

이동 애드혹 네트워크에서 목적지 개시 플러딩 기반의 AODV 프로토콜

최 현 호*

AODV Protocol Based on Destination-Initiated Flooding in Mobile Ad Hoc Networks

Hyun-Ho Choi*

요 약

본 논문에서는 기존 Ad-Hoc On-Demand Distance Vector (AODV) 프로토콜의 단점을 극복하고 라우팅 성능을 개선하기 위하여 AODV 프로토콜에 새로운 경로 유지관리 프로토콜의 추가를 고려한다. 이를 위하여 경로 설정 후 주기적으로 목적지에서 플러딩을 발생시키는 목적지 개시 플러딩 방식을 기존 AODV 프로토콜에 도입한다. 제안 라우팅 프로토콜에서는 먼저 엮듣기 동작을 통하여 오버헤드 없이 목적지에 대한 추가적인 라우팅 정보를 얻고, 소스-목적지 최단 경로 주변의 1-홉 이웃 노드를 결정한다. 이후 토폴로지 변화에 따라 최신의 경로 정보를 파악하기 위하여 목적지에서 플러딩을 시작한다. 이 목적지 개시 플러딩은 최단 경로 주변의 1-홉 이웃 노드들만 플러딩에 참여시켜 기존 플러딩 오버헤드를 현저히 줄이고, 최단 경로 주변에 다수의 대체 경로를 형성시킨다. 이를 통하여 사용 중인 전송 경로의 단절 및 변화 시 주변의 새로운 최단 경로로의 끊김 없는 라우팅이 가능하게 해준다. 시뮬레이션 결과 제안하는 목적지 개시 플러딩 기반 AODV 프로토콜은 링크 단절 빈도를 줄이고 네트워크에 발생하는 불필요한 제어 패킷 전송을 줄이면서 소스-목적지간 데이터 지연 성능을 향상시킨다.

Key Words : Routing protocol, route maintenance, AODV, destination-initiated flooding, mobile ad hoc networks

ABSTRACT

This paper considers adding a new route maintenance protocol to the typical Ad-Hoc On-Demand Distance Vector (AODV) protocol to overcome the shortcomings of AODV and improve routing performances. To this end, we enhance the existing AODV protocol by adding a destination-initiated flooding (DIF) scheme that generates a periodic flooding at a destination after the initial routing setup. The proposed routing protocol first obtains additional routing information for the destination without overhead through overhearing and determines the one-hop neighbor nodes around the shortest path between source and destination. Then, the destination starts flooding to obtain the latest route information according to the topology change. This DIF involves only one-hop neighbor nodes around the shortest path in flooding, which significantly reduces the conventional flooding overhead and forms multiple alternate paths around the shortest path. This makes it possible to seamlessly route to the newest shortest path around when the current routing path is disconnected or old-fashioned. Simulation results show that the proposed DIF-based AODV protocol reduces the frequency of link disconnection and improves the performance of end-to-end delay while reducing unnecessary control packet transmission in the network.

※ 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2016R1C1B1016261)

♦ First Author : Hankyong National University, Department of Electrical, Electronic and Control Engineering, hhchoi@hknu.ac.kr, 정회원
논문번호 : KICS2017-09-250, Received September 15, 2017; Revised September 29, 2017; Accepted October 25, 2017

I. 서 론

이동 애드혹 네트워크 (Mobile Ad hoc Network, MANET)에서 라우팅 프로토콜의 목적은 노드의 이동성 및 무선 채널의 변화에 따라 발생하는 동적 토폴로지를 오버헤드를 최소화하면서 신속하고 정확하게 파악하여 항상 소스와 목적지간 최적의 라우팅 경로를 제공하는데 있다. 이를 통하여 네트워크에 발생하는 불필요한 제어 패킷 전송을 줄여 네트워크 전송 용량을 높이면서, 소스-목적지간 데이터 전송률 및 지연 성능을 향상시키고자 하는 것이다. 이를 위하여 경로 설정 시에는 최소한의 제어 패킷을 발생시켜 신속하게 라우팅 정보를 파악하여 소스-목적지간 세션을 빠르게 설정해야 하며, 데이터 전송 중에는 변화하는 토폴로지 정보를 최소한의 오버헤드 발생으로 정확하게 파악하여 사용 경로가 단절되지 않게 하면서 더 좋은 경로를 사용할 수 있게 해주어야 한다. 또한 사용 중인 경로가 끊어진 경우에는 신속하게 라우팅 정보를 파악하여 연결 가능한 새로운 경로를 다시 제공해야 한다¹⁾.

지금까지 수많은 MANET 라우팅 프로토콜이 제안되었지만 가장 많이 인용되고 연구된 MANET 라우팅 프로토콜로는 Ad-Hoc On-Demand Distance Vector (AODV)를 뽑을 수 있다^{2,3)}. AODV 프로토콜은 반응형(reactive) 방식으로 소스에서 전송해야 할 데이터 패킷의 목적지에 대한 경로 정보가 없는 경우 route request (RREQ) 패킷을 네트워크에 플러딩(flooding) 한다. 플러딩 과정을 통하여 RREQ가 목적지에 도착하면 목적지는 route reply (RREP) 패킷을 소스에게 역으로 전송한다. RREP가 소스에 도착하면 소스-목적지간 새로운 양방향 최단 경로가 설정된다. AODV에서 RREQ와 RREP 패킷을 수신한 노드는 자신의 라우팅 테이블에 해당 패킷을 발생시킨 노드로 향하는 거리 벡터(distance vector, DV) 정보를 저장한다. 이러한 거리 벡터 정보를 이용하여 RREP 패킷 및 소스-목적지간 데이터 패킷이 라우팅 된다. 전송 경로 상의 어느 곳에서 링크 단절(link failure)이 발생하면 이를 인지한 중간 노드는 route error (RERR) 패킷을 발생시키고 이를 소스에게 전달하여 새로운 경로 설정 과정을 시작하도록 한다.

