

센서 네트워크에서 크레딧을 이용한 이기적인 노드 처리 방안

(A Mechanism for Handling Selfish Nodes using Credit in Sensor Networks)

최종원 [†] 유동희 [‡]

(Jongwon Choe) (Donghee Yoo)

요약 정보 수집이 목적인 센서 네트워크에서 자신의 전력을 절약하기 위해 다른 노드들의 패킷을 의도적으로 전송하지 않는 이기적인 노드가 포함되어 있을 수 있다. 이를 해결하기 위해 크레딧을 이용한 패킷 전송에 따른 지불 방법을 제안하였다. 센서 노드들이 패킷을 전송할 때, 크레딧을 사용하게 하고, 다른 노드들의 패킷을 전송하는 것을 통해 크레딧을 얻을 수 있게 하였다. 크레딧의 신뢰성을 보장하기 위해서 싱크 노드와 서버의 역할을 통합하고 piggybacking 기법을 이용하여 추가적인 리포트 메시지를 송신하지 않도록 하였다. 또 위치상의 이유로 다른 노드들이 전송하는 패킷을 받지 못하여 크레딧을 얻지 못하는 노드를 찾아 크레딧을 추가 지급함으로써 부당하게 대우를 받는 문제를 해결하였다. 제안된 방법의 성능 평가를 위하여 네트워크 시뮬레이션(ns2)를 이용하여 구현하였으며, 실험 결과, 이기적인 노드를 처리하기 위해 패킷 지불 방식만을 이용한 것보다 패킷 전송률이 높게 유지되었고, 시뮬레이션 시간 내 도착된 패킷의 총 개수 또한 증가하였다. 그리고 센서 노드의 전력 소모량이 줄어들어 노드들이 살아 있는 시간이 길어짐을 확인할 수 있었다.

키워드 : PIFA, Selfish node, 패킷 지불 방식, 크레딧, M-PIFA

Abstract The purpose of sensor network is gathering the information from sensor nodes. If there are selfish node that deliberately avoid packet forwarding to save their own energy, the sensor network has trouble to collect information smoothly. To solve this problem we suggest a mechanism which uses credit payment schema according to the amount of forwarding packets. Sensor nodes use credits to send their own message and they forward packets of other sensor nodes to get credits. To offer authenticity we combined the roles of sink node and server, also we used piggybaking not to send additional report message. The packet trace route is almost fixed because sensor node doesn't have mobility. In this case, it happens that some sensor nodes which don't receive forwarding packets therefore they can't get credit. So, we suggested the way to give more credits to these sensor nodes. Finally, we simulated the suggested mechanism to evaluate performance with ns2(network simulator). As a result, packet transmission rate was kept on a high rate and the number of arrival packets to sink node was increased. Also, we could verify that more sensor nodes live longer due to deceasing the energy consumption of sensor nodes.

Key words : PIFA, Selfish node, packet payment schema, credit, M-PIFA

· 본 연구는 숙명여자대학교 2007학년도 교내연구비 지원에 의해 수행되었음

[†] 종신회원 : 숙명여자대학교 정보과학부 컴퓨터과학전공 교수
choejn@sookmyung.ac.kr

[‡] 비회원 : 삼성전자 정보통신총괄 무선사업부
dh130.yoo@samsung.com

논문접수 : 2007년 11월 15일

심사완료 : 2008년 1월 16일

Copyright@2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 정보통신 제35권 제2호(2008.4)

1. 서 론

언제 어느 곳에 있어도 센서를 부착하고 있는 주변 사물들을 통해 다양한 정보를 접근, 관리할 수 있는 유비쿼터스 컴퓨팅(Ubiqitous Computing) 기술이 점점 보편화 되고 있다. 이러한 유비쿼터스 환경 구현을 위한 기반 구조로서의 센서 네트워크(Sensor Network)의 기술 개발에 대한 관심이 고조되고 있다[1].

한 지역에 구성된 센서 네트워크에 포함된 센서 노드(이후 노드로 표기함)들은 단일 조직에 의해 관리되는 것이 대부분이었다. 하지만 센서 네트워크 개발에 많은 기업들이 참여하게 되면서 동일한 지역에 여러 기업들의 노드들이 공존하게 되었고, 다양한 목적을 가진 여러 종류의 노드들이 한곳에 분포되어 있는 상황이 생겨나게 되었다. 이 때, 다른 노드들과 협력하지 않고, 자신의 전력 소모를 줄이기 위해 다른 노드들의 패킷을 싱크 노드 방향으로 포워딩 해주지 않는 노드들이 존재할 수 있다. 이러한 이기적인 노드들이 네트워크에 포함되어 있다면, 노드들이 보내는 데이터가 제대로 전달되지 못하기 때문에 노드들의 이기적인 행동을 방지하는 방법이 필요하다[2].

이동성이 있는 애드-혹 네트워크에서 이기적인 노드를 처리하는 방법으로 PPM[3], PTM[3], Sprite[4]와 PIFA[5] 등의 알고리즘이 제안되었다. 이 방법들은 이기적인 노드들을 배제하거나 패킷 포워딩에 자발적으로 참여하게 하기 위해 패킷 전송에 대한 보상 방법을 이용하였다. 다른 노드들의 패킷을 포워딩함으로써 '빈(bean)'이나 '크레딧(credit)'이라고 불리는 가상 머니를 얻고 이를 이용하여 자신의 패킷을 보낼 수 있도록 하는 것이다.

센서 네트워크에서 자신의 전력 소모를 줄이기 위해 다른 노드들의 패킷을 전달해 주지 않는 이기적인 노드들 때문에 네트워크의 효율이 떨어지는 것을 막기 위해 애드-혹 네트워크에서 제안된 PIFA의 아이디어를 센서 네트워크에 적용하여 이기적인 노드들을 처리하는 방법을 제안한다. 하지만, 센서 네트워크와 애드-혹 네트워크의 차이점으로 인해 몇 가지 문제점이 발생한다. 센서 네트워크에서의 노드들은 이동성이 없으므로 노드들의 전력이 전부 소모되지 않는 한, 라우팅 경로가 변하는 경우는 거의 없다. 또, 정보 수집이 목적인 센서 네트워크에서 각 센서 노드들은 서로 간의 통신을 하는 것이 아니라 센서 노드로부터 정보를 받아들이는 특정 노드로만 통신을 한다. 그러므로 센서 네트워크에서의 센서 노드들은 매번 거의 같은 경로를 사용하여 데이터를 전송하게 되며, 패킷 전송 경로에 포함되지 않는 일부 노드들은 다른 노드의 패킷을 전송할 기회를 얻지 못하기 때문에 '크레딧'을 얻을 방법이 없다.

