센서 네트워크에서 Event-driven 데이터의 신뢰성 있는 전송 및 버퍼 관리 기법

정회원 김 대 영*, 조 진 성*°

A Reliable Transmission and Buffer Management Techniques of Event-driven Data in Wireless Sensor Networks

Dae-Young Kim*, Jinsung Cho*° *Regular Members*

요 약

무선 센서 네트워크에서는 멀티 홉 전송동안 높은 패킷 손실률이 발생하기 때문에 신뢰성 있는 데이터 전송 방안이 필요하다. 특히, 화재 경보 시스템과 같은 event-driven 데이터가 발생하는 경우, 신뢰성 있는 데이터 전송을 위해서는 손실된 패킷을 복원하기 위한 재전송 방안이 제공되어야 한다. 손실된 데이터의 재전송은 데이터를 캐쉬하고 있는 노드에 요청이 되기 때문에, 데이터를 캐쉬하고 있는 노드는 모든 데이터 패킷을 버퍼에서 유지하고 있어야 한다. 그러나 일반적으로 센서 네트워크의 노드들은 제한된 자원을 가지 있다. 따라서 신뢰성 있는 데이터 전송을 위해서는 손실 패킷의 재전송 방안과 노드의 버퍼 관리 기법이 함께 제공되어야 한다. 본 논문에서는 전송 데이터의 신뢰도에 따라 데이터의 캐쉬지점을 결정하여 손실된 데이터를 복원하는 손실 복원 기법을 사용하는 데이터 전송에서의 효율적인 버퍼 관리기법을 제안하고, 컴퓨터 시뮬레이션을 통하여 제안하는 방안의 우수성을 검증하였다.

Key Words : Wireless Sensor Networks, Reliable Transmission, Loss Recovery, Buffer Management

ABSTRACT

Since high packet losses occur in multi-hop transmission of wireless sensor networks, reliable data transmission is required. Especially, in case of event-driven data, a loss recovery mechanism should be provided for lost packets. Because retransmission for lost packets is requested to a node that caches the packets, the caching node should maintains all of data for transmission in its buffer. However, nodes of wireless sensor networks have limited resources. Thus, both a loss recovery mechanism and a buffer management technique are provided for reliable data transmission in wireless sensor networks. In this paper, we propose a buffer management technique at a caching position determined by a loss recovery mechanism. The caching position of data is determined according to desirable reliability for the data. In addition, we validate the performance of the proposed method through computer simulations.

Ⅰ. 서 론

무선 센서 네트워크는 많은 수의 노드가 인프라가 없는 곳에 배치되어 스스로 네트워크를 구성함으로써 이루어진다. 최근의 무선 통신 기술의 발달로 다양한 기능을 탑재한 센서 노드로 구성된 센서 네트워크가

※ 이 논문은 2009년도 경희대학교 연구년 지원에 의한 결과임.
* 경희대학교 컴퓨터공학과 모바일 및 임베디드 시스템 연구실 ({kimdy,chojs}@khu.ac.kr), (° : 교신저자)
논문번호 : KICS2010-03-098, 접수일자 : 2010년 3월 9일, 최종논접수일자 : 2010년 6월 15일

가능하게 되었다. 센서 네트워크에서는 센서 노드들이 주변 환경의 상태를 모니터링하여 싱크 노드라고 불리는 서버에 모니터링한 정보를 전송하는 것으로 서비스가 이루어진다[1-3]. 이 때 싱크 노드로 전송되는 정보는 센서 노드가 수집한 주변의 상태 정보를 나타내고 있으며, 화재 경보 시스템의 예처럼 이 정보는 제공되는 센서 네트워크 서비스에서 중요한 요소로 작용한다. 즉, 원활한 센서 네트워크 서비스의 제공을 위해서는 센서 노드에서 싱크 노드까지 신뢰성이 있는 전송 방안이 필요하다.

센서 네트워크에서 신뢰성 보장을 위한 방안은 크게 두 가지로 구분이 된다: 싱크 노드에서 수집된 데이터의 양으로 신뢰도를 만족하는 방안과 소스 노드와 싱크 노드 사이에서 종단간 신뢰성 있는 전송을 보장하는 방안이 있다. 첫 번째 방안은 싱크 노드에서 수집한 데이터의 양이 신뢰도를 만족하지 못할 경우 소스 노드들은 보고 주기 (reporting frequency)를 조절하여 싱크 노드로의 잦은 데이터 업데이트를 유도하여 신뢰도를 만족시킨다[4,5]. 이 경우 광범위한 지역에서 비슷한 데이터가 발생했을 때 신뢰도를 보장하기 위한 방법이다. 그러나 소스 노드가 이벤트 기반의 데이터를 발생시킨다면, 제한된 양의 데이터가 생성되기 때문에 많은 양의 수집된 데이터로 신뢰도를 보장하는 방안은 적합하지 못하게 된다. 그래서 이벤트 기반의 데이터를 전송하기 위해서는 종단간 신뢰성 있는 전송을 보장하는 방안이 사용된다. 신뢰성이 보장된 전송을 위해서 소스 노드에서 발생한 데이터는 무선 링크 위에서 멀티 홉으로 싱크 노드까지 완전히 전송되어야 한다. 무선 링크는 높은 손실률을 가지고 있으며, 멀티 홉 전송이 이루어질수록 패킷 손실률은 지수적으로 증가하게 된다. 따라서 종단간 전송 데이터의 신뢰성을 보장하기 위해 손실된 패킷을 복원하기 위한 손실 복원 기법이 필요하다[6,7].

