

# 무선 모바일 애드 혹 네트워크에서 AODV 기반 에너지 효율 라우팅 프로토콜<sup>§</sup>

유대훈<sup>○</sup> 최웅철<sup>\*</sup> 이승형<sup>\*\*</sup> 정광수<sup>\*\*</sup>

광운대학교 컴퓨터 과학과<sup>\*</sup> 광운대학교 전자정보공학부<sup>\*\*</sup>

yo2dh<sup>○</sup>@cs.kw.ac.kr, wchoi@daisy.kw.ac.kr, {rhee, kchung}@kw.ac.kr

## AODV-Based Energy Efficient Routing Protocol in Wireless Mobile Ad Hoc Network

Daehun Yoo<sup>○</sup>, WoongChul Choi<sup>\*</sup>, Seung Hyong Rhee<sup>\*\*</sup>, KwangSue Chung<sup>\*\*</sup>

Department of Computer Science, KwangWoon University<sup>\*</sup>

Department of Electronics Engineering, KwangWoon University<sup>\*\*</sup>

### 요 약

무선 모바일 애드 혹 네트워크에서 에너지 효율을 위한 많은 라우팅 알고리즘이 연구되어 왔다. 이러한 제안된 다른 알고리즘들을 살펴보면, 에너지 효율 뿐만 아니라 성능의 향상을 위한 노력도 기울여 왔다. 일반적으로 목적지 노드가 전송받은 RREQ의 메트릭을 통하여 에너지 효율적인 경로를 선택 하게 되는데, 이것은 결국 데이터 전송 지연이 발생하게 된다. 그리고 경로가 설정된 이후에 경로를 유지하는데 있어서, 노드의 이동이나 배터리의 소진으로 링크가 끊기게 되면 경로를 다시 찾는 과정에서 제어 패킷을 다시 플러딩 하기 때문에 여러 노드의 에너지 소비를 초래하게 된다.

본 논문에서는 이러한 문제들을 해결하기 위해 기존의 AODV(On-Demand Distance Vector Routing) 프로토콜을 기반으로 다음과 같은 방법을 제안한다. 먼저 RREQ를 받은 노드가 남은 배터리의 양을 고려하여 전송의 여부를 결정하는 방법을 제안하고, 경로를 유지하는데 있어서 전송의 세기를 통해 노드의 이동성을 인지하고 링크가 끊기기 전에 다른 노드로 경로를 대체함으로써 경로 재설정을 통한 에너지 소비를 줄이는 방안을 제안한다.

### 1. 서 론

무선 모바일 애드 혹 네트워크에서 에너지 효율을 위한 많은 라우팅 알고리즘이 연구되어 왔다[1]. 이러한 알고리즘은 일반적으로 에너지 효율을 측정할 수 있는 메트릭을 통하여 경로를 선택하거나, 배터리의 양을 고려하여 네트워크의 생존 시간을 늘리기 위한 방법 등이 제안되었다. 여기서 본 논문에서는 에너지를 인지하여 네트워크의 생존시간을 늘리기 위해 경로를 정하고 에너지 효율을 위해 경로를 유지 하는 방법에 대해 제안하고자 한다.

많은 선행 연구들이 AODV (On-Demand and Distance Vector Routing) 프로토콜을 수정하여 에너지 효율적인 방법들을 적용하였다[2]. AODV 프로토콜은 노드의 이동성에 있어서도 좋은 성능을 보이는 라우팅 알고리즘이다 [3]. 이 알고리즘은 데이터 전송이 필요한 경우 경로를 탐색하게 되는 On-Demand 방식이다. 경로 탐색을 위해 송신 노드는 RREQ(Route Request) 패킷을 주변 노드에 플러딩하게 된다. RREQ를 받은 주변 노드들은 송신

