

센서네트워크에서의 효율적인 데이터 전송을 위한 적응적 혼잡 제어

이 좌 형[†] · 김 동 국^{**} · 정 인 범^{***}

요 약

무선 센서 네트워크에서의 혼잡은 데이터 손실 비율을 증가시키고, 전송 지연이 길어지는 문제점을 야기한다. 기존의 무선 센서 네트워크를 위한 혼잡 제어 방법들은 혼잡을 판단하고, 혼잡에 연관된 센서노드들을 선택하여 샘플링 주기를 조절함으로써 전송량을 줄이는 방법을 사용한다. 그러나 샘플링 주기 조절 기법은 시간적 데이터 손실에 민감한 응용에 사용하기 어려운 문제점이 있다. 본 연구에서는 무선 센서 네트워크에서의 혼잡을 해결하는 새로운 혼잡 제어 기법인 ACT를 제안한다. 제안하는 ACT기법에서는 큐 감시를 통해 다중 큐 임계값을 사용하여 네트워크의 상태를 판단한다. ACT는 혼잡 발생시 패킷 전송 간격을 조절하는 적응적 흐름 제어 기법을 통해 네트워크의 효율성을 높이며 패킷 판단 기법을 통하여 노드간의 공정성을 보장한다. 또한 압축을 이용하여 센서노드 간 전송량을 조절하도록 하여 이용가능한 데이터양을 증가시킨다. 실험을 통하여 제안된 ACT기법이 기존 프로토콜에 비해 네트워크의 패킷 전송 효율성을 향상시키며 센서노드들에게 공정성있는 데이터 전송을 제공함을 보인다.

키워드 : 센서네트워크, 혼잡, 혼잡탐지, 혼잡회피, 흐름제어, 압축

Adaptive Congestion Control for Effective Data Transmission in Wireless Sensor Networks

Joahyoung Lee[†] · Dongguq Gim^{**} · Inbum Jung^{***}

ABSTRACT

The congestion in wireless sensor network increases the ratio of data loss and causes the delay of data. The existing congestion protocols for wireless sensor network reduces the amount of transmission by control the sampling frequency of the sensor nodes related to the congestion when the congestion has occurred and was detected. However, the control method of sampling frequency is not applicable on the situation which is sensitive to the temporal data loss. In the paper, we propose a new congestion control, ACT - Adaptive Congestion conTrol. The ACT monitors the network traffic with the queue usage and detects the congestion based on the multi level threshold of queue usage. Given network congestion, the ACT increases the efficiency of network by adaptive flow control method which adjusts the frequency of packet transmission and guarantees the fairness of packet transmission between nodes. Furthermore, ACT increases the quality of data by using the variable compression method. Through experiment, we show that ACT increases the network efficiency and guarantees the fairness to sensor nodes compared with existing method.

Keywords : WSN, Congestion, Detection, Avoidance, Flow Control, Compression

1. 서 론

최근 MEMS와 마이크로프로세서 그리고 무선 통신 기술의 발전으로 무선 센서 네트워크 기술이 많이 보급되고 있

으며, 이에 대한 연구가 활발히 진행되고 있다. 센서 네트워크는 주변 환경에서 발생하는 환경 정보들을 수집하여 전달하는 것을 목적으로 적군 감시, 동식물의 모니터링, 구조물 안전 감시 등 많은 분야에 사용이 가능하다.

유선 네트워크에서는 혼잡을 제어하기 위한 많은 기법들이 연구되어왔다. 그러나 센서 네트워크에서의 센서노드는 한정된 전력을 이용하기 때문에 저전력으로 동작해야 하고, 무선 전송 방식으로 인한 시공간적 데이터 손실 등의 특징으로 인하여 다른 접근 기법이 요구된다^[1]. 혼잡으로 인해 손실된 데이터를 재전송하는 것은 전송 지연과 네트워크 혼

* 이 논문은 2008년도 정부(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2008-D00424(100901)).

† 준 회 원 : 강원대학교 컴퓨터정보통신공학과 박사과정

** 정 회 원 : (주)다이렉트 연구원

*** 종신회원 : 강원대학교 컴퓨터정보통신공학전공 교수
논문접수 : 2008년 4월 4일

수정일 : 1차 2008년 7월 4일, 2차 2008년 9월 22일

심사완료 : 2008년 10월 30일

잡을 가중시키고, 부가적인 전력을 소모하는 문제점을 초래한다^[2]. 이러한 문제를 해결하기 위해 센서 네트워크를 위한 혼잡 제어 기법들은 가장 큰 전력을 소모하는 데이터 전송 횟수를 줄임으로써 소모 전력을 줄여야 한다.

이에 따라 네트워크에 발생한 혼잡을 해결하기 위해 센서 노드에서 생성되는 데이터의 양을 줄이는 기법이 센서 네트워크에서 일반적으로 사용되고 있다^[5,6,7]. 이러한 기법들은 센서노드에서 혼잡이 탐지되면 혼잡과 연관된 센서노드들을 선별하고, 선별된 센서노드들의 샘플링 주기를 조절하여 전송량을 줄이는 방식을 사용한다. 그러나 이는 데이터의 정확한 분석을 위해 시간당 이용 가능한 데이터의 양을 필요로 하는 응용에서는 문제가 될 수 있다.