기본적으로 AODV는 주문형(on-demand) 방식으로 동작하여 네트워크 플러딩을 통하여 경로를 찾는다. 플러딩 방식은 네트워크 전체에 제어 패킷을 전파하므로 오버헤드가 매우 크다. 특히 단일 경로 라우팅 방식을 취하는 AODV의 경우 링크 단절시마다 경로

재탐색을 위하여 플러딩을 수행해야 하므로 이동성이 높은 환경일수록 플러딩 오버헤드가 더 심각한 문제로 작용한다. 또한 AODV는 경로 설정 후 경로를 유지 관리하는 별도의 동작이 마련되어 있지 않아 이에 대한 오버헤드는 존재하지 않지만, 이동성에 따른 토폴로지 변경 시 항상 최적의 경로를 사용하지 못한다는 단점이 존재한다. 이러한 단점을 극복하기 위하여 AODV 기반의 다중 경로 라우팅 방식이 몇몇 제안되었지만 동작 복잡도가 높으며, 아직 정식 프로토콜로써 표준화되지 않은 상태이다^{4,6)}.

제안 라우팅 프로토콜에서는 기존 AODV 프로토콜에서 다루어지지 않았던 경로 유지관리 프로세스에 초점을 맞춘다. 제안 방식은 기존 AODV 방식으로 설정된 최단 경로 상에서 데이터 전송 도중, 사용 경로가 나빠지거나 끊어질 것에 대비하여 주기적으로 목적지에서 플러딩을 발생시킨다. 이 목적지 개시 플러딩의 오버헤드를 최소화하기 위하여 먼저 최단 경로 주변 노드의 엷든기 과정을 통하여 1-홉 이웃 노드를 결정하고, 이 1-홉 이웃 노드들만 플러딩에 참여하도록 제한한다. 목적지에서 개시하는 주기적인 플러딩을 통하여 최단 경로 주변에 다수의 대체 경로를 제시하여 링크 단절 빈도를 줄이고 토폴로지 변화에 따라 항상 최단 경로를 사용할 수 있게 해준다. 본 논문에서는 제안 방안의 자세한 동작 과정을 설명하고, 동작 및 성능을 시뮬레이션을 통하여 검증한다.

본 논문은 다음과 같이 구성된다. II장에서는 기존 AODV 프로토콜의 동작을 설명하고 문제점을 제시한다. III장에서는 목적지 개시 플러딩을 이용하는 제안 라우팅 프로토콜을 상세히 설명한다. IV장에서 제안 라우팅 프로토콜의 성능을 보여주고, 마지막으로 V장에서 결론을 맺는다.

II. 기존 AODV 기반 라우팅 프로토콜

그림 1은 AODV에서 초기 경로 설정을 위하여 RREQ 플러딩을 수행하는 과정을 보여준다. 소스에서 발생시킨 RREQ는 주변 노드에 방송되며, 한 노드가 방송한 RREQ 메시지를 수신한 노드는 해당 RREQ가 처음 수신한 것이라면 RREQ 내에 포함된 라우팅 정보를 이용하여 자신의 라우팅 테이블 정보를 갱신하고 RREQ를 주변에 재방송한다. 나중에 수신한 RREQ 패킷은 처리하지 않고 버려진다. 이러한 방식으로 RREQ 패킷은 네트워크 전체에 퍼져나가면서 소스 노드로 향하는 경로 정보를 제공하고, 결국 어딘가에 있는 목적지 노드에 도착하게 된다. 목적지 노드

는 처음 도착한 RREQ에 대해서 RREP로 응답한다. RREP 패킷은 RREQ 플러딩 과정을 통해 만들어진 역방향 거리 벡터 정보를 이용하여 목적지 노드에서 소스 노드로 유니캐스트(unicast) 방식으로 전달된다. RREP를 수신한 노드들은 RREP를 처음 송신한 목적지로 향하는 거리 벡터 정보를 자신의 라우팅 테이블에 저장한다. 따라서 RREP가 소스 노드에 도착하게 되면 소스에서 목적지로 갈 수 있는 최단 경로를 따라 순방향 거리 벡터가 만들어진다. 소스가 RREP 수신하게 되면 소스와 목적지간에 최단 경로를 따라 양방향 거리 벡터가 형성되기 때문에 이 경로 상에서 데이터 및 확인응답(acknowledgement, ACK) 패킷의 전송이 가능하게 된다.

AODV의 가장 큰 문제점은 플러딩 시에 발생하는 제어 패킷 전송 오버헤드가 매우 크다는 점이다. 플러딩이 발생하게 되면 네트워크 전체 노드 수만큼 RREQ 패킷이 전송되어야 하므로 네트워크의 규모(즉, 네트워크에 존재하는 노드 수)에 비례하여 발생하는 오버헤드가 커지게 된다. 또한 토폴로지의 변화에 따라 전송 링크가 끊어질 때마다 플러딩이 다시 발생하므로 노드 이동성이 높은 환경에서는 링크 단절이 증가하여 발생 오버헤드가 매우 커진다⁷⁾.

또한 AODV의 문제점은 경로의 유지관리 과정이 존재하지 않는다는 점이다. AODV에서는 처음 설정된 경로가 끊어지기 전까지 계속 사용되며, 데이터 전송 도중 토폴로지가 변하여 더 좋은 경로가 만들어져도 이를 파악하거나 이용할 수 있는 방법이 제공되지 않는다. 이는 AODV가 동작 복잡도를 줄이기 위하여 경로 단절 후에야 주문형(on-demand) 방식으로 경로를 재설정하기 때문이다. 따라서 AODV는 토폴로지 변화에 따라 지속적으로 더 좋은 경로를 사용할 수 있

게 해주거나, 경로가 끊어지는 것에 대한 사전 대비를 하지 못한다. 이로 인하여 AODV는 이동성이 큰 환경에서는 플러딩 오버헤드를 가속화하게 된다⁸⁾.

아울러 AODV는 기본적으로 단일 경로(single path) 라우팅 방식을 사용하여 목적지로 가는 하나의 경로(즉, 다음 노드(next hop) 정보)만을 관리한다. 이는 라우팅 정보의 연산을 단순화하고 라우팅 테이블의 사용 메모리를 줄이는 장점이 있지만, 다중 경로(multipath) 라우팅 방식 대비 경로 단절에 매우 취약하다. 기존 AODV 표준을 기반으로 다중 경로를 관리하고 최단 경로의 단절시 대체 경로를 사용하는 방식이 제안되었다. 이러한 multipath AODV(M-AODV) 프로토콜은 초기 RREQ 플러딩시에 중간노드에서 중복되어 수신한 RREQ 정보를 이용하여 루프가 생기지 않도록 하면서 소스-목적지간에 다수의 거리 벡터를 형성시키는 방식을 사용하거나⁴⁾, 목적지에서 다수의 RREQ를 수신하고 이에 대해 각각의 RREP 응답을 소스에게 전송함으로써 소스-목적지간 겹치지 않는 다중 경로를 형성하는 방식을 사용한다^{5),6)}. 하지만 이러한 M-AODV 프로토콜은 복잡도가 높으며 실제 환경에서 사용하기에는 완성도가 높지 않아 AODV와 같이 RFC(Request for Comments) 표준으로 채택되지는 않은 상태이다.