노드와 RREQ를 전송한 노드의 정보를 기록하여 경로 정보를 기록하고 다시 주변 노드에게 Broadcasting을 한다. 이러한 과정을 통해 목적지 노드까지 RREQ를 받게 되면, 목적지 노드는 RREP(Route Reply) 패킷을 Unicast로 송신 노드에게 전송한다. 경로가 설정이 되면 이 경로를 유지하기 위해 중간 노드들은 Hello 패킷을 주기적으로 전송하여 주변노드에게 자신의 존재를 알리게 되는데, 만약 일정 시간동안 Hello 패킷을 받지 못하면, 그 노드가 전송 범위 안에 없었었다고 판단하고, RERR(Route Error) 패킷을 송신 노드에게 전송하게 된다. RERR를 받은 송신노드는 다시 RREQ를 다시 플러딩함으로써 경로 탐색 과정을 다시 실행하게 된다. 이렇게 링크가 끊기게 되면, 경로를 재설정하는 과정을 거치게 되는데 이는 데이터 전송의 지연을 더 생기게 할 뿐만 아니라 많은 양의 에너지 소비도 가져오게 된다. 여기서 경로가 설정된 노드 간에 링크가 끊기는 경우는 노드가 전송범위 밖으로 이동하거나, 배터리 소비에 의해 노드가 통신을 하지 못하게 되는 경우가 대표적이다.

본 논문에서는 이러한 문제를 예방하기 위해 배터리의 남은 양과 전송 세기를 통해 이동성을 예측하여 링크가 끊기기 전에 다른 노드로 경로를 설정하게 함으로써 경로 재설정 과정을 최소한으로 함으로써 에너지 효율적으로 데이터를 전송 할 수 있는 방안을 제안한다. 먼저 2

<sup>§</sup> 본 연구는 한국과학재단 특정기초연구 [R01 - 2005 - 000 - 10934 - 0 (2006)] 의 지원에 의해 수행되었음.

장에서는 관련 연구에 대해 논의 하고 3장에서는 문제를 인지한다. 그리고 4장에서는 문제 해결을 위한 라우팅 알고리즘을 제안하고, 5장에서는 결론과 향후과제에 대해 논의한다.

## 2. 관련 연구

### 2.1 PEER 프로토콜

PEER(A Progressive Energy Efficient Routing) 프로토콜은 Minimum Energy 라우팅 프로토콜로서, 라우팅 오버헤드와 경로 설정 시간을 줄이고, 설정된 경로 유지를 효율적으로 관리하는 방법을 제안하였다[4].

기본적으로 On-Demand 라우팅 방식을 사용하기 때문에 송신 노드는 경로를 찾기 위해 RREQ(Route Request Packet)를 Broadcasting하게 된다. 여기서 사용하는 RREQ에는 홉의 수와 노드 간에 패킷을 전송하기 위한 필요 에너지에 대한 정보를 추가 기록 한다. 즉, 최단 거리이면서, 에너지의 소비가 가장 적은 경로를 선정하도록 한다. 목적지 노드가 RREQ를 전송 받게 되면, 바로 RREP(Route Reply Packet)를 전송하지 않고, 어느 일정 시간동안 다른 RREQ를 기다린다. 이 시간 안에 다른 RREQ를 받게 되면, 그 경로를 추가하고 다시 일정 시간을 기다린다. 이렇게 해서 여러 경로 중 하나를 선정하여 RREP를 송신 노드로 전송하게 된다. 하지만 경로가 선정된 이후에 최적의 경로가 있는 RREQ가 전송되어 질 수도 있기 때문에 가장 효율적인 경로를 선정하지 못할 수도 있게 된다. 이를 해결하기 위해 PEER 프로토콜에서는 RREP를 Overhearing 하는 노드들을 이용하여 병렬적으로 빠른 경로의 수정이 가능 하도록 한다. RREP를 Overhearing한 노드들은 자신이 가지고 있는 경로가 RREP를 받은 경로보다 더 효율적인지 검사를 하고, 만약 더 최적의 경로라면 Better Route Message를 Broadcasting하여 주변노드에게 알리게 된다. 또한 설정된 경로를 유지하기 위하여 경로를 삭제(Remove), 대체(Replace), 삽입(Insert) 하는 방법을 제시하였다. 먼저 각 노드들은 이웃 노드의 RTS, CTS 프레임 혹은 Broadcast 패킷을 통하여 전송의 세기를 측정하게 된다. 이 후, IP 헤더에 이러한 정보를 저장하여 전송함으로써 링크의 상태를 모니터링 하게 된다. 노드들은 링크 에너지 테이블을 가지고 모니터링 한 정보를 비교하게 되고, 그 값들에 따라 다음과 같은 작업을 한다. 첫 번째로 경로 삭제(Remove) 작업은 그림 1 (가)에서와 같이 A 노드와 C 노드의 경로가 A, B, C로 되어 있을 때, 노드의 이동으로 A 노드에서 바로 C 노드로 전송하는 것이 에너지 소비가 더 효율적이려면, B 노드를 경유하는 경로를 삭제하는 작업이다. 두 번째로 경로 대체(Replace) 작업은 그림 1 (나)에서와 같이 A 노드에서 B 노드로 가는 경로가 B 노드로 가는 것보다 D 노드로 가는 것이 더 효율적일 때, 경로를 B 노드에서 D 노드로 대체하는 작업이다. 마지막으로 경로 삽입(Insert) 작업은 그림 1 (다)에서와 같이 A 노드에서 C 노드로 가는 경로보다 B 노드를 경유하는 것이 더 효율적인 경로 일 때, B 노드로 가는 경로를 추가하는 작업이다. 이렇게 세 가지 방