본 논문에서는 센서 네트워크에서의 혼잡을 탐지하고, 혼잡을 해결하는 새로운 혼잡 제어 기법인 ACT(Adaptive Congestion control)를 제안한다. 제안하는 ACT기법은 혼잡 탐지 기법과 혼잡 회피 기법으로 구성된다. 첫 번째 혼잡 탐지 기법은 네트워크의 혼잡을 감시하는 단계로 센서노드의 큐 상태에 따라 혼잡 상태와 비혼잡 상태를 구분하는 다중 큐 임계값 알고리즘을 사용한다. 이 알고리즘에 따라 자신의 상태를 판단하고, 자신의 상태에 따른 패킷 전송량을 설정한다^[11]. 두 번째 혼잡 회피 기법은 흐름 제어, 압축, 그리고 전달 종류 판별 세 가지로 구성된다. 첫째 흐름 제어 기법은 부모노드의 상태에 따라 자식노드의 데이터 전송 주기를 조절한다. 둘째 압축은 부모노드의 상태에 따라 압축률을 조절하여 전송량을 조절한다. 셋째 패킷 판단 기법은 균등한 전송 기회를 보장하기 위하여 자신의 생성 데이터의 양을 조절한다.

본 연구의 구성은 다음과 같다. 2장에서는 무선 센서 네트워크에서 사용하는 혼잡 제어 기법들에 대한 관련연구들을 살펴본다. 3장에서는 제안하는 혼잡 제어기법을 설명하고, 4장에서는 구현된 혼잡 제어 기법의 성능을 측정하고 결과를 분석한다. 5장에서는 본 연구의 결론을 맺는다.

2. 관련 연구

네트워크상에 혼잡이 발생하면 무선 통신상의 경쟁으로 인한 패킷 충돌이나 큐 오버플로우 등으로 인하여 데이터 손실이 발생할 수 있다. 일반적인 네트워크에서의 혼잡 제어 기법들은 무선 통신의 특징을 고려하지 않으므로 센서 네트워크에는 적합하지 않다^[11,12]. 이런 문제점을 해결하기 위하여 센서 네트워크에서 혼잡을 감지하고 혼잡을 해결하기 위한 혼잡을 제어하는 ESRT, CODA 등의 혼잡 제어 기법들이 연구되어 왔다^[5,6].

관심 지역에 대한 일정한 양의 데이터를 필요로 하는 경우에 사용하는 ESRT(Event-to-Sink Reliable Transport)은 샘플링 주기와 싱크의 요구량에 따른 수신량의 연관성을 이용하여, 최적으로 동작하는 영역을 도출한다. 시스템이 최적의 영역 안에서 동작하게 하기 위해서 싱크가 혼잡 제어를 담당하고, 폐쇄 반복 신호를 이용하여 센서노드들의 전송 주기를

조절한다. 만약 센서노드에 혼잡이 발생하면 싱크는 혼잡을 제어하기 위해 네트워크의 모든 센서노드들에게 전송량 조절 패킷을 브로드캐스팅한다. 그러나 ESRT의 브로드캐스팅 기법은 네트워크의 모든 센서노드에 전달되기 때문에 센서 네트워크내에 많은 전력이 소모되는 단점이 있다.

센서 네트워크에서의 혼잡을 감지하고 회피하기 위한 기법인 CODA(Congestion Detection and Avoidance)는 두 개의 메커니즘으로 구성되고, 세 단계를 통해 동작한다. 1 단계는 지역적 혼잡을 판단하는 단계로 센서노드들의 큐 사용량과 채널의 부하를 이용하여 최대 처리 임계값 이상이 되면 혼잡으로 판단한다. 2 단계는 센서노드 간의 흐름을 제어하는 단계로 부모의 패킷을 수신하여 부모가 혼잡 상태이면 일정시간 동안 데이터 전송을 중단한다. 네트워크의 모든 센서노드들 사이의 공평성이 보장하기 위해 싱크를 이용하여 센서노드들의 샘플링 주기를 조절하는 3 단계의 샘플링 주기 조절 메커니즘을 사용한다. 샘플링 조절 메커니즘은 혼잡이 발생할 경우 ACK를 이용하여 센서노드들의 샘플링 주기를 조절함으로써 넓은 범위의 혼잡을 제어한다. 이러한 CODA 같은 기존의 연구들은 샘플링 주기를 조절하여 전송량을 조절하기 때문에 샘플링 주기 조절로 인한 데이터의 손실이 발생하여 이용 가능한 데이터의 양이 낮아지는 단점이 있다.

CCF(Congestion control and Fairness)기법에서는 패킷 서비스 시간을 이용하여 가능한 서비스 시간과 혼잡을 감지한다^[16]. CCF에서는 서비스 레이트와 자식 노드수에 따라 전송 레이트를 조절함으로써 혼잡을 제어한다. CCF는 패킷 에러 레이트가 높거나 노드별 전송 데이터 양이 다를 경우에 효율성이 낮아질 수 있다. ARC (Adaptive Rate Control) 기법에서는 LIMD (Linear increase and Multiplicative decrease)알고리즘을 사용한다^[17]. 부모노드로 패킷을 전송한 자식노드가 부모노드가 자신의 패킷을 전송하는 것을 감지하면 자신의 전송 레이트를 서서히 증가시킨다. 반대로 부모노드가 자신의 패킷을 전송하지 못하면 혼잡으로 간주하여 자신의 전송 레이트를 급격히 감소시킨다. ARC에서는 혼잡감지나 혼잡통보를 하지 않기 때문에 제어메시지로 인한 오버헤드는 발생하지 않는다. 하지만 부모노드의 큐가 크면 패킷이 중계되는데 오랜 시간이 걸리기 때문에 혼잡에 빠르게 대응할 수 없다는 단점을 가진다. PCCP(Priority-based Congestion Control Protocol)에서는 노드별로 우선순위를 부여하며 혼잡발생시 우선순위에 따라 패킷을 전송한다^[18]. 혼잡이 심해져서 부모노드로 패킷을 전송할 수 없는 경우에는 새로운 부모노드를 찾아서 여러 경로로 패킷 전송을 시도한다^[19].

