

에너지 효율적 센서 데이터 수집을 위한 리포팅 허용 지연시간 적응형 센서 제어 기법 구현 및 성능평가

손 태 식[†] · 최 호 현^{††}

요 약

다양한 응용에 적용될 수 있는 특성을 가진 무선 센서 네트워크는 적용되는 응용에 따라 데이터 리포팅 허용 지연시간이 다양하다. 따라서, 각 응용별로 센서를 제어하는 설계 패러다임을 적용하여 에너지 효율을 최대화하고 네트워크의 생존기간을 최대화할 수 있어야 한다. 본 논문에서는 2단계 클러스터링 (Two Phase Clustering:TPC) 방식을 기반으로 하여 센서 데이터 수집시 에너지 효율을 극대화 하는 리포팅 허용 지연시간 적응형 센서 스케줄링 방안을 구현하고 성능을 평가한다. 센서 Mote상에 직접 구현된 스케줄링 기법은 제어 메시지나 시간에 민감한 센서 데이터들을 포워딩하는 직접 링크와 릴레이 링크의 두 가지 링크를 활용하며 실제 환경에서의 에너지 효율을 측정하여 센서들이 두 링크를 사용함으로써 갖는 에너지-절약 효과를 보인다.

키워드 : 에너지 절약, 클러스터링, 데이터 수집, 스케줄링, 센서 네트워크

Implementation and Performance Evaluation of Reporting Interval-adaptive Sensor Control Scheme for Energy Efficient Data Gathering

Taeshik Shon[†] · Hyo Hyun Choi^{††}

ABSTRACT

Due to the application-specific nature of wireless sensor networks, the sensitivity to such a requirement as data reporting latency may vary depending on the type of applications, thus requiring application-specific algorithm and protocol design paradigms which help us to maximize energy conservation and thus the network lifetime. In this paper, we implement and evaluate a novel delay-adaptive sensor scheduling scheme for energy-saving data gathering which is based on a two phase clustering (TPC), in wireless sensor networks. The TPC is implemented on sensor Mote hardwares. With the help of TPC implemented, sensors selectively use direct links for control and forwarding time critical sensed data and relay links for data forwarding based on the user delay constraints given. Implementation study shows that TPC helps the sensors to increase a significant amount of energy while collecting sensed data from sensors in a real environment.

Keywords : Energy Saving, Clustering, Data Gathering, Scheduling, Sensor Network

1. 서 론

센서의 제한된 에너지 보유능력으로 인해 최근 무선 센서 네트워크용 프로토콜 설계에서 가장 강조되는 것은 센서 네트워크의 생존 기간을 최대화하기 위해 센서들이 에너지를 최대한 효율적으로 사용하도록 하는 것이다. 센서 네트워크

는 대부분 정적이며 센서들이 밀도 높게 구성되어 있는 특징이 있어 여러 센서들에서 중복된 데이터가 센싱 되어 전송되는 문제를 가지며 이로 인해 불필요한 전력 소모가 발생하고 결국 네트워크의 생존기간을 감소시킨다. 일반적으로 센싱된 데이터의 전송은 센서의 동작 중 가장 많은 양의 에너지 소모를 발생 시킨다. 따라서 중복된 데이터를 센싱하여 전송하는 상황을 제거하거나, 에너지를 가장 절약하는 통신 링크를 사용하거나, 통신 횟수를 최소화하는 등의 방법으로 많은 양의 에너지를 절약하여 네트워크의 수명을 연장하게 된다. 이러한 목적을 달성하기 위해, 네트워크는 제한된 에너지 사용의 로컬 최적화를 이루도록 설계되어야 한

※ 본 논문의 구현을 위해 많은 도움을 주신 한국외대 정찬호군에게 감사의 말씀을 드립니다.

† 정 회 원: 삼성전자 Digital Media & Communications 연구소 책임연구원

†† 정 회 원: 인하공업전문대학 컴퓨터정보과 조교수(교신저자)

논문접수: 2010년 5월 7일

수정일: 1차 2010년 7월 30일, 2차 2010년 9월 7일

심사완료: 2010년 10월 5일

다. 예를 들면 클러스터링[1], 데이터 집적[2], 에너지 사용의 균형을 맞추는 라우팅[3]등은 로컬 최적화를 구성하는 방법의 좋은 예가 된다.

