

센서네트워크를 위한 홉간 순서 번호를 이용하는 신뢰성 있는 데이터 전송 지원 기법

이해준 고양우 이동만
한국정보통신대학교
(haejun, newcat, dlee)@icu.ac.kr

A Reliable Transport Support Using Hop-by-hop Sequence Numbers for a Wireless Sensor Network

Haejun Lee, Yangwoo Ko, Dongman Lee
Information and Communications University

요 약

최근 센서네트워크에서는 코드 분배와 같이 신뢰성 있는 전송을 요구하는 새로운 기능들이 추가 되고 있다. 이를 위한 기존의 프로토콜은 센서노드 간의 높은 전송 실패율 때문에 홉간 손실 복구 기법을 제안하였다. 하지만 기존 프로토콜은 홉간 손실 복구를 위해 중단간 세션의 정보를 사용하기 때문에 모든 중간 노드들이 최근 받은 패킷의 순서 번호 같은 정보를 알고 있어야만 한다. 만약, 세션 중간에 경로가 바뀌어서 새로운 노드가 참가하게 된다면 기존 프로토콜은 잘 동작할 수가 없다. 본 논문은 홉간 순서 번호 기반의 손실 복구 기법인 HRS를 제안한다. 홉간 손실 복구에서는 중단간 세션의 순서 번호 대신에 홉간 순서 번호가 사용되기 때문에 중간 노드는 중단간 세션의 정보 없이도 손실 복구에 바로 참가 할 수 있다. 시뮬레이션을 통해 제안하는 기법이 세션 중간에 경로 변화와 상관없이 잘 동작하고 경로 변화가 없더라도 기존의 프로토콜보다 대부분의 기준에서 우수하다는 것을 보였다

1. 서 론

센서 기기들의 소형화와 대량생산, 그리고 네트워크 기술의 발달로 센서네트워크가 최근 많은 각광을 받고 있다. 환경 모니터링, 야생 동물 관찰, 또는 군사감시 등을 주목적으로 하는 기존 센서네트워크 [1]에서는 센서들이 중복된 결과를 감지하는 경우가 있기 때문에 어느 정도의 데이터 손실은 허용 될 수 있다. 그래서 센서네트워크에서 신뢰성 있는 데이터 전송은 불필요하게 여겨졌다. 하지만 최근 코드의 분배 [2, 3] 나 센서 간의 협동 [4] 등과 같은 좀 더 고차원의 응용이 제안되었는데 이러한 응용을 제공하기 위해서는 신뢰성 있는 데이터의 전송이 필요하다. 예를 들어 코드 분배의 경우 하나의 패킷이라도 손실하게 되면 전체 코드가 사용할 수 없게 되기 때문에 모든 패킷의 신뢰성 있는 전송이 반드시 필요하다.

센서네트워크에서 신뢰성 있는 데이터 전송에 가장 큰 걸림돌은 높은 링크 손실률이다 [5]. 각 링크의 손실률이 매우 높기 때문에 기존의 중단간에서만 이루어지는 신뢰성 있는 전송 방식은 매우 비효율적이다. 이 문제를 풀기 위해서 각 홉에서 중간 노드들이 데이터를 저장하고 손실을 복구하여 링크 손실률을 줄이는 기법이 제안되었다 [2, 6].

기존 기법은 각 홉에서의 손실 복구를 위해 중단간 세션의 정보를 사용하기 때문에 각 중간 노드는 손실 복구를 위해 송신자로부터 마지막으로 받은 패킷의 순서 번호와 같은 중단간 세션 정보를 유지하고 있어야 한다.

서로 다른 센서들로 구성된 센서네트워크 [7, 8]의 경우 센서들은 종류에 따라서 서로 다른 코드를 필요로 할 수 있다. 즉, 코드의 분배가 모든 센서들을 대상으로 하는 것 [2, 3]이 아니라 그 각 코드가 필요한 센서들에게만 멀티캐스트를 통해 이루어지게 된다. 이 경우에 모든 중간 노드가 수신자는 아니다. 그러므로 세션 중간에 경로 변경에 의해 가입하게 된 노드들은 중단간 세션에 대한 정보가 없기 때문에 곧바로 손실 복구에 참여 할 수 없게 된다.