III. 목적지 개시 플러딩 기반 AODV 프로토콜

3.1 배경 및 프로토콜 개요

제안하는 라우팅 프로토콜은 다음과 같은 현실적인 가정에 바탕을 둔다⁹⁾. 먼저 소스-목적지간 전송 경로의 노드들의 주변에는 1-홉 거리에서 송수신이 가능한 주변 노드들이 어느 정도 존재한다고 가정한다. 즉, 네트워크상의 노드 밀도가 낮지 않아 소스-목적지간 전송 경로를 따라 전송되는 패킷들을 엿들을 수 있는 주변 노드들이 존재한다고 가정한다. 또한, 네트워크 토폴로지의 변화에 따라 현재 전송 경로를 대체할 수 있는 더 좋은 경로는 현재 사용 중인 경로를 포함하여 그 근처에서 발생한다고 가정한다. 즉, 토폴로지의 변동성이 있어도 현재 사용 중인 경로 영역을 벗어나 완전히 새로운 경로가 설정되는 최단 경로는 드물고, 새롭게 찾아진 최단 경로와 이전에 사용되었던 전송 경로 간에는 상호 관련성(즉, 지역적인 위치나 사용하는 중간 노드들이 겹침)이 존재한다고 가정한다. 마지막으로 소스와 목적지 사이에는 양방향 패킷이 전송된다고 가정한다. 소스에서 목적지로 데이터 패킷이 전송되면 이에 대한 ACK 패킷이 목적지에서 소스로 전송

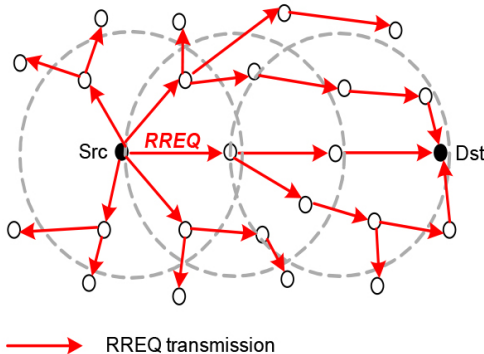


그림 1. AODV의 RREQ 플러딩 동작
Fig. 1. RREQ flooding of AODV

되거나, 세션 개시 후 목적지에서 발생되어 스스로 전달되는 데이터 패킷이 존재한다고 가정한다¹⁰⁾.

제안 라우팅 프로토콜은 무선 미디어가 지나는 브로드캐스트 속성을 이용한다. 무선 미디어를 통해 전송되는 신호는 항상 방송되는 신호이므로 신호가 미치는 영역의 노드들은 수신된 신호를 복호할 수 있다. 따라서 자신에게 해당하는 데이터가 아니더라도 노드들은 주변 신호의 엿듣기(overhearing)가 가능하다¹¹⁾. 이러한 overheard 기법은 원래 전송해야 할 패킷을 주변 노드에서 엿듣는 것뿐이므로 추가적인 오버헤드를 발생시키지 않는다. 따라서 제안 라우팅 프로토콜은 이러한 엿듣기 기법을 사용하여 별도의 오버헤드 없이 추가적인 라우팅 정보를 얻을 수 있으며, 추가적으로 엿들은 노드를 기반으로 플러딩 영역을 제한하여 기존 플러딩 오버헤드를 줄인다. 제안 라우팅 방식에서 네트워크의 각 노드는 소스-목적지 간 송수신되는 데이터 및 ACK 패킷을 엿듣는다. 이와 같이 소스-목적지 간 패킷을 엿듣는 노드는 해당 소스-목적지 경로 상의 1-홉 이웃 노드가 되며, 이들은 자신이 어떤 소스-목적지 경로에 대한 1-홉 이웃인지를 스스로 파악하고 관리한다.

제안 라우팅 프로토콜은 기존 AODV에서 사용하는 초기 경로 설정 과정을 그대로 사용한다¹²⁾. 즉, 초기 소스에서 목적지에 대한 경로 정보를 모를 때에는 기존 AODV와 동일하게 RREQ 플러딩 및 RREP 전송 과정을 통하여 초기 소스-목적지간 최단 경로를 찾아 데이터 전송을 시작한다. 하지만 제안 라우팅 프로토콜은 기존 AODV에 없던 경로 유지관리 과정을 새롭게 포함한다. 이 경로 유지관리 기법에서는 전송 경로 상에서 패킷을 엿들은 1-홉 이웃 노드를 활용하여 플러딩 영역을 효과적으로 제한한다. 이러한 경로 유지관리 과정에서 발생하는 플러딩은 기존 AODV와는 다르게 목적지에서 RREQ 플러딩이 시작되며, 플러딩에 참여하는 노드는 1-홉 이웃 노드로 제한된다. 이러한 목적지에서 개시하는 RREQ(Destination-initiated RREQ, DiRREQ) 플러딩은 주기적으로 또는 정의된 이벤트 조건에 따라 비주기적으로 발생할 수 있다. 주변 노드의 엿듣기 및 DiRREQ 플러딩 과정을 통하여 형성된 거리 벡터는 다중 경로 라우팅 방식에 따라 각 노드의 라우팅 테이블에 저장되어 관리된다. 소스에서 발생한 데이터 패킷은 이렇게 형성된 다수의 거리 벡터들을 기반으로 토폴로지 변화에도 불구하고 최단 경로를 따라 목적지로 전달된다.

참고로 목적지 개시 기반 플러딩을 통하여 다중 경로 거리 벡터를 관리함에도 불구하고 현재 전송 가능

한 경로 정보가 하나도 없을 때에만 경로 복구 과정이 수행되며, 이는 기존 AODV와 동일하게 소스 노드에서 RREQ 플러딩을 다시 수행하는 방식을 따른다.