손실 복원 기법은 손실된 패킷에 대하여 재전송을 요청하는 재전송 메커니즘을 바탕으로 하고 있다. 즉, 전송 중 손실된 데이터를 복원하기 위해서 데이터가 저장된 노드에 재전송을 요청하게 된다. 일반적으로 센서 네트워크를 구성하는 노드는 제한된 에너지와 메모리를 가지고 있기 때문에, 손실 복원 기법을 위해 메모리를 무한히 사용할 수 없으며 버퍼에 손실 복원을 위한 데이터를 오랜 시간동안 유지할 수도 없다. 만약, 버퍼가 모두 사용되어 더 이상 데이터를 저장하지 못하게 되면, 저장하지 못한 데이터의 손실이 발생하였을 때 재전송을 할 수 없게 된다. 이 경우 해당 데이터의 신뢰성 있는 전송을 보장하지 못하게 된다. 따라서 이벤트 기반의 데이터에 대한 신뢰성 있는 전송을 위해서는 손실 복원 기법과 함께 효율적인 버퍼 관리 기법이 제공되어야 한다.

본 논문에서는 소스 노드의 데이터에서 요구하는 통신 신뢰도 (Communication Reliability: CR)의 레벨에 따라 손실된 데이터 패킷의 재전송을 위한 데이터 복원 지점을 결정하는 손실 복원 기법, *Active Caching*[8]을 사용하는 데이터 전송에서, 복원 지점의 노드에서 사용할 효율적인 버퍼 관리 기법을 제공한다. 센서 네트워크에서는 소스 노드에서 발생하는 데이터와 서비스의 종류에 따라 서로 다른 통신 신뢰도 레벨을 가지기 때문에, 요구되는 통신 신뢰도 레벨 마다 서로 다른 복원 지점을 가져야 한다. 또한, 복원 지점의 노드는 버퍼 관리 기법에 데이터의 통신 신뢰도 레벨을 반영해야 한다. 즉, 데이터 복원 지점에서 저장되는 모든 데이터는 통신 신뢰도 레벨에 따라 서로 다른 시간동안 버퍼에 유지될 수 있어야 한다. 본 논문에서 제안된 이벤트 기반 데이터의 신뢰성 있는 전송을 위한 버퍼 관리 기법은 *Active Caching*을 예로 하여 설명되나, 종단간 전송 데이터의 신뢰성을 보장하기 위한 일반적인 손실 복원 기법[6,7]에도 활용될 수 있다. 또한, 컴퓨터 시뮬레이션을 통해 그 성능을 검증한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구로서 센서 네트워크에서 사용되고 있는 손실 복원 기법과 버퍼 관리 방안들을 소개하고, 3절에서 데이터의 요구되는 통신 신뢰도 레벨에 따른 손실 복원 기법을 기술하며, 4절에서 복원 지점에서 사용하는 효율적인 버퍼 관리 기법을 설명한다. 5절에서는 성능 평가를 보이고, 마지막 6절에서 결론을 맺는다.

Ⅱ. 관련 연구

종단간 데이터 전송에서 손실된 패킷들의 복원 방법에는 종단간 손실 복원(end-to-end loss recovery)과 홉간 손실 복원(hop-by-hop loss recovery)이 있다[6,7]. 종단간 손실 복원은 데이터를 발생시킨 소스 노드에 손실 패킷에 대한 재전송을 요청하는 것이고, 홉간 손실 복원은 전송 데이터를 캐쉬(cache)하고 있는 중계 노드에 재전송을 요청하는 방법이다. 종단간 손실 복원은 멀티 홉 전송에 따른 높은 패킷 손실률로 인해 종단간 전송 지연을 유발시킨다. 반면, 홉간 손실 복원은 모든 중계 노드에서 전송 데이터를 캐쉬하기 때문에 높은 메모리 요구사항을 가진다. 즉, 두 기법 사이에는 종단간 전송지연과 메모리 요구사항 측면에서

이해득실 (tradeoff) 관계를 가지게 된다. 센서 네트워크의 데이터 트래픽은 응용 서비스에 따라 다양한 레벨의 CR을 가지기 때문에 모든 종류의 트래픽을 종단간 손실 복원 또는 홉간 손실 복원 기법에 적용하는 것은 적절하지 못하다. 따라서 다양한 센서 네트워크 서비스를 지원하기 위해서는 응용 서비스의 CR에 따라 라우팅 경로 상에서 데이터의 캐쉬 지점을 다르게 선정할 수 있는 유연한(flexible) 손실 복원 기법이 필요하며, *Active Caching*[8]이 유연한 손실 복원을 위해 제공되었다. *Active Caching*은 CR에 따라 기존의 종단간 손실 복원 기법과 홉간 손실복원 기법을 포함하는 가장 일반적인 형태의 손실 복원 기법이며, 3절에서 보다 자세히 설명한다.

