

무선 센서 네트워크에서 에너지 효율적이고 신뢰성이 높은 데이터 병합을 위한 적응적 타임아웃 스케줄링 기법

정회원 백 장 운*, 남 영 진**°, 종신회원 서 대 화*

Adaptive Timeout Scheduling for Energy-Efficient, Reliable Data Aggregation in Wireless Sensor Networks

Jang Woon Baek*, Young Jin Nam**° *Regular Members*,
Dae-Wha Seo* *Lifelong Member*

요 약

무선 센서 네트워크에서 데이터 병합을 수행하는 센서노드는 자식 노드의 메시지를 기다리는 시간인 타임아웃을 스케줄링한다. 본 논문에서는 데이터 병합시 소모되는 전력을 최소화하고 데이터 정확도를 높이기 위해, 센서 네트워크에서 수집되는 데이터 패턴에 따라 타임아웃을 적응적으로 조절하는 새로운 타임아웃 스케줄링 기법을 제안한다. 제안 타임아웃 스케줄링 기법은 자식노드로부터 수신한 데이터가 특별한 이벤트 없이 기존 데이터 변동률 범위 내에 있게 되면 타임아웃 값을 줄인다. 타임아웃을 작게 하면 메시지 대기시간이 짧아지고 센서 노드의 에너지 소모가 감소한다. 센싱 데이터가 기존 데이터 변동률을 벗어나면 중요 이벤트 발생 확률이 높으므로 보다 정확한 데이터 병합 값을 얻기 위해 타임아웃 값을 증가시킨다. 시뮬레이션 결과에서 제안 기법은 기존 캐스캐이딩 타임아웃 스케줄링 기법에 비해 에너지 소모와 데이터 정확도 측면에서 우수한 성능을 보여주었다.

Key Words : Wireless sensor networks, Data aggregation, Timeout scheduling

ABSTRACT

In wireless sensor networks, a sensor node with in-network aggregation adjusts the timeout which is a waiting time to receive messages from child nodes. This paper proposes a novel timeout scheduling scheme for data aggregation in wireless sensor networks, which adaptively configures its timeout according to changing data patterns in order to improve energy efficiency and data accuracy during data aggregation. The proposed scheme decreases the timeout when the temporal difference of collected data (data variation) from children is lower than a pre-defined threshold because there is no occurrence of critical events. Conversely, it increases the timeout when the data variation is higher than the pre-defined threshold in order to fulfill more accurate data aggregation. Extensive simulation reveals that the proposed scheme outperforms the cascading-based scheme in terms of energy consumption and data accuracy.

※ 본 연구는 2008년도 경북대학교 BK21사업과 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-C1090-0801-0045)

* 경북대학교 전자전기컴퓨터학부 모바일컴퓨팅 & 임베디드시스템 연구실 ({kutc, dwseo}@ee.knu.ac.kr),

** 대구대학교 컴퓨터·IT공학부 (yjinam@daegu.ac.kr)(° : 교신저자)

논문번호 : KICS2007-12-542, 접수일자 : 2007년 12월 6일, 최종논문접수일자 : 2008년 5월 14일

I. 서 론

MEMS, 무선통신, 전자공학의 기술발전으로 무선 센서 네트워크가 여러 응용분야에서 사용되고 있다^[1]. 센서 네트워크는 물리적인 현상을 감지하고 데이터를 처리하며 데이터 통신을 할 수 있는 많은 값싼 센서노드들로 구성된다. 센서 네트워크의 가장 중요한 문제는 센서노드의 제한된 배터리를 에너지 효율적으로 사용하여 노드의 수명을 최대한 늘리는 것이다^[2]. 센서노드에서 에너지는 데이터 처리에 비해 데이터 송수신에 상대적으로 훨씬 더 많이 소모된다^[3].

데이터 송수신으로 인한 에너지 소모를 줄이기 위해 인-네트웍(In-network) 데이터 병합 기법이 사용되고 있다^[4]. 인-네트웍 병합을 적용한 센서 네트워크는 주로 트리 기반의 토폴로지를 가지고, 각 센서 노드가 주기적으로 센싱 정보를 베이스 스테이션(BS)로 전송한다. 인-네트웍 병합 기능을 가지는 센서노드는 자식노드의 수집데이터와 자신의 수집 데이터를 병합하여 하나의 메시지만 부모 노드로 전송한다. 병합을 수행하기 전에 각 노드는 자식노드의 메시지를 수신해야 한다. 이때, 자식 노드로부터 메시지를 기다리는 시간을 타임아웃이라고 한다. 타임아웃이 클수록 자식 노드로부터 메시지를 받을 확률이 높아지므로 데이터 정확도는 높아지지만 전송 지연이 커지고, 에너지 소모량이 증가한다^[5]. 일반적으로 센서노드의 라디오 상태는 전송모드, 수신모드, 대기모드, 슬립모드로 나누어진다. 타임아웃동안 센서노드는 대기모드에서 동작하므로 슬립모드에 비하여 파워소모가 훨씬 크다^[5,6]. 데이터 병합시 타임아웃을 가능한 작게 유지해야한다.