본 논문에서는 네트워크를 논리적으로 그 크기를 감소시켜 전송 횟수를 줄임으로서 에너지 소모를 감소시키는 효율적인 TPC (Two Phase Clustering) 클러스터링 기법[4]을 기반으로 하는 리포팅 허용 지연시간 적응형 센서 스케줄링 방안을 실제 환경에서 구현하고 성능을 평가한다. 이 클러스터링 기반 센서 데이터 수집 기법은 리포팅 허용 지연시간 제한과 같은 사용자나 응용의 특정 요구사항을 만족시키며 궁극적으로 네트워크의 생존기간을 연장시키기 위해 에너지 효율을 최대화 한다. 센서 Mote(일명 Mote라고도 불리워지며 Mote의 사전적인 의미는 ‘a small particle’)[5]상에 직접 구현된 TPC 기반 센서 제어 방식은 클러스터 헤드와의 직접 링크를 구성하고 동일 클러스터내의 이웃 노드와 릴레이 링크를 구성한다. 직접 링크는 제어 메시지나 시간에 민감한 센서 데이터들의 포워딩에 사용되며 릴레이 링크는 최대한 데이터를 포워딩하여 저전력 멀티홉 경로를 구성하는데 사용한다. 실제 환경에서 센서들은 이러한 두 링크를 요구된 허용 지연시간에 따라 적절히 사용함으로써 에너지 절약 효과가 기존 클러스터헤드와 멤버간의 직접 통신을 하는 1홉 클러스터링 (이하, One-hop Clustering (OCS)라 함) 대비 상당히 높음을 보인다.

이 논문의 구성은 2장에서 제안하는 구현 기법에 대해 소개하고 3장에서는 제안 내용의 설계 및 실제 구현 내용에 대해 자세히 설명한다. 이후 4장에서 구현된 기법에 대한 에너지 절약 성능 평가 결과를 보여주고 5장에서 결론을 맺는다.

2. 지연 시간 적응형 센서 지연 연구

센서 네트워크의 응용-의존적 특성을 고려할 때 응용마다 주어진 작업을 성공적으로 수행하기 위해 필요한 데이터 수집 허용 전달 지연시간은 각기 다양할 수 있다. 그러므로 센서들의 데이터 보고 간격은 사용자나 특정 응용의 허용지연시간을 만족시키고 데이터 중복성을 최대한 제거하며 에너지 절약을 최대화 할 수 있어야 한다. 이를 본 논문은 두 단계 클러스터링 (TPC)을 기반으로 한 센서 제어를 시도한다. 다음은 TPC의 두단계 (클러스터 구성과 클러스터내 릴레이링크 구성) 클러스터링에 대해 간략히 소개하며 상세한 특징 및 알고리즘에 대해서는 [4]를 참조 할 수 있다.

2.1 Cluster 구성

각 센서들은 랜덤 타이머 시간을 가지고 Cluster Head (CH)로서 선정되기 위해 경쟁하며 CH로 선정 되지 않은 각 센서노드는 자신의 클러스터를 선택하는데 에너지 소모를 최소화하기 위해서 CH까지의 거리를 기준으로 자신이 속할 클러스터를 선택한다. 즉, 자신으로부터 가장 가까운 것을 선택하게 된다. 만약 동일 거리를 갖는 다수의 CH가 존재할

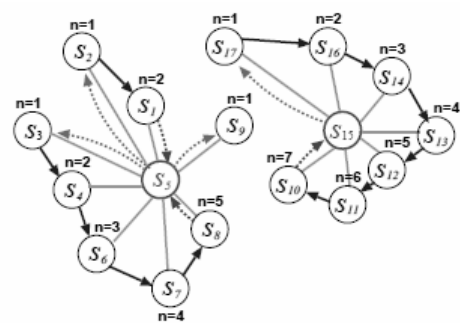
경우 한 CH를 임의로 선택한다. 센서들은 선택된 CH로 JOIN-REQUEST를 보내서 최종 Cluster Member (CM)가 된다. 클러스터 구성과정 동안에, 릴레이 링크 구성을 위한 두 번째 단계를 위해 각 센서는 어떤 이웃 노드들이 CH로서 동작하는지, 어떤 노드가 클러스터 내에서 CM으로 동작하는지, 이웃노드가 어떤 클러스터 내에 위치하는지에 대한 필요 정보를 수집한다.