비록 신뢰성 있는 데이터 전송을 위해 유연한 손실 복원 기법이 제공된다 하더라도, 일반적으로 센서 노드는 제한된 메모리 자원을 가지고 있기 때문에 재전송을 위해 캐쉬된 데이터를 오랜 시간동안 버퍼에 유지할 수가 없다. 버퍼에서 재전송을 위해 패킷을 유지하는 방법에는 크게 2가지가 있다. 첫 번째는 패킷의 RTT (Round Trip Time)를 이용하는 방법이다[9]. 이 방법에서는 데이터를 수신한 노드가 정해진 버퍼링 시간동안 재전송 요청이 없을 때 까지 버퍼에 데이터를 유지한다. 이 때 버퍼링 시간은 RTT와 버퍼링 팩터 β의 곱으로 결정된다. RTT 시간을 이용하여 예측한, 버퍼에서의 패킷 유지 시간은 손실된 패킷이 재전송 되는 경우를 반영하지 않기 때문에 정확한 패킷 유지 시간을 제공할 수 없다. 두 번째는 누적된 ACK 시퀀스 넘버를 수신한 노드가 버퍼에서 수신한 시퀀스 넘버까지의 패킷을 버퍼에서 삭제하는 방법이다[10]. 이 방법은 싱크 노드에서 손실되지 않고 연속적으로 수신한 패킷의 마지막 시퀀스 넘버를 나타내는 누적 ACK 시퀀스 넘버를 피드백 메시지에 포함시켜 소스노드에 전송을 한다. 그러면 피드백 메시지를 수신한 소스 노드는 누적 ACK 시퀀스 넘버까지의 패킷을 안전하게 버퍼에서 삭제할 수 있다. 비록 누적된 시퀀스 넘버를 이용하는 방법이 버퍼에 존재하는 패킷을 안전하게 삭제할 수 있더라도, 목적지 노드에서 소스 노드의 버퍼를 비우기 위해 피드백 메시지를 발행하는 것이 오버헤드가 될 수 있다. 따라서 효율적인 재전송 버퍼의 관리 기법은 손실된 패킷이 재전송되는 시간이 반영되어야 하고 버퍼에 유지되는 패킷들이 추가적인 메시지의 전송없이 안전하게 삭제되어야 한다.

Ⅲ. 신뢰성 있는 전송을 위한 유연한 손실 복원 기법: Active Caching

*Active Caching*은 전송해야 하는 데이터의 CR 레벨에 따라 종단간 손실 복원 기법과 홉간 손실 복원 기법의 사이에서 동작한다. *i*-th 홉에서 패킷 손실률 p_i가 주어지면, 그림 1에서 보여주는 것처럼 각 중계 노드는 소스 노드에서 부터의 패킷 전달율 (P_{tx})을 전송 데이터 패킷에 피기백(piggyback)하여, 이 정보를 바탕으로 다음 홉에서의 누적 패킷 전달율을 예측한다: *line 1*. 예측된 누적 패킷 전달율이 전송 데이터에서 요구되는 통신 신뢰도를 만족하지 못하는 경우 해당 패킷을 현재 노드의 버퍼에 캐쉬하게 된다: *line 2-4*. 이렇게 함으로써 손실된 데이터 패킷은 요구되는 통신 신뢰도에 따라 서로 다른 캐쉬 지점을 가지게 되고, 이 캐쉬 지점들에게 재전송을 요청하게 된다. 전송 패킷이 노드의 버퍼에 캐쉬되면 누적 패킷 전달율을 1로 보정하고, 다음 홉으로 전송한 후 전송 패킷의 캐쉬 여부를 판단한다. 그래서 소스 노드에서 목적지 노드까지의 누적 패킷 전달율은 항상 요구되는 통신 신뢰도를 만족할 수 있게 된다.

그림 2는 평균적인 패킷 손실률이 5%이고 데이터의 CR 레벨이 80%일 때, *Active Caching* 손실 복원 기법을 수행한 예이다. 소스 노드에서 목적지 노드까지 전송되는 동안 한 번의 데이터 저장이 발생하였고 (n_5), 모든 전송 구간에서 80%이상의 패킷 전달율을 보였다.

센서 네트워크의 응용 서비스에 대한 통신 신뢰도가 결정되면 손실 복원 기법은 요구되는 CR 레벨을

```
Reliable-Transmit(CR, i, Ptx(i-1))
1. Ptx(i) ← Ptx(i-1)×(1-pi)
2. if Ptx(i) ≤ CR
3.     then Ptx(i) ←1×(1-pi)
4.          cache data packets to a node ni
```