센서 네트워크에서 수집되는 데이터는 다음과 같은 특성을 가진다. 첫째, 센서 네트워크에서 수집되는 데이터는 보통의 경우에 이전 주기에 수집된 데이터와 현 주기에 수집된 데이터의 간의 차이가 별로 없고 지리적으로 인접한 곳에 배치된 센서노드들에서 수집되는 데이터 값은 비슷하다. 둘째, 산불이나 지진과 같은 중요 이벤트가 발생할 경우에는 이전 주기에 수집된 데이터와 큰 차이를 가진다^[9].

본 논문에서는 센서 네트워크에서 수집되는 데이터의 특성을 고려하여 인-네트웍 데이터 병합의 타임아웃을 적응적으로 조절하는 새로운 타임아웃 스케줄링 기법을 제안한다. 제안 타임아웃 스케줄링 기법은 자식 노드로부터의 수신 데이터의 변동률이 기준 변동률보다 작은 경우에는 가능한 타임아웃을

작게 유지하여 에너지 소모를 줄인다. 하지만, 수신 데이터의 변동률이 기준 변동률 범위를 넘어서서 중요 이벤트가 발생했을 가능성이 높은 경우에는 보다 정확한 데이터 병합을 수행하기 위해 타임아웃을 증가시킨다.

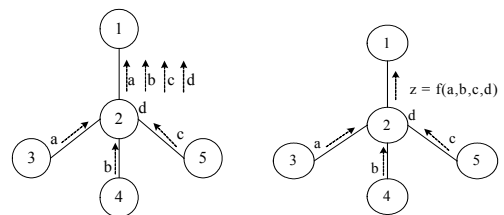
본 논문의 구성은 다음과 같다. II장에서는 인-네트웍 데이터 병합과 기존 타임아웃 스케줄링 기법에 대해 살펴본다. III장에서는 제안 타임아웃 스케줄링 기법에 대하여 자세히 설명한다. IV장에서는 시뮬레이션을 통해 성능을 분석하고, 마지막으로 V장에서는 결론을 맺고 향후 과제를 제시한다.

II. 관련 연구

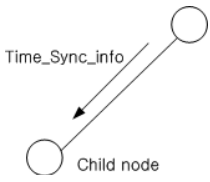
그림 1은 인-네트웍 데이터 병합의 예를 보여준다. 그림 1(가)에서 노드 2는 병합을 수행하지 않을 경우에는 4개의 메시지(a,b,c,d)를 전송해야 한다. 하지만, 그림 1(나)에서 노드 2는 자식노드로부터 받은 데이터 a, b, c와 자신이 수집한 데이터 d를 함수 f 로 병합한 z 값을 가지는 한 개의 메시지(z)만 전송하면 된다. 따라서 데이터 병합은 전송메시지를 수를 줄임으로서 에너지 소모를 줄일 수 있다^[4].

데이터 병합을 위한 동기화는 동기 메시지를 사용하여 부모노드 혹은 자식노드와의 메시지 송수신 동기를 맞추는 방법과 동기 메시지를 사용하지 않고 로컬에서 타임아웃을 계산하여 메시지 대기시간과 슬립시간을 조절하는 방법으로 나눌 수 있다.

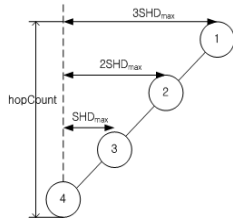
TAG^[7]와 Data Fusion^[8]에서는 인-네트웍 데이터 병합의 동기화를 위해 타이밍 정보를 포함한 동기화 메시지를 사용한다. TAG에서 각 노드는 자신이 메시지를 수신하기 위해 슬립모드에서 대기모드로 깨어나는 시각을 포함하는 타이밍 정보를 자식노드에게 알려준다(그림 2(가)). 자식노드는 그 시기에 맞추어 메시지를 전송한다. Data Fusion에서는 자식노드가 부모노드의 메시지를 받을 수 있도록 슬립모드에서 메시지 대기모드로 들어갈 시기를 알



(가) 인-네트웍 병합 없음 (나) 인-네트웍 병합
그림 1. 인-네트웍 데이터 병합 예



(가) TAG 동기화^[7]



(나) 캐스케이딩^[5]

그림 2. 기존 타임아웃 스케줄링 기법

려준다. 하지만, 이러한 동기 메시지를 사용하는 타임아웃 스케줄링 기법은 부가적인 메시지 송수신으로 인한 에너지 소모를 야기한다.

