무선 센서 네트워크를 위한 신속한 코드 전송 기법

이한선° 정광수
광운대학교 전자공학부
hslee°@adams.kw.ac.kr, kchung@kw.ac.kr

The Fast Code Propagation Scheme for Wireless Sensor Networks

Hansun Lee°, Kwangsue Chung
School of Electronics Engineering, Kwangwoon University

요 약

무선 센서 네트워크를 구성하는 센서 노드는 한번 배치되면 사람의 간섭 없이 오랜 기간 동안 동작하는데 실행중인 소프트웨어를 수정 또는 추가를 할 필요가 있다. 그러나 센서 노드를 회수하기 어려운 경우가 있기 때문에 원격 코드 업데이트 기법이 필요하게 되고, 이를 위한 신뢰성 있는 코드 전송 프로토콜에 대한 연구가 활발하게 진행되고 있다. 하지만 신뢰성만을 고려한 코드 전송 프로토콜은 코드를 안정적으로 전송하기만을 고려하기 때문에 코드를 신속하게 전송한다는 관점에 대한 고려가 부족하다는 한계를 갖는다. 그 결과 긴 코드 전송시간에 의해 불필요한 에너지 소모를 발생함으로써 센서노드의 에너지 효율을 저하시키게 된다. 본 논문에서는 기존의 코드 전송 프로토콜들이 가지는 한계를 극복하는 FCPP(Fast code propagation protocol)을 제안하였다. FCPP는 신뢰성 있는 전송뿐만 아니라 신속함을 고려한 접근 방법을 제시하고 있다. 새로 제안한 알고리즘은 RTT기반의 전송률 조절과 NACK 억제 기법으로 네트워크 상태를 반영한 전송률 조절과 에러복구에 의한 불필요한 전송지연을 피하도록 하여 네트워크의 사용률을 최대화하여 신속한 코드 전송을 가능하게 한다. 또한 ns-2 시뮬레이터를 이용한 실험을 통해 제안한 FCPP가 센서 네트워크의 코드 전송에서 신뢰성 및 신속함을 모두 만족시킬 수 있음을 확인하였다.

1. 서 론

최근 정보통신 기술의 비약적인 발전으로 기존 계산기로써의 컴퓨터는 정보단말로써 발전하여 우리의 생활에 밀접한 영향을 주고 있다. 정보통신 기술의 진보는 유비쿼터스 컴퓨팅(Ubiquitous computing)이라는 새로운 정보통신 혁명을 야기하게 되었고, 사회 발전의 흐름과 끊임없이 환경을 인간 친화적으로 바꾸고 싶어 하는 인간의 욕구와 맞물려 무선 센서 네트워크의 필요성이 제기되고 있다.

무선 센서 네트워크란 센서가 달려 있어 센싱이 가능하고 센싱된 정보를 가공할 수 있는 프로세서가 달려 있으며 이를 전송할 수 있는 무선 송수신기를 갖춘 소형 장치, 즉 센서 노드로 구성된 네트워크를 의미한다. 기존의 네트워크와 다르게 의사소통의 수단이 아니라 다수의 센서 노드를 이용하여 환경의 변화, 수질 오염, 지진 활동, 건물의 구조적 상태 등에 대한 정보를 수집하는 것을 목적으로 한다[1].

센서 노드는 한번 배치되면 사람의 간섭 없이 오랜 기간 동안 동작하지만 수집된 환경 정보가 완벽하지 않아 회수하지 않고는 완벽한 동작의 수행이 불가능한 경우가 존재한다. 그러나 실제로는 나무 꼭대기처럼 물리적으로 닿기 힘든 곳이나 둥지 안과 같이 센서 노드를 회수하는 과정에서 정보 수집에 피해를 주게 되는 다수의 시나리오가 있다. 이런 어려움에도 실행중인 소프트웨어를 수정 또는 추가 할 필요가 있기 때문에 원격 코드 업데이트 기법이 필요하게 된다. 원격 코드 업데이트는 무선 센서 네트워크의 확장성을 제공하고 소프트웨어의 유지보수와 디버깅을 가능하게 한다. 부가적으로, 대규모의 네트워크에 서도 멀티 홉 전송기법을 사용하여 인위적인 코드 업데이트 과정을 자동화할 수 있다[2].