본 논문에서는 센서네트워크에서의 신뢰성 있는 데이터 전송을 위한 홉간 순서 번호 기반의 손실 복구 기법(이하 HRS)을 제안한다. 홉간 손실 복구를 위해 각 홉에서의 실제 패킷의 전달 순서를 나타내는 홉간 순서 번호가 고안 되었다. 패킷의 손실은 이 순서 번호를 기반으로 탐지되기 때문에 새로운 노드가 가입한다 하더라도 바로 손실 복구에 참여 할 수 있다. 또한 각 패킷은 각 홉에서 새로 순서 번호를 부여 받기 때문에 "propagation of a NACK event" [2]와 같은 문제를 발생하

지 않는다.

어떤 크기의 코드라도 손실 없이 효율적으로 전송하기 위해서 HRS는 전송 확인을 고의적으로 지연시킨다. 패킷을 받는 노드는 ACK 회신 타이머를 만들고 그 타이머가 끝나기 전까지 다음 패킷이 오지 않을 경우에만 ACK 패킷을 보낸다. 패킷 손실은 다음 패킷이 있을 경우에는 NACK에 의해 탐지되고 없을 경우에는 ACK에 의해 탐지된다. 이런 방식을 사용함으로써, HRS는 NACK방식의 효율성을 가지면서 어떤 크기의 코드라도 손실 없이 전송할 수 있다.

성능 평가를 위해 ns-2 [9] 기반의 시뮬레이션을 통해 HRS를 PSFQ[2]와 비교하였다. 실험 결과, HRS는 PSFQ보다 모든 면에서 성능이 좋고 세션 중간에 경로 변화가 있다 하더라도 문제없이 동작함을 보였다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구를 소개하고, 3장에서 센서네트워크에서 신뢰성 있는 데이터 전송을 위한 고려사항을, 4장에서는 HRS를 자세히 기술하고, 5장에서는 기존 연구와 성능 비교를 한다. 마지막으로 6장에서 본 연구의 결론과 앞으로의 연구 방향을 제시한다.

2. 관련 연구

센서네트워크에서의 신뢰성 있는 데이터 전송을 위해 여러 연구가 있었다. 노드 간의 높은 링크 손실률을 고려한 홉간 손실 복구 기법[2, 7], 네트워크 내의 모든 노드를 대상으로 한 코드 분배 기법 [3] 등이 제안 되었다.

PSFQ [2]는 홉간 손실 복구에서 "propagation of a NACK event" 문제를 풀기 위해 "pump slowly fetch quickly"기법을 제안하였다. 각 홉에서 손실 복구에 의해 패킷의 순서가 바뀔 수 있다. 이렇게 바뀐 순서로 다음 노드에 전달되게 되면 실제로 손실된 패킷이 없음에도 불구하고 다음 노드는 패킷이 손실되었다고 생각하게 된다. 예를 들면 1, 2, 3, 4의 순서 번호를 가진 패킷을 전송할 때 3번 패킷이 손실되고 4번 패킷에 의해 그 손실이 발견되고 복구되었다고 한다면 그 다음 노드에게는 1, 2, 4, 3의 순서로 전달 되게 된다. 때문에 다음 노드는 4번 패킷을 받았을 때 실제로 그 전 노드는 3번 패킷을 보내지 않았음에도 불구하고 3번 패킷이 손실되었다고 생각하게 된다. 이 문제를 해결하기 위해 PSFQ는 만약 3번 패킷이 손실되고 4번 패킷에 의해 발견 및 복구 될 경우 3번이 복구되어 전달 될 때까지 4번 패킷을 전달하지 않는다. 또한 데이터 패킷의 전달은 상대적으로 천천히 하고 손실의 복구는 빠르게 함으로

써 전체 세션이 너무 지연되는 것을 막는다. 하지만 손실 패킷의 복구 전에는 그 다음 패킷을 전송할 수 없기 때문에 세션이 지연되고 패킷의 전달을 멈추고 있다 하더라도 그 전 노드에서는 계속 패킷이 전달되기 때문에 더 많은 패킷을 버퍼링하여야 하는 문제가 있다.

