워크 전체에서 드롭된 패킷의 양을 측정한 결과이다. 싱크 노드 이동으로 인해 발생하는 드롭만을 측정하기 위해 큐에서 오버플러우가 발생하지 않도록 하였다. 이를 위해 라우터의 큐 사이즈를 충분히 큰 값으로 결정하였다. 실험 결과, 모바일 PULSE는 모든 이벤트 발생 간격에서 드롭되는 데이터 패킷의 양이 크게 줄었다. 특히, 그래프에서 600ms 구간을 보면 약 2배 정도 줄었음을 확인할 수 있다. 드롭 패킷의 양 감소

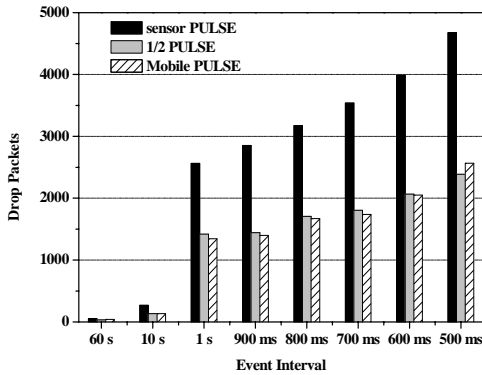


그림 5. 드롭되는 패킷의 양
Fig. 5. The number of drop packets

효과는 이벤트 간격이 작을수록 크게 나타난다. 이벤트 간격이 작을수록 데이터 패킷 전송의 주기가 짧아져 전송하는 데이터 패킷의 수가 많기 때문이다. 이것은 단위시간 동안 더 많은 드롭이 발생하게 한다. 그러므로 복구 시간이 단축되면 더 많은 양의 드롭을 줄이는 효과를 얻을 수 있다.

한편, 센서 네트워크에서는 경로 복구 시간을 단축하는 하면서도 전력 소모가 적어야 한다. 모바일 PULSE는 중간 펄스 주기가 추가되기 때문에 노드의 전력 소모량은 증가할 것으로 예상할 수 있다. 이 시간 동안 노드가 활성 상태로 되어 있어야 하고, 추가적인 중간 펄스 패킷의 브로드캐스트를 수행하기 때문이다. 그러므로 이에 대한 비교 실험이 요구된다. 이를 위해 센서 PULSE와 모바일 PULSE의 전력 소모량을 비교하는 실험을 진행하였다^[11]. 두 프로토콜의 타이밍 파라미터 값은 동일하게 하고, IV장 1절에서 언급한 네트워크 토폴로지에서 평균 전력 소모량을 측정하였다. 전력 소모량은 노드마다 차이를 보이기 때문에 전체 노드의 평균 전력 소모량을 측정하였다. 그림 6은 각 이벤트 발생 간격에 따라 평균 전력 소모량을 측정할 결과이다.

이 실험 결과를 보면, 모바일 PULSE의 평균 전력 소모량이 센서 PULSE에 비해 크게 증가하지 않는다는 것을 알 수 있다. 즉, 프로토콜의 수정이 전력 효율에 미치는 영향은 매우 작다고 할 수 있다. 이것은 노드들이 추가로 활성 상태를 유지해야 하는 시간이 100ms로 짧고, 중간 펄스 패킷의 전송에 관련된 노드가 싱크 노드로부터 2-3홉 거리에 있는 일부 노드로 한정되기 때문이다. 즉, 중간 펄스 주기의 추가로 더 많은 펄스 패킷이 전송됨에도 불구하고 이로 인한 전력 소모가 매우 적다. 이것은 그림 7의 실험 결

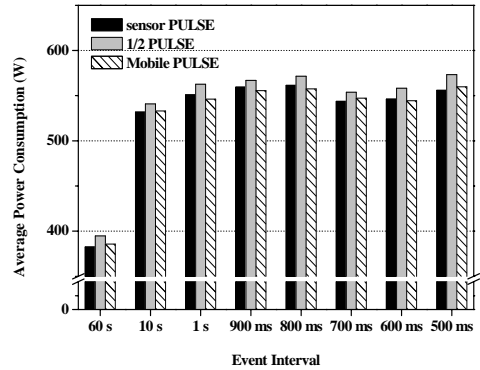


그림 6. 평균 전력 소모량
Fig. 6. The average power consumption

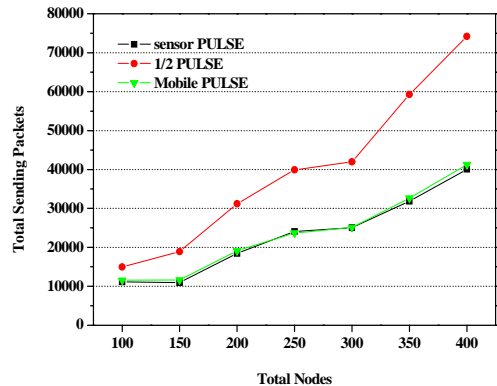


그림 7. 전송된 컨트롤 패킷의 수
Fig. 7. The number of transmitted control packets

과에 잘 나타난다.