* 본 연구는 한국과학재단 특정기초연구[R01-2005-0000-10934-0(2005)]의 지원에 의해 수행되었음.

원격 코드 업데이트 과정에서 패킷 손실은 코드 업데이트의 실패를 야기하기 때문에 신뢰성 있는 전송 프로토콜이 필요하게 되고, 이를 위한 연구가 활발하게 진행되고 있다[2~7]. 이러한 연구들은 신뢰성 있는 코드 전송을 위해 손실된 패킷에 대한 안정적인 재전송을 목적으로 하지만 코드 업데이트에 소모되는 시간에 대한 고려가 부족하다는 한계를 갖는다. 모든 센서 노드는 원격 코드 업데이트 과정에서 동작 상태로 에너지를 소모하게 되므로 긴 코드 업데이트 시간은 많은 에너지를 소모하여 네트워크의 수명을 단축시키게 된다. 이러한 문제점을 개선하기 위해 신속한 코드 업데이트에 대한 연구가 요구된다.

본 논문에서는 무선 센서 네트워크 환경에서 신속하고 신뢰적인 코드 전송을 위한 새로운 프로토콜인 FCPP(Fast code propagation protocol)를 제안한다. FCPP는 신속한 코드 전송을 위해 네트워크 상태에 따른 전송률 조절과 NACK(Negative acknowledgment) 기반의 에러복구에 의한 불필요한 전송지연을 피하도록 설계되었다. 그럼으로써 신뢰성 있는 코드 전송뿐만 아니라 신속한 코드 전송을 가능하게 한다.

본 논문의 2장에서는 무선 센서 네트워크 환경에서 코드 전송을 위한 고려사항과 관련 연구에 대하여 기술하였고, 3장에서는 기존 연구들의 단점을 개선하기 위해 새롭게 제안한 FCPP의 알고리즘에 대해 기술하였으며, 4장에서는 시뮬레이터를 이용하여 제안한 알고리즘의 성능을 검증하고 기존 코드 전송 프로토콜과의 성능 비교 결과를 기술하였다. 마지막으로 5장에서는 결론을 맺었다.

2. 관련연구

무선 센서 네트워크에서 코드 업데이트를 위해 전송된 코드 패킷의 일부분이 손실될 경우 나머지 코드 패킷은 쓸모없게 되기 때문에 신뢰성 있는 전송 프로토콜이 요구된다. 현재 유선 네트워크에서 사용되는 신뢰성 있는 전송 프로토콜인

TCP(Transmission control protocol)는 패킷 손실의 원인을 네트워크의 혼잡(Congestion)으로 보고 혼잡제어(Congestion control)에 초점을 두고 있다. 그러나 무선 센서 네트워크에서 패킷 손실은 무선 채널간의 간섭(Interference)이나 낮은 전력에 의해 발생하기 때문에 이를 고려한 신뢰성 있는 전송 프로토콜이 필요하게 된다.

PSFQ(Pump slowly, fetch quickly)[3]는 NACK 기반의 신뢰성 있는 전송 프로토콜로 데이터 저장을 통해 순차적인 패킷 전송을 보장한다. 느리게 데이터를 전송(Pump slowly)함으로써 손시로딘 패킷을 복구하기 위한 충분한 시간을 보장하고, 빠르게 손실된 패킷을 복구(Fetch quickly)함으로써 신뢰적인 재전송을 한다.

Deluge[4]는 코드를 페이지 단위와 패킷 단위로 분할하여 전송하며 전체 코드의 일부만 가진 노드가 코드를 전송하는 소스 노드가 되어 동작한다. 분할된 코드 패킷은 파이프라인(Pipelining)으로 전송되어 코드 전송 시간을 감소시킨다.

XNP는 무선 센서 네트워크를 위한 OS(Operating system)인 TinyOS에 포함되어 있으며 TinyOS의 응용들을 무선 환경을 통해 원격으로 코드를 업데이트하기 위해 개발되었다. XNP는 다일 홉을 통한 코드 전송만 지원하기 때문에 센서 노드들이 배치된 무선 센서 네트워크 환경에서 실제로 사용할 수 없다.