또한 PSFQ는 중단간 세션의 정보를 이용한 홑간 손실 복구를 한다. 하지만 모든 중간 노드가 중단간 세션의 정보를 가지고 있어야 하기 때문에 세션 중간에 경로가 변하게 되면 제대로 동작할 수 없다.

3. 설계시 고려사항

이 절에서는 본 논문에서 사용된 용어의 정의와 설계할 때 고려해야 할 사항에 대해 설명한다.

3.1. 용어 정의

- 중단간 순서 번호 : 각 패킷에 대해 송신자가 전체 메시지에서 패킷의 순서를 수신자에게 알리기 위해 붙이는 번호.
- 홑간 순서 번호 : 각 중간 노드에서 패킷의 전달 순서에 의해 새로 붙이는 번호. 이 번호는 한 중간 노드와 그것의 다음 홑 노드 사이에서만 의미가 있음.
- 홑간 ACK, NACK과 Repair (hACK, hNACK과 hRepair) : 홑간 손실 복구에 쓰이는 컨트롤 메시지. 중단간 세션의 컨트롤 메시지를 구분하기 위해서 'h'를 덧붙임.
- 홑간 ACK 회신 타이머 : hACK 메시지를 보내는 것을 지연시키고 다음 패킷이 타이머가 끝나기 전에 도착하면 취소되는 타이머.

3.2. 경로 변화에 상관없는 효율적인 신뢰성 보장

센서 노드는 제한된 배터리로 동작하기 때문에 언제나 경로가 변할 가능성이 높다. 특히 모든 노드가 수신자가 아닌 경우 새로 가입하는 중간 노드가 수신자가 아닐 가능성이 높다. 이 노드들은 송신자가 마지막으로 보냈던 패킷의 순서 번호를 모르기 때문에 손실 복구에 바로 참여할 수 없다. 그러므로 홑간 손실 복구는 경로의 변화에 상관없이 잘 동작하여야 한다.

3.3. 실제 패킷이 손실된 곳에서의 손실 복구

중단간 순서 번호를 홑간 손실 복구에 사용할 때의 문제점 중 하나는 실제 각 홑에서의 패킷의 전달 순서와 송신자에서 붙이는 순서가 다를 수 있기 때문에 실제 손실 되지 않은 패킷이 손실된 것으로 잘못 알게 되는 경우가 생긴다는 것이다 [2]. 그렇기 때문에 홑간 손실 복구는 그 홑에서 일어난 패킷의 손실만을 복구하는 것이 바람직하다.

3.4. 적은 컨트롤 부담

센서네트워크에서 가장 중요한 기준은 에너지 효율성이다. 이를 위해서는 손실 복구에 필요한 컨트롤 부담이 최소화 되어야 한다. 일반적으로 NACK기반의 기법들이 효율적이라고 알려져 있지만 모든 메시지가 손실되었거나 마지막 몇 개의 메시지가 손실 되었을 때는 손실을 발견하지 못한다. 단 하나의 패킷이라도 손실 될 경우 전체의 메시지가 쓸모없어지기 때문에 NACK기반의 효율성을 유지하면서 모든 패킷이 손실 없이 전달되어야 한다.

4. HRS

HRS는 크게 홑간 순서 번호 붙이기와 hNACK과 지연된 hACK의 혼합기법으로 이루어진다. 전자는 매 홑에서 패킷에 순서 번호를 부여함으로써 앞장의 칫재와 둘째 요구사항을 만족시킨다. 후자는 hNACK과 hACK메시지를 혼합하여 사용함으로써 NACK기법의 효율성을 가지는 동시에 순수한 NACK기법의 대표적인 문제점인 모든 패킷의 손실이나 마지막 몇 개의 패킷의 손실을 막음으로써 셋째 요구사항을 만족시킨다.

