

AODV 기반 모바일 Ad-hoc 네트워크의 노드 Disjoint 다중경로 라우팅 프로토콜

준회원 김 태 훈*, 종신회원 정 상 화*^o, 준회원 강 수 영*, 종신회원 유 영 환**

A Node-Disjoint Multi-Path Routing Protocol in AODV-based Mobile Ad-hoc Networks

Tae-Hun Kim* *Associate Member*, Shang-Hwa Chung*^o *Lifelong Member*,
Su-Young Kang* *Associate Member*, Young-Hwan Yoo** *Lifelong Member*

요 약

본 논문에서는 모바일 노드들의 움직임이 심한 MANET 환경에서 데이터의 전송을 보다 안정적이고 신뢰성 있게 하기 위해 소스 목적지 간의 다중 경로를 찾는 새로운 방법을 제시하고자 한다. 새로운 다중경로 라우팅은 먼저 AODV에 기반을 두어 주경로를 설정하고 주경로로 설정된 노드는 보조경로 탐색에 참여하지 못하도록 하여 주경로와 보조경로를 노드 비 겹침(disjoint)하게 형성한다. 또한 주경로가 설정된 후 바로 데이터 전송을 시작하고 데이터 전송이 되고 있는 중에 백그라운드로 보조경로 탐색을 실행하여 경로 탐색 시에 발생하는 데이터 전송의 지연을 주경로 탐색 시에만 발생하도록 한다. 주경로와 보조경로 중 하나의 경로가 단절되면 즉시 다른 경로로 데이터 전송을 하게 되고 단절된 경로는 경로유지를 통해 다시 노드 비 겹침 경로를 찾는다. Qualnet 기반 시뮬레이션을 수행한 결과, 제한한 라우팅 프로토콜을 사용한 경우 주경로가 단절되었을 때 62.5%정도 보조경로가 존재하여 보조경로를 통한 지연 없는 데이터 전송을 하며, AODV, AODV-Local Repair에 비해 패킷 전송률이 2~3% 향상되고, 종단 간 지연시간이 10%정도 줄어든다.

Key Words : AODV, Multi-path Routing, Node-disjoint, MANET, Transmission Delay

ABSTRACT

In this paper, we propose a new multi-path routing protocol to provide reliable and stable data transmission in MANET that is composed of high-mobility nodes. The new multi-path routing establishes the main route by the mechanism based on AODV, and then finds the backup route that node-disjoint from the main route by making add nodes in the main route not participate in it. The data transmission starts immediately after finding the main route. And the backup route search process is taking place while data is transmitted to reduce the transmission delay. When either of the main route or the backup route is broken, data is transmitted continuously through the other route and the broken route is recovered to node-disjoint route by the route maintenance process. The result of the simulation based on the Qualnet simulator shows that the backup route exists 62.5% of the time when the main route is broken. And proposed routing protocol improved the packet transmission rate by 2~3% and reduced the end-to-end delay by 10% compared with AODV and AODV-Local Repair.

* 이 논문은 2009년 교육과학기술부로부터 지원받아 수행된 연구임 (지역거점연구단육성사업/차세대물류IT기술연구사업단)

* 부산대학교 컴퓨터공학과 임베디드 네트워크 시스템 연구실(shchung@pusan.ac.kr)(^o:교신저자)

** 부산대학교 컴퓨터공학과 유비쿼터스 컴퓨팅 연구실(ymomo@pusan.ac.kr)

논문번호 : KICS2009-08-338, 접수일자 : 2009년 8월 7일, 최종논문접수일자 : 2009년 12월 10일

I. 서 론

모바일 애드 혹 네트워크(Mobile Ad-hoc Network: MANET)는 모바일 노드들로 구성되어 있는 무선 애드 혹 네트워크의 일종으로, 각 모바일 노드들은 기존의 유선 백본망이나 기지국에 의존하지 않고 동적으로 토폴로지를 구성하여 통신이 가능하다. MANET 환경의 모바일 노드들은 전력과 프로세싱, 메모리 자원에 제약이 있고 이동성이 높다. 모바일 노드들은 동적으로 네트워크 환경에 들어와서 토폴로지를 구성하기도 하고, 네트워크 환경에서 빠져 토폴로지를 변화시키기도 한다. 이러한 MANET 환경에서 모바일 노드간의 통신을 위해 다양한 라우팅 프로토콜이 제안되어 있으며, 애플리케이션 및 환경에 따라 적절한 라우팅 프로토콜이 사용되어야 한다.