이러한 문제를 해결하기 위해 동기화 메시지를 사용하지 않는 캐스케이딩 타임아웃 스케줄링 기법이 제안되었다^[5]. 그림 2(나)와 같이 캐스케이딩 타임아웃 스케줄링 기법에서는 별도의 동기화 메시지의 사용 없이 각 노드의 홉수에 따라 자식노드보다 홉간지연(Single-Hop-Delay, SHD) 만큼 타임아웃을 증가시킴으로써 데이터 병합을 위한 동기화를 한다. 캐스케이딩 기법은 자식노드의 메시지 수신을 보장하기 위해 타임아웃을 홉간지연의 최대값(SHD_{max})을 사용하였다. 하지만, 실제 홉간지연은 패킷 처리, 스케줄링 및 채널 예약 등의 네트워크 환경에 따라 가변적이다^[3]. 보통의 경우, 실제 홉간지연은 SHD_{max} 값 보다 훨씬 작으므로 타임아웃을 항상 SHD_{max} 로 유지하는 것은 불필요한 에너지 소모를 야기한다. 또한, 센서 네트워크 트래픽이 증가할 경우에는 실제 홉간지연이 SHD_{max} 보다 큰 경우가 발생할 수도 있다. 데이터 병합에서 타임아웃 이후에 도착하는 메시지는 손실될 것으로 간주하므로, 중요한 이벤트 정보를 포함한 메시지가 손실될 수 있다.

본 논문에서는 각 센서 노드가 로컬에서 타임아웃을 스케줄링함으로써 TAG와 Data Fusion에서 동기 메시지 송수신으로 인한 에너지 소모를 없애고, 캐스케이딩 기법에서 고정된 홉간지연(SHD_{max})을 이용한 로컬 타임아웃 스케줄링으로 인한 발생하는 문제점을 해결하기 위한 적응적 타임아웃 스케줄링 기법을 제안한다.

III. 제안 알고리즘

본 논문에서는 센서 네트워크에서 수집되는 데이터의 패턴을 고려하여 인-네트워크 데이터 병합의 타

임아웃을 적응적으로 조절하는 새로운 타임아웃 스케줄링 기법을 제안한다.

센서 네트워크에서 수집되는 데이터의 패턴을 고려하여 본 논문에서는 데이터 변동률을 이용하여 중요이벤트 발생유무를 판단한다. 데이터 변동률(dv_i)은 이전 주기의 데이터($prev_i$)와 현 주기의 데이터($curr_i$)간의 시간적 변동률로 정의된다. 즉, 데이터 변동률(dv_i)은

$$dv_i = |(curr_i - prev_i)/curr_i| * 100(\%)$$

으로 계산된다. 자식 노드로부터 수신한 데이터 및 직접 센싱한 데이터들 중에서 데이터 변동률이 기준 변동률(DV) 보다 큰 경우, 중요 이벤트가 발생한 것으로 간주한다. 기준 변동률(DV)은 응용 애플리케이션에 따라 적정 값으로 설정할 수 있다.

중요 이벤트 발생의 여부를 판단하기 위해서는 충분한 샘플수를 가져야한다. 대부분의 자식노드의 메시지는 평균 홉간지연(SHD_{avg}) 내에 도착한다^[10]. 제안 기법은 평균 홉간지연(SHD_{avg})동안 수신한 메시지의 데이터 변동률을 검사하여 중요 이벤트의 발생여부를 판단한다.

제안 타임아웃 스케줄링 기법은 자식 노드로부터의 수신 데이터의 변동률이 기준 변동률보다 작은 경우에는 가능한 타임아웃을 작게 유지하여 에너지 소모를 줄인다. 하지만, 수신 데이터의 변동률이 기준 변동률 범위를 넘어서 중요 이벤트가 발생했을 가능성이 높은 경우에는 보다 정확한 데이터 병합을 수행하기 위해 타임아웃을 증가시킨다.

제안 타임아웃 스케줄링 기법의 동작은 그림 3과 같다. 기본적으로 제안 타임아웃 스케줄링 기법은 각 센서노드($node_i$) 상에서 동작한다. 부모노드($node_{par(i)}$)부터 쿼리를 받은 센서노드($node_i$)는 자신의 홉수($hopCount_i$), 타임아웃($Timeout_i$), 여유분(MRG_i)와 기준 데이터 변동률(DV)를 초기화한 후, 타임아웃 타이머(T_{cur})를 구동하고, 자식노드들로부터 메시지를 기다린다. 센서노드는 SHD_{avg} 동안 도착한 데이터의 변동률을 업데이트하면서 부분적인 병합을 수행한다. SHD_{avg} 동안 수신한 데이터의 변동률이 DV 보다 작으면 더 이상 자식노드의 메시지를 수신하지 않고 최종병합을 수행한다. 데이터 변동률이 DV 보다 크면 자식노드가 사용한 타임아웃 마진을 뺀 SHD_{max} 로 증가시킨다. 이때, 타임아웃 이내에 모든 자식노드들로부터 메시지를 수신하면 바로 최종 병합을 수행한다. SHD_{max} 값은

```

procedure Timeout_Schedule(Query from nodepar(i))
// running at nodei;
extract hopCountpar(i), SHDmax, SHDavg,
and DV from Query;
hopCounti = hopCountpar(i) + 1;
δ = (SHDmax - SHDavg);

while TRUE do;
Timeouti = SHDavg; dvi = MRGi = 0;
start TIMER with Tcur = 0;
repeat
receive data & MRGchild(i) from nodechild(i);
do partial data aggregation(curri);
update dvi;
if (all messages arrived) goto transmit;
until Tcur ≤ Timeouti;

//if dvi is low enough, stop receiving data
if (dvi ≤ DV), then goto transmit;

//an event occurred, set the timeout to SHDmax
//subtract the amount used by its child nodes
Timeouti = SHDmax - Max{MRGchild(i)};
repeat
receive data from nodechild(i);
do partial data aggregation(curri);
if (all messages arrived) goto transmit;
until Tcur ≤ Timeouti;

// unless data are received from all childs
// finally, use the timeout margin
MRGi = δ2; Timeouti = Timeouti + MRGi;
repeat
receive data from nodechild(i);
do partial data aggregation(curri);
if (all messages arrived) goto transmit;
until Tcur ≤ Timeouti;

transmit:
finally do aggregation with its sensing data;
previ = curri;
send curri & MRGi to nodepar(i);
sleep until next period;
end Timeout_Schedule
    
```