4.1 홑간 순서 번호 붙이기

HRS에서 각 패킷은 매 홑을 지날 때마다 각 홑에서의 전달 순서에 의해 새로 홑간 순서 번호를 부여 받고 패킷을 받는 노드는 바로 전 홑의 노드와 자신 사이의 홑간 순서 번호를 기반

으로 손실된 패킷을 찾는다. 모든 노드는 주위 1홑 노드들에 대한 홑간 순서 번호를 가지고 각 홑에 패킷을 보내게 될 경우에 순서 번호를 1씩 더해가며 패킷에 실어 보낸다. 이 홑간 순서 번호는 중단간 세션에 상관없이 각 노드가 손실 복구에 참여하는 한 계속 유지 된다.

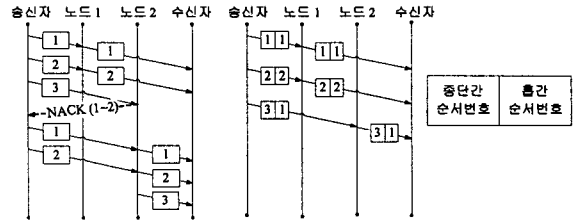


그림 1. 경로 변화가 있을 때의 패킷전달

그림 1은 중단간과 홑간 순서 번호를 사용할 때를 비교한 것이다. 두 경우 모두 셋째 패킷을 보낼 때 경로가 변하여 전의 2개의 패킷은 노드 1에게 보내고 셋째 패킷은 노드 2에게 보내게 된다. 중단간 순서 번호를 사용하는 경우에는, 노드 2가 처음 셋째 패킷을 받았을 때 그 전 노드가 최근에 보냈던 패킷의 순서 번호를 모르기 때문에 처음 2개의 패킷은 손실된 것으로 간주한다. 홑간 순서 번호를 사용하게 되면, 노드 2가 둘째 패킷을 받게 되더라도 전 노드와 노드 2 사이의 홑간 순서 번호가 1 이기 때문에 그전에 패킷의 손실이 없었다는 것을 정확히 알 수 있다.

홑간 순서 번호의 공백은 실제 그 홑에서 손실된 패킷을 가리키는 것이기 때문에 "propagation of a NACK event" [2] 문제 같은 불필요한 NACK을 발생시키지 않는다.

4.2 hNACK과 지연된 hACK의 혼합 기법

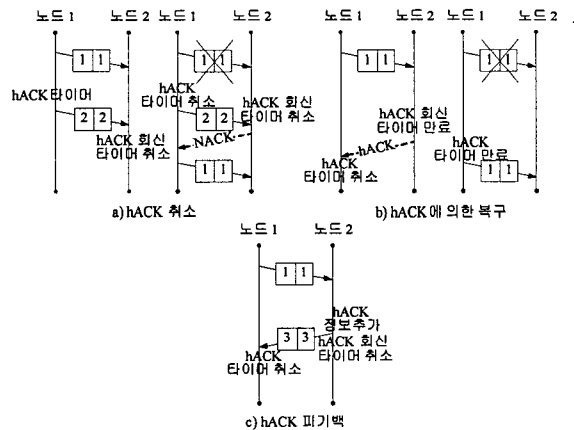


그림 2. 지연된 hACK기법

NACK기반의 손실 복구 기법은 효율적인 반면에 모든 메시지가 손실되었거나 마지막 몇 개의 패킷이 손실되었을 경우 손실을 발견하지 못하는 결정적 단점을 갖고 있다. NACK기반에서는 성공적으로 전달된 다음 패킷에 의해 이전 패킷의 손실이 탐지된다. 그렇기 때문에 이런 문제점을 피하기 위해서는 가장 마지막 패킷은 반드시 전달이 되어야 한다.

HRS는 기본적으로 순서 번호의 공백으로 손실된 패킷을 발견하는 NACK기반이지만 마지막 패킷의 전달을 보장하기 위해 hACK 타이머와 hACK 회신 타이머를 사용한다. 이것은 그림 2에 나타나 있다. 각 중간 노드에서 패킷을 보낼 때 hACK 타이머를 만들고 그 패킷을 받은 노드는 hACK 회신 타이머를 만든다. 그림 2의 a)에서 같이 만약 두 개의 타이머가 끝나기

전에 다음 패킷이 전달되면 두 타이머는 취소된다. 이전 패킷이 손실되었다 하더라도 다음 패킷에 의해 손실이 발견될 수 있기 때문이다. 그림 2. b)와 같이 다음 패킷이 없다면 패킷을 받은 노드가 hACK 회신 타이머가 끝날 때 hACK 패킷을 보내기 때문에 이것을 가지고 이전 패킷의 손실 복구 할 수 있다. 만약 그림 2. c)와 같이 다른 방향에서 hACK 회신 타이머가 끝나기 전에 데이터 패킷이 전달된다면 이 패킷에 hACK에 대한 정보를 실어 보냄으로써 hACK 패킷을 줄일 수 있다.