법으로 경로를 유지하며, 몇 가지의 규칙을 가진다. 우선 경로 대체(Replace)와 경로 삭제(Remove) 작업은 처리의 효율을 위해 두 개의 홉까지로 제한 한다. 그리고 경로 대체(Replace)와 경로 삽입(Insert) 작업은 주변 노드에게 변경된 경로 정보를 알리기 위해 특정한 제어 메시지(Control Message) 패킷을 전송하여 작업을 진행한다. 마지막으로 경로 대체(Replace)와 경로 삭제(Remove) 작업은 경로 삽입(Insert) 작업 보다 높은 우선순위를 가진다. 즉, 경로 삽입(Insert) 작업을 요청 받으면, 일정 시간 동안 다른 요청이 있는지 검사한 후, 다른 요청을 받으면 Insert 작업을 하지 않고 요청 받은 작업을 수행한다. 그리고 경로 대체(Replace)와 경로 삭제(Remove) 작업이 같이 오면, 에너지 효율이 더 높은 작업을 선택한다.

본 논문에서는 PEER 프로토콜에서 사용하는 경로 수정 방식을 사용하였다. 여기서 사용하는 방식은 데이터 전송에 대한 Overhearing를 사용하는데, 이 경우 데이터를 양방향으로 전송하지 않으면 충분한 정보를 받지 못하게 된다. 또한 제한한 프로토콜에서는 이동성을 고려해야 함으로써 주기적인 정보가 필요하게 된다. 이를 위해 Hello Message를 이용한 전송 세기를 사용하여 적용하였다.

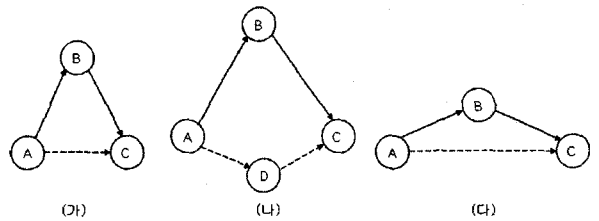


그림 1 (가) Remove, (나) Replace, (다) Insert

### 2.2 LEAR 프로토콜

LEAR(Localized Energy Aware Routing) 프로토콜은 잔여 배터리의 양을 고려한 라우팅 알고리즘이다[5]. 이 프로토콜은 에너지 효율적인 경로를 목적지가 선택하기 위해서 기다리는 문제를 해결하고자 하였다. 또한, 라우트 캐시의 장점으로 성능의 향상을 도모하기 위해 에너지를 고려하면서도 라우트 캐시를 사용할 수 있는 방법을 제안한다. 이 프로토콜은 다음과 같이 동작한다.