2.2 클러스터 내 릴레이 링크 구성

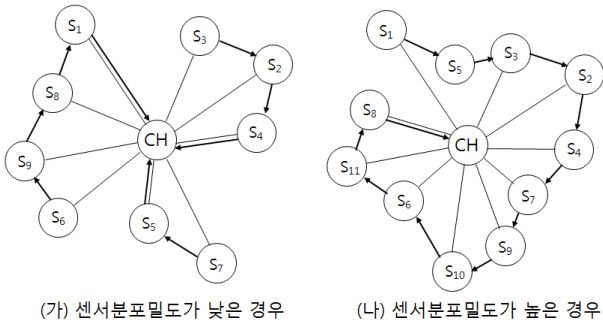
각 CM들은 CH와의 직접링크보다 거리상으로 짧은 릴레이링크 구성을 위해 클러스터내 한 이웃노드를 선택한다. 릴레이 링크 구성의 목적은 지연 시간에 대한 제한 사항을 최대한 만족시키면서 수집된 센싱 데이터의 중복성을 제거하며 CH까지 전달하는데 사용되는 에너지 소모를 최소화하는 것과 CH와 CM간의 통신 횟수를 줄여 CH의 부하를 분산시키는 것이다.

(그림 1)은 구성된 클러스터 내에서 CM간 릴레이 링크를 구성하는 예를 보인다. 여기서 직선링크는 직접링크, 화살표 직선은 릴레이 링크를, n은 각 CM이 갖게 되는 SEQID를 나타내며 밖으로 향하는 화살표 점선은 CH의 릴레이 링크 구성 요청을 나타낸다. 일단 릴레이 링크 구성에 참여하여 센서 데이터를 수신 받는 릴레이 포인트 노드가 되면 그 노드는 다른 노드의 릴레이 포인트로 선택되지 못하며 자신의 릴레이 포인트만을 선택할 수 있다. 클러스터 내에서 센서의 분포밀도에 따라 어떤 CM은 릴레이 포인트를 찾는데 실패하는 경우도 발생 할 수 있기 때문에 최종 클러스터내의 릴레이 링크 구성은 (그림 2)에서 보이는 형태가 된다.

CM들은 릴레이 포인트로 센싱 데이터를 포워딩하며 릴레이 포인트는 자기의 센싱 데이터와 수신한 데이터를 수집/집적하여 자신의 릴레이 포인트로 전송한다. 하지만 릴레이 포인트가 존재하지 않을 경우 바로 직접링크를 사용하여 CH로 전송한다. 릴레이 링크나 직접 링크 중에 어느 것을 사용할지의 선택은 사용자나 응용이 정한 허용지연시간 제한에 따라 CH에 의해 제어되며 아래에 제어방법에 대해 자세히 설명한다.



(그림 1) 클러스터의 구성과 클러스터내의 릴레이링크 구성 (화살표선은 릴레이링크, 직선은 직접링크를 나타냄)
(Fig. 1) Construction of Relay Links. Directed lines and lines Represent Relay Links and Direct Links, Respectively

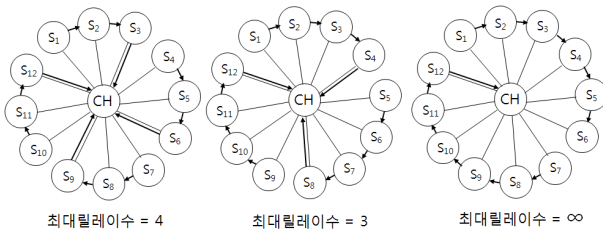


(가) 센서분포밀도가 낮은 경우 (나) 센서분포밀도가 높은 경우

(그림 2) 센서분포밀도에 따른 릴레이 링크구성 예 (Fig. 2) Use of Relay Links Depending on Node Density

2.3 리포팅 허용 지연시간 제약에 따른 센서 제어

데이터 보고 지연시간은 응용의 종류에 따라 정적인 경우도 있고 동적인 경우도 있다. 동적인 경우에 사용자들은 환경적 상황이나 이미 전송된 센싱 데이터의 분석에 따라 제어 메시지를 보내서 센서의 행위 (예를 들면, 센싱이나 데이터 보고 간격)를 제어한다. 반면에 정적인 경우에는, 센서들은 초기에 주어진 임계값에 기반하여 데이터 보고 주기를 결정한다. 예를 들어, 초기 데이터 보고 간격 $\Delta T_1 = 30$ 초라 가정 하자. 이 초기 간격은 모든 CM이 릴레이 링크를 통해 센싱 데이터를 포워드할 수 있을 만큼 충분히 커서 최대 에너지 절약 효과를 위해 모든 CM은 릴레이 링크를 사용하도록 제어된다. 현재 주어진 $\Delta T_1 = 30$ 를 가지고 센서들이 주어진 업무를 수행하는 동안, 사용자는 전달된 센싱 데이터를 분석하여 새로운 보고 간격, 예를 들면 $\Delta T_2 = 12$ 초인 요구사항을 CH로 보낸다. 허용지연시간 제한요건을 받은 CH는 이 제한요건을 만족할 수 있는 릴레이의 최대 수를 계산한다. 예를 들어 최대 릴레이 수가 4라 하면 CH는 4-릴레이 제어 메시지를 브로드캐스트 하고 CM은 $SEQID \bmod 4$ 의 결과가 "0"이면 직접 링크를 그 외는 릴레이 링크를 사용하여 자신의 센싱 데이터를 CH에 전달한다. (그림 3)은 최대 릴레이수가 3, 4, 그리고 무제한(∞)일 경우에 릴레이 링크를 이용한 데이터 수집 예를 보인다.