그런데 기존의 네트워크 환경에서 각 노드들은 거의 토폴로지 구성을 변화시키지 않기 때문에 기존에 제시되어 있는 라우팅 알고리즘은 애드 혹 네트워크 내에서 토폴로지가 다양하게 변화하는 환경에는 적절하지 않다. 모바일 노드로 구성된 환경에서는 빈번한 토폴로지 변화로 인해 경로의 재검색이 자주 일어날 뿐만 아니라 안정적으로 경로를 유지하는 것이 불가능하다. 따라서 MANET 환경에서의 라우팅은 모바일 노드의 중요한 특징인 이동성을 고려하여, 데이터 전송의 안전성과 신뢰성을 제공할 수 있어야 한다. 이러한 필요성에 의해 기존의 단일경로 라우팅인 AODV^[1]를 기반으로 한 다양한 다중경로 라우팅이 제안되었으며, 그 예로 RREQ 패킷이 전송되어 온 노드를 기록하여 다중경로를 찾는 AODMV^[2], RREP 패킷이 전송되는 것을 오버히어링(overhearing)하여 다중경로를 찾는 AODV-BR^[3], AODVM^[4] 등이 있으며, 위의 다중경로 라우팅 프로토콜은 모두 링크 disjoint를 기반으로 하고 있다.

이 논문에서는 MANET 환경에서 AODV를 기반으로 한 새로운 다중경로 라우팅 알고리즘을 제안한다. 이 라우팅 알고리즘은 먼저 경로탐색을 통해 주경로를 찾고, AODV에서 RREQ ID를 이용하여 RREQ 패킷의 중복전송을 방지하는 방법을 사용하여 주경로와 노드 disjoint 한 보조경로를 찾는다. 이를 이용하여 기존의 AODV와 AODV-Local Repair보다 패킷 전송률을 높이고 중단 간 지연시간을 줄이고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련

연구를 소개하고, 3장에서는 본 논문이 제시하고자 하는 새로운 다중경로 라우팅 프로토콜에 대해서 설명하고 4장에서는 시뮬레이션을 통한 실험 결과를 분석하고 마지막으로 5장에서 결론을 내도록 한다.

II. 관련 연구

2.1 AODV

AODV^[1]는 애드 혹 네트워크의 on-demand 라우팅 프로토콜로서 각 노드들이 유지하고 있는 라우팅 테이블을 통해 홉 단위 통신을 한다. AODV의 경로 탐색 과정은 소스가 목적지로의 경로가 필요할 때 실행된다. 소스는 데이터를 보내려하는 목적지에 대한 경로를 찾기 위해 RREQ 패킷을 flooding한다. 노드가 RREQ 패킷을 받으면, 그 노드는 자신이 RREQ 패킷의 목적지인지 검사하여 자신이 목적지가 아니면 목적지를 향해 RREQ 패킷을 다시 flooding하고, 목적지가 맞으면 해당 노드는 RREP 패킷을 생성하여 RREQ 패킷이 온 경로를 따라 소스 노드로 RREP 패킷을 보낸다. 소스 노드가 RREP 패킷을 받으면 소스에서 목적지까지의 경로가 설정되고 이 경로를 통해 소스는 목적지로 데이터 전송을 시작한다.

AODV의 경로 유지 방식은 노드가 다음 홉으로의 전송 시 링크가 단절된 것을 발견하면 노드는 소스로 RERR 패킷을 보낸다. RERR 패킷은 링크를 따라가면서 각 노드의 라우팅 테이블에서 링크가 끊어진 목적지로 가는 모든 경로를 삭제한다. 소스가 RERR 패킷을 받고, 계속해서 목적지로의 경로가 필요하면 경로 탐색 과정을 재실행한다.

AODV의 local repair는 경로가 단절 되었을 때, 링크를 복구(repair)하기 위해 링크가 끊어진 직전 노드에서 목적지로 RREQ 패킷을 전송하여 경로를 재탐색한다. 이를 사용하면 기존의 경로 유지 방식보다 더 빠른 경로 재설정이 가능하고 경로가 단절되어 보내지 못한 데이터 패킷을 다시 생성된 경로를 통해 바로 재전송 할 수 있다.