3. 적응적 혼잡 제어 기법: ACT

ACT는 센서 네트워크에서의 혼잡 제어 기법으로서 혼잡 탐지 메커니즘과 혼잡회피 메커니즘으로 구성된다. 첫째 혼잡 탐지 메커니즘에서는 각 센서노드들이 패킷을 생성할 때와 패킷을 수신할 때 혼잡을 감시한다. 각 센서노드들은 다

중 큐 임계값 알고리즘에 따라 혼잡 및 비혼잡 상태를 판단하고, 자신의 상태를 전송 패킷에 설정한다.

둘째 혼잡회피 메커니즘은 흐름 제어, 압축, 그리고 패킷 판단 기법으로 구성된다. 부모노드의 상태에 따라 자식노드는 흐름 제어 기법을 이용하여 전송 간격을 조절하여 부모노드로 전송하는 데이터의 양을 조절하고, 동시에 압축을 이용하여 자신이 생성하는 데이터의 양을 조절한다. 혼잡 감시와 동시에 센서노드는 단위 시간마다 자신의 상태를 판단하여 혼잡 상태일 경우 자신이 생성한 패킷과 자식노드들로부터 싱크로 증계하는 패킷의 비율을 확인하여 압축률을 조절한다.

3.1 혼잡 탐지

3.1.1 다중 큐 임계값

ACT는 다중 큐 임계값을 사용하여 혼잡의 정도를 구분하고, 이를 바탕으로 보다 효율적인 혼잡 제어를 수행한다. 다중 큐 임계값은 최대 큐 크기에 따라 최대 임계값(max_{th}), 중간 임계값(mid_{th}), 최소 임계값(min_{th})을 설정하여 큐의 상태를 네 개의 영역으로 구분하고, 현재 센서노드의 큐 사용량(cur)의 변화에 따라 단위 시간 i (EPOCH INTERVAL) 동안의 평균 큐 사용량(avg_i)을 $avg_i = 1/T \sum_{t=1}^T cur_t$ 에 따라 계산한다. 단위 시간(i)은 전송 패킷이 생성될 때까지의 시간을 나타내며, 단위 시간(i) 동안의 평균 큐 사용량(avg_i)에 따라 현재 센서노드의 상태가 결정된다.

<표 1>의 다중 큐 임계값 알고리즘에서와 같이 영역 1(STATE1)은 $avg < min_{th}$ 인 혼잡이 발생하지 않는 비혼잡 영역이고, 영역2(STATE2), 영역3(STATE3)은 $min_{th} \leq avg < max_{th}$ 인 영역으로, 약한 혼잡 상태이다. 그에 반해 영역 4(STATE4)는 $max_{th} \leq avg$ 인 심한 혼잡 영역을 나타낸다. Midth는 중

<표 1> 다중 큐 임계값 알고리즘

```

if(avg < minth) // STATE1
    if(state1 == FALSE or cnt1 > duration1) CS = '00'; cnt1 = 0;
    state1 = TRUE; cnt1++;
elseif(minth ≤ avg < midth) // STATE2
    if(state3 == TRUE) CS = '01';
    state1 = state3 = FALSE; state2 = TRUE; cnt1 = 0;
elseif(midth ≤ avg < maxth) // STATE3
    if(state2 == TRUE) CS = '10';
    state2 = state4 = FALSE; state3 = TRUE; cnt1 = 0;
elseif(maxth ≤ avg) // STATE4
    if(state4 == FALSE or cnt1 > duration1) CS = '11'; cnt1 = 0;
    state4 = TRUE; cnt1++;

Variables:
avg: 평균 큐 사이즈
cnt1: STATE1의 지속 횟수
cnt4: STATE4의 지속 횟수
Fixed parameters:
duration1: STATE1 지속 임계값
duration4: STATE4 지속 임계값
    
```

간 임계값으로 큐 상태가 영역 3으로 변하면 노드는 큐로 들어오는 패킷들 중 일부를 선택적으로 제거한다. 만약 혼잡이 심화되어 영역 4로 변하면 큐가 가득차다는 것을 의미하기 때문에 모든 수신 패킷을 제거한다.

3.1.2 지속 임계값

다중 큐 임계값은 영역이 변화하는 경우에만 혼잡 상태를 판단하기 때문에 하나의 영역에 지속적으로 머무르게 되면 혼잡을 제어하지 못하는 문제가 있으므로 상태의 변화가 없는 경우에도 혼잡을 제어하기 위한 방법이 필요하다. ACT에서는 혼잡 지속 임계값을 설정하여 상태의 변화가 없을 경우에도 혼잡 제어가 가능하게 한다.

지속 임계값은 패킷이 큐로 들어오면 해당 영역의 지속값을 증가시키고, 만약 지속값이 지속 임계값 이상일 경우 해당 영역의 상태에 따라 CS bits를 설정하고, 값을 0으로 초기화한다. 지속 임계값은 혼잡이 발생하여 지속적으로 유지될 경우에 사용된다. 기존의 방식들에서는 혼잡발생을 알리는 패킷을 한번만 전송하기 때문에 자식노드들이 부모 노드의 상태를 파악하기 어려운 문제점이 있었다. 하지만 ACT의 경우에는 혼잡이 지속되면 이를 주기적으로 자식노드들에 통보하여 혼잡상태가 지속되고 있음을 알림으로써 자식노드들이 혼잡을 회피하기 위한 노력을 지속할 수 있도록 한다. 지속 임계값은 다중 임계값의 수, 센서노드의 큐 크기, 네트워크 토폴로지, 네트워크의 전체 센서노드의 수 등에 크게 영향을 받는다. 따라서 최적의 지속 임계값은 큐 오버플로우 비율이 가장 작은 경우의 값을 설정한다.