최대릴레이수 = 4 최대릴레이수 = 3 최대릴레이수 = ∞

(그림 3) 응용에서 요구 되어지는 데이터 보고 허용 지연시간에 따른 릴레이경로 구성 예 (Fig. 3) Reporting delay-adaptive Relay Path Construction

3. 설계 및 구현

리포팅 허용 지연시간 적응형 센서 제어 기법의 구현 및

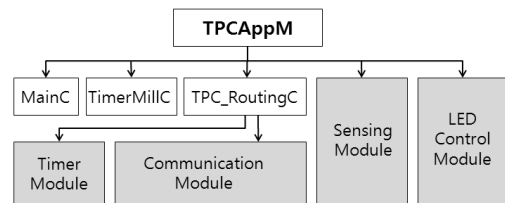
<표 1> ZigbeX 센서노드 제원 <Table 1> ZigBex Sensor Mote Specification

MCU	CPU: ATmega128L (7.4Mhz) Flash Memory : 128Kb, SRAM : 4Kb
RF 모듈	CC2420 (RF:2.4GHz)
Sensor	온습도, 조도, Photo Diode
OS	TinyOS 2.x

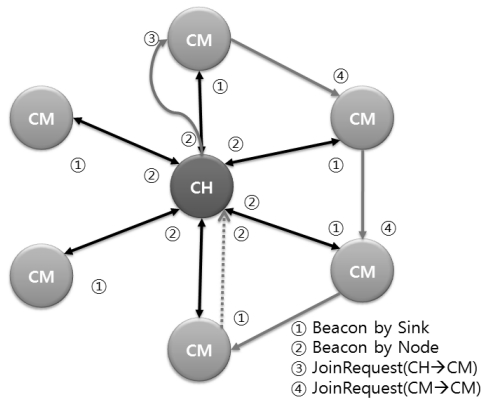
성능평가를 위해 한백전자[6]의 유비쿼터스 센서 네트워크 시스템인 ZigbeX[5]을 사용하였고, 제원은 <표 1>과 같다. ZigbeX 플랫폼에 TinyOS와 nesC를 사용하여 TPC 구조의 센서네트워크 어플리케이션 (TPCAppM)의 전체적인 구성은 (그림 4)와 같다.

어플리케이션에서 사용하는 MainC 컴포넌트에 각각의 센서 Mote가 수집해야 할 데이터와 수집 주기, Host PC로 데이터를 전송하는 기능을 구현하였다. 각각의 센서 Mote는 TCP_RoutingC 컴포넌트를 활용하여 TPC를 구성하고, MainC가 전달한 데이터를 릴레이 포인트 또는 클러스터 헤드에 포워딩하는 역할을 한다. 통신 메시지 구성을 위해서는 Communication 모듈에서 제공하는 인터페이스는 ActiveMessageC, AMSenderC와 AMReceiverC를 사용하였다 (자세한 정의는 [5]를 참조). 센서 Mote의 개수와 실험 환경에 제한적이어서 실질적인 대규모 센서네트워크를 구성할 수 없었기 때문에 물리적인 구현 네트워크는 CH 역할을 하는 노드 1개와 클러스터 멤버 역할을 하는 노드 최대 8개로 구성하였다.

TPC 구성을 위한 라우팅 프로토콜은 두 단계로 구성되는데, 먼저 CH를 중심으로 CM들이 스타형 네트워크를 형성하는 단계와, 앞 단계에서 구성한 스타형 네트워크에 부가적인 에지의 집합을 추가하여 체인 또는 불완전한 휠 (Incompleted Wheel) 구조를 갖는 네트워크로 재구성하는 단계로 나누어 TCP_RoutingC 컴포넌트에 구현하였다. 첫 번째 단계에서 각각의 센서 Mote들이 자신의 정보를 주변에 알리는 두 가지 종류의 Beacon 메시지를 전송한다. Beacon 메시지를 수신한 주변의 센서 Mote들은 Beacon 메시지로부터 주변 Mote들에 대한 정보를 추출하여 라우팅 테이블에 저장한다. 그리고 두 번째 단계에서는 구성된 라우팅 테이블을 이용하여 네트워크에 부가적인 에지들의 집합을 추가하여 TPC 클러스터를 구축한다. (그림 5)는 TPC를 구성하는 단계에 필요한 Beacon 메시지들의 전송과정을 간략하게 나타낸 것이다.