3.2 제안 라우팅 프로토콜의 상세 동작

그림 2-7의 예를 통하여 제안하는 라우팅 프로토콜의 동작을 설명한다. 그림 2는 기존 AODV 방식에 의해 초기 경로 설정 과정 후 소스와 목적지 간에 데이터 및 ACK 패킷을 전송하는 동작을 보여준다. 초기 경로 설정은 기존 RREQ 플러딩 방식을 동일하게 따르며, 이를 통해 소스-목적지간 최단 경로가 설정된다.

그림 3은 목적지에서 스스로 전달되는 ACK 패킷을 최단 경로 주변의 노드들이 엿듣는 동작을 보여준다. 목적지 노드 및 경로 상의 중간 노드에서 전송되는 ACK 패킷은 무선 미디어를 통해 경로 상의 주변 이웃 노드가 엿들을 수 있다. 엿들은 주변 노드들을

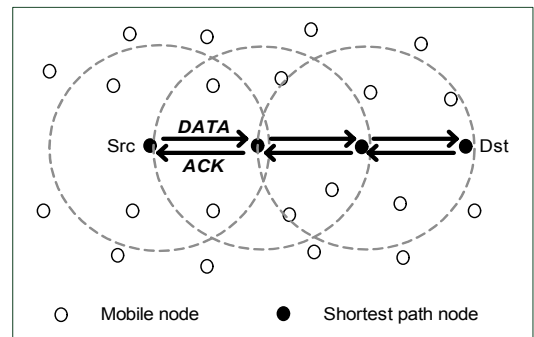


그림 2. AODV의 초기 경로 설정 과정 후 소스와 목적지 간에 데이터 및 ACK 패킷을 전송하는 동작
Fig. 2. Transmission operation of data and ACK packets between source and destination after initial route setup by AODV

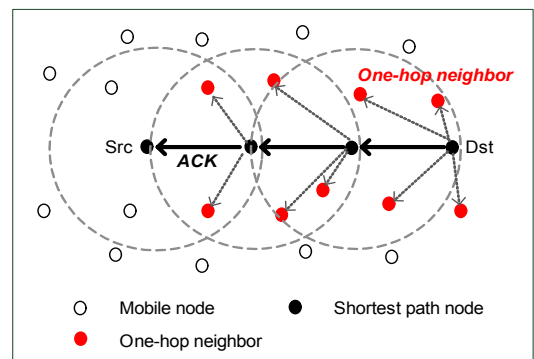


그림 3. 목적지에서 스스로 전달되는 ACK를 최단 경로 주변의 1-홉 이웃 노드가 엿듣는 동작
Fig. 3. Overhearing operation of one-hop neighbors around the shortest path by the ACK feedback from the destination to the source

해당 소스-목적지의 최단 경로 상의 1-홉 이웃 노드라 정의한다. 1-홉 이웃 노드들은 목적지에서 소스로 전달되는 패킷을 엿들음으로써 특별한 오버헤드 없이 스스로 자신이 해당 경로의 1-홉 이웃 노드인지 아닌지를 판단할 수 있다. ACK 패킷 내에는 이 패킷을 만든 목적지 노드의 주소정보, 이 패킷이 최종 전달되어야 할 소스 노드의 주소 정보, 이 패킷을 송신한 송신 노드의 주소 정보, 목적지까지의 홉 수 또는 라우팅 비용(routing cost) 정보, ACK 패킷의 순서번호 등이 포함되어 있어 이를 수신한 노드는 이 정보들을 바탕으로 목적지 노드로 향하는 순방향 거리 벡터 정보를 만들 수 있다. 목적지에서 ACK 패킷 외에 스스로 데이터 패킷을 발생시킨다면, ACK 패킷 대신 데이터 패킷을 똑같이 엿들어 앞서 기술한 바와 같이 목적지 노드로 향하는 순방향 거리 벡터 정보를 만들 수 있다. 만약 목적지에서 스스로 전달되는 ACK 패킷이나 데이터 패킷을 발생시키지 않는 세션의 경우, 제안 방식에서는 목적지에서 위의 정보를 포함하는 제어 패킷을 자체적으로 만들어 주기적으로 소스로 전송한다.

그림 4는 엿듣기 동작 후 최단 경로상의 노드와 주변 1-홉 이웃 노드에서 형성된 순방향 거리 벡터를 보여준다. 기존 RREP 응답을 통해 만들어진 최단 경로상의 순방향 거리 벡터에 엿듣기 동작을 통해 1-홉 이웃 노드들에서 형성된 순방향 거리 벡터 정보가 추가된다. 따라서 최단 경로 주변에는 해당 목적지로 향하는 다수의 라우팅 정보(즉, 순방향 거리 벡터 정보)가 형성된다.

그림 5는 목적지에서 DiRREQ를 플러딩하는 과정을 보여준다. DiRREQ의 플러딩은 데이터 전송 과정에서 목적지에서 주기적으로 발생시킨다. DiRREQ

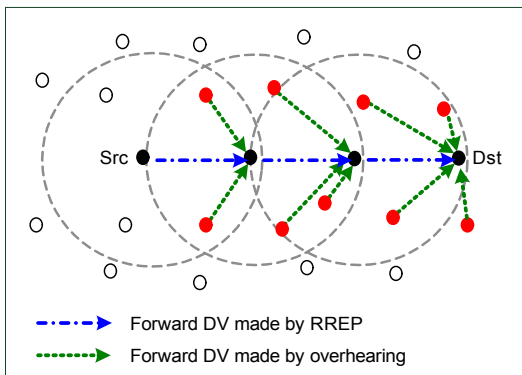


그림 4. 엿듣기 동작 후 최단 경로상의 노드와 주변 1-홉 이웃 노드에서 형성된 순방향 거리 벡터
Fig. 4. Forward DV formed at shortest path nodes and one-hop neighbors after overhearing operation

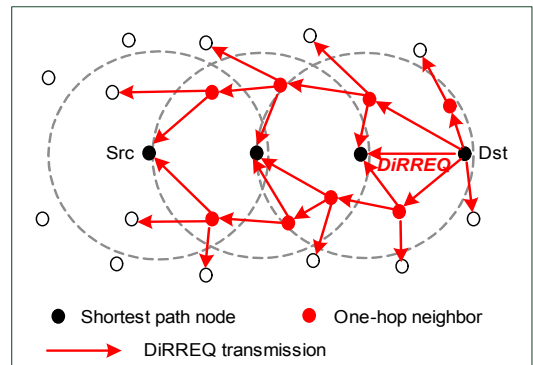


그림 5. 목적지에서 DiRREQ를 플러딩하는 동작
Fig. 5. Flooding operation of DiRREQ initiated by destination

