

실시간 센서네트워크 응용을 위한 혼잡 회피 기법

정원우^o 차호정
연세대학교 컴퓨터학과
(wwjung^o, hjcha)^o@cs.yonsei.ac.kr

A Congestion Avoidance Mechanism for Real-Time Sensor Network Applications

Won-woo Jung^o Hojung Cha
Dept. of Computer Science, Yonsei University

요 약

실시간 센서네트워크 응용은 사용자의 요구에 따른 실시간성을 보장하여야 한다. 센서네트워크에서 혼잡 문제를 다룬 기존의 연구들은 소스 노드 혹은 중간 노드에서의 전송률 제한 및 전송 지연을 통해 트래픽을 억제하는 방식을 사용하기 때문에 실시간성을 보장하지 않는다. 본 논문에서는 일정 시간 동안 전송되는 실제 데이터 수준을 유지하는 동시에 네트워크 트래픽을 억제하는 방식을 통해 실시간 센서네트워크 응용에 적절한 혼잡 회피 기법을 제안한다. 이는 네트워크에 존재하는 데이터를 실시간성을 기준으로 분류하여, 각각 병합 및 첨부 방식을 통해 하나의 패킷에 다수의 데이터를 포함시키는 기법이다. 시뮬레이션 결과에 의하면 본 기법의 적용에 의해 실시간 응용의 성능은 전송 성공률과 실시간성 측면에서 그렇지 않은 경우에 비해 높은 성능을 보여주고 있다.

1. 서론

실시간 센서네트워크 응용은 센서 필드에서 수집된 정보를 최대한 빠른 시간 안에 사용자에게 전달해야 하는 응용으로써 이동 대상 위치 추적, 재난 방지를 위한 실시간 환경 모니터링 등이 이에 속한다. 이와 같은 응용들은 비실시간 센서네트워크 응용에 비해 보다 많은 제약 조건을 갖는다. 이는 해당 응용에 대한 사용자의 요구 수준에 따라 정해진다. 즉, 사용자는 실시간 응용에서, 특정 허용 시간 이내에 일정 수준의 실시간성, 발생 이벤트에 대해 고정된 보고 주기를 기대한다.

센서네트워크에서의 혼잡 문제는 이벤트 발생 시, 해당 이벤트에 대한 전송 요구량이 전체 네트워크의 사용 가능한 자원보다 클 때 발생한다.[1] ESRT[2]는 이와 같은 사실에 근거하여, 싱크 노드에서 중앙 집중적인 방식을 통해 일정 주기마다 소스 노드의 보고 주기를 적절히 조절하여 주는 방식으로 혼잡 문제를 해결하는 기법을 제안하고 있다. CODA[3]는 싱크에서 소스 노드로 전송하는 ACK 메시지의 수신 수준을 기반으로 소스 노드가 전송률 제한함으로써 전체 센서 필드에서의 지속적인 혼잡 문제를 해결하려는 정책과 함께, 순간적으로 지속되는 지역적 혼잡 상태 해결을 위해 중간 노드에서 혼잡 완화 시까지 전송 수준을 억제하는 방식을 제안한다. Kang 등[4]은 복수의 이벤트에 대한 데이터들이 별개의 싱크 노드로 전송되는 경우, 이들의 경로가 중복되는 지점에서 발생하는 혼잡 문제를 해결하기 위한 기법을 제안하고 있다. 즉, 혼잡 지역에서 복수의 경로를 설정하여 트래픽을 분산시키는 방법으로 혼잡 문제를 완화할 수 있다고 주장한다.

실시간성이 없는 센서네트워크 응용의 경우, 하나의 이벤트에 대해 다수의 노드가 중복적으로 보고를 시도하는 것이 일반적이다. 그러나 위치 추적 시스템을 비롯한 실시간 센서네트워크 응용에서는 하나의 이벤트에 대해 오직 하나의 노드만이 보고를 시도한다. 따라서 혼잡 완화를 위해 소스 노드의 보고 주기를 변경하는 기법은 사용자가 요구하는 전송 요구량을 만족시키지 못하는 결과를 야기하므로 적용이 불가능하다. 이와 함

께 실시간 응용은 생성된 데이터가 일정 시간 이내에 싱크 노드에 도착해야 하는 실시간 제약조건을 갖는다. 따라서 중간 노드에서 무조건적으로 전송을 제한하는 기법은 실시간성을 보장하지 못하므로 사용이 배제되어야 한다. 또한 실시간 응용을 포함한 대부분의 센서네트워크 응용은 하나의 싱크 노드가 다수의 이벤트를 보고 받는 것이 일반적인 상황이므로, 각각의 이벤트에 대해 독립적인 싱크 노드가 존재한다는 가정에 의한 트래픽 분산 기법은 적용이 불가능하다.