먼저 ROUTE\_REQ를 풀러딩 할 때 에너지의 소비를 줄이기 위해 목적지까지 Broadcasting 하는 중간노드가 자신의 배터리의 양을 체크하여 ROUTE\_REQ의 전송 여부를 결정한다. 이것은 미리 결정된 Threshold 값에 좌우된다. 이 값보다 배터리의 양이 적으면 중간노드는 ROUTE\_REQ를 전송하지 않는다. 만약에 목적지 노드까지 이러한 문제 때문에 ROUTE\_REQ가 도달하지 않아 경로를 설정하지 못했다면, 송신 노드는 재전송 ROUTE\_REQ로 설정하여 다시 재전송한다. 중간 노드가 이러한 ROUTE\_REQ를 받으면 현재 이미 결정된 Threshold 값 보다 작은 값으로 비교하여 목적지 노드까지 ROUTE\_REQ가 도달할 수 있도록 한다. 이렇게

ROUTE\_REQ의 Broadcasting하는 노드의 수를 줄임으로써 에너지 소비를 줄이게 된다.

이 프로토콜은 기존의 라우트 캐시도 배터리의 양을 고려하게 만들기 위해서, ROUTE\_CACHE 라는 패킷을 제한한다. 기존의 라우트 캐시 방식은 중간 노드가 ROUTE\_REQ를 받았을 때, 바로 ROUTE\_REP를 전송하게 되는데 이 경우 LEAR 프로토콜에서 제한한 배터리의 양의 고려를 할 수가 없게 된다. 그렇기 때문에 그림 2와 같이 노드 B가 목적지 노드 D까지의 경로를 C1과 C2를 통해 전송하는 정보를 라우트 캐시가 가지고 있을 때, 송신 노드가 ROUTE\_REQ를 전송하면 B 노드까지는 Broadcasting으로 전송을 하고 노드 B 이후부터는 목적지 노드 D까지 ROUTE\_CACHE 패킷을 Unicast로 전송한다. 이렇게 Unicast로 전송하는 과정에서 중간 노드는 제한한 방식으로 이미 결정된 Threshold 값과 잔여 배터리의 양을 비교하여 그 이상이 되는 경우에 한해서 ROUTE\_CACHE를 목적지 노드까지 전달하게 된다. 만약 목적지 노드까지 ROUTE\_CACHE를 전달하는 과정에서 잔여 배터리의 양이 Threshold 값 보다 적은 경우에는 ROUTE\_CACHE를 송신한 노드에게 CANCEL\_ROUTE\_CACHE 패킷을 전송한다. 이를 받은 노드는 라우트 캐시에서 삭제 경로 정보를 삭제하게 된다. 또한 목적지 노드에게는 DROP\_ROUTE\_CACHE를 전송함으로써 기존에 가지고 있는 라우팅 정보를 삭제하도록 만든다.

본 논문에서는 LEAR 프로토콜에서 사용하는 지역적 에너지 인지 방식을 사용하였다. 이 방법은 경로를 탐색할 때 플러딩하는 패킷의 수를 감소시킬 수 있는 장점을 가진다. 하지만 배터리의 양에 따라 Threshold 값이 달라지기 때문에 하드웨어의 영향을 받아 값이 달라지게 된다. 그래서 값을 설정하는데 있어서 어려움이 있다. 또한, 이 방식대로 ROUTE\_REQ를 전송하지 않아서 송신 노드가 경로를 찾지 못하게 되면 라우트 탐색 과정을 다시 거쳐야 하기 때문에, 더 많은 에너지 소비가 일어날 수가 있다. 그래서 제안한 프로토콜에서는 Threshold 값을 배터리 양의 비율에 따라 적용 할 수 있도록 했다. 또한, ROUTE\_REQ를 무조건 전달하지 않은 방식이 아니라, 주변노드의 ROUTE\_REQ를 받은 수를 고려하여 전달할 수 있도록 수정하여 적용하였다.