3.1.3 혼잡 전파

혼잡 제어를 위해 혼잡 발생시 마다 혼잡을 알리는 제어 패킷을 전송하면 혼잡이 가중되기 때문에 일반 데이터 패킷에 센서노드의 상태를 전달하는 혼잡 상태 표시 기법이 필요하다^[10]. ACT에서는 <표 2>에서와 같이 각 센서노드는 패킷의 헤더에 2 bits의 CS(Congestion State) bits를 설정한다^[12].

평균 큐 사용량을 확인한 뒤, 혼잡이 없는 원활한 상태인 경우에는 CS bits을 '00'로 설정하고, 혼잡이 해제되면 '01'로 설정한다. 혼잡이 발생하면 '10'로 설정한다. 혼잡이 심화되어 더 이상 큐에 패킷을 넣을 수 없는 경우에는 하위노드로 패킷전송 중단을 알리는 '11'로 설정한다.

3.2 혼잡 회피

3.2.1 흐름 제어

ACT에서 센서노드는 주기적으로 자식노드수(NUM_{CN})를

<표 2> 데이터 패킷의 혼잡 상태 표시

CS Bits	혼잡상태 표시
00	원활 상태
01	혼잡 해제
10	혼잡 발생
11	전송 중지

상위노드로 전달한다. 여기서 자식노드수는 센서노드가 전달하는 모든 패킷의 생성노드(Source Node)를 의미한다. 센서노드는 중계하는 패킷의 헤더 중 생성노드 필드를 확인하여 자식노드수를 계산한다. 자식노드는 부모노드의 자식노드수(NUM_{ICN})를 수신하여 자신의 자식노드수(NUM_{OCN})으로 나누어 부모노드의 네트워크 부하중 자신이 차지하는 비중

(GNT, Gravity of Network Traffic)을 $GNT = \frac{NUM_{OCN}}{NUM_{PCN}}$ 계산한다.

흐름 제어는 혼잡 발생시 자식노드의 전송 간격을 지연시키는 방법이다. 혼잡 탐지의 결과에 따라 부모노드가 CS bits를 설정한 후 전송하면 이를 수신한 자식노드들은 자신의 현재 전송 주기(PT , a Period of Transmission, default=0)를 $PT_n = (1 + \alpha)PT_{n-1}$ 와 같이 설정한다. α 는 지연계수로 CS bits이 '01'일 경우에는 $\alpha = 0$ 으로 설정하여 현재 상태를 유지한다. CS bits가 '10'일 경우에는 $\alpha = GNT$ 로 설정하여 전송간격을 늘려 부모노드로 전송되는 패킷수를 줄여 혼잡을 완화시킨다. '11'일 경우에는 전송주기를 최대치로 설정하여 부모노드로 패킷이 전송되지 않도록 한다. '01'을 수신한 다음에 '00'을 수신하면 부모노드에서 혼잡이 해결되어 상태가 원활한 것이므로 $\alpha = (-1) * GNT$ 로 설정하여 전송량을 증가시킨다.

3.2.2 압축

ACT에서는 혼잡이 발생할 경우 압축을 이용하여 전송량을 줄임과 동시에 데이터의 시간적 손실을 막을 수 있다. ACT에서는 센서노드가 데이터를 감지하면, 얻어진 값은 양자화 단계를 거치게 된다. 양자화는 데이터 크기의 손실이 발생하기 때문에 혼잡이 발생하지 않으면 양자화를 수행하지 않고, 혼잡이 발생할 경우에만 수행한다. 이것은 혼잡으로 인한 데이터 손실보다 양자화를 통해 혼잡을 제어함으로써 더 높은 전송 효율을 얻을 수 있기 때문이다. 차분펄스 부호변조를 수행한다. ACT에서는 알고리즘의 복잡도를 줄이기 위해 시간 t 의 감지 값 V_t 과 이전 시간의 감지 값 V_{t-1} 사이의 차이를 구한다. 차분펄스부호변조를 사용할 경우 변화가 없는 구간에서는 대부분의 데이터가 0의 값을 가지게 되고, 이후 가변길이 부호화를 적용하여 데이터의 양을 줄임으로써 전송하는 패킷의 양을 줄인다^[14].

세 가지 압축기법들 중에서 차분펄스부호변조와 가변길이 부호화는 무손실 압축기법이며 양자화는 손실 압축기법으로 양자화 단계를 조절하면 생성되는 데이터의 양을 조절할 수 있다. 양자화 파라미터 (QP, Quantization Parameter)값이 증가하면 생성되는 데이터의 양이 줄어들기 때문에 상위노드에서 혼잡이 발생하는 경우 하위노드들은 양자화 파라미터를 증가시켜 생성되는 데이터의 양을 줄여 혼잡을 완화시킨다. 상위노드로부터 수신한 패킷에 CS bit가 '10'으로 혼잡이 발생하였음을 나타내면 $QP_n = QP_{n-1} + 1$ 로 설정하여 생성되는 패킷수를 감소시키며 '11'을 수신하면 QP를 최대치로 설정하여 생성되는 패킷수를 최소화한다. 상위노드로부

터 혼잡해제를 나타내는 '01'을 수신하면 QP는 일정한 값을 유지시킨다. CS bit가 '00'으로 설정되어 있으면 상위노드의 상태가 원활한 것이므로 생성되는 패킷수를 증가시키기 위 $QP_n = QP_{n-1} - 1$ 로 설정한다.