그림 1. 노드 n_i에서 수행되는 *Active Caching* 손실 복원 기법의 알고리즘

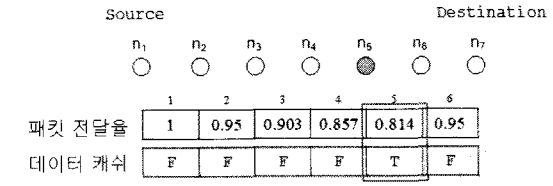


그림 2. *Active Caching*의 사용 예

만족하기 위해 필요한 메모리 요구사항을 제시한다. 가장 높은 신뢰도를 요구하는 서비스에 대해서는 홉간 손실 복원 기법과 동일하게 동작하며, 가장 낮은 신뢰도를 요구하는 서비스에 대해서는 종단간 손실 복원 기법과 동일하게 동작하게 된다. 높은 신뢰도를 만족하기 위해서는 데이터를 캐쉬하는 중계 노드의 수가 늘어나지만 그 만큼 재전송에 의한 전송 지연은 줄어든다. 데이터를 캐쉬하고 있는 중계 노드의 수가 많지 않으면 손실 복원을 위한 재전송 메시지는 여러 홉을 거쳐 캐쉬 지점에 도달하게 되고, 복원되는 패킷들 역시 멀티홉 전송을 해야 하기 때문에 전송 지연은 증가하게 된다.

Ⅳ. 제안하는 신뢰성 있는 데이터 전송을 위한 버퍼 관리 기법

신뢰성 있는 데이터 전송을 위해 종단간 손실 복원과 홉간 손실 복원이 제안되었다. 종단간 손실 복원 기법은 멀티홉 전송동안의 높은 손실률 때문에 100%의 신뢰도를 보장하기 위해 종단간 긴 전송 지연을 유발시킨다. 그래서 높은 신뢰도을 유지하며 전송 지연 시간을 줄이기 위해 홉간 손실 복원 기법이 제안되었으나, 경로상의 모든 중계 노드에서 데이터를 캐쉬해야 하기 때문에 높은 메모리 요구사항을 가진다. 그러나 센서 네트워크 응용들은 다양한 신뢰도를 가지며, 다양한 종단간 허용 전송 지연을 가지기 때문에 홉간 손실 복원 기법과 종단간 손실 복원 기법 사이에서 유연하게 동작할 수 있는 손실 복원 기법이 제공되어야 한다.

신뢰성 있는 데이터 전송을 위해 유연한 손실 복원 기법이 제공됨에도 불구하고, event-driven 데이터 트래픽은 전송 중 여러 중계노드에서 저장되기 때문에 높은 메모리 요구사항을 가지며, 센서 노드의 제한된 메모리 인해 재전송을 위한 패킷들을 버퍼에 오랜 시간동안 유지할 수 없다. 뿐만 아니라, 여러 개의 데이터 플로우 (flow)를 가지는 중계 노드는 더 많은 메모리를 요구사항을 가진다. 따라서 신뢰성 있는 전송을 위해서 각 센서 노드는 저장된 패킷들을 언제까지 유지할 지에 대한 버퍼 관리 기법을 가지고 있어야 한다.

제안된 버퍼 관리 기법은 버퍼에 캐쉬된 패킷들을 유지하기 위한 시간을 계산하기 위해서, 다음 캐쉬 지점까지의 재전송을 포함한 완전한 데이터 전송에 필요한 패킷 전송 횟수를 예측하여 사용한다. K개의 패킷을 h홉 동안 전송하게 될 때, 손실이 발생하지 않은 경우 전송 횟수는 Kh가 되고, 전송 중에 패킷 손실이 발생하게 되면 전체 전송 횟수는 Kh에 재전송에 필요한 전송 회수를 더한 것으로 결정된다. 이 때, 손실된 패킷이 재전송되는데 패킷 재전송은 손실된 패킷이 모두 전달될 때까지 반복된다.

패킷이 전송되는 동안 i번째 홉에서의 패킷 손실률이 p_i라고 할 때, s번째 센서 노드에서 h홉 동안 패킷 전송률은 식(1)과 같다

$$P_{tx}(s,h) = \prod_{i=s}^{s+h-1}(1-p_i). \quad (1)$$

식(1)의 $P_{tx}(s,h)$을 3.1절의 손실 복원 기법에 적용하면 현재 노드 s에서 다음 캐쉬 지점까지의 홉수 h를 알 수 있다. 그림 3은 제안하는 손실 복원 기법을 적용하여 h를 계산하기 위한 함수를 나타낸다.

h홉 동안 성공적으로 전송된 패킷 수 $\Gamma_s(k,s,h)$는 k개 패킷 중에서 성공적으로 전달된 m개의 패킷에 대한 Bernoulli trials $P_s(k,m,s,h)$로 다음과 같이 표현할 수 있다

$$\Gamma_s(k,s,h) = \sum_{m=1}^{k}\left(mP_s(k,m,s,h)\right). \quad (2)$$

그리고 $P_s(k,m,s,h)$는 식(3)과 같다

$$P_s(k,m,s,h) = \binom{k}{m}P_{tx}(s,h)^m\left(1-P_{tx}(s,h)\right)^{k-m}. \quad (3)$$

식(2)를 이용하면, 다음 캐쉬 지점으로의 재전송해야 하는 손실 패킷의 수를 예측할 수 있다. 데이터 패킷들이 다음 캐쉬 지점으로 완전히 전송될 때까지 필요한 시간은 그림 4의 함수를 사용하여 h홉 동안 발생하는 모든 손실 패킷들의 재전송 횟수를 계산함으로써 예측할 수 있게 된다.