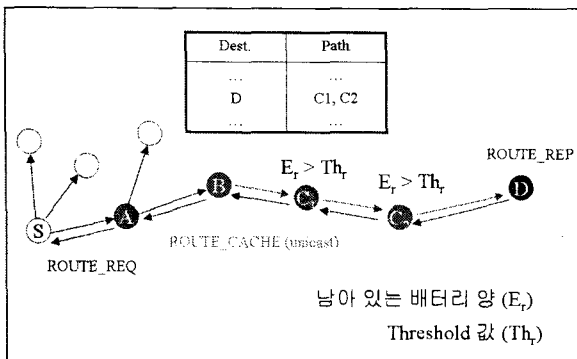


그림 2 라우트 캐시 작동 원리

### 3. 문제점 인지

On-Demand 라우팅 알고리즘은 이동성이 있는 네트워크 환경에서 좋은 성능을 보여준다. 이 중에서도 많이 언급되고 있는 AODV 라우팅 프로토콜에 대해 논의를 하고자 한다. 먼저 이 알고리즘은 경로를 탐색하는데 있어서 플러딩을 사용한다. 이것은 목적지 노드의 위치가 어디 있는지 모르기 때문인데, 이는 많은 에너지 소비를 가져온다. 또한 이는 경로 유지 과정에서도 문제를 발생시킨다. 경로를 유지 하는 노드들 중 하나가 노드의 이동이나 배터리의 소진 등의 이유로 통신을 할 수 없는 상태가 되면, 송신 노드로 RERR 패킷을 전송하여 링크가 끊어졌음을 알린다. 그리고 송신 노드는 목적지 노드까지의 경로를 찾기 위해 경로 탐색 과정을 가지게 되는데, 위에서 언급한 것처럼 또 다시 에너지 소비를 가져오게 된다. 또한, 2.2절에서 언급한 대로 에너지 효율적인 경로를 찾기 위해서 여러 경로의 정보를 RREQ를 통해 목적지가 받게 되는데, 목적지는 더 좋은 경로를 찾기 위해서 일정 시간동안 다른 RREQ를 기다리게 된다. 이는 데이터 전송 지연 시간이 더 늘어나기 때문에 성능에 영향을 미치게 된다.

### 4. 제안된 라우팅 프로토콜

제안된 프로토콜은 경로 탐색 과정에 있어서 제어 패킷의 수를 줄이기 위해 LEAR 프로토콜의 방식을 사용한다. 배터리가 거의 없는 상태의 노드라면 경로가 설정된다고 하더라도, 곧 통신할 수가 없는 상태가 되기 때문에 다시 경로 탐색 과정을 할 것이기 때문이다. LEAR 프로토콜에서는 이를 위해 배터리 양에 따른 Threshold 값을 사용하는데 이 값은 하드웨어에 따라 설정을 달리 해주어야 하기 때문에 어려움이 따를 수 있다. 그래서 제안된 프로토콜에서는 배터리 양에 따른 Threshold 값으로 설정하지 않고 전체 배터리의 양과 남아 있는 배터리의 양의 비율로써 Threshold 값을 설정하고자 한다. 즉, RREQ를 받은 노드가 이 값보다 작다면, RREQ를 Broadcasting 하지 않고 기다리도록 한다. 여기서 기존 LEAR 프로토콜은 기다리지 않고 버리게 되는데, 제안된 프로토콜은 기다린 후, 다음과 같은 작업을 진행한다. 배터리의 양이 적은 노드는 주변노드의 같은 종류의 RREQ를 주변 노드가 다시 Broadcasting 하는 것을 Overhearing 하여 받은 횟수를 기록한다. 여기서 우리는 N을 최소한에 받아야 할 RREQ 패킷의 횟수로 정의한다. 만약 일정 시간동안 N의 값만큼 RREQ를 받지 못하면, 배터리 때문에 전송을 기다리고 있는 노드는 RREQ를 전송하게 된다. 이는 LEAR 프로토콜의 경우처럼 송신 노드가 경로를 찾지 못하여 RREQ를 재전송 하는 것을 예방한다.