이러한 문제점을 해결하기 위해, 센서 네트워크에서 이용하는 싱크 노드(Sink Node)와 크레딧을 관리하기 위해 사용하는 크레딧 관리 서버의 역할을 하나로 합쳐서 이용하는 방법을 제안한다. 이 방법을 사용하면 각 노드들이 크레딧에 관한 정보를 제공하기 위해 추가적인 패킷을 전송할 필요가 없어 되어 노드들이 별도의 패킷을 전송하는데 사용하는 전력을 절약하고, 네트워크의 오버헤드를 줄일 수 있다. 또한, 이기적인 노드가 아님에도 불구하고 포워딩 패킷을 전송할 기회를 얻지 못해서 자신의 패킷을 전송하기 위해 필요한 크레딧 조차 얻지 못하게 되는 노드에게 크레딧을 추가적으로 지급하는 방법을 통해 문제점을 해결하고자 한다.

노드들이 추가적인 리포트 메시지를 보내지 않게 되어 노드들의 전력이 절약되며, 노드들의 수명 시간 또한 길어져 네트워크가 오래 지속될 수 있었고, 그에 따라 목적지까지 전송되는 전체 패킷의 개수가 증가되는 결과를 시뮬레이션을 통해 얻을 수 있었다. 이기적인 노드가 아닌 노드들이 불이익을 받는 경우를 찾아 이를 해결함으로써 해당 노드가 지속적으로 패킷 전송에 참여할 수 있게 하였다. 그래서 네트워크가 유지되는 동안 보다 다양한 노드들에게 패킷을 받는 결과를 얻을 수 있었고, 이를 통해 센서 네트워크 구성 목적인 노드들의 정보 수집을 보다 효율적으로 하는 것을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 PPM, PTM, Sprite, PIFA의 정의와 동작 원리에 대해 알아본다. 3장에서는 PIFA의 문제점을 살피고 그 문제점을 해결하기 위한 방법을 제안한다. 4장에서는 시뮬레이션을 통하여 본 논문이 제안한 기법이 기존 기법보다 성능이 향상됨을 보인다. 마지막으로 5장에서는 결론을 맺는다.

2. 관련 연구

유선 네트워크와 다른 이동 애드-혹 네트워크(MANET : Mobile Ad hoc Network)환경에서 AODV[6]나 DSR[7]와 같은 라우팅 프로토콜이 이용된다. 이런 라우팅 프로토콜은 모든 노드들이 자발적으로 패킷 포워딩에 참여한다고 가정하고 있다. 그러나 하나의 애드-혹 네트워크가 여러 관리 기구에 속한 노드들로 구성될 경우에 발생할 수 있는 노드들의 이기적인 행동에 제한을 하고자 다양한 프로토콜이 제안되었다[5].

이 절에서는 본 논문에서 제안하는 기법의 기반이 되는 이동 애드-혹 네트워크에서 이기적인 노드 관리 방안을 제안한 PPM과 PTM[3], Sprite[4], PIFA[5]에 대해 기술한다.

2.1 PPM과 PTM

이기적인 노드 제한 방법에 상업적 거래 개념을 처음 도입한 2가지 방법이 있는데 그것은 PPM(Packet

Purse Model)과 PTM(Packet Trace Model)이다[3].

PPM은 패킷 포워딩에 대한 대가로 ‘빈’(bean)이라는 지불 수단을 사용한다. 우선, 패킷을 보내고자 하는 노드는 목적지까지의 ‘빈’의 사용량을 추정한다. 그리고 패킷을 전송할 때, ‘빈’을 같이 보내게 된다. 패킷이 목적지에 도착할 때까지 거치게 되는 노드들은 패킷을 전달해 줄 때, 패킷에 담겨있는 ‘빈’을 가져가게 된다. 이렇게 ‘빈’을 이용하여 패킷을 목적지까지 보내는 방법이 PPM이다.

이 방식의 문제점은 소스 노드가 패킷을 전송할 때, 목적지까지의 ‘빈’의 사용량을 추정해야 한다는 것이다. 이를 제대로 예측하지 못하면 ‘빈’이 남아서 낭비되거나, 모자라서 패킷이 제대로 전달되지 못하게 된다. 그렇기 때문에 도착지까지의 ‘빈’ 소모량을 예측하기 위해서 DSR[2]와 같이 패킷의 전달 경로를 미리 알 수 있는 방식에만 사용될 수 있다.

두 번째 방식은 PTM(Packet Trade Model)이다. 패킷이 전달되는 동안에 패킷에 담겨 있는 ‘빈’을 노드를 거쳐 갈 때마다 계산하는 PPM과 달리 PTM은 목적지에 패킷이 도착한 후, 패킷이 전송되면서 사용된 ‘빈’의 양을 한 번만 계산한다. 따라서, 이 방식은 패킷을 이전 노드(previous node)에서 사서, 다음 노드(next node)에 파는 방식이라고 할 수 있다. 예를 들어, 이전 노드에서 오는 패킷을 1개의 ‘빈’을 주고 사서 다음 노드에게 2개

의 ‘빈’을 받고 파는 형식이다. 결과적으로 목적지 노드가 패킷이 도착할 때까지 지나왔던 노드들이 요구한 ‘빈’을 전부 지불하게 된다.

이 방식의 장점은 패킷을 전송할 때, 필요한 ‘빈’의 양을 미리 예측하지 않아도 된다는 것이다. 하지만, 소스 노드가 패킷 전송에 대해 비용을 지불하지 않으므로, 불필요한 데이터를 보내어 네트워크에 많은 트래픽을 만드는 DoS(Denial of Service) 공격의 대상이 될 수 있다.