본 논문에서는 실시간 센서네트워크 응용에 적용 가능한 혼잡 관리 정책을 제안한다. 이는 기존의 관련 연구와는 달리 소스 노드 혹은 중간 노드에서의 전송률 제한 혹은 지연 등의 기법을 배제하고, 중간 노드에서 하나의 패킷에 다수의 데이터를 포함시키는 방식을 통해 네트워크의 혼잡 상태를 완화시킨다. 이를 위해 생성 후 경과 시간을 기준으로 각 데이터의 실시간성을 평가하여 실시간 데이터와 비실시간 데이터로 분류한 후, 이들을 병합하여 다수의 실시간 데이터를 포함한 새로운 패킷을 생성하거나, 전송 패킷의 여유 공간에 비실시간 데이터를 첨부하는 방식을 사용한다.

2. 실시간 응용을 위한 혼잡 제어 기법

비실시간 응용과 달리 위치 추적 시스템을 비롯한 실시간 센서네트워크 응용을 위한 혼잡 문제 해결을 위해서는 다음과 같은 제약 조건들을 고려해야 한다. 우선 특정 수준 이상의 실시간 제약이 존재하여 사용자는 특정 시간 이내에 데이터를 전송 받기를 원한다. 다음으로 센서 필드 내에 특정 이벤트에 대한 데이터는 유일하다. 따라서 하나의 데이터가 손실되는 것은 해당 이벤트에 대한 정보 전체가 손실되는 것을 의미한다.

그림 1은 위의 제약 조건들을 고려하여 설계한 위치 추적 시스템의 백본 노드 구조이다. 백본 노드는 위치 추적 시스템에서 소스 노드로부터 싱크 노드로까지의 데이터 전송을 담당한다. 백본 노드에 적용된 혼잡 제어 정책은 새로운 데이터를 실시간성을 기준으로 분류하여, 병합 혹은 지연 후 첨부 정책을 통해 한 패킷에 다수의 데이터를 포함시켜 네트워크의 트래픽을 감소시키는 것이다.

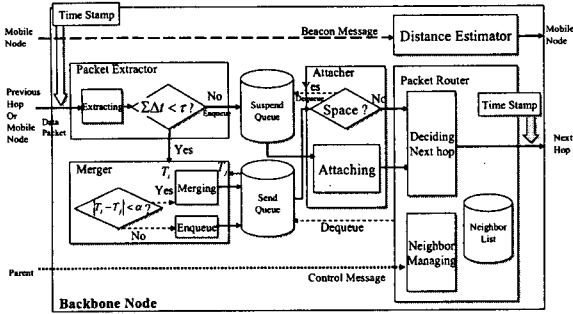


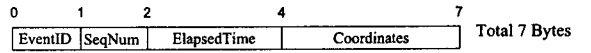
그림 1. 정책 적용을 위한 백본 노드 구조

백본 노드에 새롭게 전송된 패킷은 하나 혹은 그 이상의 데이터를 포함한다. 이들은 실시간성을 기준으로 분류된다. 즉, 데이터의 생성 이후 경과 시간 (Δt)과 실시간성 평가의 기준이 되는 특정 시간 (τ)을 비교하여, τ 를 초과하지 않는 경우 실시간성이 유효한 데이터로, 초과하는 경우 실시간성을 잃은 데이터로 분류된다. 분류된 데이터는 각각 즉각적인 전송 단계와 공간이 있는 다른 패킷에 침부를 기다리기 위한 지연 단계를 수행하게 된다. 이러한 정책을 통해 실시간성이 유효한 데이터는 그렇지 않은 데이터보다 우선적으로 전송될 것을 기대할 수 있다. 동시에 실시간성을 잃은 데이터 ($\Delta t \geq \tau$)에 의해 혼잡 문제가 심화되는 것을 방지할 수 있다.