플러딩 과정에서 DiRREQ를 수신한 네트워크상의 모든 노드들은 DiRREQ에 포함된 정보를 기반으로 라우팅 테이블 정보를 갱신한다. 하지만 DiRREQ를 수신한 노드들 중에서 다시 이를 네트워크상에 플러딩하는 노드는 해당 전송 경로 주변의 1-홉 이웃 노드들로 제한된다. 즉, 해당 소스-목적지 경로에 대한 1-홉 이웃 노드들만 수신한 첫 번째 DiRREQ를 재방송하게 되고, 이외의 노드들은 이를 재방송하지 않는다. 또한 DiRREQ 플러딩 과정 중 같은 순서번호를 갖는 DiRREQ를 수신한 노드들은 DiRREQ에 포함된 정보를 기반으로 라우팅 테이블 정보를 추가적으로 갱신하고, 해당 DiRREQ 패킷은 재방송하지 않고 버리게 된다. 이와 같이 같은 순서 번호를 갖는 DiRREQ 패킷은 라우팅 테이블 갱신에만 사용되고 1-홉 이웃 노드와 다른 노드들에서 더 이상 플러딩되지 않는다. 참고로 DiRREQ 패킷에 포함되는 정보는 RREP 정보와 동일하며, 단지 기존 RREQ가 아닌 목적지에서 발생한 DiRREQ 패킷이라는 구분 표시만 해주면 된다.

목적지에서 발생하는 DiRREQ 플러딩은 기존의 소스에서 발생하는 RREP 플러딩과는 달리 네트워크 전체에 DiRREQ가 퍼져나가지 않고 1-홉 이웃 노드들이 존재하는 영역에만 DiRREQ가 전달되게 된다. 이와 같이 소스-목적지 경로 주변으로 제한된 영역에만 플러딩이 수행되는 방식은 기존 RREP 플러딩 방식에서 발생하는 제어 패킷 전송 오버헤드를 현저히 줄이면서 소스-목적지 주변에서 발생하는 최신의 최단 경로 정보를 신속하게 파악할 수 있게 해준다.

그림 6은 DiRREQ 플러딩 과정을 통하여 형성된 순방향 거리 벡터 정보를 보여준다. 기존 소스에서 발생하는 RREP 플러딩과 RREP 응답을 통하여 만들어진 최단 경로 노드 상의 순방향 거리 벡터와 1-홉 이

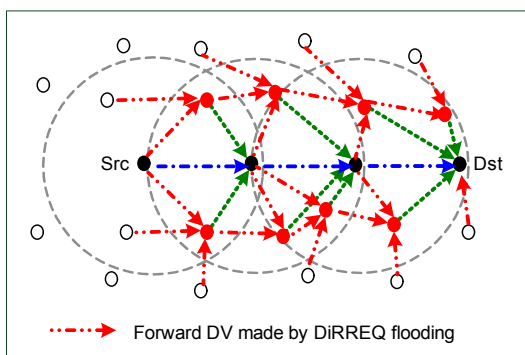
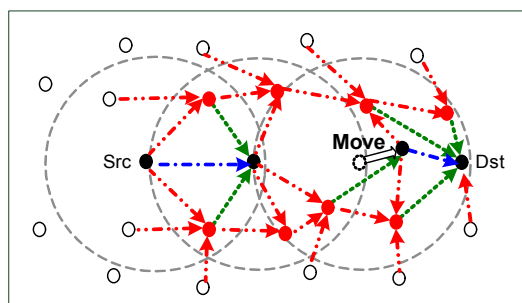


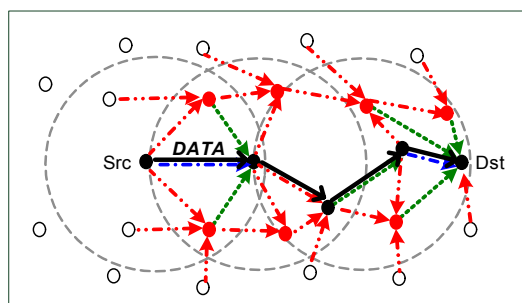
그림 6. DiRREQ 플러딩에 의해 형성된 순방향 거리 벡터
Fig. 6. Forward DV formed by DiRREQ flooding

웃 노드들의 엷들기를 통하여 만들어진 순방향 거리 벡터에 DiRREQ 플러딩 과정을 통하여 형성된 순방향 거리 벡터가 추가된 것을 볼 수 있다. 보는 바와 같이 해당 목적지로 향하는 다양한 순방향 거리 벡터가 형성되어 소스-목적지간 다양한 다수의 경로가 최단 경로 주변에 형성된다. 따라서 소스에서 발생한 데이터 패킷은 이렇게 형성된 다수의 거리 벡터 정보를 이용하여 다양한 전송 경로를 선택하여 목적지로 전달될 수 있다. 참고로 각 노드는 기존 AODV 프로토콜과는 달리 하나의 목적지에 대하여 다수의 거리 벡터를 관리하기 위하여 라우팅 테이블을 확장하여 사용한다. 즉, 각 노드의 라우팅 테이블에는 하나의 목적지에 대하여 패킷 전달이 가능한 다음 노드 정보들이 기록되고, 각각의 다음 노드를 통하여 최종 목적지에 도착하는데 걸리는 비용 정보들과 이러한 정보를 얻는데 이용되었던 패킷의 순서 번호가 저장된다. 순서 번호는 같은 패킷을 두 번 이상 수신하지 않기 위하여 사용되거나 저장된 정보가 상대적으로 얼마나 오래되었는지를 파악하는 용도로 사용된다.