그림 4에서 보여주는 것처럼, 전송해야 하는 전체 데이터 패킷의 수가 K일 때, 재전송 패킷 수 r과 재전

```
Hop-Count(CR)
1. h ← 1
2. loop: Ptx(s,h) > CR
3.       h ← h + 1
4. end loop
```

그림 3. 다음 캐쉬 지점까지의 홉수 계산 함수

```
ReTx-Count(K,s,h)
1.  k ← K, packets ← 0
2.  r ← 0, q ← 0, success ← 0
3.  loop: packets < K
4.        success ← [Γ_s(k,s,h)]
5.        packets ← packets +success
6.        if packets < K
7.             then q ← q +1
8.                  k ← k - success
9.                  r ← k × h + r
10. end loop
```

그림 4. 다음 캐쉬 지점까지의 재전송 패킷 수를 예측하기 위한 함수

송 요청 메시지의 수 q는 K 패킷에 대한 $\Gamma_s(k,s,h)$의 계산을 반복함으로써 K개의 패킷을 완전히 전송하기 위해 필요한 재전송 횟수를 예측할 수 있다.

r과 q를 획득한 캐쉬 지점의 노드는 패킷의 크기(S)와 데이터 전송률(R)을 고려하여 자신의 버퍼에서 해당 데이터 패킷을 위한 저장 시간(T_b)을 결정한다

$$T_b = \frac{(K \times h + r + q \times h) \times S}{R}. \qquad (4)$$

이렇게 각 캐쉬 노드에서 T_b동안 버퍼에 캐쉬된 데이터를 유지하면, 데이터의 전송 신뢰도를 보장하기 위해 필요한 시간동안만 버퍼를 사용할 수 있게 된다.

비록 T_b의 예측으로 인해 버퍼가 적절한 시간동안 데이터를 유지하더라도, 여러 개의 데이터 플로우에 의해 노드의 버퍼가 가득 찰 수 있다. 버퍼 오버플로우가 발생하면 버퍼에서 폐기 (drop)된 패킷들은 손실되게 되고, 재전송 요청에 응답할 수 없어 신뢰성 있는 전송이 불가능하게 된다. 따라서 버퍼가 최대 임계값 (high threshold)을 넘어서게 되면, 캐쉬 지점의 노드는 유입되는 트래픽의 전송률을 조절해야 한다. 제안하는 버퍼 관리 기법에서는 전송 트래픽의 이전 캐쉬 지점에 전송률을 조절하기 위한 메시지를 전달하면, 이전 캐쉬 지점의 노드는 전송률을 낮추고 수정된 전송률을 바탕으로 남아있는 데이터의 버퍼링 시간을 수정한다. 또한 가득 차있던 버퍼가 최소 임계값 (low threshold)으로 떨어지면, 다시 이전 캐쉬 지점에 메시지를 전달하여 낮춰진 전송률을 원래 상태로 되돌린다.

1) $n-0.5 \le x < n+0.5$ 일 때, $[x]$는 n.

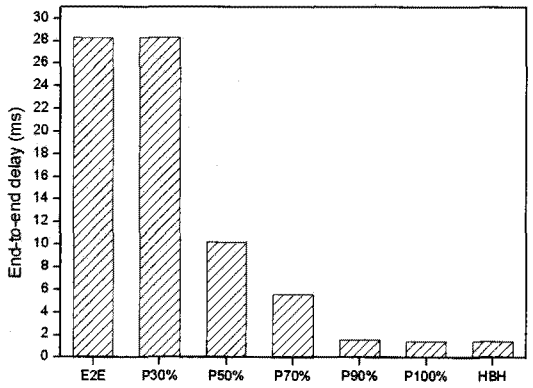
V. 성능 분석

4절에서 센서 네트워크에서 event-driven 데이터의 신뢰성 있는 전송을 위한 효율적인 버퍼 관리 기법을 고안하였다. 이 절에서는 end-to-end loss recovery (E2E), hop-by-hop loss recovery (HBH)[6,7] 와의 컴퓨터 시뮬레이션 결과를 통해 유연한 손실 복원 기법인 *Active Caching* 에서의 종단간 지연과 메모리 요구사항에 대한 이해득실 (tradoff)을 분석한다. 또한, RTT를 기반으로 하는 버퍼 관리 기법[9]과 누적된 ACK 시퀀스를 이용하는 버퍼 관리 기법[10]의 컴퓨터 시뮬레이션 결과를 통해 제안하는 버퍼 관리 기법에 대한 성능 분석을 수행한다. 컴퓨터 시뮬레이션을 위해서 시뮬레이터는 SMPL 라이브러리[11]를 이용해서 구현되었으며, SMPL은 C언어 기반의 event-driven 시뮬레이터이다.

5.1 손실 복원 기법 실험

20개의 센서 노드가 일렬로 분포되어 있는 환경에서 소스 노드는 30bytes 크기의 데이터 패킷 40개를 전송하며, 이 때 무선 채널에서의 평균 패킷 손실률은 5%, 데이터 전송률은 250kbps로 가정하였다. *Active Caching*의 CR은 30%, 50%, 70%, 90%, 100%로 설정하였으며, 시뮬레이션에서 P30%, P50%, P70%, P90%, P100%로 나타내었다. 데이터 전송 중 손실이 발생하면, 손실 복원 기법들은 NACK 메커니즘에 의해 재전송을 요청하게 된다. E2E는 소스 노드에, HBH는 이전 노드에 재전송을 요청하게 된다. 또한 *Active Caching*은 그림 1의 알고리즘을 이용하여 주어진 CR에 따라 재전송 지점을 다르게 결정하고, 그곳에서 재전송을 요청하게 된다.