이와 같이 여러 개의 패킷을 보낼 때 대부분의 hACK 패킷은 억제 되고 마지막 몇 개의 패킷은 hACK에 의해 신뢰성 있게 전달되기 때문에 HRS는 메시지 크기에 상관없이 효율적으로 메시지 전달의 신뢰성을 보장 할 수 있다.

5. 성능 평가

5.1 시뮬레이션 환경

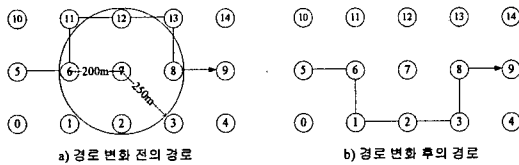


그림 3. 센서노드의 배치

시뮬레이션은 ns-2[9]를 사용 하였고 PSFQ[2]와 비교하였다. 그림 3은 시뮬레이션을 위한 센서 노드의 배치와 경로를 나타낸다. 그림 3의 a)와 b)는 경로 변화전과 후의 경로이고 경로 변화 시점은 임의로 선택하였다. 라우팅 프로토콜에 의한 부작용을 방지하기 위해 소스 라우팅이 사용 되었다. IEEE 802.11이 MAC 프로토콜로 사용 되었고 MAC 레이어에서의 손실 복구는 사용하지 않았다. 송신자와 각 노드에서의 전송률은 초당 10패킷이고 총 100개의 패킷이 전송되었다. 최종 값은 총 10번의 시뮬레이션 결과의 평균을 사용하였다.

5.2 시뮬레이션 결과

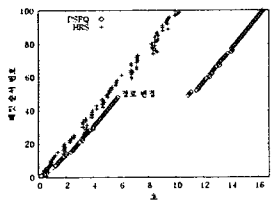


그림 4. 경로 변화가 있을 때의 패킷 도착순서

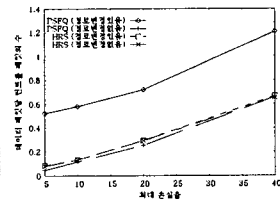


그림 5. 컨트롤 오버헤드

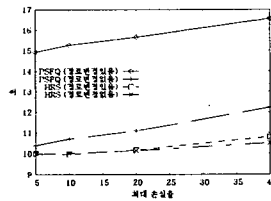


그림 6. 종료 시간

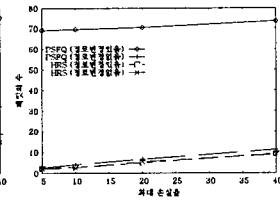


그림 7. 최소 버퍼 요구

그림 4는 51번째 패킷을 보낼 때 경로 변화가 있을 때의 패킷 전달 순서와 시간을 나타낸 것이다. HRS는 경로변경에 영향을 거의 보이지 않고 중간노드에서 중단간 순서번호와 상관없이 패킷들을 전달하기 때문에 패킷들의 도착순서가 상당히 뒤바뀌는 편이다. PSFQ의 경우 경로가 변경되었을 때 한동안 제대로 동작하지 않는 것을 볼 수 있다. 이것은 경로 변경 후에 지금까지의 모든 패킷을 다시 복구해야하기 때문이다.

그림 5는 컨트롤 오버헤드를 나타낸다. 경로변경이 없고 순

실률이 낮을 때는 PSFQ가 근소한 차이로 적는데 이것은 HRS는 최소한 마지막 패킷에 대해 hACK을 쓰기 때문이다. 하지만 손실률이 높아질수록 PSFQ는 더 많은 "proactive NACK" [2]을 쓰게 되므로 차이는 줄어들고 게다가 PSFQ는 모든 패킷이 손실 되었을 경우 신뢰성을 보장하지 못한다. 경로 변경이 있을 땐 HRS는 거의 차이를 보이지 않지만 PSFQ는 이전 모든 패킷을 복구해야 하기 때문에 큰 차이를 보인다.