3. FCPP(Fast Code Propagation Protocol)

무선 센서 네트워크 환경에서 코드 전송을 우한 대다수의 연구들은 신뢰성 있는 코드 전송에 초점을 두고 있다. 특히 기존 연구에서 PSFQ는 안정적인 전송을 위해, 길고 고정된 패킷 전송간격을 사용하여 에러복구를 위한 시간을 보장하기 때문에 긴 코드 전송시간을 가지게 된다. 그러나 모든 센서 노드는 코드를 전송하는 동안 에너지를 소모하기 때문에 긴 코드 전송시간은 불필요한 에너지 소모를 발생함으로써 센서노드의 에너지 효율을 저하시키게 되는 단점이 있다.

본 논문에서는 NACK 기반의 환경에서 엿듣기(Overhearing)를 통한 전송률 조절 알고리즘과 기존의 NACK 기법의 문제점을 해결하기 위한 NACK 억제 기법을 사용하여 네트워크 상황에 따른 전송률 조절과 불필요한 전송지연을 피하여 신속한 코드 전송을 수행하는 FCPP를 제안한다.

3.1 전송률 조절 알고리즘

FCPP는 네트워크의 상태에 따라 전송률을 조절하기 위해 RTT(Round Trip Time)를 기반으로 전송간격을 조절한다. ACK 기반의 전송 프로토콜에서는 데이터 패킷을 전송하고 ACK 패킷을 반송하는 시간의 차이로써 RTT를 계산하게 되는데, NACK 기반의 전송 프로토콜에서는 손실된 패킷에 대해서만 NACK를 수신할 수 있기 때문에 ACK 기반의 전송 프로토콜과 같은 RTT 측정 방법이 불가능하다. 그렇지만 무선 환경에서는 이웃 노드가 전송하는 패킷을 엿듣기가 가능하기 때문에 노드에서 패킷을 전송한 시간과 이웃 노드에서 해당 패킷을 재전송하는 시간의 차를 이용하여 RTT를 측정할 수 있다.

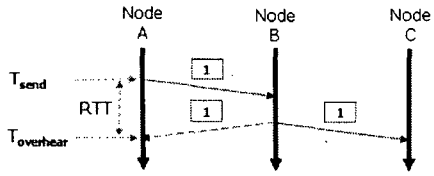


그림 3.1 엿듣기를 통한 RTT 측정

$$RTT_{estimate} = T \tag{3.1}$$

$$RTT_{n+1} = \alpha \times RTT_n + (\alpha - 1) \times RTT_{estimate} \tag{3.2}$$

$$T_{interval} = h \times \frac{RTT}{2} \approx h \times OTT \tag{3.3}$$

계산된 RTT를 기반으로 전송률을 조절하기 위해 식 3.3과 같이 홉수 h에 RTT의 1/2을 곱한 값으로 전송간격을 위한 전송 타이머의 값을 설정한다. RTT의 1/2은 OTT(Oneway trip time)로, 한 노드에서 다음 이웃 노드까지 패킷을 전송하는데 걸리는 시간을 의미하게 되고 홉수 h를 곱함으로써 h 홉까지 패킷을 전송하는데 걸리는 시간을 나타낸다. 따라서 식 3.3과 같은 전송간격으로 패킷을 전송함으로써 h 홉까지 패킷을 전송한 후 다음 패킷을 전송하도록 한다.

3.2 NACK 억제 기법

NACK 기반의 전송 프로토콜은 손실 이벤트 전달을 방지하기 위해 순차적인 전송을 보장하고 불필요한 전송 지연을 유발하게 된다. FCPP는 불필요한 전송 지연을 제거하여 코드 전송을 신속하게 수행하기 위해 비순차적으로 코드를 전송하게 한다. 비순차적으로 코드를 전송할 경우 손실 이벤트가 이웃 노드로 전달되기 때문에 이를 방지하기 위한 방법으로 NACK 억제 기법을 제안한다.