그림 7은 노드의 이동에 따라 토폴로지가 변화하여 사용 중인 데이터 전송 경로의 단절이 발생하였을 때 자동적으로 다른 주변 최단 경로로 라우팅 되는 경우를 보여준다. 그림 7(a)와 같이 어떤 링크가 단절되었음을 해당 노드에서 인지하면 해당 거리 벡터 정보들은 라우팅 테이블에서 삭제된다. 사용 중이던 데이터 전송 경로가 단절이 발생한 경우 그림 7(b)와 같이 데이터 패킷은 최단 경로 주변에 형성되어 있던 순방향 거리 벡터 정보를 이용하여 자동으로 다른 최적 경로를 사용하여 전달된다. 이와 같이 데이터 패킷은 사용 중인 최단 경로가 단절되거나 사용하는 경로의 비용이 다른 경로에 비하여 커지면 자연스럽게 비용이 적은 다른 경로를 사용하여 전달된다.



(a)



(b)

그림 7. (a) 노드 이동에 따른 전송 경로의 단절이 발생한 경우와 (b) 주변 대체 경로로 데이터가 전달되는 동작
Fig. 7. (a) Link failure by node movement and (b) Rerouting of data packets to alternative path

그림 8은 제안하는 목적지 개시 플러딩 기반 라우팅 프로토콜의 전체 동작 순서도를 보여준다. 먼저 소스에서 전송할 데이터가 발생하면 이를 목적지로 전송할 라우팅 경로가 있으면 해당 경로를 따라 데이터를 전송한다. 만약 목적지까지의 경로가 존재하지 않으면 이를 찾기 위해 초기 라우팅 설정 절차로 RREQ 플러딩을 수행한다. 이후 소스와 목적지 사이에 경로가 설정되면 이를 이용하여 소스에서 목적지로 데이터를 전송하고, 목적지는 데이터에 대한 응답으로 소스에게 ACK 패킷으로 응답한다. 목적지에서 스스로 ACK 패킷이 최단 경로를 따라 전송될 때 주변에 있던 노드들은 이 ACK 패킷을 엷들고 자신이 해당 소스-목적지의 최단 경로에 대한 1-홉 이웃 노드임을 판단하고 목적지까지의 경로 정보를 갖도록 자신의 라우팅 테이블을 갱신한다. 이후 사전에 설정했던 DiRREQ 타이머가 만료되면 목적지에서 DiRREQ 플러딩을 개시하여 최단 경로 주변에 대체 경로가 형성되도록 한다.

프로토콜을 복잡도 측면에서 비교해보면, 기존 AODV 프로토콜의 경우 네트워크 내 전체 노드 수가 N 이고 초당 플러딩 발생 확률이 p 라고 할 때 초당 제

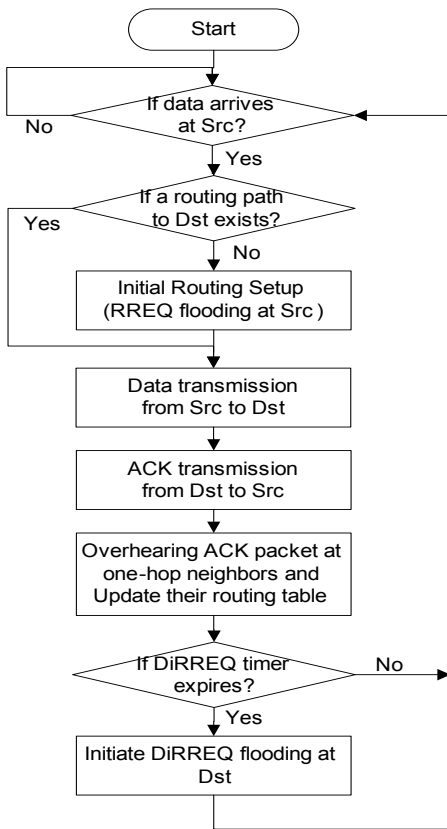


그림 8. 제안 라우팅 프로토콜의 순서도
Fig. 8. Flow chart of proposed routing protocol

어 패킷 횟수는 pN 이 되어 $O(pN)$ 의 복잡도를 갖는다. 반면 제안 프로토콜의 경우, 1-홉 이웃 노드 수를 N , 초당 플러딩 발생 확률을 p' , DiRREQ 플러딩 발생 주기를 T 라고 할 때, $p'N + \frac{N}{T}$ 의 초당 제어 패킷 전송 횟수를 갖는다. 제안 방식에서는 주기적인 DiRREQ 플러딩 동작으로 인하여 p' 는 p 보다 매우 작은 값을 가지며 0에 근접하여 복잡도가 $O(\frac{N}{T})$ 로 근사화 될 수 있다. 따라서 $N < N$ 의 관계에서 두 프로토콜의 복잡도는 라우팅 환경에 의해 결정되는 p 값과 제어 파라미터 T 의 두 파라미터에 의존한다.

IV. 모의실험 결과

표 1은 사용한 시뮬레이션 파라미터를 보여준다. 1000 m²의 정사각형에 50개의 노드를 균일(uniform) 랜덤하게 분포시키고 100개의 토폴로지에 대해서 시뮬레이션을 수행하였다. 이동성 모델로는 random

표 1. 시뮬레이션 파라미터
Table 1. Simulation parameters

Name	Value
# of simulated topologies	100
Simulation time	60 s
# of nodes (N)	50
# of sessions	1
Width = Height	1000 m
Communication range	283 m(=2*width/√N)
Mobility model	Random Waypoint
Speed interval	[0, 5~30] m/s
Pause interval	[0, 30] second
Direction interval	[-180, 180] (degrees)
Period of data packet generation	0.25 s
Period of DiRREQ (T)	1, 2, 3 s
ROUTE_TIMEOUT timer	3 s
TTL for control packets	20

waypoint (RWP) 모델을 사용하고 최대 속도를 5-30 m/s로 변화시켜가면서 이동성에 따른 성능 변화를 살펴 보았다. 60초의 시뮬레이션 시간동안 0.25초마다 데이터 패킷이 발생하며 DiRREQ 플러딩은 1, 2, 3초의 주기를 가지고 발생한다. 생성된 라우팅 정보가 지워지는데 걸리는 시간 ROUTE_TIMEOUT은 3초로 설정되며 모든 제어 패킷의 TTL 값은 20으로 설정되었다¹¹.

라우팅 효과를 확인하기 위해 초기에 출발지-목적지 간 최단 경로 홉 수가 3홉 이상이 되는 경우만 결과에 반영하였다¹³. 또한 초기 경로 설정을 위한 플러딩 시에 AODV 표준의 확장된 링 탐색(expanding ring search) 기법을 사용하지 않고 RREQ가 네트워크 전체에 플러딩 되도록 TTL 값을 충분히 크게 하였다. 아울러 RREP는 중간 노드에서는 발생하지 않고 목적지에서만 발생한다고 가정하였다¹⁴. 성능 평가를 위하여 MATLAB 시뮬레이터를 사용하였다.