그림 7은 총 노드 수에 따라 전송된 컨트롤 패킷의 양을 측정한 그래프이다. 컨트롤 패킷의 수만을 측정했기 때문에 데이터 패킷을 생성하는 이벤트 간격과 관계가 없다. 반면, 전체 노드의 수에 따라 브로드캐스트되는 패킷의 양과 패킷이 전달되는 홉 수가 달라지기 때문에 전체 노드의 수를 변화시켰다. x축은 전체 노드의 수이고, y축은 컨트롤 패킷의 전송량을 의미한다. 이 실험의 결과에 의하면, 모바일 PULSE의 컨트롤 패킷 전송량은 센서 PULSE와 거의 비슷하다. 중간 펄스 패킷은 싱크 노드에서 2-3홉 떨어진 노드에게만 브로드캐스트되기 때문에 전체 컨트롤 패킷 수에 비하면 그 양이 매우 적다. 그러므로 중간 펄스 패킷의 전송 횟수 증가는 전체 네트워크의 전력 소모량에 거의 영향을 미치지 않는다. 즉, 그림 6과 그림 7의 실험 결과와 같이 모바일 PULSE 프로토콜은 기존의 센서 PULSE 프로토콜과 비슷한 수준의 전력량 감소 효과를 얻을 수 있다.

이들 실험을 통해, 모바일 PULSE는 센서 PULSE에 비해 더 나은 성능을 보임을 알 수 있다. 즉, 전력 소모량에서 기존 센서 PULSE와 거의 대등한 성능을 보이면서 싱크 노드의 이동으로 인한 경로 오류를 더 빠르게 복구할 수 있다. 이것은 모바일 PULSE가 싱크 노드의 이동에 대해서 기존 센서 PULSE보다 더 나은 성능을 보임은 물론, 고정된 싱크 노드를 가진 센서 네트워크에서도 별다른 수정없이 사용할 수 있다는 것을 의미한다.

4.3 펄스 주기의 단축과 모바일 PULSE 프로토콜

PULSE 프로토콜은 특성상 펄스 주기에 따라 경로 갱신 주기가 달라진다. 그러므로 싱크 노드가 이동하더라도 이 펄스 주기를 조절함으로써 경로 복구 시간을 단축할 수 있다. 이런 경우에 우리가 제안한 모바일 PULSE 프로토콜과 비교하여 어떤 차이가 있는지 알아보기 위해 실험을 진행하였다. 이를 위해 센서 PULSE의 펄스 주기를 1/2로 줄여 모바일 PULSE와 비슷한 시간에 경로 복구가 가능하도록 수정하였다. 그리고 역시 같은 네트워크 토폴로지에서도 트랩된 데이터 패킷의 양, 평균 전력 소모량을 측정하였다. 각 실험의 결과는 IV장 2절의 그래프에 1/2 PULSE의 이름으로 포함되어 있다. 모바일 PULSE와 같아지도록 펄스 주기를 수정한 것이기 때문에 경로 복구 시간은 따로 측정하지 않았다.

1/2 PULSE는 데이터 패킷의 손실 면에서 모바일 PULSE와 비슷한 성능을 보이고 있다. 그림 5에 나타난 바와 같이, 트랩된 데이터 패킷의 양은 센서 PULSE보다 2배 가까이 줄었고 모바일 PULSE와는 거의 비슷하다. 즉, 단순히 펄스 주기를 줄이는 것만으로도 경로 복구 시간 단축이 가능하다.