3.2.3 패킷 판단

ACT에서의 각 센서노드는 패킷 종류 판별 주기(Interval_{PD}, Interval for Packet Decision)마다 자신이 생성한 패킷의 수와 자신이 전달하는 패킷의 수를 확인하고, 자식노드 수(NUM_{CN})에 따라 압축률을 조절하여 공평성을 보완하는 패킷판단 알고리즘을 사용한다. 큐에서 전송하는 전체 패킷의 수 (NUM_{TP})중에서 자신이 생성한 패킷수(OP, Own Packet)가 노드당 평균 패킷 수(PPN, Packet Per Node)보다 클 경우 압축단계에서 양자화 파라미터를 증가시켜 생성되는 패킷수를 감

소시킨다. 즉 $PPN = \frac{NUM_{TP}}{NUM_{CN+1}}$ 로 구할 때 $OP > PPN$ 이면 QP를 증가시키고 반대인 경우에는 QP를 감소시킨다.

4. 실험 결과 및 분석

4.1 실험 환경

본 논문에서 제안한 ACT를 센서네트워크용 운영체제인 TinyOS를 이용하여 구현하였고, TOSSIM과 TinyViz를 이용하여 실험하였다^[9]. 100개의 센서노드를 고정 위치에 배치하고, 이웃하는 센서노드와의 통신 거리를 1홉으로 설정하여 네트워크의 모든 센서노드들은 통신이 가능하도록 하였다. 각 센서노드는 임의의 데이터를 감지하며, 감지한 데이터는 부모노드로 데이터를 전송한다. 라우팅 프로토콜로는 TOSSIM에서 기본으로 제공하는 MintRoute를 사용하였다^[2]. 센서노드는 CrossBow사에서 제작 판매하는 Micaz 센서노드를 기본으로 하였다^[15]. Micaz 모트의 사양은 <표 3>과 같다. 시뮬레이션의 세부 설정은 <표 4>와 같이 설정하였다. ACT의 성능 비교를 위하여 기존의 혼잡 제어 기법인 CODA를 동일한 환경에서 구현하여 서로간의 성능을 평가하였다.

4.2 성능 척도

본 논문에서는 네트워크의 성능을 평가하기 위해 세 가지

<표 3> 시뮬레이션용 센서모트 사양

MCU	ATMEGA 128L 8MHz
Memory	4K RAM / 128K FLASH
RF	Chipcon CC2420 IEEE 802.15.4/ZigBee 2.4GHz Frequency band 250Kbps Transmit data rate -24dBm to 0 dBm RF power 20m to 30m indoor Range

<표 4> 시뮬레이션 환경 설정

매개변수	값
하위, 중간, 상위 임계값	24, 36, 48
큐 크기	64
패킷 크기	41
노드당 샘플링 수	5000
고정 샘플링 주기	62ms
최대 지연 시간	255ms
전송 가능 범위	1 hop
유지 임계값	64
패킷 판별 주기	3000ms

의 평가 요소들을 사용한다. 첫째 효율성(Efficiency, η)은 네트워크의 전체 데이터 전송량과 싱크의 수신율에 따라 효율적인 데이터 전송을 평가한다. 둘째 각 센서노드의 데이터 전송량과 싱크의 수신율에 따른 센서노드별 데이터 수집에 대한 공평성(Fairness, ϕ)을 평가한다. 셋째 이용 가능한 데이터의 양(Quality of Data, QoD)은 센서노드들이 감지한 데이터의 양과 싱크에서 수신한 데이터의 양을 이용하여 이용 가능한 데이터의 양을 평가한다^[7, 13].

효율성(η)은 식 (1)으로 나타내어진다. U 는 싱크가 수신한 패킷을 전송한 센서노드의 집합, P 는 네트워크의 모든 노드의 집합, $hops_n$ 는 싱크와 센서노드 n 과의 홉 수, r_n 는 센서노드 n 으로부터 싱크가 수신한 패킷 수, $S_c(u)$ 는 센서노드 u 자신이 생성한 데이터의 전송 패킷 수, $S_f(u)$ 는 센서노드 u 가 자식들로 받은 패킷을 싱크로 전달하는 패킷 수이다.

$$\eta = \frac{\sum_{n \in U} (hops_n \times r_n)}{\sum_{n \in P} (S_n(u) + S_f(u))} \quad \text{식 (1)}$$

공평성(ϕ)은 식 (2)와 같이 나타낸다.

$$\phi = \frac{(\sum_{i=1}^N r_i)^2}{N \sum_{i=1}^N r_i^2} \quad \text{식 (2)}$$

N 은 전체 센서노드 수를 나타내며 r_i 는 i 번째 노드로부터 수신한 패킷 수를 나타낸다.

이용 가능한 데이터의 양(QoD)은 식 (3)과 같이 구할 수 있으며, i 는 일정한 시간 간격을 나타내고, d_i 와 d_i' 는 i 시간 동안 사용자가 요구하는 데이터의 양과, 센서노드가 감지하는 데이터의 양을 각각 나타내고, 데이터 손실률(δ_i)은 $\delta_i = |d_i - d_i'|/d_i$ 와 같다. 이용 가능한 데이터의 양(QoD)는 아래 식 (3)과 같고, T 는 단위 실험 시간을 나타

낸다.