그림 9는 노드 이동성에 따른 제안 라우팅 프로토콜의 성능을 (a) 단절 횟수, (b) 데이터 전달률, (c) 중단 간 지연, (d) 지터(jitter), (e) 오버헤드 측면에서 보여준다. 이동성이 증가할수록 연결 경로의 단절 횟수가 다 같이 증가한다. 하지만 기존 AODV와 M-AODV 방식에 비하여 목적지 개시 플러딩을 사용하는 제안 방식의 경우 단절 횟수가 현저히 낮아짐을 볼 수 있다. 이는 주기적으로 목적지 개시 플러딩을 하는 경우 최단 경로 주변에 다중 경로를 형성하여 사

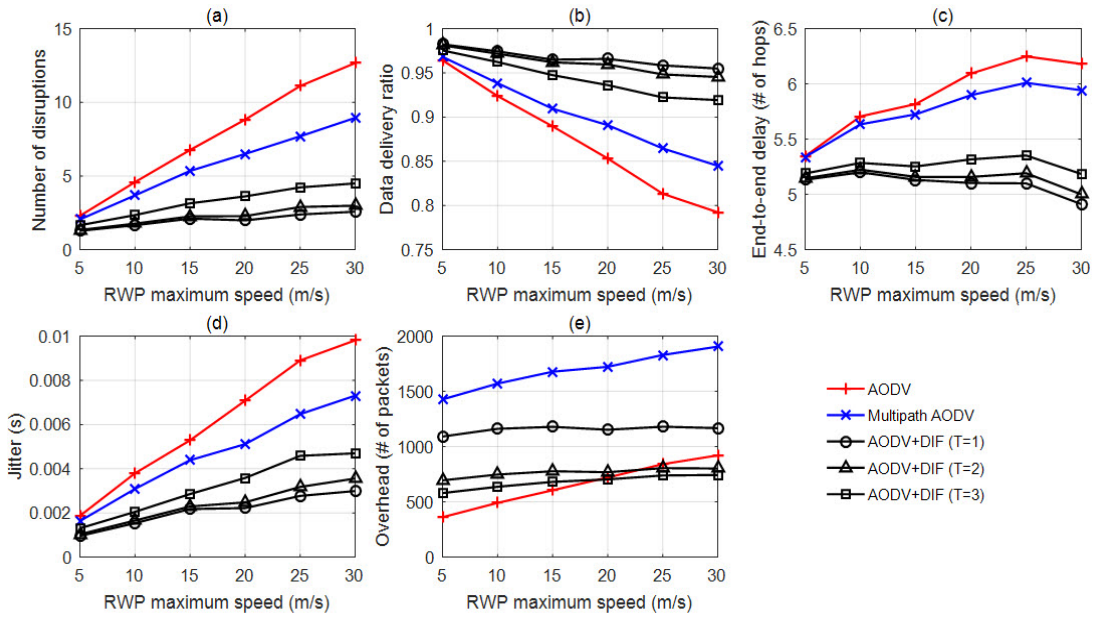


그림 9. RWP 이동성 모델의 최대 속도에 따른 제안 라우팅 프로토콜의 성능: (a) 단절 횟수, (b) 데이터 전달률, (c) 종단 간 지연, (d) 지터, (e) 오버헤드

Fig. 9. Performances of the proposed routing protocols according to the maximum speed in the RWP mobility model: (a) number of disruptions, (b) data delivery ratio, (c) end-to-end delay, (d) jitter, (e) overhead

용 중인 경로가 단절되어도 다른 주변 경로를 사용하여 데이터가 전송되기 때문이다. 목적지 개시 플러딩 주기(T)가 적을수록 경로 유지 관리를 보다 자주 하게 되므로 단절 횟수가 가장 낮음을 볼 수 있다.

데이터 전달률은 소스에서 발생한 전체 데이터 패킷 개수 대비 목적지에 도착한 데이터 패킷의 개수로 정의된다. 각 노드는 경로 단절시 목적지로 전달해야 할 데이터 패킷을 보관 후 재전송하지 않고 바로 폐기하기 때문에 데이터 전달률은 단절 횟수 성능에 직접적으로 영향을 받는다. 따라서 데이터 전달률의 경우 단절 횟수 성능과 비슷하게 제안 방안의 성능이 크게 개선됨을 볼 수 있다.

종단 간 지연은 목적지까지 성공적으로 도착한 데이터 패킷이 전달되는데 걸리는 시간으로 데이터 전송에 사용되는 경로가 얼마나 짧은지를 나타낸다. 제안 방안의 경우 기존 AODV 및 M-AODV 보다 낮은 지연 성능을 보여준다. 또한 목적지 개시 플러딩 주기가 짧을수록 새로운 경로를 자주 갱신하기 때문에 더 낮은 지연 성능을 갖는다. 일반적으로 모든 라우팅 프로토콜에서 이동성이 증가할수록 지연이 증가하는데, 본 결과에서는 지연 성능에 성공적으로 전달된 데이터 패킷만을 고려하고 경로 단절시 새로운 최단 경로를 찾기 때문에 25 m/s 이상의 속도에서 모든 라우팅

프로토콜의 종단 간 지연이 더 낮아지는 경우가 발생한다.

지터 성능은 목적지에 도착하는 데이터 패킷의 간격 변화로 다음과 같이 정의된다^[9].

$$jitter = \frac{\sum_{i=2}^n |(t_i - t_{i-1}) - (t_{i-1} - t_{i-2})|}{n-2} \quad (1)$$

여기에서 t_i 는 i 번째 데이터 패킷의 도착시간을 나타내고, n 은 목적지에 도착한 전체 데이터 패킷의 개수를 나타낸다. 마찬가지로 이동성이 증가함에 따라 패킷의 도착시간 간격의 변화가 커져 지터 성능이 나빠지게 된다. 제안 방안의 경우 $T=2$ 이상인 경우에 기존 AODV 및 M-AODV 보다 더 좋은 지터 성능을 보여준다.