그림 3. 제안 타임아웃 스케줄링 기법

$$SHD_{max} = Max(sd) + Up(td + pd + qpd)$$

으로 산출 된다^[5]. 여기서, *sd*는 백오프 지연, *td*는 전송 지연, *pd*는 전파지연, 그리고 *qpd*는 큐잉 및 프로세싱 지연을 의미한다.

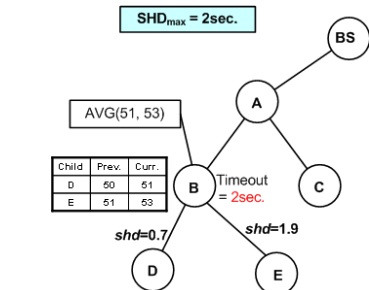
모든 메시지를 수신하지 못할 경우, 부모노드의 타임아웃 마진의 절반(δ2)을 자신의 타임아웃에 추가하여 자식노드의 메시지를 기다린다. 그리고 사용한 타임아웃의 양을 부모노드에게 알린다. 그러므로 자식노드에서 사용된 타임아웃 마진을 고려한 각 노드의 타임아웃(Timeout_i)는

$$Timeout_i = SHD_{max} - Max(MRG_{child(i)})$$

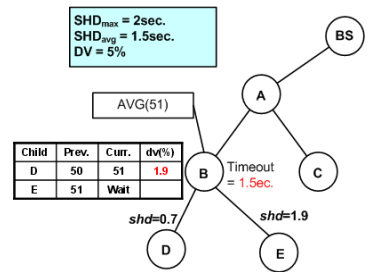
이다. 대부분의 자식노드의 메시지가 SHD_{avg} 내에 도착하므로 각 노드(node_i)의 타임아웃 마진(δ)은

SHD_{max}와 SHD_{avg}의 차로 초기화된다. 추가한 타임아웃이 만료되면 제안 기법은 더 이상 자식노드의 메시지를 받지 않고, 최종 병합을 수행하고 센서노드는 다음 주기까지 슬립한다.

그림 4은 데이터 변동률을 적용한 제안 기법과 기존 캐스케이딩 기법을 비교한 것이다. 그림 4(가)의 캐스케이딩 기법에서 노드 B의 타임아웃은 SHD_{max}(2sec)로 설정한다. 그림 4(가)에서 노드 B는 1.9초에 모든 자식노드의 메시지를 받아서 데이터를 병합하여 부모노드 A로 전송한다. 그림 4(나)의 제안 기법에서 노드 B는 타임아웃을 SHD_{avg}(1.5sec)로 설정한다. 그림 4(나)에서 노드 B는 1.5초 동안 수신한 데이터의 변동률이 기존 변동률보다 작으므로 더 이상 메시지를 수신하지 않고 노드 D의 수신 데이터와 노드 E의 이전 주기에 수신된 데이터를 병합하여 부모노드 A로 전송한다. 그 결과, 그림 4의 예제에서 제안 기법은 캐스케이딩 기법에 비해 20%(0.4초) 정도 메시지 대기 시간을 줄일 수 있다. 데이터 병합의 결과를 살펴보면, 캐스케이딩 타임아웃 스케줄링을 적용한 경우에 데이터 병합의 결과 AVG(51, 53)은 52이고, 제안 타임아웃 스케줄링을 적용한 경우에는 데이터 병합의 결과 AVG(51)은 51이다. 제안 타임아웃 스케줄링 기법의 데이터 병합 결과와 실제 수집데이터의 차이는 응용 애플리



(가) 캐스케이딩 타임아웃 스케줄링



(나) 제안 타임아웃 스케줄링

그림 4. 데이터 변동률을 이용하는 제안 타임아웃 스케줄링 예제

케이션이 정의한 기준 변동률 범위 내에 속한다. 그림 4의 예제를 통해, 데이터 변동률이 충분히 작을 때, 제안 타임아웃 스케줄링 기법은 기존 캐스케이딩 타임아웃 스케줄링 기법에 비해 데이터 병합의 정확도 측면에서 거의 저하 없이 메시지 대기시간을 상당히 줄임으로써 에너지 소모를 상당히 줄일 수 있음을 알 수 있다.