그러나 이 방식에는 전력 소모량 면에서 큰 단점이 존재한다. 그림 6에 의하면, 1/2 PULSE는 모든 이벤트 간격에서 전력 소모가 더 많은 것을 알 수 있다. 이것을 수치로 나타내면 다음 표 2와 같은 전력 소모량의 차이를 보인다. 이 결과에 의하면 1/2 PULSE는 모바일 PULSE에 비해 최소 7.8W에서 최대 13.9W의 전력을 더 소모한다. 또한, 모바일 PULSE의 경우 중간 펄스 패킷을 주변 2-3홉에만 브로드캐스트하지만, 1/2 PULSE는 모든 노드에게 브로드캐스트한다. 이로 인해, 전송하는 컨트롤 패킷의 수에서 그림 7과 같은 결과가 나타난다. 특히, 전체 노드 수가 증가함에 따라 컨트롤 패킷의 수는 더 큰 폭으로 증가한다. 모바일 PULSE는 싱크 주변에 위치한 적은 수의 노드에게만

표 2. 프로토콜의 전력 소모량의 차이

Table 2. The difference of the power consumption in protocols

	모바일 PULSE - 센서 PULSE	1/2 PULSE - 센서 PULSE	1/2 PULSE - 모바일 PULSE
60 s	3.126	12.187	9.060
10 s	1.225	9.094	7.869
1 s	-4.826	11.631	16.457
900 ms	-3.720	7.639	11.360
800 ms	-3.790	10.162	13.952
700 ms	3.636	10.300	6.663
600 ms	-1.698	12.285	13.983
500 ms	4.046	17.570	13.524

펄스 패킷을 전송하므로 컨트롤 패킷의 증가량이 매우 적다. 반면, 1/2 PULSE는 모든 노드에게 브로드캐스트하기 때문에 펄스 패킷의 수가 크게 증가한다. 이처럼 1/2 PULSE는 많은 펄스 패킷의 전송으로 전력 소모량이 증가한다. 그러므로 노드 수가 많고 전력 소모를 최소화해야 하는 센서 네트워크에서 1/2 PULSE는 적합하지 않다.

이와 같이 기존 센서 PULSE의 펄스 주기 단축은 빠른 경로 복구가 가능하지만, 잦은 펄스 패킷의 전송으로 인해 전력 소모가 많아지는 단점이 있다. 하지만 우리가 제안하는 모바일 PULSE는 기존 센서 PULSE와 비슷한 수의 컨트롤 패킷만으로 경로 복구 시간을 단축시킬 수 있다. 이것은 전력 소모량을 기존 센서 PULSE와 비슷한 수준으로 유지할 수 있게 한다. 즉, 모바일 PULSE는 경로 복구 시간 단축과 소모 전력량 감소의 두 목적을 모두 달성할 수 있다. 그러므로 싱크 노드의 이동을 고려한 센서 네트워크에서는 단순히 펄스 주기를 짧게 하는 방법보다 모바일 PULSE를 이용하는 것이 더 효율적이다.

V. 관련 연구

싱크 노드의 이동을 고려한 센서 네트워크에서의 라우팅 프로토콜 연구는 싱크 노드에서 경로 생성을 시작하는 방법과 소스 노드에서 시작하는 방법으로 나뉜다^[2]. 싱크 노드에서 경로를 생성하는 프로토콜로는 Directed Diffusion^[13], SAFE^[14], MobiRoute^[15] 등이 있고, 소스 노드에서 경로를 생성하는 프로토콜에는 SEAD^[16], TTDD^[17], TTCG^[18] 등이 있다.

MobiRoute는 Berkeley MintRoute^[19] 프로토콜을 노드의 이동을 고려해 확장한 프로토콜이다. 두 프로토콜 모두 거리-벡터 알고리즘(distance-vector algorithm)에 기반한다. MobiRoute는 싱크 노드의 이동을 고려하여

4가지 상태 정보(pre-move, move, pre-sojourn, sojourn)를 정의한다. 싱크 노드는 상태를 sojourn에서 pre-move로 변경하고 비콘 메시지를 전송한다. 비콘 메시지의 전송이 완료되면 싱크 노드는 move 상태로 변환하고 이동을 시작한다. 각 센서 노드는 첫 비콘 메시지를 수신하면 자신의 상태를 sojourn에서 move 상태로 변환한다. 싱크 노드는 이동이 멈추면, 자신의 상태를 pre-sojourn으로 변경하고 더 빠른 전송 빈도(speed-up rate) 경로 메시지(route message)를 전송한다. 싱크 노드로부터 직접 이 메시지를 받은 노드는 자신의 상태를 pre-sojourn으로 변경하고 메시지를 다른 노드에게 전송한다. 그 후 타이머에 의해 모든 노드는 sojourn 상태가 되고, 새로운 경로를 통해 데이터 전송을 수행한다. 그러나 싱크 노드가 계속해서 이동하면 데이터 전달이 어렵고, 싱크 노드는 자신이 이동 중인지 멈춘 상태인지를 알아야 하는 단점이 있다.