그림 5는 *Active Caching*과 E2E, HBH 사이의 성능 비교를 보여준다. 메모리 요구사항을 전송 경로 위에서 데이터를 캐쉬하는 노드의 비율로 나타낼 때, HBH가 가장 많은 메모리 요구사항을 가진다. 반면에, E2E는 소스 노드에서만 데이터를 캐쉬하기 때문에 가장 적은 메모리 요구 사항을 가진다. 그러나, E2E의 경우 재전송 요청 시간이 오래 걸리고, 재전송 중에 손실되는 데이터가 많이 발생하기 때문에 데이터 전송을 완료하기까지 많은 시간이 걸린다. 따라서 다양한 종류의 CR을 가지는 데이터에 E2E 또는 HBH를 동일하게 적용하는 것은 오히려 효율을 떨어뜨리게 된다. 그림 5에서 나타나는 것처럼 Active Caching은 CR에 따라 기존의 손실 복원 기법인 E2E와 HBH 사이에서 동작하며, 전송 데이터의 CR에 따라서 서로



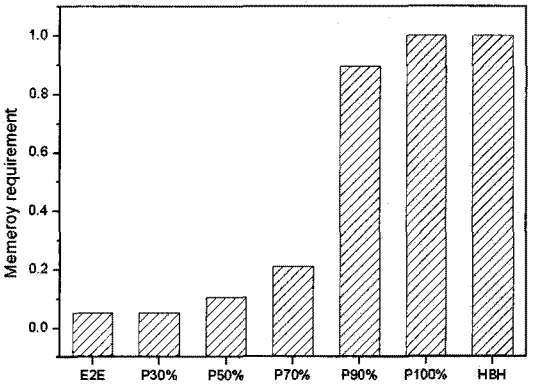


그림 5. 손실 복원 기법의 성능 비교

다른 메모리 요구사항과 종단간 전송 지연시간을 가진다. 그래서 다양한 전송 데이터에 대해 서로 다른 데이터 캐쉬 지점을 적용할 수 있다. 즉, 전송 데이터는 CR에 따라 메모리 요구사항을 최소화하기 위한 종단간 전송 지연을 일부 허용함으로써 전송 신뢰성과 함께 효율성도 보장 받을 수 있게 된다.

5.2 버퍼 관리 기법 실험

100개의 센서 노드가 분포되어 있는 센서필드에서 6개의 소스 노드가 데이터 전송을 시도하는 환경에서 손실된 패킷을 복원하기 위해 *Active Caching*을 사용하였다. 소스 노드들의 이벤트 데이터는 Poisson 분포를 가지고 발생되며, 이벤트 마다 30bytes 크기인 데이터 패킷 20개를 생성시킨다. 또한 전송 데이터의 CR은 70%와 100% 사이에서 결정되도록 하였다. 이 때, 무선 채널에서의 평균 패킷 손실률은 5%, 데이터 전송률은 250kbps로 가정하였다. 이 절에서는 동일한 손실 복원 기법에서 서로 다른 버퍼 관리 기법의 성능을 비교한다.

그림 6은 제안한 버퍼 관리 기법과 RTT를 기반으로 한 버퍼 관리 기법의 버퍼에서 데이터 패킷을 유지하는 시간을 나타낸다. CR이 낮을수록 데이터가 버퍼에서 유지되는 시간이 길어진다. 제안하는 버퍼 관리 기법의 경우 'RTT-based'[9]보다 짧은 시간동안 전송 데이터를 버퍼에서 유지하며, 이 시간은 하나의 RTT 시간보다도 짧은 것으로 나타난다. 'RTT-based'는 β가 1일 때, 1 RTT동안 재전송 요청이 발생하면 1 RTT를 더 기다리게 된다. 제안한 버퍼 관리기법은 손실된 데이터의 재전송 시간을 고려하여 버퍼에서의 데이터 유지 시간이 결정되기 때문에, 짧은 시간동안 전송 데이터를 버퍼에서 보관하더라도 재전송에 통해 손실없이 데이터 전송을 완료할 수 있다.