2.2 다중경로 라우팅의 특징

다중경로 라우팅은 소스와 목적지 사이에 여러 경로를 찾는 프로토콜이다. 이런 소스와 목적지 간의 다양한 경로 설정은 자주 변화하고 예측이 어려운 MANET 환경에서 부하 분배(load-balancing), 장애 복구(fault tolerance) 그리고 더 높은 대역폭을 제공한다. 다양한 경로를 통해서 트래픽을 분산시킬

으로써 부하 분배가 가능하게 하여, 혼잡상황이나 병목현상을 감소시킬 수 있다. 장애 복구 관점에서 다중경로 라우팅은 소스와 목적지간의 여러 경로를 가지고 있으므로, 하나의 경로가 단절되어도 다른 경로를 통해 데이터 전송을 계속할 수 있다. 이를 통해 경로의 단절로 인해 발생하는 데이터 전송 실패와 지연시간을 줄일 수 있다.

다중경로 라우팅의 구성은 소스와 목적지 노드간의 다중 경로를 찾는 경로 탐색, 이러한 다중 경로를 계속 유지하기 위한 경로 유지 그리고 다중 경로에 데이터를 할당하는 트래픽 할당으로 구성되어 있다. 다중경로 라우팅 프로토콜은 경로 탐색을 위해 노드 비 겹침, 링크 비 겹침 또는 비 겹침이 없는 경로를 찾는다. 노드 비 겹침은 노드와 링크가 완전히 구분되는 경로를 찾는다. 링크 비 겹침은 공통되는 링크는 없지만 경로별로 겹치는 노드는 존재할 수 있다. 비 겹침이 없는 경우는 노드 또는 경로가 겹치게 된다. MANET 환경에서는 노드의 이동성으로 인해 링크가 단절되거나 노드의 실패로 일부 또는 전체 경로가 단절되는데, 비 겹침이 없거나 링크 비 겹침의 경우에는 주경로와 보조경로가 같이 사용하는 노드나 링크에 의해 경로가 단절되면 주경로와 보조경로가 같이 끊어지게 된다. 이에 반해 노드 비 겹침은 주경로와 보조경로가 서로 다른 노드와 링크를 사용하므로 하나의 경로가 단절되어도 다른 경로를 통해 데이터전송이 가능하다.

경로를 지속적으로 유지하기 위해 단일 경로 라우팅은 경로가 단절될 경우 경로 복구를 수행한다. 따라서 새로운 경로가 생성될 때까지 데이터 전송이 중단되기 때문에 그 동안 데이터 전송의 지연이 발생하게 된다. 그에 반해 다중경로 라우팅은 여러 경로 중 하나가 단절되거나 모든 경로가 다 단절되면 경로 유지를 실행한다. 모든 노드가 끊어지기 전에 경로 유지를 수행하면, 항상 여러 경로를 가지게 되어 경로가 끊어져서 발생하는 데이터 전송의 지연시간을 줄일 수 있다. 반면에 다양한 경로 중 하나가 끊어진다고 해서 매번 경로 유지를 실행하게 되면 RREQ 패킷과 RREP 패킷에 대한 전송 증가로 인한 라우팅 오버헤드가 커진다.

2.3 AODV를 기반으로 한 다중경로 라우팅

AODV를 바탕으로 다중경로 라우팅을 구현한 대표적인 예는 AOMDV^[2], AODVM^[4] 그리고 AODV-BR^[3]이 있다. AOMDV (Ad-hoc On-demand Multiple Distance Vector Routing)는 다중경로를 찾기 위해

노드에서 여러 개의 RREQ 패킷을 받아서 처리하는데, RREQ 패킷의 처음 전송 경로를 구분하기 위해 RREQ 패킷에 first hop이라는 필드를 두어 RREQ 패킷을 받은 제일 처음 노드를 저장한다. 중간노드들은 모든 경로로 오는 RREQ 패킷을 받아서 패킷에 기록되어 있는 first hop과 자신에게 저장되어 있는 first hop list를 비교하여 중복되지 않는 RREQ 패킷을 사용한다. 해당 RREQ 패킷이 중복되지 않으면 라우팅 테이블에 있는 다음 홉(next hop)과 홉 카운트(hop count)를 갱신하는데, 서로 다른 경로로 온 RREQ 패킷에 대해서 모두 저장해야 하기 때문에 이를 리스트 형태로 저장한다. 목적지 노드는 서로 다른 노드로부터 자신에게 도착한 RREQ 패킷에 대해서 RREP 패킷을 보내고 이 RREP 패킷들은 중간의 노드들에 저장되어 있는 여러 개의 경로를 통해 소스 노드로 전송되어져 다중 경로를 생성한다.