TTDD는 가상의 좌표(grid)를 이용해 싱크 노드까지 데이터를 전송한다. 보낼 데이터를 가진 센서 노드(source)는 데이터 알림 메시지를 자신이 속한 좌표 셀의 4군데 가상의 전달 지점(dissemination point)로 전송한다. 이때, 이 지점에 가장 가까운 센서 노드가 전달 노드(dissemination node)가 된다. 전달 노드는 소스 노드에 대한 정보를 저장하고, 수신한 알림 메시지를 다른 셀의 전달 노드에게 보낸다. 이 과정은 알림 메시지 안에 있는 지점에 가장 가까운 전달 노드에 도달할 때까지 반복된다. 그 후, 싱크 노드는 데이터가 필요할 경우 데이터를 요구하는 쿼리를 전송한다. 이 쿼리는 싱크 노드에 가장 가까운 전달 노드에게 전송된다. 전달 노드는 자신이 가진 소스 노드의 정보를 통해 원하는 데이터를 가진 소스를 찾고, 해당 소스의 알림 메시지를 전달해 준 인접 셀의 전달 노드에게 요청 쿼리를 전달한다. 이 과정을 반복하여 소스까지 도달하면, 소스는 쿼리가 전달된 방향으로 데이터를 전송한다. 그러나 싱크 노드를 제외한 모든 센서 노드가 자신의 위치를 알아야 하는 단점이 있다.

또한, TTDD는 여러 영역에서 Interest 이벤트가 빈번하게 발생하는 경우 가상 좌표계 생성에 요구되는 패킷들의 증가로 전력 효율성이 떨어지는 단점이 있다. 이를 해결하기 위해 TTCG 프로토콜이 제안되었다. 이 프로토콜은 하나의 좌표계를 더 추가하여 3개의 좌표계를 이용한다. 추가된 좌표계는 모바일 노드를 위한 것으로 TTDD에서 생성한 좌표계와 동일한 좌표계를 이룬다. 이 기법은 TTDD에서 사용되던 각각의 전달 노드들을 고정된 싱크 노드로 사용한다.

이들 노드는 자신의 가상 좌표 셀 안에 포함된 센서 노드들로부터 데이터를 수집한다. 모바일 노드는 자신의 셀 안에 있는 고정 싱크 노드에게 데이터를 요청하여 수집한다. 이 때, 모바일 노드는 고정 싱크 노드의 알림 메시지(Announcement message)를 주기적으로 수신하여 자신의 위치를 파악한다. 이 방식은 반복적인 좌표계 구성으로 인한 전력 낭비를 방지할 수 있지만, 고정된 싱크 노드에 의존하기 때문에 이들 노드의 전력 소모가 크고 이들의 오류가 데이터 경로의 오류로 직결되는 단점이 있다.

VI. 결 론

우리의 연구는 두 가지 가정에서 출발한다. 첫 번째는 싱크 노드가 이동할 수 있다는 것이고, 두 번째는 경로 변화가 생기는 노드는 싱크 노드 주변으로 한정된다는 것이다. 이 두 가정을 통해 싱크 노드와 2~3홉 떨어진 노드와의 경로만 복구하면 데이터 전송이 재개된다는 것을 유추할 수 있다. 이것이 제안한 모바일 PULSE 프로토콜의 기본 바탕이 된다.

모바일 PULSE 프로토콜은 기존의 센서 PULSE 프로토콜에 중간 펄스 주기를 추가하는 간단한 작업으로 경로 복구 시간을 단축할 수 있다. 또한, 중간 펄스 패킷을 싱크 노드 주변의 2~3홉 범위 안에만 브로드캐스트하기 때문에 추가적인 패킷 전송이 많지 않다. 즉, 모바일 PULSE 프로토콜은 센서 네트워크 내에서의 싱크 노드 이동을 가능하게 하면서 소모 전력량의 증가를 억제할 수 있다. 이처럼 프로토콜의 수정으로 인한 영향이 작기 때문에 고정된 싱크 노드를 사용하는 센서 네트워크에서도 기존 센서 PULSE와 성능 차이가 적다. 이는 모바일 PULSE 프로토콜은 모바일 싱크 노드를 가진 센서 네트워크와 고정된 싱크 노드를 가진 센서 네트워크 모두에 적합함을 의미한다.