NACK 억제 기법은 패킷을 이웃 노드로 전송할 때 노드에서 순차적으로 수신을 희망하는 패킷의 순차번호를 명시한다. 이 번호는 노드에서 손실된 패킷 없이 완전하게 받은 패킷의 번호를 의미하고 이웃 노드에게 손실된 패킷을 복구할 수 있는 최대 패킷 번호를 알려주는 기능을 한다. 노드에서 패킷을 비순차적으로 수신하였을 경우, 손실된 패킷을 복구하기 위한 동작을 하게 된다. 패킷의 희망 순차번호가 노드에서 수신을 희망하는 순차번호보다 크게 되면 해당 패킷을 전송한 노드에 손실된 패킷을 저장하고 있기 때문에 NACK를 전송하여 손실된 패킷을 복구하게 된다. 패킷의 희망 순차번호가 노드에서 수신을 희망하는 순차번호보다 작다면 해당 패킷을 전송한 노드는 손실된 패킷을 저장하지 않고 있기 때문에 NACK를 전송하지 않는다.

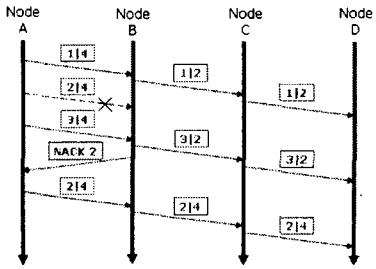


그림 3.2 NACK 억제 기법의 동작

4. 실험 및 성능평가

본 장에서는 새로 제안한 FCPP의 성능 평가를 위해 LBNL(Lawrence berkely national laboratory)의 ns-2(Network simulator)[5]를 사용하여 다양한 실험을 수행하였다.

4.1 실험 환경

본 실험은 100m × 100m 크기의 네트워크 구역을 가정하고, 20m 간격으로 5 × 5 그리드 토폴로지를 구성하였다. 코드 전송 프로토콜로 제안한 FCPP와 성능을 비교하기 위한 기존 프로토콜로 PSFQ를 사용하였다. BS에서는 2.5Kbytes의 코드를 50Bytes 씩 50개의 패킷으로 나누어 전송한다.

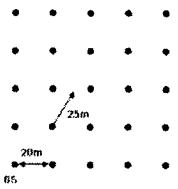


그림 4.1 실험 환경

무선 구간은 ns-2의 에러 모델을 사용하여 무선 링크에서 송수신되는 패킷에 대하여 일정하게 0~50%의 패킷 에러율을 적용하였고, 25m의 전송범위와 2Mb의 대역폭을 갖는 Simple CSMA/CA(Carrier sense multiple access/collision avoidance) MAC(Media access control)으로 설정하였다.

4.2 성능실험

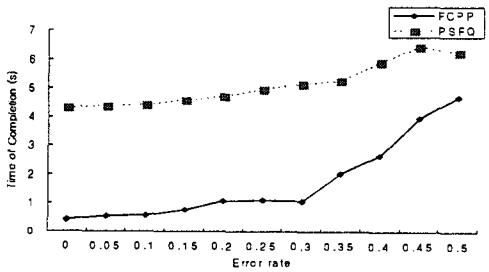
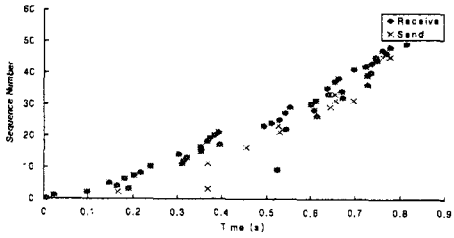