그림 3의 경로 탐색 과정을 살펴보면, 송신 노드 S는 경로를 찾기 위해 RREQ를 주변 노드에게 전송한다. 노드 A는 자신의 배터리 비율이 Threshold 값보다 크기 때문에 다시 주변 노드로 RREQ를 Broadcasting 한다. 여기서 RREQ를 전송 받은 노드 B는 배터리 비율이 Threshold 값 보다 작기 때문에, 전송하지 않고 기다리

게 된다. 그 이후 일정시간 동안 주변노드가 전송하는 RREQ를 Overhearing하여 그의 수를 라우트 캐시 테이블에 누적하여 기록 한다. 일정 시간 동안 누적된 수가  $N_r$ 보다 적다면, 노드 B는 RREQ를 주변 노드에게 Broadcasting 하게 된다. 이는 LEAR 프로토콜과 비교해서 RREQ를 재전송하는 것을 줄여준다. 만약, LEAR 프로토콜의 사용 할 때, 최악의 경우 여러 홉을 거쳐 거의 목적지 노드 D 가 가까이에 있는 노드가 배터리의 양 때문에 전송하지 못한다면, RREQ의 재전송은 많은 에너지 소비를 일으키는 문제를 발생하게 된다.

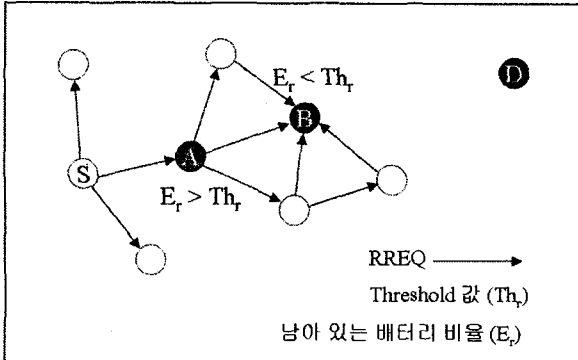


그림 3 경로 탐색 과정에서 RREQ 전송 과정

다음으로 경로 유지를 위해 전송 세기와 남아 있는 배터리 비율을 라우트 캐시에 추가로 기록한다. 경로를 바꾸는 작업이기 때문에 데이터를 전송할 수 있는 충분한 배터리의 양이 있어야 함으로, Threshold 값 보다 에너지 비율 값이 큰 노드만이 해당된다. 전송 세기를 기록하는 노드와 절차는 다음과 같다. 먼저 RREQ를 전송 받은 노드는 전송 받은 노드의 정보와 함께 전송 세기를 기록하여 자신의 주변 노드에게 Broadcasting 한다. 그 이후 RREP를 주변 노드에게 받으면, 마찬가지로 RREP를 전송한 노드의 정보와 함께 전송의 세기를 기록한다. 이렇게 되면 결국 경로가 설정된 노드의 주변 노드는 두 노드가 연결된 전송 세기를 알 수가 있게 된다. 예를 들어 설명하면, 그림 4에서와 같이 송신 노드 S에서 RREQ를 전송하기 시작하여, 노드 A와 B를 거쳐 목적지 노드 D로 전달하는 상황이다. 주변 노드 N은 노드 B의 RREQ에서 노드 A에서 B까지의 전송 세기의 정보를 알게 된다. 그리고 RREP를 통해 노드 A의 RREP에서 노드 B에서 A까지의 전송 세기에 대한 정보를 알게 된다. 이렇게 두 노드의 전송세기와 자신과 경로가 설정된 두 노드간의 전송 세기를 알 수 있게 된다. 이제 노드 A와 노드 B는 경로 상태를 체크하기 위해서 Hello 패킷에 자신의 배터리의 비율을 포함하여 전송한다. 이를 전송 받은 주변 노드는 전송 세기를 통해서 이동성과 배터리의 비율을 검사한다.