본 논문은 모바일 PULSE의 경로 복구 능력과 전력 소모량을 보임으로써 모바일 센서 네트워크에 모바일 PULSE가 적합함을 증명하였다. 제안하는 모바일 PULSE는 경로 복구 시간을 단축하였으나 전력 소모량에서 약간의 증가를 가져왔다. 향후 중간 펄스 주기의 실행 시기와 횟수에 따른 영향을 고려한 계속적인 연구가 진행된다면 전력 소모량을 더욱 감소시킬 수 있다. 본 논문은 모바일 PULSE 프로토콜을 제안하고, 모바일 센서 네트워크 환경에 적합한 프로토콜로서의 가능성을 제시하고 있다.

참 고 문 헌

- [1] Baruch Awerbuch, David Holmer, Herbert Rubens, Kirk Chang, and I.-J. Wang, "The Pulse Protocol: Sensor Network Routing and Power Saving", In Proceedings of *Military Communications Conference (MILCOM 2004)*, Nov. 2004.
- [2] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks", In Proceedings of *21st International Annual Joint Conference of the IEEE Computer and Communications Societies*, June 2002.
- [3] S. Singh and C. Raghavendra, "Pamas: Power aware multi-access protocol with signaling for ad hoc networks", In *ACM Computer Communications Review*, 1999.
- [4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks", In *IEEE Communications Magazine*, 40(8):102 - 116, August 2002.
- [5] Baruch Awerbuch, David Holmer, and Herbert Rubens, "The Pulse Protocol: Energy Efficient Infrastructure Access", In Proceedings of *The 23rd Conference of the IEEE Communications Society INFOCOM*, 2004.
- [6] Baruch Awerbuch, David Holmer, and Herbert Rubens, "The Pulse Protocol: Mobile ad hoc network performance evaluation", In Proceedings of *Wireless On-demand Network Systems and Services 2005 (WONS 2005)*, January 2005.
- [7] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," In Proceedings of *2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
- [8] 이 신형, 유혁, "M-PULSE: 싱크 노드의 이동성 지원을 위한 센서 네트워크 라우팅 프로토콜", 한국정보과학회 추계학술대회, 2003
- [9] "NS-2 Network Simulator", <http://www.isi.edu/nsnam/ns/>
- [10] L.M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment", In Proceedings of *INFOCOM*, pp.1548-1557, April 2001.
- [11] 이치영, 이신형, 유혁, "모바일 싱크 노드를 가진 센서 네트워크를 위한 에너지 효율적인 라우팅 기법", 한국정보과학회 종합학술대회, 2008.
- [12] Kwang-il Hwang, and Doo-seop Eom, "Adaptive Sink Mobility Management Scheme for Wireless Sensor Networks", In Proceedings of *Ubiquitous Intelligence and Computing Third International Conference (UIC 2006), Lecture Notes in Computer Science*, Vol. 4159, pp.487-487, 2006.
- [13] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion for Wireless Sensor Networking", In *IEEE/ACM Transaction on Networking*, Vol. 11, 2003.
- [14] Sooyeon Kim, Sang H. Son, John A. Stankovic, Shuoqi Li, and Yanghee Choi, "SAFE: A Data Dissemination Protocol for Periodic Update in Sensor Networks", In Proceedings of *the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)*, 2003.
- [15] Jun Luo, Jacques Panchard, Michal Piorowski, Matthias Grossglauser, and Jean-Pierre Hubaux, "MobiRoute: Routing towards a Mobile Sink for Improving Lifetime in Sensor Networks", In Proceedings of *International Conference on Distributed Computing in Sensor Systems*, June 2006.
- [16] Hyung Seok Kim, Tarek F. Abdelzaher, and Wook Hyun Kwon, "Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks", In Proceedings of *Embedded Networked Sensor Systems*, 2003.
- [17] Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, and Lixia Zhang, "TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks", In *Wireless Networks*, Volume 11, Issue 1-2, January 2005.
- [18] S. Sho, K. Kim, W. Jun, J. Kim, S. Kim, and J. Lee, "TTCG: Three-Tier Context Gathering Technique for Mobile Devices", In Proceedings of *the 5th international conference on Pervasive services*, pp.152-162, 2008.
- [19] Alec Woo, Terence Tong, and David Cullar, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks", In Proceedings of *the 1st international conference on Embedded Networked Sensor Systems*, 2003.