그림 7은 6개의 소스 노드에서 데이터 전송이 발생하였을 때, 평균 버퍼 요구 사항을 나타낸다. 이 때 소스 노드에서 전송되는 데이터의 CR은 70%와 100%의 사이에서 랜덤하게 결정되도록 하였다. 초당 발생되는 이벤트가 많아질수록 버퍼 요구 사항은 증가한다. 누적된 ACK 시퀀스를 이용하는 피드백 (feedback) 기반

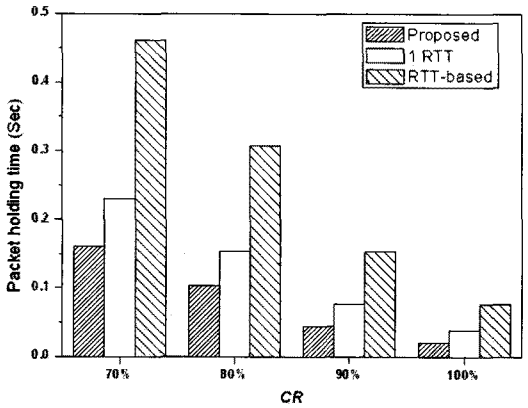


그림 6. 버퍼에서의 데이터 유지 시간

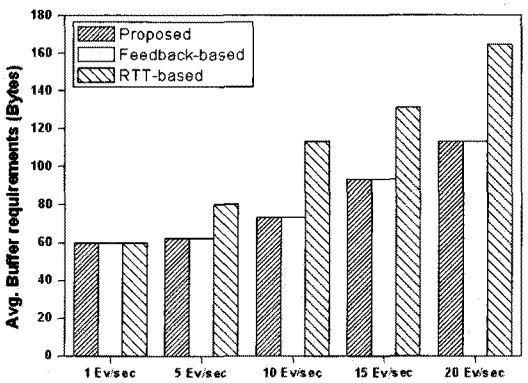


그림 7. 평균 버퍼 요구 사항

인 'Feedback-based'[10]는 전송 성공한 데이터 패킷에 대해서 버퍼에서 재빨리 삭제할 수 있지만, 누적된 ACK 시퀀스에 대한 피드백 메시지를 요구한다. 그러나 RTT를 이용한 'RTT-based'[9]는 비교적 긴 시간동안 전송 데이터를 버퍼에 유지하지만, 추가적인 메시지의 전송을 요구하지 않는다. 제안한 버퍼 관리 기법은 'RTT-based'처럼 추가적인 메시지 전송은 요구하지 않으면서, 'Feedback-based'에서와 같이 적은 버퍼 요구 사항을 가짐으로써 효율적으로 버퍼를 사용할 수 있는 방안을 제공한다.

추가적인 메시지 전송은 더 많은 에너지의 소비를 이끌기 때문에, 네트워크의 수명을 감소시킬 수 있다. 따라서 센서 네트워크의 시스템은 추가적인 메시지의 발송을 최대한 줄여야 한다. 표 1은 CR에 따른 에너지 소비량을 나타낸다. 에너지 소비량을 나타내기 위해 LEACH[12]의 에너지 모델이 사용 되었다. 노드간 거리를 10m로 가정하면, 소스 노드에서 싱크 노드까지 전송 횟수가 τ 일 때 l-bit 데이터를 전송하는 경우 소비되는 에너지는 $E=(0.11\times10^{-6})\times l\times\tau$ Joule로 나타난다. CR이 낮을수록 재전송 횟수의 증가로 인해 에너지 소모가 증가한다. 'Feedback-based'[10]의 경우 버퍼 관리에서 사용되는 추가적인 메시지로 인해 제안한 기법과 'RTT-based'[9]의 기법보다 많은 에너지가 사용됨을 알 수 있다. 모든 버퍼 관리 기법이 손실된 패킷의 복원을 위해 *Active Caching*의 기법을 사용하기 때문에 제안한 기법과 'RTT-based' 사이에서 재전송 횟수의 차이는 발생하지 않는다. 따라서 제안한 기법과 'RTT-based'는 동일한 에너지를 소비하지만, 'RTT-based'가 캐쉬된 패킷들에 대해 더 긴 버퍼 점유 시간을 허용한다.

이렇게 제안한 버퍼 관리 기법이 데이터 패킷의 불필요한 버퍼 점유를 방지함으로써 버퍼의 효율을 높인다. 이로 인해 재전송이 요구되는 데이터를 효율적으로 유지, 관리함으로써 데이터의 신뢰성 있는 전송을 제공할 수 있다. 뿐만 아니라, 추가적인 메시지의 사용을 줄여 불필요한 에너지 소모도 줄임으로써 중계 노드의 생존을 증가시켜 신뢰성 있는 전송을 오랫동안 유지할 수 있도록 한다.

표 1. 종단간 데이터 전송에서 발생하는 에너지 소비량

CR	Proposed	Feedback	RTT
100%	104.5 *mJ*	109.3 *mJ*	104.5 *mJ*
90%	106.7 *mJ*	111.4 *mJ*	106.7 *mJ*
80%	119.3 *mJ*	125.1 *mJ*	119.3 *mJ*
70%	122.0 *mJ*	128.3 *mJ*	122.0 *mJ*