(그림 4) 애플리케이션 컴포넌트 구성도 (Fig. 4) Application Component Architecture



(그림 5) TPC 구성에 필요한 Beacon 메시지
 (Fig. 5) Use of Beacon Messages for TPC Formation

TPC 클러스터를 구축하는 세부 단계의 구현은, 먼저 첫 번째 단계에서 CH는 자신을 중심으로 한 CM들의 스타형 네트워크를 구성하기 위해 자신의 상태를 다른 센서 Mote들에게 알리는 Beacon 메시지를 전송한다. Beacon 메시지는 CH가 전송하는 BeaconSink와 CM이 전송하는 BeaconNode로 나뉜다. CH는 타이머가 트리거 될 때 마다 BeaconSink를 전송하고, BeaconSink를 수신한 노드들은 CH와의 거리가 1홉(hop) 떨어진 노드들로써 클러스터에 참여하는 CM이 된다(그림 5-①).

스타형 네트워크의 에지들이 이 과정에서 형성된다. BeaconSink를 수신한 CM들은 타이머가 트리거 될 때 마다 BeaconNode 메시지를 전송하여 자신이 클러스터 내부에 존재함을 CH와 다른 CM들에게 알린다. 두 가지 종류의 BeaconSink와 BeaconNode 메시지를 수신한 모두 수신한 센서 Mote들은 Beacon 메시지로부터 메시지를 송신한 센서 Mote의 고유 ID와 LQI(Link Quality Information), RSSI(Received Signal Strength Indicator)값, CH와의 거리(hop)을 얻어와 라우팅 테이블에 저장한다(그림 5-②). 이 값들은 다음 단계에서 릴레이 링크를 구성하는데 필요한 자료로 사용된다.

두 번째 단계에서 CH는 CM들이 보낸 BeaconNode 메시지로 부터 얻어낸 노드들의 정보로 라우팅 테이블을 구성하고, 라우팅 테이블에서 릴레이 포인트로 사용된 적이 없으면서 LQI와 RSSI값이 가장 나쁜 센서 Mote를 찾아 JoinRequest 메시지를 전송한다(그림 5-③).

JoinRequest 메시지는 Beacon 메시지와 동일한 구조를 가지며, Beacon 메시지의 FrameControl 값을 사용하여 그 종류를 구별한다. LQI와 RSSI값이 가장 나쁜 센서 Mote는 CH의 주파수 반경 내에서 무선 통신 비용이 가장 많이 드는 센서 Mote로 앞으로 테일(Tail) 노드라고 한다. (그림 6)은 전체 시스템 구현의 예를 보여주기 위한 일부 코드로서 CH가 TPC에 참여 가능한 노드들 중에서 테일 노드를 선택하는 부분에 대한 설명을 돕기 위한 실제 코드와 각 기능에 대한 설명이다. CM이 수신한 Beacon 메시지의 FrameControl의 값이 Join-Request이면 CH로부터 네트워크에 참여할 것

```

task void SelectTail()
{
    uint8_t i;
    uint8_t NextAddress_index = 0xFF;
    uint8_t MinLQI=0xFF;
    char MinRSSI = 127;
    // JoinRequest는 비콘 메시지를 이용해서 보낸다.
    BeaconFrame BF;
    BF.SrcAddr = TOS_NODE_ID;
    // 릴레이 포인트가 아니면 통신 비용이
    // 가장 큰 노드(테일 노드)를 선택한다.
    for ( i = 0 ; i < NumNeighborTable ; i++ ){
        if(NTableList[i].ChildAddress == UnkownAddress
            && NTableList[i].Naddr != SinkAddress
            && NTableList[i].Naddr != UnkownAddress) {
            if(MinLQI >= NTableList[i].LQI) {
                if(MinRSSI >= NTableList[i].RSSI)
                {
                    MinLQI = NTableList[i].LQI;
                    MinRSSI = NTableList[i].RSSI;
                    NextAddress_index = i;
                }
            }
        }
    }
    // JoinRequest Message를 만들어 전송한다.
    if (NextAddress_index != 0xFF)
    {
        BF.FrameControl = BeaconRequest;
        BF.HopNum_from_Sink = 1;
        memcpy(call SendBeacon.getPayload(&BFframe),
            &BF, sizeof(BeaconFrame));
        if (call SendBeacon.send
            (NTableList[NextAddress_index].Naddr,
            &BFframe, sizeof(BeaconFrame)) == SUCCESS)
            TX_Type = BF.FrameControl;
    }
}
    
```