$$QoD = \frac{1}{T} \sum_{i=1}^T 1 - \delta_i \quad \text{식 (3)}$$

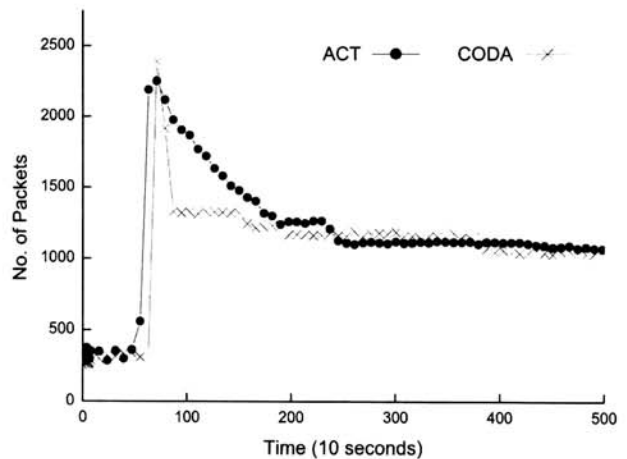
4.3 생성 패킷 수

(그림 1)에서 패킷 생성량이 최대치로 증가하였다가 바로 감소하는 것을 볼 수 있는데 이는 패킷 생성량 증가로 인해 발생한 네트워크에 혼잡이 탐지가 되어 패킷 생성량을 줄이기 때문이다. CODA에서 혼잡이 탐지되어 혼잡 발생을 하위로 통보하면 하위노드들이 AIMD(Additive Increase Multiplicative Decrease)를 적용하여 패킷 생성량을 줄이기 때문에 패킷 생성량이 급격히 줄어드는 것을 볼 수 있다. 이에 반해 ACT에서는 혼잡 발생시 적응적 흐름제어를 통하여 중간 노드들에서 전송간격을 조절하기 때문에 혼잡 발생이 CODA에 비해 천천히 전파되어 생성되는 패킷량이 서서히 감소하는 것을 볼 수 있다.

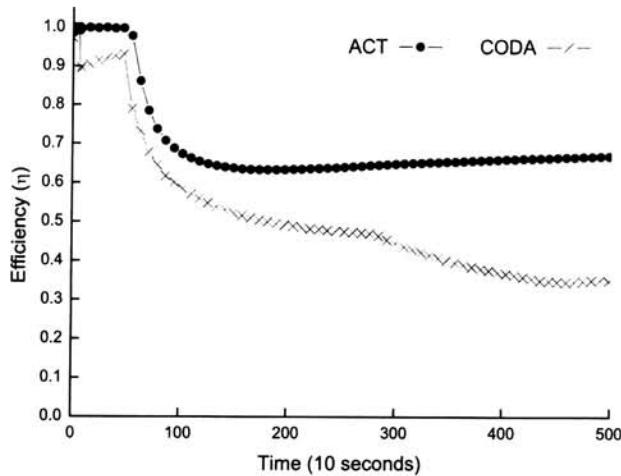
4.4 효율성

혼잡발생시 가장 먼저 이루어지는 것은 혼잡이 발생한 노드에서 패킷을 버리는 것이다. 패킷이 싱크로 전송되는 중간에 혼잡으로 인하여 손실되면 패킷이 손실되기 전까지의 전달과정은 불필요하게 이루어진 것으로 자원이 낭비된 것으로 볼 수 있다. CODA의 경우 센서노드에서 혼잡이 발생하면 주위노드들이 일정시간 동안 전송을 중지하여 혼잡이 완화되도록 한다. 하지만 일정시간 후에는 다시 모든 노드들이 빠르게 패킷을 전송하기 때문에 혼잡이 다시 심해지는 악순환이 발생한다. 따라서 혼잡이 발생할 가능성이 높은 싱크 주변의 노드들에서는 혼잡이 반복적으로 발생하여 패킷이 지속적으로 손실되는 문제점이 있다. 네트워크 효율성을 나타내는 (그림 2)에서 CODA의 효율성이 시간이 지남에 따라 점차 낮아지는 것은 이런 이유에서이다. CODA는 단기적으로는 혼잡을 해결할 수 있으나 장기적으로 혼잡이 반복되어 효율성이 낮아지는 문제점을 가진다.