그림 5에서는 데이터 정확도를 높이기 위해 타임아웃 마진을 적용한 제안 기법과 기존 캐스케이딩 기법을 간단한 예제를 이용하여 비교한 것이다. 그림 5(가)의 캐스케이딩 기법에서 노드 B의 타임아웃은 $SHD_{max}(2sec)$ 로 설정한다. 그림 5(가)에서 노드 B는 SHD_{max} 동안 자식 노드 D의 메시지만 수신하고, 자식 노드 E의 메시지는 손실된 것으로 간주하고 데이터 병합을 한다. 그림 5(나)의 제안 기법에서 노드 B는 데이터 변동률이 기준 변동률 보다 크므로 타임아웃을 $SHD_{avg}(1.5sec)$ 에서 $SHD_{max}(2sec)$ 로 증가시킨다. 노드 B는 $SHD_{max}(2sec)$ 동안 자식 노드 E의 메시지를 수신하지 못했음을 확인하고 타임아웃 마진(0.25sec)을 추가노드 E의 메시지 수신 확률을 높인다. 그림 5(나)에서 노드 B는 2.25초의 타임아웃을 사용함으로써 자식노드 D와 E의 메시지를 모두 수신

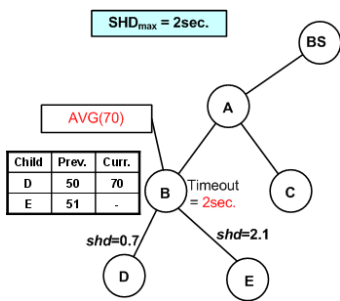
한다. 데이터 병합의 결과를 살펴보면, 캐스케이딩 타임아웃 스케줄링을 적용한 경우에 데이터 병합의 결과 $AVG(70)$ 은 70이고, 제안 타임아웃 스케줄링을 적용한 경우에는 데이터 병합의 결과 $AVG(70, 72)$ 은 75이다. 캐스케이딩 타임아웃 스케줄링 기법의 경우에는 실제 수집 데이터의 병합 결과와 큰 차이가 나지만, 제안 타임아웃 스케줄링 기법은 타임아웃 마진을 사용함으로써 실제 수집 데이터의 병합 값과 동일한 결과를 가질 수 있다. 그림 5의 예제를 통해, 중요 이벤트 발생으로 데이터 변동률이 기준 변동률 보다 크면, 제안 타임아웃 스케줄링 기법은 부모 노드의 타임아웃 마진을 사용함으로써 기존 캐스케이딩 타임아웃 스케줄링 기법에 비해 중요한 데이터를 보다 많이 수신할 수 있고 데이터 병합의 정확도를 높일 수 있음을 알 수 있다.

V. 성능 평가

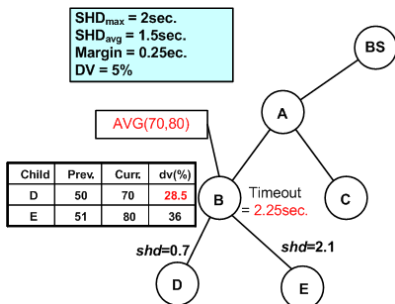
캐스케이딩 타임아웃 스케줄링 기법과 제안 타임아웃 스케줄링 기법의 성능을 비교하기 위하여 NS-2 네트워크 시뮬레이터상에 인-네트웍 데이터 병합을 구현하였다¹¹⁾.

5.1 시뮬레이션 환경

표 1은 실험에 사용된 파라미터를 보여준다. 시뮬레이션에서 100개의 센서 노드가 랜덤하게 $1000 * 1000$ 그리드에 배치되었고, 센서 노드의 전송 범위는 50 미터, 링크 대역폭은 115kbps로 각각 설정하였다⁵⁾. MAC 프로토콜은 무선 센서 네트워크의 특성을 고려한 저전력 MAC 프로토콜인 SMAC¹²⁾을 사용하였고, 송신모드의 파워 소모는 14.88mW, 수신모드의 파워 소모는 12.50mW, 대기모드의 파



(가) 캐스케이딩 타임아웃 스케줄링



(나) 제안 타임아웃 스케줄링

그림 5. 타임아웃 마진을 이용하는 제안 타임아웃 스케줄링 예제

표 1. 시뮬레이션 파라미터

파라미터	값
노드의 수(N)	100개
라디오 전송 범위	50m
링크 대역폭	115kbps
송신 전력	14.88mW
수신 전력	12.50mW
대기 전력	12.36mW
슬립 전력	0.016mW
패킷 사이즈	30bytes
데이터 수집 주기	1sec ⁵⁾
중요 이벤트 발생 횟수	10, 20, 50, 100