위에 설명한 2개의 서비스가 공통으로 가지는 문제점은 각 노드가 사용하는 ‘빈’의 신뢰성 문제이다. ‘빈’의 사용량이 각각의 노드에 저장되기 때문에 사용된 ‘빈’을 다시 사용하거나 패킷을 전송할 때 가져야 할 양보다 많은 양의 ‘빈’을 가져갈 수도 있다. 제안된 PPM이나 PTM에서는 노드가 손상 방지 모듈(tamper resistant security module)을 가지고 있는 것으로 가정하고 있지만, 이런 문제점 때문에 이 서비스들이 잘 사용되지 않고 있다.

2.2 Sprite

Sprite(Simple Cheat-Proof, Credit-based System)는 패킷 전송에 대한 보상으로 크레딧(credit)이라는 가상 머니를 사용하고, 그림 3과 같이 CCS(Credit Clearance Service)라는 서버와 모바일 노드로 구성된다. CCS는 앞에서 말한 PPM과 PTM의 문제점인 ‘빈’의 신뢰성 문제를 해결하고자 만들어진 것으로 MANET 외부와

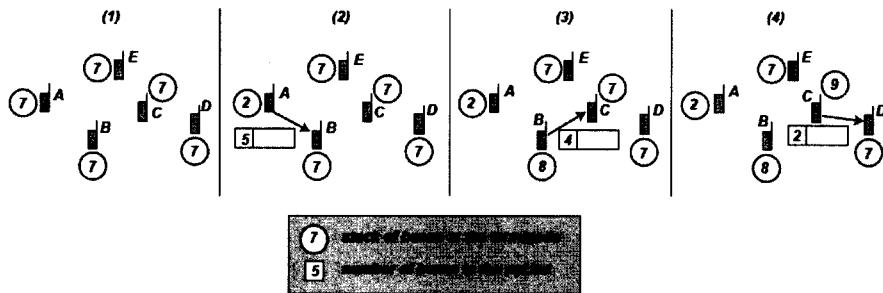


그림 1 The Packet Purse Model

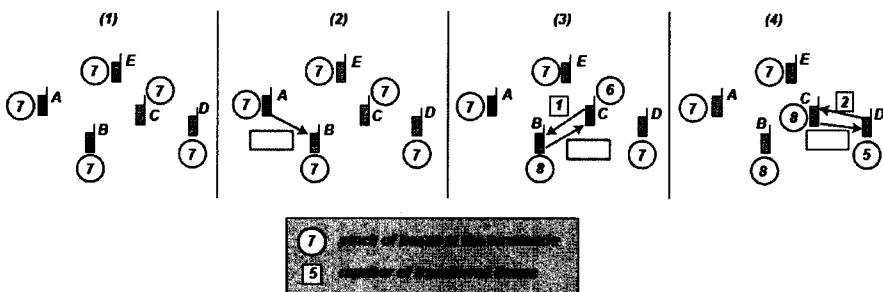


그림 2 The Packet Trade Model

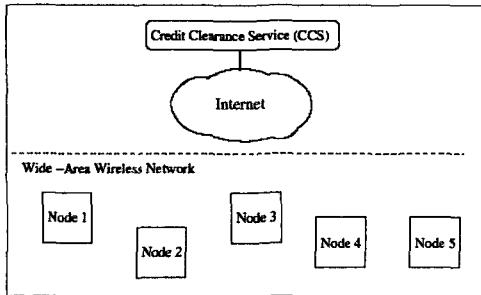


그림 3 Sprite의 구조

연결되어 노드들의 크레딧을 관리하는 서버이다[4].

Sprite에서는 노드가 자신의 메시지를 전송할 때 크레딧을 사용하고, 그 크레딧을 얻기 위해 다른 노드들의 패킷을 전송한다. 이 크레딧에 관한 정보를 관리하는 CCS는 각 노드로부터 관련 정보를 받는다. 각 노드는 다른 노드의 메시지를 전송하는 정보를 콤팩트 플래시(Compact Flash) 카드 같은 저장 장치에 저장해 놓고 빠른 전송이 가능하거나 여유 전력이 있을 때마다 CCS에게 리포트 메시지를 보낸다. 노드는 대역폭과 저장 장치를 절약하여 사용하기 위해 영수증 개념의 작은 메시지(receipt)를 사용한다. CCS는 노드들로부터 받은 영수증을 이용하여 리포트 패킷을 포워딩한 노드의 크레딧은 증가시키고, 패킷을 보내는 노드의 크레딧은 감소시킨다.

더 많은 크레딧을 받기 위해 CCS를 속이는 방법들이 생겨났고, 이것을 방지하기 위해 Sprite는 기만 방지(cheat-proof) 알고리즘을 포함하였다. 이것은 패킷을 보내는 노드에게 목적지까지 가기 위해 필요한 크레딧의 양보다 많은 양의 크레딧을 차감하고 각 노드들의 영수증 내용이 일치하면 나머지를 되돌려 주는 방식으로 동작한다.

Sprite은 크레딧에 관한 정보를 각 노드가 관리하지 않아서 신뢰성을 얻게 되었지만, 각 노드들이 패킷 전송에 대한 내용을 CCS에게 모두 보고해야 하므로 네트워크상의 큰 오버헤드가 발생하게 된다. 그리고 2.1절에서 언급한 PPM과 마찬가지로 소스에서 목적지까지의 전체 경로에 대한 정보가 필요하므로 DSR과 같은 소스 경로 배정 프로토콜에서만 사용할 수 있다는 단점을 가지고 있다.

2.3 PIFA

PIFA(Protocol Independent Fairness Algorithm)는 라우팅 경로를 미리 알 수 있는 소스 라우팅 프로토콜에만 사용할 수 있는 PPM, Sprite와 달리 프로토콜의 종류에 상관없이 사용할 수 있는 방법이다. PIFA는 Sprite와 같이 크레딧을 이용한 패킷 지불 방식을 사용

하고, 이 크레딧에 대한 정보를 저장하는 CDB(Credit DataBase)를 관리하는 CM(Credit Manager)으로 불리는 서버로 구성된다[5].