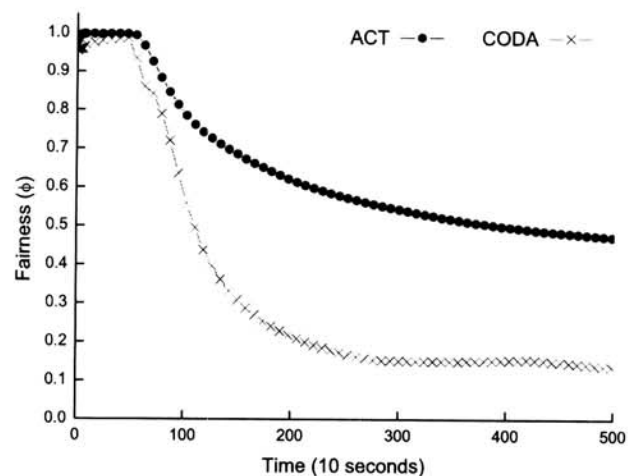
이에 반해 ACT는 센서노드에 혼잡이 발생하면 주위 노



(그림 1) 생성 패킷 수



(그림 2) 네트워크 효율성



(그림 3) 공정성

드들이 적응적 흐름제어를 통해 전송간격을 늘려 혼잡을 완화시키고자 한다. 여기서 ACT와 CODA의 큰 차이점은 CODA는 일시적으로만 전송간격을 증가시키지만 ACT는 혼잡이 해결될 때까지 전송간격을 점진적으로 늘려나간다는 것이다. 이는 혼잡이 발생한 노드에서 혼잡이 지속되면 혼잡탐지기법 중의 지속임계값을 통하여 혼잡이 지속되고 있음을 주위노드들로 통보할 수 있기 때문이다. 혼잡이 지속되고 있다는 통보를 받는 주위노드들은 전송간격을 더욱 늘려 혼잡을 완화시키고자 노력한다. 여기서 전송간격이 늘어나면 해당노드에도 혼잡이 발생하여 다시 하위노드들로 혼잡발생이 통보된다. 이런 과정을 여러 단계에 걸쳐서 수행하면 혼잡이 점차 하위로 전달되어 최종적으로 트리의 제일 하단까지 도달하면 네트워크 중간에서는 패킷손실이 거의 발생하지 않으며 하단의 일부 노드에서만 패킷이 손실된다. (그림 2)에서 보면 ACT의 효율성도 초기에는 낮아지지만 CODA에 비해서는 20%이상 높게 유지하는 것을 볼 수 있으며 시간이 지남에 따라 약간씩 증가하여 최종적으로는 CODA에 비해 두배이상 높은 효율성을 나타냄을 볼 수 있다.

4.5 공정성

CODA의 경우 혼잡이 발생한 노드가 취하는 동작은 하위노드들로 혼잡 발생을 통보하는 것 뿐이다. 상위노드로부터 혼잡발생을 통보 받은 하위노드들은 패킷 전송간격과 샘플링 주기를 조절하여 상위노드로 전달되는 패킷량을 줄여 혼잡을 해결하고자 한다. 하지만 정작 혼잡이 발생한 노드 자체에서 생성되는 패킷에 대해서는 고려하지 않기 때문에 자체 생성되는 패킷수는 줄어들지 않는다. 노드들 간의 패킷 전송의 공정성을 보여주는 (그림 3)을 보면 CODA의 경우 혼잡이 발생하면 노드간 공정성이 급격히 낮아지는 것을 볼 수 있다. 이는 앞서 언급하였듯이 네트워크의 혼잡은 하위로 전파되는데 혼잡이 발생한 노드들에 패킷생성율은 줄지 않기 때문에 상위노드들과 하위노드들간에 패킷생성 및 전송에 차이가 심해지기 때문이다.

이에 반해 ACT에서는 혼잡회피기법 중에 패킷 판단을 통하여 자신이 생성하는 패킷과 하위노드들로부터 전송되는

패킷의 비율을 판단하여 자신이 생성하는 패킷량을 조절하기 때문에 혼잡발생시 하위노드들의 패킷량이 감소하면 자신이 생성하는 패킷량도 감소시킨다. 따라서 혼잡이 하위노드들로 전파되어 하위로부터 전송되는 패킷량이 감소하면 상위노드들도 패킷생성량을 감소시키기 때문에 CODA에 비해 높은 공정성을 보장한다. (그림 3)에서 ACT도 CODA와 마찬가지로 혼잡발생시 공정성이 감소하는 것을 볼 수 있지만 공정성이 10%정도로 감소하는 CODA비해 ACT는 50%이상의 공정성을 유지하는 것으로 나타나 CODA에 비해 5배 이상 좋은 성능을 나타내고 있다.

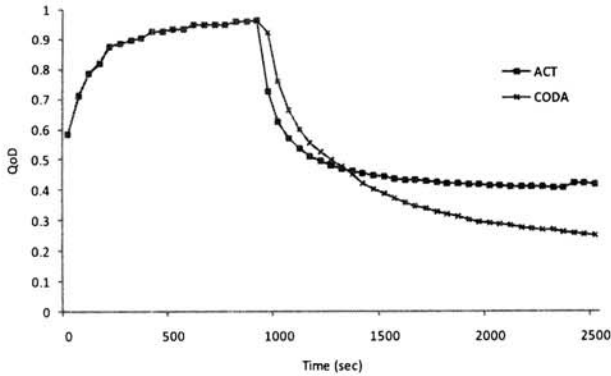
4.6 이용 가능한 데이터의 양

혼잡발생시 패킷이 손실되는 것을 줄이기 위해서는 근본적으로 센서노드들이 생성하는 패킷의 양을 줄여야 한다. CODA와 같은 기존의 혼잡제어 기법들에서는 혼잡발생시 데이터 샘플링 주기를 늘리는 방식을 주로 사용하고 있다. 데이터 샘플링 간격이 늘어나면 시간당 생성되는 데이터의 양이 줄고 따라서 생성되는 패킷양도 감소하는 효과를 가져올 수 있다. 하지만 샘플링 주기를 조절하면 시간당 생성되는 데이터의 양이 줄기 때문에 시간당 고정적인 샘플링 주기를 필요로 하는 응용에 적용하기 어려운 문제점이 있다. ACT는 압축기법을 적용하여 데이터의 시간적 손실을 제거한다.

(그림 4)는 싱크노드에서 수신한 패킷에서 추출한 이용가능한 데이터의 양을 보여준다. CODA와 ACT 모두 혼잡이 발생하면서 패킷이 손실되기 때문에 이용가능한 데이터의 양이 급격히 감소하는 것을 볼 수 있다. 하지만 시간이 지나면서 지속적으로 데이터의 양이 감소하는 CODA에 비해 ACT는 일정한 수준으로 수렴하는 것으로 볼 수 있으며 최종적으로 거의 2배에 가까운 데이터를 보장함을 볼 수 있다.

5. 결 론

센서 네트워크에서의 혼잡 현상은 데이터 손실을 가져오고, 결과적으로 부가적인 전력 소모와 전송 지연이 발생한



(그림 4) 이용가능한 데이터의 양

다. 또한 멀티홉 환경에서는 싱크에서 수신하는 전체 센서 노드의 데이터는 불균등한 데이터 수신율을 보이게 된다. 본 연구에서는 센서 네트워크에서의 혼잡을 탐지하고, 혼잡이 발생하는 센서노드 및 주변 센서노드의 부하를 줄여 혼잡을 해결하는 흐름 제어를 제안하였다.