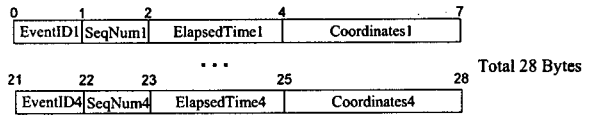
실시간성을 유지하고 있는 데이터 ($\Delta t < \tau$)는 현재 전송 큐 (Send Queue)에 존재하는 패킷들에 포함된 데이터와 생성 후 경과 시간을 비교된다. 만약 둘의 생성 시간 차이가 특정 값 (α)보다 작으면, 둘은 병합(Merge)된다. 만약 둘의 생성 시간 차이가 α 보다 크면, 데이터는 새로운 패킷을 생성하여 전송 큐에 삽입된다. 따라서 전송 큐에 존재하는 하나의 패킷에는 생성 시간이 비슷한 다수의 데이터가 포함된다.

지연 후 침부는 Packet Extractor에서 분류된 데이터들 중 실시간성을 잃은 데이터들 ($\Delta t \geq \tau$)에 대상으로 한다. 이들은 이미 실시간성을 잃은 상태이므로 싱크로의 전송을 서두를 필요가 없다. 따라서 이들은 현재의 백본 노드 내에 일시적으로 보관되어 불필요한 트래픽의 증가를 억제하게 된다. 즉, 지연 큐(Suspend Queue)에 보관된다. 이들이 전송되는 시점은 실시간성이 있는 데이터들로 이루어진 패킷이 전송 큐에서 나와 (dequeue) 전송을 준비하는 시점이다. 만약 해당 패킷에 여유 공간이 있다면, 해당 공간이 지연 큐의 데이터들로 채워진 후 패킷은 다음 노드(next hop)로 전송되게 된다. 결과적으로 지연 후 침부는 실시간성을 기준으로 유효한 패킷의 전송을 원활하게 해주는 동시에 유효하지 않은 패킷에 대해서도 손실을 최대한 막아주는 역할을 한다. 실시간성을 잃은 데이터들을 단순히 포기하지 않는 이유는 위치 추적 시스템에서 사용자가 원하는 정보는 대상의 현재 위치와 더불어 대상의 이동 경로도 포함될 것이라는 가정에 의해서이다. 싱크에 연결된 단말은 실시간성을 잃은 데이터가 도착하였을 경우, 해당 대상의 과거 부분을 완성하여 이동 경로에 대한 정보를 보강하게 된다.

그림 2는 위치 추적 시스템에서 본 제안 정책을 적용하기 위한 데이터 및 패킷 구조의 예를 보여주고 있다. 그림 2(a)는 대상의 위치 정보를 위한 데이터 구조의 예이다. 그림 2(b)는 하나의 패킷에 4개의 데이터가 포함된 예를 보여주고 있다. 실제 TinyOS[5] 기반의 MICA2[6] 센서 노드를 이용할 경우 하나의 패킷에 포함될 수 있는 데이터의 최대 크기는 29 bytes이다. 따라서 그림 2 (b)는 하나의 위치 정보를 표현하는 데이터의 크기가 7 bytes 일 경우 하나의 패킷에 데이터가 최대로 포함된 상황을 보여주고 있다.



(a) 이동 노드의 위치 정보를 위한 데이터 구조



(b) 하나의 패킷에 4개의 데이터가 포함된 예

그림 2. 데이터 및 패킷 구조

3. 실험

3.1. 인자 도출을 위한 사전 실험

제안하는 혼잡 정책의 실제 구현을 위해서는 최대 전송 허용 시간 (τ)과 병합 기준 시간 (α)에 해당하는 인자값이 필요하다. 본 논문에서는 시뮬레이션 결과를 바탕으로 해당 인자값을 결정하였다.

실험은 TinyOS가 제공하는 센서네트워크 시뮬레이터인 TOSSIM[7] 상에서 이루어졌다. 시뮬레이션을 위한 데이터 및 패킷의 구조는 그림 2와 같다. 각 노드 간 거리는 40ft(=12m)이다. MICA2의 유효 통신 거리는 약 15m이므로 이와 같은 설정에서 각각의 노드는 상하좌우 총 네 개의 노드를 자신의 이웃 노드로 인식하여 직접 통신한다. 노드의 개수는 총 100 개이며 이들은 108m×108m의 센서 필드에 격자형으로 위치한다. TOSSIM 상에서는 이동 대상의 시뮬레이션이 불가능하므로, 위치 정보 데이터는 센서 필드 내의 전체 노드 중 30%의 노드가 패킷을 생성하여 싱크 노드로 전송하도록 하였다. 싱크 노드와 패킷 생성 노드들의 위치는 표 1과 같다.