네트워크상의 모든 노드는 자신이 포워딩한 패킷의 수를 CM에게 리포트 메시지를 이용하여 일정 주기 간격으로 보고하며, CM은 이 메시지를 바탕으로 이 정보가 신뢰할 수 있는지를 검증한다. CM은 보고 메시지에 포함된 정보를 가지고 이를 보고의 신빙성 여부 알아보기 위해 세 가지 검사를 수행한다. 첫 번째, 노드의 출력 패킷의 수와 이웃 노드의 입력 패킷 수가 일치하는지를 확인한다. 두 번째, 노드의 총 입력 패킷 수와 자신이 목적지인 패킷의 수의 차이가 포워딩 해준 패킷의 수가 같은지 검사한다. 세 번째 검사는 이웃 노드로부터 온 출력 패킷 중에서 이 노드로부터 송신된 패킷의 수와 해당 이웃으로부터의 입력 패킷들 중에서 이 이웃에서 송신된 패킷의 수가 같은지 확인하는 것이다. 이 정보들을 가지고 CM은 리포트 메시지들로부터 유추해 낸 토폴로지를 이용해 노드들이 전송한 보고가 신빙성이 있는지, 노드들이 CM에게 거짓 보고를 하는지를 검증한다.

CM은 각 노드가 보내는 리포트 메시지를 통해 포워딩 해준 패킷의 개수와 비례해서 각 노드에게 크레딧을 지급해 준다. 노드는 자신이 메시지를 전송하기 위해서는 크레딧을 사용하게 되고 크레딧을 얻기 위해서 다른 노드들의 패킷을 전송해 주는 자발적인 참여를 하게 되는 것이다. PIFA에서 소스 노드는 자신이 사용하게 될 크레딧의 양을 예측하지 않아도 되므로 자신이 어떤 경로를 통해서 패킷을 전송하게 되는지 알 필요가 없다. 그러므로 프로토콜에 상관없이 이용이 가능하다.

하지만 이 방식 역시 각각의 노드들은 자신의 패킷 포워딩에 대한 정보를 포함한 리포트 메시지를 CM에게 주기적으로 보내야 하기 때문에 네트워크에 오버헤드가 생기게 된다.

3. 센서 네트워크에서 크레딧을 이용한 이기적인 노드 처리방안

이 절에서는 이동성이 없는 센서 네트워크 환경에서 이기적인 노드를 처리하는 방법을 제안하고, 그 때 생기는 문제점을 보완하는 기법을 제안한다.

3.1 센서 네트워크에서 PIFA 적용 시 문제점과 해결방안

센서 네트워크는 센싱(sensing)이 가능하고 수집된 정보를 가공하는 프로세서가 달려있으며 이 정보를 전송하는 소형 무선 송수신 장치 즉, 센서 노드(sensor node)와 이를 수집하여 외부로 내보내는 싱크 노드(sink node)로 구성된다[8].

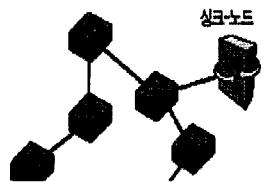


그림 4 센서 네트워크의 구조

센서 네트워크는 애드-혹 네트워크에 비해 제한된 리소스의 단순한 구조와 하나의 목적을 가지며, 데이터 중 심적이라는 특징을 가지고 있다. 이 밖에 많은 차이점이 있지만, 주목할 점은 센서 네트워크의 센서 노드들은 이동성이 없다는 점이다. 또, 센서 네트워크의 센서 노드들은 자신의 목적을 가지고 그와 관련하여 수집한 데이터들을 싱크 노드로 주기적으로 전송하는 임무를 가지고 있다.

이런 센서 네트워크 환경에서도 애드-혹 네트워크와 마찬가지로 자신의 배터리를 절약하기 위해 다른 노드들과 협력하지 않고 다른 노드들이 전송하는 패킷을 포워딩 하지 않는 노드들이 존재할 수 있다. 이런 노드들을 처리하기 위해 2절에서 언급한 애드-혹 네트워크 환경에서 제안된 알고리즘 중 하나인 PIFA을 이용하였다.

3.1.1 센서 네트워크에서 PIFA을 이용할 때의 문제점

본 논문은 PIFA와 마찬가지로 크레딧을 이용하여 패킷 전송에 대한 보상을 해주는 방식을 사용한다. 처음에 일정액의 크레딧을 주고 센서 노드 자신이 생성한 패킷을 전송하면 크레딧을 감소시키고, 다른 노드들의 패킷을 포워딩해주면 크레딧을 증가시킨다. 각 노드들의 크레딧을 관리하기 위해 CM(Credit Manager) 노드라는 서버를 사용한다.

PIFA는 이기적인 노드들이 자발적으로 다른 노드들의 패킷을 포워딩하게 함으로써 노드들의 에너지가 공평하게 소모되도록 하며, 라우팅 프로토콜의 종류와 무관하게 사용될 수 있다는 장점을 가지고 있다. 하지만 크레딧을 관리하는 서버를 사용하여 크레딧을 따로 관리하기 때문에 각 노드들은 크레딧을 획득하기 위해 서버에게 패킷 포워딩에 관한 정보를 포함하는 메시지를 주기적으로 보내게 된다. 이것은 네트워크상의 많은 오버헤드를 만들게 되며, 또한 각 노드의 배터리를 많이 소모하게 만든다.

또한 PIFA는 각각의 노드들이 이동성을 가지고 있는 애드-혹 네트워크 환경에서 이용할 수 있게 구현되었지만 센서 네트워크 환경에서의 대부분의 센서 노드들은 이동성을 가지고 있지 않다. 그렇기 때문에 패킷을 전달하는 라우팅 경로가 거의 변경되지 않는다. 크레딧은 다른 노드들이 전송하는 패킷을 포워딩 해 주는 경우에만 획득할 수 있기 때문에 싱크 노드로 가는 경로에 포함

되지 않는 노드들은 크레딧을 얻을 수 없다. 그러므로 이러한 노드들은 자신이 초기에 받은 크레딧을 다 사용하게 되면, 더 이상 자신의 데이터를 전송할 수 없게 된다. 이기적인 노드가 아님에도 자신의 패킷 전송에 제약을 받게 되는 것이다. 이렇게 되면 노드들이 자신의 배터리가 남아 있음에도 불구하고 더 이상 자신의 역할을 수행할 수 없게 되므로, 센서 네트워크를 구성한 목적인 센서를 통한 데이터 획득을 극대화 할 수 없다.