제안된 기법은 혼잡 탐지, 혼잡 제어 메커니즘으로 구성하였다. 혼잡 탐지 메커니즘은 큐 임계값과 유지 임계값을 설정하여 혼잡을 탐지하였다. 혼잡 제어 메커니즘에서는 큐에 입력되는 데이터의 전달 종류를 판단하여 더 높은 공평성을 제공하도록 하였다. 또한 혼잡 탐지의 결과에 따라 흐름 제어 기법과 압축을 동작시켜 혼잡을 제어하도록 하였다. 흐름 제어 기법을 통해 싱크 주변 센서노드들의 큐 오버플로우를 방지함으로써 수집된 데이터들이 싱크에 도달할 수 있는 효율성을 개선하였다. 또한 데이터 전송량 조절을 위해 압축을 사용함으로써 기존의 샘플링 주기 조절 기법보다 모든 센서노드들로부터 공평하게 데이터가 수집될 수 있게 하였다. 실험을 통하여 데이터의 양이 많이 요구되는 응용의 경우 제안된 기법을 사용하여 전체 센서노드들의 데이터 전송의 효율성이 향상되었고, 각 센서노드들이 수집한 데이터들에 대한 공평성 또한 향상되었다.

참 고 문 헌

[1] J. Zhao, R. Govindan, "Understanding Packet Delivery Performance In Dense Wireless Sensor Networks", Proceedings of the 1st International Conference on Embedded Networked Sensor Systems(SenSys), 2003.
 [2] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks" In Proceedings of the first international conference on Embedded networked sensor systems, pp.14-27, 2003.
 [3] N. Xu, S. Rangwala, K. Kant Chintalapudi, D. Ganesan, A. Broad, R. Govindan and D. Estrin, "A wireless sensor network for structural monitoring" in Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, USA, pp.13-24,

2004.
 [4] A. Woo, D. E. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks" in Seventh Annual International Conference on Mobile Computing and Networking, pp.221-235, July, 2001.
 [5] Y. Sankarasubramaniam, O. Akan and L. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks", in MobiHoc'03, Annapolis, Maryland, USA, pp.177-188, June, 2003.
 [6] C-Y. Wan, S. Eisenman and A. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks" in Proceedings of the 1st International Conference on Embedded Networked Sensor Networks, Los Angeles, California, USA, pp.266-279, November, 2003.
 [7] B. Hull, K. Jamieson, H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks" In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems(SenSys), 2004.
 [8] C. Intanagonwivat, R. Govindan, and D.Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks" In MobiCom 2000.
 [9] "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications" In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys), 2003.
 [10] A. Duresi, M. Sridharan, C. Liu, M. Goyal, "Traffic Management using Multilevel Explicit Congestion Notification", In Proceedings of 5th World Multiconference on Systemics, Cybernetics and Informatics, 2001.
 [11] B. A. Warneke and S. J .Pister, "MEMS for Distributed Wireless Sensor Networks", 9th IEEE International Conference on Electronics, Circuits, and Systems.
 [12] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, pp.397-413, August, 1993.
 [13] M. Amirjoo, S. H. Son, and J. Hansson, "QoS Adaptation for Achieving Lifetime Predictability of WSN Nodes Communicating over Satellite Links," International Conference on Networked Sensing Systems (INSS), 2007.
 [14] Z. Nian Li, and M. S. Drew, "Fundamentals of Multimedia", Prentice-Hall, 2004.
 [15] <http://www.xbow.com/>
 [16] C.-T. Ee and R. Bajcsy, "Congestion control and fairness for many-to one routing in sensor networks," in Proc. ACM Sensys, Nov., 2004.
 [17] A. Woo and D. C. Culler, "A transmission control scheme for media access in sensor networks," in Proc. ACM Mobicom, July, 2004.
 [18] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y.

Hu, "Upstream congestion control in wireless sensor networks through cross-layer optimization," in IEEE Journal on select areas in Communications, Vol.25, No.4, 2007.

- [19] M. Hossein Yaghmaee, and D. Adjeroh, "A new priority based congestion control protocol for wireless multimedia sensor networks", in international symposium on a world of wireless, mobile and multimedia networks, 2008.



이좌형

e-mail : jhlee@snslab.kangwon.ac.kr
 2003년 강원대학교 정보통신공학과(공학사)
 2005년 강원대학교 컴퓨터정보통신공학과
 (공학석사)
 2005년~현 재 강원대학교 컴퓨터정보통신공학과 박사과정

관심분야: 멀티미디어 시스템, 센서 네트워크



김동국

e-mail : dkkim@snslab.kangwon.ac.kr
 2005년 강원대학교 컴퓨터정보통신공학과
 (공학사)
 2008년 강원대학교 컴퓨터정보통신공학과
 (공학석사)
 2008년~현 재 (주)다이렉트 연구원

관심분야: 센서 네트워크, 정보검색



정인범

e-mail : ibjung@snslab.kangwon.ac.kr
 1985년 고려대학교 전자공학과 학사
 1985년~1995년 (주) 삼성전자 컴퓨터 시스템사업부 선임 연구원
 1994년 한국과학기술원 정보통신공학과 석사
 2000년 한국과학기술원 전산학과 박사

2001년~현 재 강원대학교 컴퓨터정보통신공학전공 교수

관심분야: 멀티미디어 시스템, 센서네트워크