워 소모는 12.36mW, 그리고 슬립모드의 파워소모는 0.016mW로 설정하였다^[5]. 패킷의 사이즈는 30 바이트로 정하였다. 센서 노드에서 수집된 데이터는 매 초마다 주기적으로 BS로 전송된다. 중요 이벤트가 임의의 지역에서 발생하도록 하고 발생 횟수를 조절하였다. 데이터 수집으로 인한 네트워크 트래픽 이외에, SHD_{max} 보다 큰 휴간지연이 발생하도록 무선 센서 네트워크의 랜덤 지역에 과부하 트래픽을 주입하였다. 시뮬레이션은 1000초 후에 종료되도록 하였다. 식 (1)에 의해 산출되는 SHD_{max} 값은 [5]를 참조하여 0.03 초로 정하였고, SHD_{avg} 값은 반복된 실험을 통하여 0.02 초로 설정하였다. 시뮬레이션에서 베이스 스테이션은 전체 무선 센서 네트워크로 쿼리를 브로드캐스트 한다. 쿼리를 받은 센서 노드는 그림 5에서 기술한 것과 같이 타임아웃, 마진, 기준 변동률을 초기화 한다. 성능 평가를 위해 캐스캐이딩 타임아웃 스케줄링 기법과 기준 데이터 변동률이 0%, 5%일 때의 제안 타임아웃 스케줄링 기법을 비교 분석하였다. 시뮬레이션에서 데이터 변동률은 0%보다는 크고 정상시에는 5%보다 작으며 중요이벤트 발생시 5%보다 크도록 모델링 하였다.

5.2 평균 소모 에너지 비교

센서 노드의 소모 에너지는 센서노드가 송수신하는 메시지 수에 비례한다. 센서 노드의 에너지 소모는 슬립모드에 비해 대기/수신 모드에서 파워소모가 훨씬 크다. 그림 6은 캐스캐이딩 기법과 제안 기법의 평균 에너지 소모량을 비교한 것이다. 제안 기법(DV=0%)의 평균 소모 에너지는 캐스캐이딩 기법의 평균 소모 에너지와 유사하다. 매 주기마다 수집되는 데이터 값이 동일한 경우는 거의 없으므로, 제안 기법(DV=0%)의 실제 타임아웃은 SHD_{avg} 가 아니라

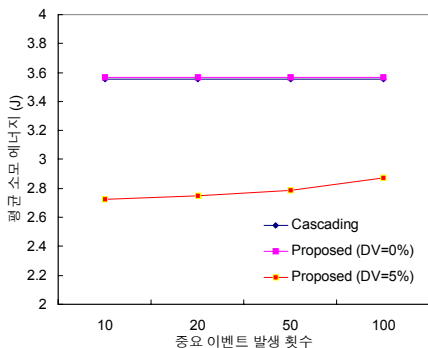


그림 6. 타임아웃 스케줄링 기법에 따른 평균 소모 에너지(J) 비교: 캐스캐이딩 타임아웃 스케줄링 기법, 제안 기법(DV=0%), 제안 기법(DV=5%)

SHD_{max} 로 설정되기 때문에 캐스캐이딩 타임아웃 스케줄링 기법과 유사한 결과를 가진다. 제안 기법(DV=5%)의 평균 소모 에너지는 캐스캐이딩 기법에 비하여 23% 정도 에너지 소모가 작음을 볼 수 있다. 시뮬레이션 동안 중요 이벤트가 드물게 발생하기 때문에, 데이터 변동률을 이용하는 제안 기법(DV=5%) 사용하는 센서노드는 대부분의 경우에 SHD_{avg} 동안만 메시지를 대기한다. 하지만, 캐스캐이딩 타임아웃 스케줄링 기법을 사용하는 센서 노드는 거의 SHD_{max} 동안 메시지 수신을 위한 대기모드로 동작한다. 그 결과, 제안 기법(DV=5%)은 캐스캐이딩 타임아웃 스케줄링 기법에 비해 대기모드에서 소모되는 에너지가 ($SHD_{max}-SHD_{avg}$)에 비례하여 더 작고, 수신 메시지의 수도 줄기 때문에 메시지 수신으로 인한 에너지 소모를 감소시킨다. 그림 6에서, 제안 기법(DV=5%)은 중요 이벤트의 발생 횟수가 증가할수록 에너지 소모량이 증가한다. 중요이벤트가 발생하면 제안 기법은 타임아웃을 SHD_{max} 으로 증가시키고 필요한 경우에는 부모노드의 타임아웃 마진을 추가하여 자식노드의 메시지를 대기하기 때문에, 중요이벤트가 자주 발생할수록 에너지 소모량이 증가한다. 하지만, 실제 센서 네트워크 환경에서 중요이벤트는 자주 발생하지 않기 때문에 제안 알고리즘은 캐스캐이딩 기법에 비행 훨씬 에너지 효율적일 것으로 예상된다.