3.1.2 문제점 해결 방안

이 절에서는 센서 노드와 싱크 노드로 이루어진 센서 네트워크에서 PIFA 알고리즘을 적용했을 때 발생한 문제점을 보완하기 위한 방법을 제안한다.

PIFA를 적용한 센서 네트워크는 센서 노드와 싱크 노드로 이루어져 있으며, 각 노드들의 크레딧 정보를 저장하는 CDB(Credit Data Base)를 관리하는 CM(Credit Manager)를 가지고 있다. 본 논문에서 다루는 센서 네트워크에서는 센서 네트워크에서의 각 센서 노드들이 자신이 획득한 정보를 주기적으로 싱크 노드에게 전송하게 되어 있다. 이러한 특성을 이용하여, 각 노드들이 획득한 정보를 주기적으로 보낼 때, 크레딧에 관한 정보를 추가해서 보내는 것으로 설정했다(piggybacking). 즉, 싱크 노드가 CM의 역할을 포함하며 각 노드들이 수집한 데이터를 모으는 역할을 하는 동시에 노드들의 크레딧을 관리하기 위해 각종 패킷의 개수에 대한 정보를 받는 역할을 한다. 여기서 싱크 노드는 기존 네트워크에 연결되었거나, 센서 노드로부터 받은 정보를 충분히 처리할 수 있는 능력을 갖고 있으며 배터리가 충분한 노드가 선정된다. 이 노드가 각 노드로부터 받은 정보를 통해 크레딧을 계산하고 그에 따라 이기적인 노드들을 걸러내어 블랙리스트를 만들고 이것을 노드들에게 전송하여, 이기적인 노드들로 하여금 강제적으로 패킷 포워딩에 참여하게 하는 CM의 역할도 같이 수행하게 하는 것이다.

보완된 이기적인 노드 처리 흐름도는 그림 5와 같다. 센서 노드들은 블랙리스트를 확인하고(4), 그 리스트에 포함된 노드들이 보내는 패킷은 포워딩 하지 않는다(6). 여기서 PIFA는 일반 노드들이 이기적인 노드의 패킷을 포워딩 해주지 않게 하는 방법으로 블랙리스트에 포함되어 있는 노드들이 소스인 패킷을 드롭(drop)시키는 방식을 사용하였지만, 패킷의 TTL을 1로 변경시키고 전송하게 하는 방식으로 변경하였다. 이렇게 설정한 이유는 고의적으로 패킷을 드롭 시킨 경우와 블랙리스트에 포함된 노드의 패킷을 드롭 시킨 경우를 구분하기 위함이다. 본 논문에서는 해당 노드에게 들어온 패킷과 나가는 패킷의 수를 비교하여 크레딧을 주는데, 블랙리스트에 포함된 노드들의 패킷을 따로 구분하지 않고 드

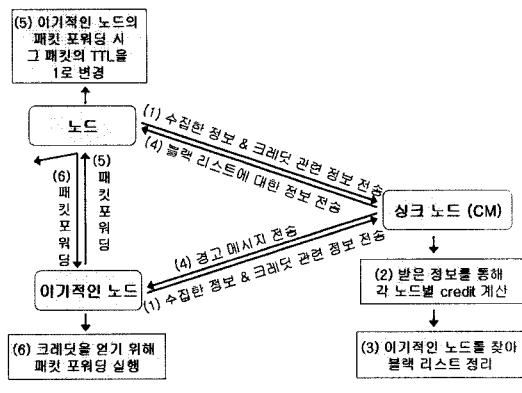


그림 5 보완된 이기적인 노드 처리 흐름도

롭 시킨다면, 해당 노드에서 드롭된 패킷의 수가 증가하여, 이 노드 또한 블랙리스트에 올라가는 상황이 발생될 수 있다. 이를 예방하기 위해 고의적으로 패킷을 드롭시키는 이기적인 노드의 경우와 블랙리스트에 포함된 노드들의 패킷을 드롭시키는 경우를 구분한 것이다.

여기서 각 노드들이 자신의 크레딧을 관리하지 않고 크레딧을 관리하는 서버를 따로 사용함으로써 크레딧에 대한 신뢰성을 높인다. 노드들이 자신의 크레딧을 각자 관리하면, 패킷을 보내고 크레딧을 감소시키지 않거나 패킷을 포워딩해 주고 많은 양의 크레딧을 증가시키는 등의 속임수를 사용할 수 있다. 이를 방지하기 위해 CM에 의해서만 크레딧이 계산되고 그 값을 저장하게 한다.

CM과 싱크 노드를 합쳐서 싱크노드가 CM 역할을 포함하게 하고, 싱크노드에게 주기적으로 보내는 데이터에 크레딧 계산을 위한 정보를 추가적인 필드로 포함시킴으로써, 노드들이 추가적인 리포트 메시지를 보내지 않게 하였다. 이를 통해 노드들의 전력을 절약할 수 있고, 네트워크의 트래픽을 줄일 수 있다.

크레딧 관련 정보는 추가적인 필드를 이용하여 보내지게 되는데, 그 형식은 그림 6과 같다.

싱크 노드는 그림 6에 포함된 정보를 바탕으로 이웃 노드에게 보낸 패킷 수의 합 ($\sum O$)에서 자신이 보낸 패킷 수 ($\sum S$)의 차이와 비례하여 크레딧을 얻을 수 있게 계산한다.

$$\text{credit} \propto \sum O - \sum S \quad (1)$$

센서 네트워크 환경에서의 노드들은 이동성을 가지고 있지 않기 때문에 패킷 전송 경로는 거의 변경되지 않는다. 이럴 경우, 각 노드와 싱크 노드 사이의 경로에 포함되지 않는 노드들이 생기며, 이런 노드들은 포워딩되는 패킷을 거의 받지 못한다. 따라서, 다른 노드의 패킷을 포워딩 해줄 수 있는 기회를 얻지 못해서 크레딧

20 Bytes							
RID	NID	I	O	IT	OT	S	OFN
4 Bytes	4 Bytes	2 Bytes					

그림 6 크레딧 관련 추가 필드

- RID : 보고 노드의 ID
- NID : 이웃 노드의 ID
- I : NID에게 받은 패킷 수
- O : NID에게 보낸 패킷 수
- IT : NID에게 받은 패킷 중에 TTL 값이 0인 패킷 수
- OT : NID에게 보낸 패킷 중에 TTL 값이 1인 패킷 수
- S : NID에게 보낸 패킷 중에 RID가 소스인 패킷 수
- OFN : NID에게 받은 패킷 중에 NID가 소스인 패킷 수

을 얻을 수 없게 된다. 다른 노드의 패킷 포워딩에 비례하여 크레딧을 계산하고 이에 따라 이기적인 노드를 구별하는데, 이런 노드들은 이기적인 노드가 아님에도 불구하고 이기적인 노드 취급을 받게 된다. 이러한 노드들의 문제점을 해결하기 위해 우선 이기적인 노드 취급을 받는 노드를 찾는 방법을 제시하였다.