VI. 결 론

무선 센서 네트워크에서 이벤트 기반의 데이터가 발생할 경우, 데이터의 신뢰도를 보장하기 위해 종단간 신뢰성 있는 전송 방법이 사용되어야 한다. 종단간 신뢰성 있는 전송 방법은 전송시 손실되는 데이터 패킷을 위한 손실 복원 기법을 제공해야 하고, 손실 복원을 위해 전송 데이터를 저장하는 노드를 위한 버퍼 관리 기법을 가지고 있어야 한다. 본 논문에서는 데이터의 중요도에 따라 재전송을 위한 전송 데이터의 저장 위치를 다르게 결정하는 손실 복원 기법인 *Active Caching*[8]위에서의 효율적인 버퍼 관리 기법을 제공하여 무선 센서 네트워크의 event-driven 데이터를 위한 신뢰성 있는 전송 방안을 보이고, 컴퓨터 시뮬레이션을 통해 그 효율성을 검증하였다. 제안하는 버퍼 관리 기법은 다음 데이터 캐쉬 지점까지 발생되는 재전송 횟수를 확률적으로 계산하고, 이를 바탕으로 버퍼에서의 데이터 저장 시간을 예측하여 사용한다. 이렇게 함으로써 버퍼 관리를 위해 추가적인 메시지를 사용하지 않고 버퍼에서의 데이터 유지 시간을 최소화할 수 있었다. 또한, *Active Caching*은 일반적인 손실 복원 방안을 제공하기 때문에 그 위에서 동작하는 제안한 버퍼 관리 기법 또한 일반적으로 활용할 수 있다.

본 논문에서 분석한 재전송을 위한 손실 복원 기법인 *Active Caching*은 손실 복원 기법 실험에서 전송 데이터의 중요도 (CR)에 따라 메모리 요구 사항과 종단간 지연사이에서 유연하게 동작됨을 보여주었으며, 제안한 버퍼 관리 기법은 버퍼 관리 기법 실험에서 재전송이 필요한 시간동안 버퍼에 데이터를 유지함으로써 버퍼의 효율을 높여 재전송이 완료되지 않은 데이터가 버퍼에서 폐기 (drop)되는 것을 방지하는 것을 보여주었다. 제안된 방안을 활용하는 것으로 무선 센서 네트워크에서의 데이터는 싱크 노드까지 손실없이 완전하게 전송될 수 있다.

참 고 문 헌

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survery on Sensor Networks," IEEE Communications Magzine, Vol.40, No.8, pp.102-114, August 2002.

[2] D. culler, D. Estrin, and M. Srivastava,

"Guest editors' introduction: Overview of sensor networks," IEEE Computer, Vol.37, No.8, pp.41-49, August 2004.

[3] H. Karl, Protocols and Architectures for Wireless Sensor Networks, John Wiley & Sons, 2005.

[4] Y. Sankarasubramaniam, O. B. Akan and I. F. Akyildiz, ''ESRT: event-to-sink reliable transport in wireless sensor networks,'' Proc. of ACM International Symposium on Mobile Ad hoc Networking and Computing, pp.177-188, 2003.

[5] K. S. Yim, J. Kim and K. Koh, "An Energy-Efficient Reliable Transport for Wireless Sensor Networks," Lecture Notes in Computer Science, Vol.3090, pp.54-64, 2004.

[6] C. Y. Wan, A. T. Campbell and L. Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," Proc. of ACM International Workshop on Wireless Sensor Networks and Applications, pp.1-11, September 2002.

[7] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," Proc. of IEEE International Workshop on Sensor Network Protocols and Applications, pp. 102-112, May 2003.

[8] D. -Y. Kim and Jinsung Cho, "Active Caching: A Transmission Method to Guarantee Desired Communication Reliability in Wireless Sensor Networks," IEEE Communications Letters, Vol.13, No.6, pp.378-380, June 2009.

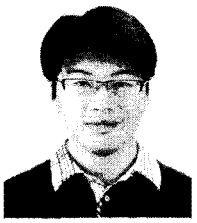
[9] Z. Xiao, K. P. Birman, and R. V. Renesse, "Optimizing Buffer Management for Reliable Multicast," Proc. of IEEE International Conference on Dependable Systems and Networks, pp.187-196, 2002.

[10] J. Paek and R. Govindan, "RCRT: Rate-Controlled Reliable Transport for Wireless Sensor Networks," Proc. of ACM International Conference on Embedded Networked Sensor Systems, pp.305-319, 2007.

[11] M. H. MacDougall, Simulating Computer Systems, Techniques and Tool, The MIT Press, 1987.

[12] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishman, "Energy-efficient communication protocol for wireless microsensor networks," Proc. of The 33rd Hawaii International Conference on System Sciences, January 2000.

김 대 영 (Dae-Young Kim) 정회원



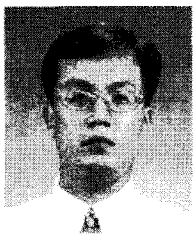
2004년 2월 경희대학교 전자공학과

2006년 2월 경희대학교 컴퓨터공학과 석사

2006년 3월~현재 경희대학교 컴퓨터공학과 박사과정

<관심분야> 모바일 네트워크, 센서 네트워크, 임베디드 시스템

조 진 성 (Jinsung Cho) 정회원



1992년 2월 서울대학교 컴퓨터공학과

1994년 2월 서울대학교 컴퓨터공학과 석사

2000년 2월 서울대학교 컴퓨터공학과 박사

1997년 4월~8월 IBM T.J. Watson Research Center Visiting Researcher

1999년 9월~2003년 2월 삼성전자 책임연구원

2003년 3월~현재 경희대학교 컴퓨터공학과 조교수

<관심분야> 모바일 네트워크, 센서 네트워크, 임베디드 시스템