(그림 6) 릴레이 포인트로 사용된 적이 없으면서 CH와의 통신 비용이 가장 큰 노드를 선택하는 SelectTail 함수 (그림 5-③의 구현)

(Fig. 6) SelectTail function code to select a sensor that has not been selected as a relay point and to initiate a relay path construction (refer to Figure 5-③)

을 요청받은 것이다. 이 CM은 라우팅 테이블에서 릴레이 포인트로 사용된 적이 없으면서 LQI와 RSSI값이 가장 좋은 노드를 데이터 릴레이 포인트로 네트워크에 참여하도록 JoinRequest 메시지를 전송한다(그림 5-④).

LQI와 RSSI값이 가장 좋은 센서 Mote는 무선 통신에 필요한 비용이 가장 적게 드는 센서 Mote다. 만약에 데이터 릴레이 포인트로 사용된 CM과 CH의 거리(hop)가 미리 정의된 MaxHop이상이 되거나, CH를 데이터 릴레이 포인트로 선택하면, 해당 CM은 더 이상 릴레이 포인트를 찾지 않고 CH로 데이터를 전송한다. 이는 데이터 처리과정에서 발생할 수 있는 오버헤드를 고려한 것으로, CH가 정하는 최대 릴레이 수와, 각각의 CM에서 최대 릴레이수와 SEQ ID에 대하여 이루어지는 modulo 연산을 간략화한 것이다. CH는 클러스터에 속한 모든 CM이 테일 노드 또는 데이터 릴레이 포인트로 사용될 때 까지 두 번째 단계의 처리 과정을 반복하여 첫째 단계에서 구성된 스타형 네트워크를 체인 또는 불완전한 휠의 형태로 재구성한다.

앞서의 두 단계 과정을 통해서 구축된 클러스터의 테일 노드는 매 3000ms 마다 주변의 온도, 습도, 조도, 전력값을

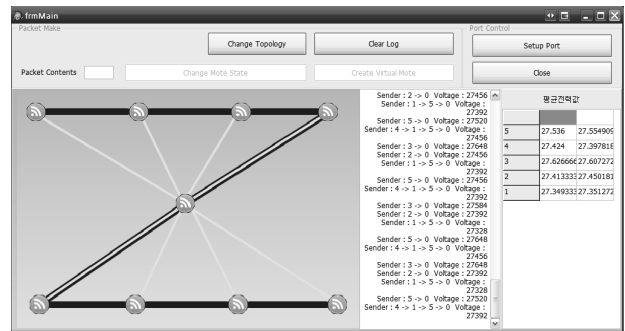
센싱해 데이터를 얻는다. 테일 노드는 얻어온 정보를 데이터 패킷의 UpperData1에 저장하고, DstAddr에 CH의 센서 Mote ID를, SrcAddr에 자신의 센서 Mote ID를 입력해 데이터 릴레이 포인트에 전달한다. 데이터 패킷을 수신한 센서 Mote가 CH이면 호스트 PC에 RS232C Serial Port를 통해서 데이터 패킷을 전송한다. 데이터 패킷을 수신한 노드가 CM이면 데이터 패킷이 거쳐 온 데이터 릴레이 포인트 정보와 각 릴레이 포인트의 센서들이 센싱한 데이터를 데이터 패킷에 저장해 자신의 데이터 릴레이 포인트에 전송하는 과정을 반복하여 CH까지 데이터를 포워딩한다. 이때 실제로 무선으로 전송되는 TinyOS의 message_t 구조체의 크기는 헤더와 테일을 포함하여 총 118바이트이다. 테일 노드가 아닌 CM은 데이터 패킷을 수신하였을 때만 자신이 센싱한 데이터를 포워딩 하고, 스스로 CH를 향해 데이터를 전송하진 않음에 유의한다.