이기적인 노드가 아닌 일반 노드들은 다른 노드들로부터 받은 총 패킷의 수 ($\sum I$)와 자신이 소스(source)인 패킷의 총 개수 ($\sum S$)를 합한 것에서 패킷의 TTL이 0이어서 드롭한 패킷의 총 개수를 뺀 것 ($\sum C$)이 다른 노드에게 전송한 총 패킷의 개수 ($\sum O$)와 같다.

$$\sum I + \sum S - \sum C = \sum O \quad (2)$$

하지만 이기적인 노드는 고의적으로 다른 노드로부터 전송되어 온 패킷을 포워딩하지 않기 때문에 좌변이 우변보다 큰 (3)번 식이 적용된다.

$$\sum I + \sum S - \sum C > \sum O \quad (3)$$

(2)번 식이 성립하고, 다른 노드로부터 받은 패킷의 총 합 ($\sum I$)이 자신이 보내려고 하는 패킷의 수 ($\sum S$) 보다 작고, 크레딧이 제한된 값보다 작은 노드가 우리가 찾고자 하는 이기적인 노드가 아님에도 불구하고 이기적인 노드 취급을 받는 노드이다.

$$\begin{aligned} \sum I + \sum S - \sum C &= \sum O \\ \sum I < \sum S \end{aligned} \quad (4)$$

$$\text{credit} < \text{제한값}$$

(4)번 식을 이용해 이기적인 노드가 아닌 노드들 중에 포워딩할 패킷을 받지 못해서 자신이 보낼 패킷을 보낼 수 있는 만큼의 크레딧을 얻지 못하는 노드를 찾아서, 그 노드에게 일정양의 크레딧을 보충해 줌으로써 문제를 해결하였다. 이를 통해 이기적인 노드가 아닌 노드가 부당하게 불이익을 받는 것을 최소화 하고 데이터

전송에 참여할 수 있도록 하여 센서 네트워크의 데이터 수집에 대한 효율성을 높일 수 있다.

4. 실험 및 결과

이 절에서는 3절에서 제안한 센서 네트워크에서의 이기적인 노드 처리 방법에 대한 실험 및 성능 평가 내용을 다룬다.

성능 평가를 위해서 본 논문에서는 센서 네트워크 범위를 한정하고, 각 노드들은 주기적으로 자신이 수집한 데이터를 싱크 노드에게 보내도록 설정하였다. 또한 이기적인 노드들은 서로 공모하지 않으며, 두 노드 간의 단일 흡 패킷 전송을 항상 성공한다고 가정한다.

4.1 실험 환경 및 구현

정확한 성능 비교를 위해 리눅스 환경에서 ns2 시뮬레이션을 이용하여 실험하였다. 이 절에서는 실험 환경과 구현 방법을 기술한다[14].

표 1 실험 환경 설정

설정 조건	MICA2 설정	실험 설정
노드 전송 범위	10 m	200 m
초기 노드 에너지	4 Joule	0.5
송신할 때 에너지 소모량	5.3 ~ 26.7 mA	0.660 Joule
수신할 때 에너지 소모량	7.4 mA	0.395 Joule
실험 환경 조건		실험 설정
센서 네트워크 범위	500m * 500m	
네트워크 구성 형태	flat topology	
수신 신호 세기 모델	TwoRayGround	
MAC 프로토콜	IEEE 802.11n	
라우팅 프로토콜	AODV	
센서 노드 수	99	
싱크 노드 수	1	
APP	CBR	
packet size	512 Bytes	

성능 비교를 위해 ns2 시뮬레이션으로 구현하였다. 실험 환경은 각 노드는 이동성이 없으며, 이기적인 노드율을 20%로 설정하였다. 노드끼리 전송할 때는 전송 실패가 없다고 가정하고, 이기적인 노드로 설정된 노드는 자신이 받은 패킷은 무조건 드롭(drop)하도록 설정하였다.

PIFA에서 이기적인 노드를 제한하는 방법을 센서 네트워크에서 구현한 것과 그 때의 문제점을 해결한 방법을 포함하여 구현한 것을 비교하였다. 본 실험에서는 노드끼리 공모하여 CM을 속이거나 거짓 보고를 하지 않는 것으로 가정하였기에, PIFA에서 구현된 리포트 메시지의 신뢰성 검증 알고리즘은 구현하지 않았다. PIFA를 구현한 알고리즘은 다음과 같다.

싱크 노드에서 크레딧 계산은 그림 7에 나온 것과 같

```
[싱크 노드 - 크레딧 계산]
credit(i) = 20 - n_sendpkt(i) * 2 + n_output(i) - n_sendpkt(i);
if (credit(i) < 0)
    blacklist[] = i;
[이기적인 노드]
if (RID != saddr(p))
{
    if (20 - (n_sendpkt * 2) + n_output - n_sendpkt < 5)
        forward(p);
    else
        drop(p);
}
[센서 노드 공통]
if (saddr(p) == blacklist[])
{
    drop(p);
}
```

그림 7 PIFA 구현 알고리즘

다. 만약 크레딧이 '0'보다 작으면 그 노드는 블랙리스트에 포함시킨다. 이기적인 노드는 자신이 보내는 패킷이 아니면 드롭하도록 설정하였다. 하지만 자신의 계산된 크레딧이 5보다 작게 되면 크레딧을 얻기 위해 다른 노드의 패킷을 포워딩하게 설정하였다. 그림 7의 이기적인 노드 설정에 나온 조건식에서 크레딧이 '0'이 아닌 '5'보다 크면 포워딩 하게 해 놓은 것은 싱크 노드까지 패킷 전달 개수를 전송하는데 걸리는 시간을 고려한 것이다. 센서 노드 공통으로 구현된 상황은 패킷의 소스가 블랙리스트에 포함되어 있는 노드라면, 그 패킷을 드롭하게 설정하였다.