경로를 대체하는 과정은 다음과 같다. 첫 번째로, 배터리 양에 의한 노드 대체 작업이다. PEER 프로토콜에서 언급한 경로 대체(Replace) 작업과 유사하다. 그림 4에서 만약 주변 노드 N이 노드 A와 밀접하게 있다면, A의

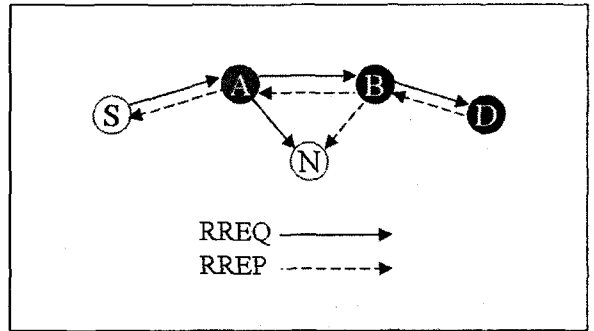


그림 4 경로가 설정된 노드의 전송 세기의 Overhearing

전송세기가 매우 클 것이고, 송신 노드 S와 노드 B의 Hello 패킷을 받을 수가 있게 된다. 여기서 최대 전송 세기의 값으로  $P_{max}$  를 정의한다. 주변 노드 N이 전송세기가  $P_{max}$  값이 보다 크다고 할 때, 노드 A의 Hello 패킷으로 배터리 비율이 이미 설정된 Threshold 값보다 적고 주변 노드 N이 많은 양의 배터리 비율을 가지고 있다면, 주변 노드 N은 NODE\_REPLACE 패킷을 전송한다. 이 패킷 안에는 송신노드 ID, 목적지 노드 ID, 대체 노드 ID, 자신의 ID가 포함되어 있다. 이를 전송 받은 노드 S와 B는 경로 정보를 수정하고 주변 노드였던 N를 통해 데이터를 전송하게 된다. 두 번째로, 이동성에 따른 노드 삽입 작업이다. 이것은 PEER 프로토콜의 경로 삽입(Insert) 작업과 유사하다. 이 작업을 하는 시기는 그림 5와 같이 경로가 설정된 이후에 목적지 노드 D가 이동한 경우이다. 이 때, 노드 D의 이동으로 노드 B는 결국 통신이 실패되어 경로가 끊기게 된다. 이를 예방하기 위해서 다음과 같은 노드 삽입 작업을 한다. 목적지 노드 D의 Hello 패킷을 통해 노드 B는 전송 세기를 기록하면서 전에 기록한 전송 세기와 비교하여 증감을 기록한다. 여기서 최소 전송 세기의 값  $Th_p$ 를 정의한다. 노드 B는 전송 세기의 증감 값으로 다음의 전송 세기를 예측하고 전송 세기가  $Th_p$ 보다 작아지게 되면, 다음에 전송할 Hello 패킷에 노드 D의 ID와 세기를 포함하여 전송한다. 이것을 전송 받은 주변노드 N이 목적지 노드 D에서 받은 전송 세기가 크다면 노드 N은 NODE\_INSERT 패킷을 전송한다. 이 패킷 안에는 송신노드 ID, 목적지 노드 ID, 이전 노드 ID, 다음 노드 ID, 자신의 ID를 Broadcasting 하여 주변 노드에게 알린다. 이것을 받은 노드 B와 노드 D는 자신의 경로 정보를 수정하게 되고 데이터를 노드 N를 통해 통신하게 된다.