4. 실험 및 성능평가

4.1 성능 평가를 위한 실험 환경

앞서 구현한 TPCAppM 프로그램으로 TPC의 성능을 평가하기 위하여 CH로부터 선택된 테일 노드는 매 3000ms마다 ZigbeX의 온/습도 센서, 적외선 센서, 전압 회로로부터 데이터를 얻어와 헤더를 포함하여 매번 118Bytes 크기의 패킷을 전송하도록 하였고, 데이터 릴레이 포인트가 되는 노드의 RSSI 값에 따라 전송에 필요한 전력 소모량을 17.4mA 부터 15.2mA, 11.2mA, 8.5mA까지 4단계로 조절하였다. 또한 TPC로 구성된 클러스터에서 릴레이 제어 메시지(MAXHOP 상수)는 4로 고정하여, 테일 노드와 CH 사이의 거리가 최대 4홉이 되도록 하였다. CH와 CM이 반복해서 전송하는 BeaconSink와 BeaconNode 메시지의 간격은 6000ms로 설정하였고, OCS 클러스터에서는 노드들의 전송 파워를 최대(17.4mA) 고정하여 TPC 클러스터와 실제 전력 소모량을 비교하였다. OCS 기반 클러스터 구성을 위해서 TCP_RoutingC의 릴레이 제어 메시지(MAXHOP 상수)를 1로 조절해 CM이 테일 노드가 되도록 하였다.

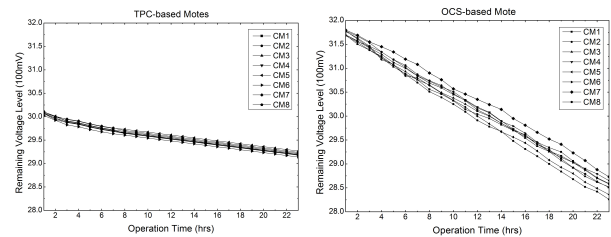
CH가 시리얼 통신으로 호스트 PC에 전달하는 센서 Mote들의 센싱 데이터와 구성된 네트워크를 시각적으로 표현하기 위한 호스트 어플리케이션(그림 7 참조)은 Borland RAD Studio 2009로 구현하였고, 이는 윈도우 XP 이상의 환경에서 동작한다. 중앙에 존재하는 센서 Mote가 CH이며, 각각의 노드들 사이에 얇은 실선은 스타형 네트워크를 이루는 경로이고, 굵은 실선은 TPC를 이루는 경로이다.



(그림 7) 호스트측 컨트롤 어플리케이션 캡처이미지
(Fig. 7) Snapshot of Host Control Application

4.2 전력 소모량 측정

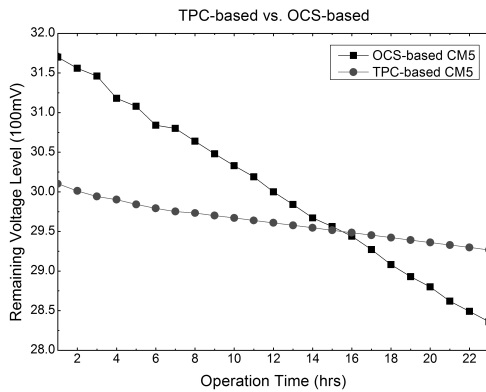
OCS 클러스터와 TPC 클러스터의 전력소모량을 비교하기 위하여, 각 센서 Mote들을 동일한 장소에 배치하고 OCS와 TPC로 클러스터를 구성하여 각각 23시간 동안 수집된 정보를 CH로 전달하도록 하였다. 전력 소모량의 측정을 위하여 매 1시간마다 센서 Mote에 연결된 건전지의 전압을 전압계로 직접 검사하였다. 이때 $V=IR$ 이므로 전압은 잔여 전류량에 비례함에 유의한다. 서로 인접한 두 노드들 사이의 최소 거리는 3m, 최대 거리는 8m로 구성하였다. <표 2>는 TPC와 OCS 기반의 각 노드들의 23시간이 지난 후의 잔여전력량의 비교를 나타낸 것이고, (그림 8)의 좌, 우는 각각 TPC와 OCS로 구성한 클러스터의 초기 전압과 23시간 후의 전압레벨을 나타낸다. 그리고 (그림 9)는 TPC와 OCS로 구성된 클러스터에서 동일한 위치에 있던 5번 센서 Mote의 전력소모량을 직접 비교한 것이다. TPC와 OCS로 각각 23시간 동안 진행한 실험에서 TPC로 구성한 클러스터가 사용한 센서 Mote들의 평균 전력량은 0.09V, OCS 클러스터가 사용한 센서 Mote들의 평균 전력량은 0.341V로, TPC로 구성한 클러스터는 OCS로 구성한 클러스터 대비



(그림 8) TPC-based Mote (좌)와 OCS-based Mote (우) 초기전압과 23시간 동작후의 전압 레벨
(Fig. 8) Remaining Voltage level in TPC-based Mote (Left) and OCS-based Mote (Right) after 23-hour operation