PIFA 구현 알고리즘을 CM이 아닌 이기적인 노드에 구현한 이유는 패킷 전송에 관련된 것이므로 서버에서 크레딧을 계산하여 노드에 알려주고 패킷 전송을 하게 하는 것보다 크레딧을 계산하는 바탕이 되는 패킷 개수를 이용하여 노드에서 직접 하게 하는 것이 효율적이라고 판단하였기 때문이다.

센서 네트워크에서 간략하게 구현한 PIFA를 보완한 알고리즘은 패킷 전송 경로에 포함되지 않는 노드들이 의도적으로 패킷을 포워딩하지 않는 것이 아님에도 포워딩할 패킷을 받지 못하여 크레딧을 얻지 못하는 것을 예방하기 위해 제안하였다.

그림 8에서 이기적인 노드가 아님에도 이기적인 노드 취급을 받는 노드를 찾기 위해 사용한 조건은 3. 1. 2절에서 유도한 것이다. 이런 노드에 해당하는 경우, 해당 노드의 크레딧을 '10'만큼 증가 시켜 주었다. 그리고 센서 노드에서 패킷의 소스가 블랙리스트에 포함되어 있는 노드라면, 드롭 시키는 것 대신에 그 해당 패킷의 TTL값을 1로 변경시켜주었다. 이것은 이기적인 노드가 고의적으로 다른 노드의 패킷을 드롭 시키는 것과 이기

[싱크 노드 - 크레딧 계산 추가]	
if ((n_input(i) + n_sendpkt(i) = n_output(i) + n_inttl(i)) && (n_input(i) < n_sendpkt(i)) && (credit < 0))	{
credit(i) = credit(i) + 10;	}
[센서 노드]	
if (saddr(p) == blacklist[])	{
ih_ttl(p) = 1;	}

그림 8 센서 네트워크에서 구현된 PIFA를 보완한 알고리즘

적인 노드의 패킷을 전송시키지 않겠다는 정책에 따라서 일반 노드들이 이기적인 노드의 패킷을 드롭 시키는 것을 구분하기 위해서이다.

4.2 실험 결과

4.2.1 패킷 전송률 비교

기존 기법과 제안 기법 성능을 비교하기 위한 실험 환경은 4.1에 제시하였다. 하나의 MAP에 싱크노드를 포함한 100개의 노드를 랜덤하게 위치시켜 네트워크를 구성하였다. 이 노드들은 주기적으로 512 바이트 크기의 CBR 패킷을 생성하여 싱크 노드에게 전송한다. 이 CBR 패킷을 전송할 때, 크레딧에 관한 데이터가 추가되어 보내지게 된다. 싱크 노드는 크레딧에 관한 데이터를 받아 크레딧을 계산한다.

본 실험에서는 이기적인 노드만을 적용한 경우, 센서 네트워크에서 간단하게 PIFA를 적용한 경우, 그리고 PIFA를 보완해서 적용한 경우 이렇게 3가지로 나누어 구현하고 실험하였다.

그림 9는 시뮬레이션을 실행시켰을 때, 시간의 흐름에 따른 패킷 전송률을 보여준다. 3가지 토플로지에서 각각의 시뮬레이션을 100초 간격으로 실행시킨 후, 그 때마다 싱크노드에서 받은 패킷의 개수를 통해 전송된 패킷의 개수를 계산하였다. 이를 이용해 각 조건 별로 패킷 전송률을 얻어 내었다. 시뮬레이션 시간 범위는 50초부

터 500초로 하였으며, 최대 시뮬레이션 시간을 500초로 설정한 이유는 그 이후에 전송된 패킷의 개수가 더 이상 증가하지 않았기 때문에 의미가 없다고 판단, 제외하였다.

그림 9는 실험을 통해 얻은 데이터의 평균을 내어 패킷 전송률을 그래프로 나타낸 것이다. PIFA를 보완한 소스는 200초까지 80%의 전송률을 유지하면서 PIFA만 적용한 소스보다 높은 전송률을 계속 유지하고 있는 것을 볼 수 있다. PIFA를 보완한 소스는 이기적인 노드를 적용했을 때보다 항상 10% 이상 높은 전송률을 유지하는 것을 볼 수 있다.

4.2.2 센서 네트워크의 토플로지 유지 시간 비교

이 절에서는 센서 네트워크에서 적용된 PIFA와 여기서의 문제점을 보완한 M-PIFA의 토플로지 유지 시간을 비교한다. 이를 위해 시뮬레이션 실행 시간 동안 10초 단위로 살아있는 노드의 개수를 비교하였다.

보완되기 전의 PIFA는 CM에게 크레딧에 관한 정보를 포함하는 메시지를 따로 보내게 된다. 이것은 네트워크에 많은 오버헤드를 발생시키며, 각 노드의 배터리 또한 많이 소모하게 된다. 하지만 보완된 PIFA에서는 각 노드들이 주기적으로 메시지를 보낼 때, 크레딧에 관한 리포트 메시지를 같이 보내게 함으로써 네트워크의 오버헤드를 만들지 않게 할 뿐 아니라 노드의 배터리를 절약할 수 있게 한다. 그림 10은 실험에서 얻은 데이터를 그래프로 나타낸 것이다.

초기에는 보완전의 PIFA가 더 많은 노드가 살아 있지만, 실행 시간이 지남에 따라 보완된 PIFA가 더 많은 노드를 유지하고 있음을 알 수 있다. 이것은 더 오래 네트워크의 토플로지를 유지할 수 있음을 나타내며, 그에 따라 네트워크에 있는 노드에게 더 많은 정보를 얻을 수 있다는 것을 의미한다. 즉, PIFA 알고리즘을 개선하여 적용함으로써 같은 환경의 노드를 더 효율적으로 사용할 수 있다는 것을 알 수 있다.