	싱크 위치 (노드 번호)	패킷 생성 노드 배치
설정 1	중앙 (5)	임의 분산
설정 2	모서리 (0)	임의 분산
설정 3	중앙 (5)	밀집
설정 4	모서리 (0)	밀집

표 1. 싱크 위치 및 패킷 생성 노드 배치 설정

각각의 적정값들은 다양한 인자값으로 설정한 실험을 통해 싱크 노드까지의 전송 성공률과 지연 시간의 경향을 기준으로 선택되었다. α 를 1000ms로 고정된 상태에서, τ 값이 1000ms 이상이 되었을 때 전송 성공률에서 성능 감소를 보였다. 지연 시간은 τ 를 기준으로 한 실시간성 측면에서는 τ 가 증가할수록 좋은 성능을 보였으나, 절대 시간 측면에서는 1000ms 이하에서 최적의 성능을 보였다. 따라서 최적값은 1000ms 이하에서 결정되는 것이 타당하며, 정책의 성능 평가를 위해 선택한 값은 1000ms이다. τ 를 1500ms로 고정된 상태에서, α 가 40ms 미만 이 되었을 때 전송 성공률 면에서 급격한 성능 감소를 나타냈다. 지연 시간은 다양한 α 에 대해 뚜렷한 성능 차이를 나타내지 않았다. 따라서 α 는 40ms 이상의 값으로 선택되어야 하며, 본 논문의 성능 평가 부분을 위해 선택한 값은 300ms이다.

3.2. 성능평가

제안 정책의 성능은 싱크까지의 전송 성공률과 지연 시간의 경향 측면에서 평가되었다. 시뮬레이션을 위한 설정은 3.1과 동일하며, 싱크 노드의 위치와 패킷 생성 노드의 배치는 표 1의 설정 3을 기준으로 하였다. 각 인자값은 3.1의 결과를 토대로

는 1000ms, α 는 300ms로 설정하였다.

전송 성공률은 B-MAC[8] 기반의 ACK 메시지를 사용하지 않는 경우와, 사용하여 재전송이 이루어지는 상황에서 정책을 적용한 경우와 적용하지 않은 경우를 각각 비교하였다. 이와 함께 정책을 적용하지 않은 경우 노드가 비정상적으로 큰 전송 큐(2000×36bytes)를 갖고 있는 상황에 대한 평가도 이루어졌다.

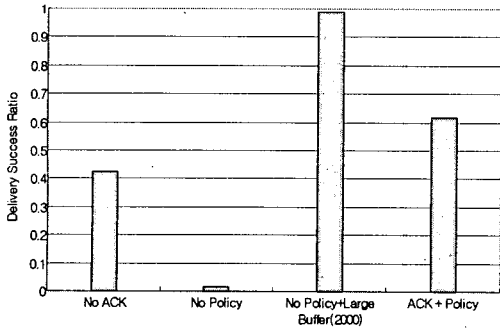
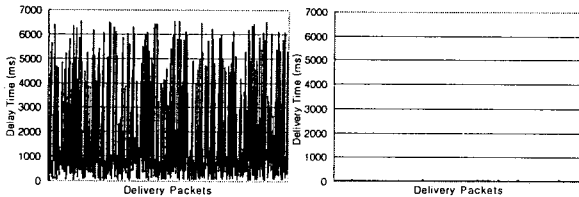


그림 3. ACK 및 정책 적용 여부에 따른 전송 성공률

그림 3은 각 경우의 전송 성공률을 보여주고 있다. ACK와 정책을 동시에 사용한 경우가 약 60% 정도로, ACK를 사용하지 않는 경우와 정책을 적용하지 않은 경우에 비해 높은 수준의 성능을 나타냈다. 큰 전송 큐를 사용하였을 경우가 전송 성공률 면에서 모든 경우에 비해 가장 높은 성능을 보였으나, 실제 도착 패킷의 90% 이상이 35000ms이후에 도착하므로 실시간성 측면에서 의미가 없다. 지연 시간은 ACK를 사용하여 정책을 적용한 경우와 ACK를 사용하지 않은 경우에 대해 비교 평가가 이루어졌다.