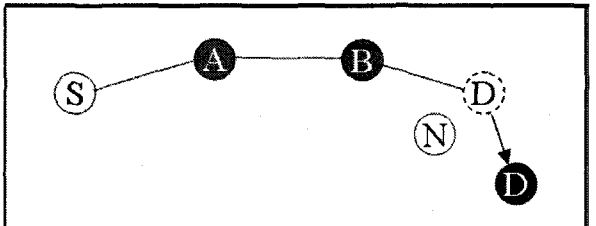


그림 5 경로가 설정된 이후에 노드의 이동

마지막으로 노드 삭제 작업이다. 이 작업은 송신 노드와 목적지 노드의 경로가 설정된 주변 노드에서만 동작할 수 있다. 그림 6를 살펴보면, 그림 5와 같이 노드가 이동하는 상황이지만, 처리가 달라야 한다는 알 수 있다. 송신 노드 S가 노드 A로 이동하는 경우이다. 이 때, 송신 노드 S는 노드 A와 매우 근접해 있기 때문에 높은 전송 세기를 받게 된다. 받은 전송 세기가  $P_{max}$  값 보다 크면, 송신 노드 S는 NODE\_REMOVE 패킷을 주변 노드에게 알린다. 이 패킷 안에는 송신노드 ID, 목적지 노드 ID, 삭제 노드 ID, 자신의 ID를 포함한다. 이 패킷을 받은 노드 A와 B는 경로를 갱신하고, 송신 노드 A는 데이터를 노드 B를 통해 전송한다.

이렇게 세 가지의 작업을 통해 경로를 유지한다. 이 작업들은 PEER 프로토콜과 유사하지만, 데이터 패킷을 통해 받은 정보로만 비용을 갱신하기 때문에 경로의 변경이 늦어 질 수가 있다. 본 논문에서는 이를 위해 데이터 패킷뿐만 아니라 Hello 패킷을 통해 정보를 갱신하기 때문에 더 빠른 경로 변경을 할 수가 있다. 또한 전송 세기와 배터리의 양을 고려함으로써 이동성을 예측하기 때문에 PEER 프로토콜과는 구분된다.

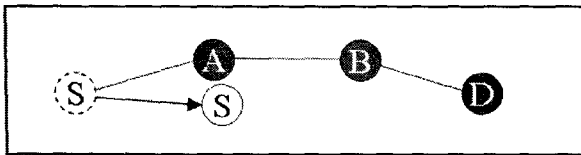


그림 6 경로가 설정된 이후에 노드의 이동

### 5. 결론 및 향후과제

모바일 애드 혹 무선 네트워크 환경에서 AODV는 인기가 많은 On-Demand 방식 프로토콜 중 하나이다. 이동성에서도 좋은 성능을 가지고 있지만, On-Demand 방식의 단점이라고 할 수 있는 경로 탐색 과정의 에너지 소비는 해결해야 할 과제 중 하나이다. 이러한 제어 패킷을 줄이기 위해 본 논문에서는 경로 탐색을 할 때 제어 패킷을 줄이기 위한 방법으로 LEAR 프로토콜의 방식을 수정하여 적용하였다. 또한 경로가 끊기기 전에 이동성과 배터리의 양을 고려하여 다른 경로로 대체함으로써 경로 재설정 과정을 최소화하고자 하였다. 이 프로토콜에 대한 신뢰성 있는 검증은 위해 Ns-2 시뮬레이션으로 구현 중에 있다.

향후 과제로 경로를 유지하는데 있어서 여러 네트워크 토폴로지 시나리오 테스트를 하고자 한다. 이를 통해 최악의 경우를 알아내고, 이를 바탕으로 신뢰성 있는 경로 유지를 위해 알고리즘을 수정, 보완 할 예정이다.

### 6. 참고 문헌

- [1] IETF MANET Working Group, <http://www.ietf.org/html.charters/manet-charter.html>
- [2] Hsin-mu Tsai, N. Wisitpongphan, O.K. Tonguz, "Link-Quality Aware Ad Hoc On-Demand Distance Vector Routing Protocol," *Wireless Pervasive Computing*, 2006 1st International Symposium on, pp. 16-18, Jan. 2006.
- [3] C. Perkins, E. Belding-Royer and S. Das, "Ad-Hoc On-demand Distance Vector(AODV) Routing," *IETF RFC 3561*, 2003.
- [4] J. Zhu and X. Wang, "PEER: a Progressive Energy Efficient Routing Protocol for Wireless Ad Hoc Networks," *INFOCOM.2005*, pp.1887-1896, Mar. 2005.
- [5] K. Woo, C. Yu, H.Y. Youn, and B. Lee, "Non-Blocking, Localized Routing Algorithm for Balanced Energy Consumption in Mobile Ad Hoc Networks," *MASCOTS 2001*, pp. 117-124, Aug. 2001.