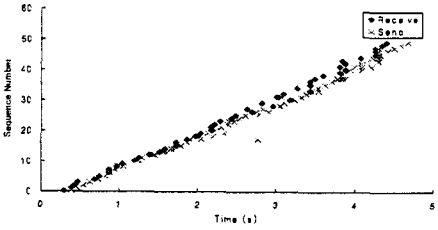
그림 4.2 패킷 에러율에 따른 코드 전송시간 비교

FCPP는 네트워크 상황에 따른 전송률 조절과 NACK 억제 기법으로 코드 패킷을 신속하게 전송하도록 한다. 따라서 기존의 코드 전송 프로토콜인 PSFQ와 코드 전송을 완료하는 시간을 비교함으로써 코드 전송의 신속성을 검증 할 필요가 있다. 그림 4.2는 FCPP와 PSFQ에서 패킷 에러율에 따른 코드 전송시간을 보여준다. 패킷 에러율이 30% 이하일 경우, FCPP의 코드 전송시간은 PSFQ의 20% 정도의 전송시간을 가진다. FCPP가 네트워크 상태에 따라 전송률을 조절하고 에러 복구를 위한 전송 지연시간을 제거함으로써 보다 빠르게 코드를 전송할 수 있게 하기 때문이다. 하지만 패킷 에러율이 30% 이상의 경우 FCPP의 코드 전송시간이 급격하게 증가하는 것을 볼 수 있다. 노드에서 마지막 코드 패킷이 손실되었을 경우 비순차적이 패킷수신이 일어나지 않아 에러 복구를 하지 못하고 에러 복구 타이머가 완료되어 NACK를 전송하여 손실된 코드 패킷을 재전송 받기 때문에 패킷 에러율이 증가 할수록 코드 전송을 완료하는 시간이 길어지게 된다.

순차번호를 분석하기 위해서 5 × 5 그리드 토폴로지의 정 가운데 노드에서 시간에 따른 순차번호의 변화를 확인해 보았다. 그림 4.3에서 FCPP와 PSFQ가 코드 패킷전송을 완료한 시간은 각각 0.81초와 4.68초이다. 그림 4.3(a)은 FCPP에서 시간에 따른 순차번호를 분석한 것으로 NACK 억제 기법을 사용하여 에러 복구에 의한 전송 지연을 제거하기 위해 수신된 패킷을 바로 전송하여 비순차적으로 패킷을 전송하는 것을 볼 수 있다. 그림 4.3(b)은 PSFQ에서 시간에 따른 순차번호를 분석한 것으로 패킷이 손실되었을 경우 손실된 패킷을 복구하기 전까지 패킷 전송을 하지 않는 것을 확인 할 수 있다.



(a)



(b)

그림 4.3 시간에 따른 순차번호
(a) FCPP (b) PSFQ

5. 결론 및 향후과제

본 논문에서는 기존의 코드 전송 프로토콜들이 가지는 한계를 극복하는 FCPP(Fast code propagation protocol)을 제안하였다. FCPP는 신뢰성 있는 전송뿐만 아니라 신속함을 고려한 접근 방법을 제시하고 있다. 새로 제안한 알고리즘은 RTT기반의 전송률 조절과 NACK 억제 기법으로 네트워크 상태를 반영한 전송률 조절과 에러복구에 의한 불필요한 전송지연을 피하도록 하여 네트워크의 사용률을 최대화하여 신속한 코드 전송을 가능하게 한다.

시뮬레이터를 이용한 실험을 통해서 FCPP의 기본적인 신뢰성 있는 코드 전송 성능을 검증하고 기존의 PSFQ(Pump slowly, fetch quickly)와 성능 비교를 통하여 신속한 코드 전송뿐만 아니라 안정적인 전송 측면에서도 대등한 결과를 보였다. 실험결과를 통해 제안하는 FCPP가 센서 네트워크의 코드 전송에서 신뢰성 및 신속함을 모두 만족시킬 수 있음을 확인하였다.

참고문헌

[1] 채동현, 한규호, 임경수, 안순신, "센서 네트워크의 개요 및 기술동향," 정보과학회 논문지, 제 22권, 제 12호, pp. 5-12, 2004. 12.

[2] T. Stathopoulos, J. Heidemann and D. Estrin, "A Remote Code Update Mechanism for Wireless Sensor Networks," Technical Report CENS-TR-30, November 2003.

[3] C. Wan, A. Campbell, and L. Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," In Proceedings of ACM WSNA, pp. 1-11, July 2002

[4] J. Hui and D. Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale," In Proceedings of ACM SenSys, pp. 81-94, November 2004.

[5] The network simulator ns-2, http://www.isi.edu/nanam/ns/