오버헤드 성능은 라우팅 프로토콜에서 발생하는 전체 제어 패킷의 개수로 확인할 수 있다. 기존 AODV의 경우 이동성이 증가함에 따라 경로 단절이 많이 발생하고 이로 인한 플러딩 발생으로 인하여 오버헤드가 선형적으로 크게 증가한다. M-AODV의 경우 다중 경로 관리시 필요한 RREP의 전송 개수가 증가하여 기본적으로 AODV 보다 제어 패킷의 전송 개수가 많으며 AODV와 마찬가지로 이동성 증가에 따

라 오버헤드가 선형적으로 증가한다. 반면 제안 방안은 이동성이 증가하여도 오버헤드가 크게 증가하지 않아 주기적으로 목적지 개시 플러딩을 수행함에도 불구하고 최대 속도 20 m/s 이상의 경우에 더 낮은 오버헤드 성능을 보인다. 하지만 이동성이 낮은 경우에 제안 방안은 AODV 보다 오버헤드가 높는데, 이는 제안 방안이 기존 AODV와는 달리 주기적으로 목적지 개시 플러딩을 수행하기 때문이다. 따라서 제안 방식의 경우 오버헤드를 최대한 줄이기 위해서는 이동성이 낮은 경우에는 플러딩 주기를 크게 하여 목적지 개시 플러딩 빈도를 줄이도록 해야 한다.

V. 결 론

제안 라우팅 프로토콜은 기존 AODV 프로토콜에 경로 유지관리 프로토콜을 추가하여 AODV의 단점을 극복하고 라우팅 성능을 개선하였다. 이를 위하여 먼저 엮듣기 동작을 통하여 오버헤드 없이 목적지에 대한 추가적인 라우팅 정보를 얻고, 소스-목적지 최단 경로 주변의 1-홉 이웃 노드를 결정한다. 이후 토폴로지 변화에 따라 최신의 경로 정보를 파악하기 위하여 목적지에서 플러딩을 시작한다. 이 목적지 개시 플러딩 방식은 최단 경로 주변의 1-홉 이웃 노드들만 플러딩에 참여하도록 제한하여 기존 플러딩 오버헤드를 현저히 줄이고, 사용 중인 전송 경로의 단절 및 변화시 주변의 새로운 최단 경로로의 자연스러운 라우팅이 가능하게 해준다. 시뮬레이션 결과 이러한 제안 방식의 다중 경로 관리 방식은 링크 단절 빈도를 줄이고 네트워크에 발생하는 불필요한 제어 패킷 전송을 줄이면서 소스-목적지간 데이터 지연 성능을 향상시킨다.

References

- [1] L. Hanzo and R. Tafazolli, "A survey of QoS routing solutions for mobile ad hoc networks," *IEEE Commun. Surv. & Tuts.*, vol. 9, no. 2, pp. 50-70, 2007.
- [2] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mob. Comput. Syst. and Appl.*, p. 90, Washington DC, USA, Feb. 1999.
- [3] C. E. Perkins, *Ad hoc On-Demand Distance Vector (AODV) Routing*, IETF RFC 3561, 2003.
- [4] J. Zhou, H. Xu, Z. Qin, Y. Peng, and C. Lei, "Ad hoc on-demand multipath distance vector routing protocol based on node state," *Commun. and Netw.*, vol. 5, no. 3, pp. 408-413, Sept. 2013.
- [5] N. Jaisankar and R. Saravanan, "An extended AODV protocol for multipath routing in MANETs," *IACSIT Int. J. Eng. and Technol.*, vol. 2, no. 4, pp. 394-400, Aug. 2010.
- [6] T.-H. Kim, S.-H. Chung, S.-Y. Kang, and Y.-H. Yoo, "A node-disjoint multi-path routing protocol in AODV-based mobile ad-hoc networks," *J. KICS*, vol. 34, no. 12, pp. 1371-1379, Dec. 2009.
- [7] S. A. Ade and P. A. Tijare, "Performance comparison of AODV, DSDV, OLSR and DSR routing protocols in mobile ad hoc networks," *Int. J. Inf. Technol. and Knowledge Management*, vol. 2, no. 2, pp. 545-548, Jan. 2010.
- [8] M. Ayash, M. Mikki, and K. Yim. "Improved AODV routing protocol to cope with high overhead in high mobility MANETs," in *Proc. IEEE Sixth Int. Conf. Innovative Mob. and Internet Serv. in Ubiquitous Comput. (IMIS)*, pp. 244-251, Palermo, Italy, Sept. 2012.
- [9] H.-H. Choi, B. Roh, M. Han, and J.-R. Lee, "Regional route maintenance protocol based on local pheromone diffusion for mobile ad hoc networks," *Springer Mob. Netw. and Appl.*, Online published first, Sept. 2017.
- [10] J. Hsu, et al., "Performance of mobile ad hoc networking routing protocols in realistic scenarios," in *Proc. IEEE Military Commun. Conf.*, pp. 1268-1273, Boston, USA, Oct. 2003.
- [11] H.-H. Choi, B. Roh, H. S. Choi, and J.-R. Lee, "Bio-inspired routing protocol for mobile ad hoc networks," *J. KICS*, vol. 40, no. 11, pp. 2205-2217, Nov. 2015.
- [12] F. Ducatelle, "Adaptive routing in ad hoc wireless multi-hop networks," Dissertation for the Degree of Doctor of Philosophy, Lugano, Switzerland, May 2007.

- [13] H.-H. Choi, H.-G. Lee, and J.-R. Lee, "A new energy-aware source routing protocol for maximization of network lifetime in MANET," *IEICE Trans. Inf. and Syst.*, vol. E97-D, no. 2, pp. 335-339, Feb. 2014.
- [14] M. Woo, "An ant-based routing method using enhanced path maintenance for MANETs," *J. KICS*, vol. 35, no. 9, pp. 1281-1286, Sept. 2010.

최 현 호 (Hyun-Ho Choi)



2001년 2월 : KAIST 전기 및
전자공학과 졸업
2003년 2월 : KAIST 전기 및
전자공학과 석사
2007년 2월 : KAIST 전기 및
전자공학과 박사
2007년 3월~2011년 2월 : 삼성
종합기술원 전문연구원

2011년 3월~현재 : 국립환경대학교 전기전자제어공
학과 부교수

<관심분야> 차세대 이동통신시스템, 애드혹 네트워
크, 저전력 통신 프로토콜, 매체접속제어, 분산자
원관리, 생체모방 알고리즘