그림 6은 세션의 종료 시간을 나타낸다. PSFQ는 중간 노드에서 이전 패킷이 손실될 경우 복구하여 전달하기 전에는 다음 패킷을 전달하지 않기 때문에 손실률이 높아질수록 종료시간이 상대적으로 많이 지연된다. HRS의 경우는 패킷 전달을 멈출 필요가 없기 때문에 PSFQ보다 일찍 종료되고 손실률이 커져도 PSFQ만큼 지연되지 않는다. 경로변경이 있을 경우 HRS는 차이가 거의 없는 반면 PSFQ는 컨트롤 오버헤드의 경우와 같은 이유로 많이 느려진다.

그림 7은 각 프로토콜이 각 시뮬레이션 시나리오에서 사용하는 최소한의 버퍼의 크기를 나타낸다. PSFQ에서는 이전 패킷을 복구하는 동안 다음 패킷들을 전달할 수 없지만 패킷들은 이전 노드로부터 계속 전달되어져 오기 때문에 자연히 요구하는 버퍼의 크기가 늘어나고 이것은 경로변경이 있을 경우 더 심해진다. HRS의 경우는 중간에 전달을 멈출 필요가 없기 때문에 경로 변경에 상관없이 가장 작은 버퍼를 요구한다.

6. 결론

본 논문에서는 센서 네트워크에서의 신뢰성 있는 데이터 전송을 위한 홉간 순서 기반의 손실 복구 기법을 제안하였다. 기존의 중단간 순서 기반의 홉간 손실 복구 기법에 비해 각 홉마다 새롭게 홉간 순서 번호를 부여함으로써 세션 중간에 경로가 변해서 새로운 중간 노드가 참가한다고 하더라도 문제없이 동작한다. ACK 메시지를 지연시키고 억제시킴으로써 NACK기반의 효율성을 지키면서 메시지 크기에 상관없이 전송할 수 있다. 시뮬레이션을 통해 경로 변화에 상관없이 잘 동작하고 경로가 변하지 않더라도 대부분의 측면에서 PSFQ보다 나은 성능을 보인다는 것을 입증하였다.

향후 hACK 회신 타이머 값에 따른 성능 차이의 분석과 실제 센서네트워크 테스트베드에서의 실험을 계획 중이다.

참고 문헌

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, *Wireless sensor networks: A survey*, in Computer Networks, vol. 38, pp. 393-422, 2002.
- [2] C.-Y. Wan, A. T. Campbell, *PSFQ: A reliable transport protocol for wireless sensor networks*, in proc. of ACM WSN'02, Sept. 2002.
- [3] Levis, N. Patel, D. Culler, and S. Shenker, *Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks*, in proc. Of USENIX/ACM NSDI'04, Mar. 2004
- [4] Liu, J.; Chu, M.; Liu, J. J.; Reich, J. E.; Zhao, F, *State-centric programming for sensor and actuator network systems*, in IEEE Pervasive Computing. vol. 2, no. 4, pp. 50-62, Oct./Dec. 2003
- [5] Jerry Zhao, Ramesh Govindan, *Connectivity Study of a CSMA based Wireless Network*, Technical Report TR-02-774, USC/ISI, Los Angeles, CA, 2002.
- [6] F. Stann and J. Heidemann, *RMST: Reliable data transport in sensor networks*, in proc. of IEEE SNPA'03, May. 2003.
- [7] E. Welsh, W. Fish, P. Frantz, *GNOMES: A Testbed for Low-Power Heterogeneous Wireless Sensor Networks*, in IEEE ISCAS, 2003
- [8] L. Subramanian and R. H. Katz, *An Architecture for Building Self Configurable Systems*, in proc. IEEE MobiHoc, August 2000.
- [9] *The Network Simulator - ns-2*. <http://www.isi.edu/nsnam/>