<표 2> TPC와 OCS 클러스터의 잔여 전력량
<Table 2> Remaining Energy Level in both TPC-based and OCS-based Clusters

	Mote1	Mote2	Mote3	Mote4	Mote5	Mote6	Mote7	Mote8
TPC	96.95%	96.96%	96.92%	96.99%	97.12%	96.95%	97.01%	96.94%
OCS	89.23%	88.81%	89.20%	89.69%	89.17%	89.38%	89.36%	89.13%



(그림 9) TPC와 OCS-based 센서 Mote 5번의 전력 소모량 비교
(Fig. 9) Energy Consumption Comparison between TPC-based and OCS-based motes

26.57%의 전력량을 사용하는 높은 효율을 보임을 알 수 있다. (그림 9)에서 알 수 있듯이 TPC로 구성한 클러스터는 OCS로 구성한 클러스터에 비하여 적은 전력을 소비하여, 네트워크의 생존시간을 높일 수 있음을 알 수 있다.

5. 결 론

이 논문에서는 두 단계 클러스터링 (TPC) 기반의 리포팅 허용 지연시간 적응형 데이터 수집 방법을 소개하고 이를 구현하여 그 결과를 평가하였다. 이 논문의 목표는 실제 환경에서 센서 Mote들이 제안된 데이터 수집 방법을 기반으로 네트워크 생존시간을 극대화하는 에너지 효율형 정도를 평가하는 것이다. 이 목표를 달성하기 위한 TPC 클러스터링은 기존의 클러스터 헤드와 멤버간 1홉을 사용하여 구성한 네트워크, 즉 OCS, 클러스터에 비하여 동일 동작 시간에 약 26.57%의 전력만을 사용하는 높은 효율을 보였다. 이는 CM들이 센싱 데이터를 수집하고 CH를 향해 포워딩 하는 동안 클러스터 유지에 필요한 에너지를 절약해 생존시간을 극대화 할 수 있음을 의미한다. 특히 밀도가 높은 클러스터에서 CM들은 무선 통신에 필요한 비용이 적게 드는 릴레이 포인트를 많이 선택할 수 있기 때문에, 최대 릴레이 포인트 개수가 커질수록 네트워크의 생존시간이 늘어날 것으로 보인다.

향후 CH에서 사용자의 데이터 보고 시간 요구사항에 따라 최대 릴레이수를 계산하는 방법을 개발하여 제안된 TPC 방식에 포함시키는 연구와 함께 CM이 센싱한 데이터 자체의 허용 지연시간을 평가하여 동적으로 릴레이수를 계산하는 연구를 진행할 계획이다.

참 고 문 헌

[1] R. Virrankoski and A. Savvides. (2005) TACS: Topology Adaptive Spatial Clustering for Sensor Networks. In Proceedings of IEEE Int'l Conference on Mobile Ad hoc and

Sensor Systems (MASS), pp.1-10.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin (2000), Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In Proceedings of ACM MOBICOM, pp.56-67.
 [3] W. Choi, et al., Angle-based Dynamic Path Construction for Route Load Balancing in Wireless Sensor Networks. In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), pp.2474-2479, 2004.
 [4] W. Choi, et al., Cluster-based Delay-adaptive Sensor Scheduling for Energy-saving in Wireless Sensor Networks, Journal of the Korea Society for Simulation, Vol.19, No.3, pp.47-59, 2009.
 [5] (주)한백전자 기술연구소, ZigbeX를 이용한 유비쿼터스 센서 네트워크 시스템(제3판), 2008.
 [6] Hanbak Electroncis, <http://www.hanback.co.kr>



손 태 식

e-mail : ts.shon@samsung.com

2000년 아주대학교 정보 및 컴퓨터공학부 (학사)

2002년 아주대학교 정보통신공학과(공학 석사)

2005년 고려대학교 정보보호학과(공학박사)

2004년~2005년 2월 Research Scholar, Univ. of Minnesota

2005년~현 재 삼성전자 Digital Media & Communications 연구소 책임연구원

관심분야: Wireless/Mobile Network Security, WSN/WPAN, Anomaly Detection/Machine Learning



최 호 현

e-mail : hchoi@inhac.ac.kr

1994년 서강대학교 컴퓨터공학과(학사)

1996년 서강대학교 컴퓨터공학과(석사)

2005년 서강대학교 컴퓨터공학과(박사)

2005년~2009년 삼성전자 통신연구소 책임연구원

2009년~현 재 인하공업전문대학 컴퓨터정보과 조교수

관심분야: USN, Wireless Mesh Network, Mobile Ad hoc Network, M2M, Social Network