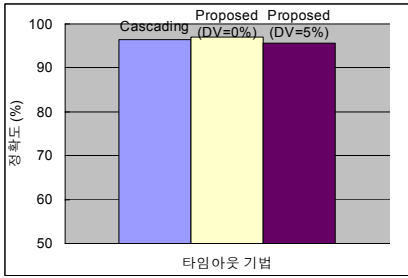
5.3 데이터 정확도 비교

무선 센서 네트워크에서 채널 충돌이나 프로세싱 지연으로 인한 메시지 손실이나 지연은 데이터 병합의 정확도에 영향을 준다^[13]. 특히, 센서 네트워크의 데이터 병합의 경우에는 타임아웃 이후에 도착하는 메시지는 손실된 것으로 간주한다. 본 논문에서 데이터 정확도는 실제 모든 센서노드에서 수집된 데이터의 병합값(X)과 인-네트워 데이터 병합을 통해 전송된 데이터 값(X')의 유사한 정도를 정확도 정의하고,

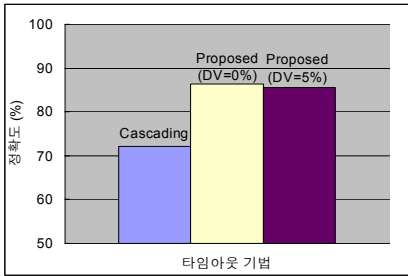
$$Accuracy = \sum_i^N \left(1 - \left| \frac{X_i - X'_i}{X_i} \right| \right) / N \times 100\%$$

으로 계산한다.

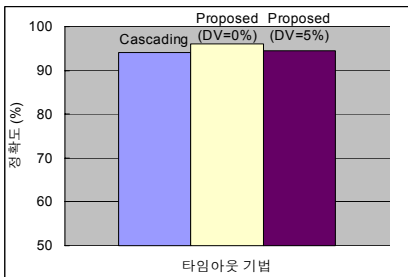
그림 6은 중요 이벤트 발생 횟수를 100으로 했을 때, 보통의 경우와 중요 이벤트 발생의 경우에 대하여 각 타임아웃 기법에 따른 데이터의 정확도를 비교한 것이다. 중요 이벤트가 발생하지 않은 보통 이벤트의 경우, 캐스캐이딩 기법의 타임아웃은 SHD_{max} 이고 제안 기법의 타임아웃은 데이터 변동률



(가) 보통 이벤트(non-critical events)



(나) 중요 이벤트(critical events)



(다) 전체 이벤트

그림 7. 타입아웃 스케줄링 기법에 따른 상대 왜곡 비교: 캐스케이딩 타입아웃 스케줄링 기법, 제안 기법(DV=0%), 제안 기법(DV=5%)

이 기준 변동률 범위내의 속하므로 SHD_{avg} 으로 동작한다. 그림 7(가)에서 제안 기법의 타이아웃이 캐스케이딩에 비해 작지만 정확도 측면에서 유사한 결과를 가짐을 볼 수 있다. 중요 이벤트가 발생하지 않은 보통의 경우에는 각 센서노드에서 수집되는 데이터는 이전 데이터 값과 유사하고, 인접한 센서 노드들의 수집 데이터 값이 비슷하기 때문에 정확도 측면에서 성능의 저하가 거의 없다.

그림 7(나)는 중요 이벤트 발생시 각 타이아웃 스케줄링 기법이 정확도를 비교한 것이다. 제안 기법(5%)가 캐스케이딩 기법에 비해 정확도 측면에서 13%정도 성능이 우수함을 알 수 있다. 중요 이벤트

가 발생한 경우, 캐스케이딩 기법의 타이아웃은 초기에 설정한 SHD_{max} 로 항상 일정하게 유지되고, 제안 기법의 타이아웃은 SHD_{max} 내에 모든 자식노드로부터 메시지를 받지 못할 때, 상위노드의 여유분을 추가 타이아웃으로 사용할 수 있다. 센서 네트워크 내에 트래픽 증가시 채널 충돌로 인해 홉간 지연이 길어질 경우, 캐스케이딩 기법은 고정된 타이아웃을 가지기 때문에 지연으로 인한 데이터 손실을 막을 수 없다. 하지만 제안 기법은 타이아웃이 부족할 경우에 상위노드의 여유분을 추가로 사용하여 지연으로 인해 메시지 손실을 줄일 수 있다. 중요 이벤트 발생시에는 각 센서 노드에서 수집되는 데이터 값의 차가 크기 때문에, 가능한 많은 자식 노드의 메시지를 수신해야 정확한 병합 값을 얻을 수 있다.

그림 7(다)는 전체 이벤트에 대한 각 타입아웃 기법의 데이터 정확도는 유사한 결과를 가짐을 볼 수 있다. 시뮬레이션 동안 발생하는 대부분의 이벤트는 보통 이벤트이기 때문에 전체 이벤트에 대한 데이터 정확도는 보통 이벤트의 정확도와 유사하다.