5. 결 론

센서 네트워크의 활용에 대해 관심이 높아지면서 보

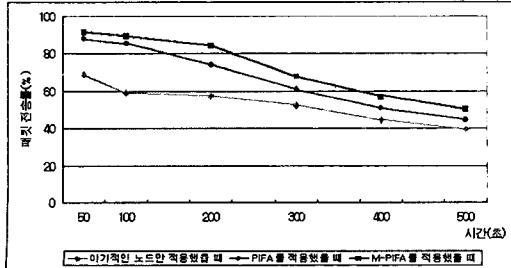


그림 9 조건 별 패킷 전송률 그래프

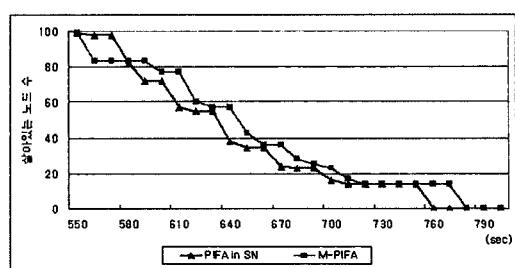


그림 10 실행 시간에 따른 노드 수의 변화

다 효율적으로 정보를 얻을 수 있는 많은 방법들이 연구되고 있다. 이러한 흐름에 맞춰 이기적인 노드들이 포함되어 있는 센서 네트워크의 성능을 높이기 위해 크레딧을 이용한 패킷 지불 방법을 이용한 방법을 제안하고 성능을 평가하였다.

본 논문에서는 PPM, PTM, Sprite, PIFA와 같은 애드-혹 네트워크에서 이기적인 노드를 처리하는 기술에 대해 간략히 살펴보고, 센서 네트워크 환경에서 언급된 기술을 이용하여 이기적인 노드를 처리해 보고자 하였다. 이 때, 발생된 문제점을 파악하고, 이를 보완한 기술을 제안하였다.

이기적인 노드를 처리하기 위해서 자신의 패킷을 보낼 때 크레딧을 사용하게 하고, 크레딧을 얻기 위해서는 다른 노드들의 패킷을 전송하게 하여 다른 노드들의 패킷을 의도적으로 전송하지 않아서 패킷 전송을 막는 일을 예방하였다.

센서 네트워크의 노드들은 이동성이 없기 때문에 패킷의 전송 경로가 고정되어 있으므로 이기적인 노드가 아님에도 불구하고 패킷 포워딩을 해주지 못하는 노드가 발생하는 문제점이 생긴다. 이를 해결하기 위해 이런 노드들을 패킷 전송 개수를 이용하여 찾고, 추가적인 크레딧을 지불해 주는 것으로 해결하였다. 또한, 추가적인 데이터 전송을 필요로 하는 크레딧 관리 서버를 따로 두는 것 대신에 싱크 노드에 크레딧 관리 서버의 역할을 포함시킴으로써 추가 필드를 이용하여 크레딧 정보를 전송할 수 있게 하였다.

실험을 통해 패킷 전송률과 노드의 전력 소모량에 대한 결과를 얻었다. 이기적인 노드가 포함된 센서 네트워크에서 단순히 크레딧을 이용해 이기적인 노드를 처리한 것 보다 그에 따른 문제점을 해결한 방안이 패킷 전송률이 더 높게 유지되었으며, 총 패킷 전송 개수도 더 많았다. 그리고 싱크 노드가 크레딧 관리 서버의 역할을 함께 하면서 크레딧 정보를 전송하기 위해 리포트 메시지를 보내지 않아도 되기 때문에 센서 노드들은 전력을 절약할 수 있었다.

본 논문에서 제안된 방법을 활용하여 센서 네트워크를 운영하면, 지형에 대한 정보나 날씨 정보 등의 정보 수집을 목적으로 하는 센서 네트워크에서는 보다 효율적으로 노드들을 활용하여 정보를 수집할 수 있을 것이다.

참 고 문 헌

- [1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks," IEEE Communications magazine 2002.
- [2] T. Roughgarden and E. Tardos, "How bad is selfish routing?" Journalof ACM, Vol.49, No.2, pp. 236-259, 2002.
- [3] L. Buttyan and J.-P. Hubaux, "Enforcing Service Availability in Mobile Ad-Hoc Networks," ACM MobiHoc, pp. 87-96, 2000.
- [4] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," IEEE INFOCOM, pp. 1987-1997, 2003.
- [5] 이춘재, 한옥표, 정영준, "Ad-hoc Network에서 동적 임계값을 이용한 부하균등 라우팅 프로토콜," 정보통신논문지 제 8판, Vol.2, 2004.
- [6] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad hoc On-demand Distance Vector(AODV) Routing," IETF RFC 3561, July 2003.
- [7] D.-B. Johnson, D. A. Maltz, and Y. C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," IETF Internet Draft, July 2004.
- [8] C.-K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall PTR, 2001.
- [9] 안상현, 유영환, 이재훈, "이동 애드혹 네트워크에서 라우팅 방식과 무관한 이기적인 노드 관리 방안," 정보과학회, 2003.
- [10] M. Franklin and M. Reiter. Fair exchange with a semi-trusted third party. In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 16, April 1997.
- [11] H.-Y. Hsieh and R. Sivakumar, "Performance comparison of cellular and multi-hop wireless networks: A quantitative study," in Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS) 2001, Cambridge, MA, June 2001.
- [12] S. Buchegger and J.-Y. L. Boudec, "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks," in 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, 2002.
- [13] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks," in Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), IEEE, June 2002.
- [14] J. Hill and D. Culler, "Mica: a wireless platform for deeply embedded networks," IEEE Micro 22(6): 12-24, 2002.
- [15] Network Simulator-ns-2, <http://www.isi.edu/nsnam/ns/>.



최종원

1984년 서울대학교 컴퓨터공학(학사), 1986년 서울대학교 컴퓨터공학(석사), 1992년 노스웨스턴대학교 컴퓨터공학(박사). 1993년~현재 숙명여자대학교 정보과학부 교수. 관심분야는 라우팅, 멀티캐스트, 이동통신, 미래 인터넷 등



유동희

2005년 2월 숙명여자대학교 정보과학부(학사), 2007년 2월 숙명여자대학교 컴퓨터과학(석사), 2007년 2월~현재 삼성전자 정보통신총괄 무선 사업부 연구원. 관심분야는 센서네트워크, 무선통신, GSM 등