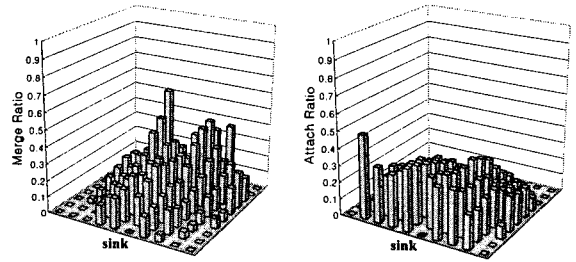


(a) 정책 적용 시 지연 시간 (b) ACK를 사용하지 않은 경우 지연 시간

그림 4. 지연 시간 경향

그림 4는 각각의 경우에 대해 싱크에 도착한 순서대로 패킷 1000개의 지연 시간을 보여주고 있다. 절대 시간 측면에서는 ACK를 사용하지 않은 경우가 우수한 성능을 보여주고 있다. 그러나 정책 적용 시를 기준으로 상대적인 실시간성을 유지하고 있는 패킷들은 전체 도착 패킷의 약 60% 이상이었다. 전송 성공률 면에서 정책을 적용하였을 때가 ACK를 사용하지 않은 경우에 비해 성능이 우수하므로 상대적인 실시간성에서는 두 경우가 큰 차이를 보이지 않는 것으로 해석할 수 있다. 즉, ACK를 사용하지 않은 경우 손실된 패킷의 양만큼이 정책을 적용함에 따라 실시간성을 잃은 상태의 패킷으로 싱크에 도달하고 있는 것이다. 그림 5는 전체 센서 필드에서 제안한 정책의 발생 빈도를 보여주고 있다. 각 막대는 해당 위치의 노드에서 데이터가 다른 패킷과 병합 혹은 첨부되는 비율을 나타낸다. 병합은 이벤트 발생 지역에서, 빈번하게 발생하며, 첨부는

싱크에 가까운 노드에서 발생 빈도가 높다. 이러한 경향성은 다음과 같이 해석이 가능하다. 즉, 데이터가 싱크에 가까워짐에 따라 경과 시간이 증가하고 이로 인해 첨부의 발생 비율은 싱크로 접근하면서 높아지게 되는 것이다. 병합의 경우 이벤트 발생 지역의 높은 트래픽에 의해 해당 지역에서 급증한 후, 싱크에 접근할수록 병합 및 첨부에 의한 트래픽의 감소로 인해 줄어드는 경향을 보이고 있는 것이다.



(a) 병합 (b) 첨부

그림 5. 정책 경향

4. 결론

본 논문은 위치 추적 시스템을 비롯한 실시간 센서네트워크 응용에서 혼잡 제어 기법을 제안하였다. 실시간 응용은 각 이벤트에 대한 데이터가 유일하며, 실시간 제약조건을 가지므로, 비실시간 응용을 기준으로 한 기존의 연구들이 제안한 기법들은 적용할 수 없다. 본 논문은 생성 시간을 기준으로 다수의 데이터를 병합하거나 실시간성을 잃은 데이터들을 지연 후 다른 패킷에 첨부하는 방식을 통해 전체 센서 필드에서 혼잡 문제를 완화시키는 기법을 제안하였다. 병합 기법에 의해 혼잡 지역에서 트래픽 감소를 기대할 수 있으며, 지연 후 첨부 기법에 의해 실시간성을 잃은 데이터들을 포기하지 않으면서 이에 의한 불필요한 트래픽 증가를 막을 수 있다. 시뮬레이션 결과에 의하면 제안 정책은 전송 성공률 면에서 다른 비교 대상들에 비해 우수한 성능을 보이고 있으며, 전송 시간 면에서도 정책을 적용하지 않은 경우에 비해 전반적으로 높은 성능을 보여주고 있다.

참고문헌

- [1] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "Infrastructure tradeoffs for sensor networks," *In Proc. of WSNA 2002*, pp. 49--58, Atlanta, September, 2002.
- [2] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," *In Proc. of MobiHoc '03*, Annapolis, Maryland, June, 2003.
- [3] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," *In Proc. of ACM Sensys 2003*, Los Angeles, USA, November, 2003.
- [4] J. Kang, Y. Zhang, B. Nath, and S. Yu, "Adaptive Resource Control Scheme to Alleviate congestion Control in Sensor Networks," *In Proc. of the 1st Workshop on Broadcast Advanced Sensor Networks (BASENETS)*, San Jose, CA, October, 2004.
- [5] <http://www.tinyos.net>
- [6] <http://www.xbow.com>
- [7] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *In Proc. of ACM Sensys 2003*, Los Angeles, USA, November, 2003.
- [8] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *In Proc. ACM Sensys 2004*, Baltimore, MD, November, 2004.