VI. 결론 및 향후 연구과제

본 논문에서는 센서 네트워크에서 데이터 병합을 수행할 때, 데이터 정확도를 유지하면서 에너지를 효율적으로 사용하는 새로운 타입아웃 기법을 제안 하였다. 제안 타입아웃 스케줄링 기법은 자식 노드로부터 수신한 데이터의 변동률이 기준 변동률보다 작은 경우에는 가능한 타이아웃을 SHD_{avg} 로 유지하면서 에너지 소모를 줄인다. 하지만, 수신 데이터의 변동률이 기준 변동률 범위를 넘어서서 중요 이벤트가 발생했을 가능성이 높은 경우에는 보다 정확한 데이터 병합을 수행하기 위해 타이아웃을 SHD_{max} 으로 증가시키고, 필요한 경우에는 부모노드의 타이아웃 마진(MRG)을 추가로 사용한다. 시뮬레이션을 통하여 제안한 기법을 실험하였고 기존 캐스케이딩 타입아웃 스케줄링 기법과 비교하였다. 실험 결과에서 제안 기법(DV=5%)는 캐스케이딩 타입아웃 스케줄링 기법에 비해 평균 소모 에너지 측면에서 23.3%, 중요 이벤트 발생시 데이터 정확도 측면에서 13% 정도의 성능이 더 우수함을 보여 주었다. 향후 과제에서는 센서 노드의 고장에 대응할 수 있도록 제안한 타입아웃 스케줄링 기법을 향상시키기 위한 연구를 수행할 계획이다.

참 고 문 헌

[1] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler, "The Emergence of Networking Abstractions and Techniques in TinyOS," *USENIX/ACM Symposium on Networked Systems Design and Implementation*, 2004.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, 38, 2002.

[3] Y. Yao, and J. Gehrke, "Query Processing for Sensor Networks," *Conference on Innovative Data Systems Research*, 2003.

[4] S. Madden, R. Szewczyk, M. Franklin and D. Culler, "Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks," *IEEE Workshop on Mobile Computing and Systems Applications*, 2002.

[5] I. Solis and K. Obraczka, "In-Network Aggregation Trade-offs for Data Collection in Wireless Sensor Networks," *International Journal of Sensor Networks*, 1(2), 2006.

[6] S. Motegi, K. Yoshihara, and H. Horiuchi, "DAG based In-Network Aggregation for Sensor Network Monitoring," *Symposium on Applications and the Internet*, 2006.

[7] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a Tiny Aggregation Services for Ad-Hoc Sensor Networks," *Symposium on Operating Systems Design and Implementation*, Dec. 2002.

[8] W. Yuan, S. Krishnamurthy, and S. Tripathi, "Synchronization of Multiple Levels of Data Fusion in Wireless Sensor Networks," *IEEE Global Communications Conference*, 2003.

[9] M. Sharaf, J. Beaver, A. Labrinidis and P. Chrysanthis, "Balancing Energy Efficiency and Quality of Aggregate Data in Sensor Networks," *The VLDB Journal*, 2004.

[10] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot, "Analysis of Measured Single-Hop Delay from an Operational Backbone Network," *IEEE INFOCOM*, June, 2002.

[11] VINT, "The Network Simulator NS-2," <http://www.isi.edu/nsnam>, Nov. 2005.

[12] W. Ye, J. Heidemann and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *IEEE INFOCOM*, 2002.

[13] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks", *IEEE INFOCOM*, May, 2007.

백 장 운 (Jang Woon Baek)

정회원



2002년 2월 경북대학교 전자전기공학부 졸업
 2004년 2월 경북대학교 전자공학과 석사
 2004년 3월~현재 경북대학교 전자공학과 박사과정
 <관심분야> 센서네트워크, 모바일 컴퓨팅, 임베디드 시스템

남 영 진 (Young Jin Nam)

정회원



1992년 2월 경북대학교 전자공학과 졸업
 1994년 2월 포항공과대학교 전기공학과 석사
 2004년 3월 포항공과대학교 컴퓨터공학과 박사
 1994년~1998년 한국전자통신연구원 컴퓨터연구단

2004년~현재 대구대학교 컴퓨터IT공학부 조교수

<관심분야> 네트워크 스토리지, 임베디드 S/W, 무선네트워크

서 대 화 (Dae-Wha Seo)

중신회원



1981년 2월 경북대학교 전자공학과 졸업
 1983년 2월 한국과학기술원 전산학과 석사
 1993년 2월 한국과학기술원 전산학과 박사
 1983년~1995년 한국전자통신연구원 시스템S/W연구실

2004년~현재 경북대학교 임베디드S/W센터장
 1995년~현재 경북대학교 전자전기컴퓨터학부 교수

<관심분야> 임베디드 S/W, 병렬처리, 분산운영체제