AODVM (AODV Multipath)은 중간 노드들은 전송 받은 모든 RREQ 패킷을 기록하고, 목적지 노드는 받은 모든 RREQ에 대해 RREP를 전송한다. 이 때, 각 노드는 이웃의 모든 노드들이 전송하는 RREP 패킷을 오버헤어링하여 자신이 가지고 있는 RREQ 테이블에서 이웃에 대한 정보를 삭제한다. 이를 통해 노드들이 들을 수 있는 범위 내에서는 제일 처음 전송되는 RREP 패킷에 대한 경로만이 생성되고 나머지 경로는 자동으로 사라지게 된다.

마지막으로 AODV-BR (AODV Backup Routing)은 주경로를 생성하기 위해 RREP가 전송되는 동안 주변 노드들은 RREP가 전송되는 것을 오버헤어링하여 보조 경로를 생성하고 유지한다. 이를 통해 주경로가 끊어졌을 때 해당 노드는 주변 노드로 데이터를 전송하고, 주변 노드는 이전에 들었던 RREP 패킷에 대한 정보를 이용하여 생성된 보조 경로를 통해 데이터 전송을 계속한다.

AOMDV와 AODVM, AODV-BR은 경로 비 겹침한 다중경로를 생성한다. 하지만 AOMDV는 라우팅 테이블에 하나의 목적지에 대한 여러 개의 다음 홉과 홉 카운트를 저장하는 것과 first hop list를 따로 저장하는 것에 대한 오버헤드가 있다. AODVM과 AODV-BR은 다른 노드가 전송하는 패킷을 오버헤어링해야 하는 단점이 있다. 그리고 AODVM의 경우 RREP 패킷의 전송경로에 따라서 다른 경로가 생성되지 않을 수도 있으며 AODV-BR은 소스 노드부터 목적지 노드까지 여러 개의 경로를 가지는 것이 아니라 주경로 주변에 있는 노드를 통한 우회

경로를 유지하는 것이다.

III. Multi-Path AODV 알고리즘

이 논문에서는 AODV 프로토콜을 바탕으로 하는 새로운 Multi-Path AODV(MP-AODV) 알고리즘을 제안한다. MP-AODV는 MANET 환경을 배경으로 하고 있는데, 이 MANET 환경은 노드들이 임의로 움직이며 네트워크를 구성한다. 이러한 네트워크 환경에서는 데이터를 보내기 위해 설정된 경로가 노드의 움직임으로 인해 단절되어 데이터의 전송이 저하되는 현상이 발생하게 된다. MP-AODV는 이러한 문제를 해결하기 위해 소스노드와 목적지 노드 간의 노드 비 겹침 다중경로를 설정하도록 하였다. 노드 비 겹침 다중경로를 설정함으로써 노드의 움직임으로 인해 하나의 경로가 단절되어도 다른 경로에는 영향이 미치지 않아 계속해서 데이터를 보낼 수 있게 된다.

AODMV나 AODVM의 경우에는 다중경로를 한번에 찾고 난 후, 데이터 전송을 하기 때문에 초기에 경로탐색에 따른 오버헤드(overhead)가 발생하고 데이터 전송이 늦춰지게 된다. MP-AODV는 이러한 문제점을 해결하기 위해 소스노드가 목적지노드까지의 주경로를 찾고 나면 즉시 데이터 전송을 시작한다. 이후 데이터 전송이 되고 있는 상태에서 두 번째 경로인 보조경로를 찾기 위해서 두 번째 경로탐색을 시작한다. 이렇게 주경로와 보조경로를 탐색하는 과정을 분리시키고 데이터 전송과 같이 백그라운드로 보조경로를 탐색하도록 함으로써 다중경로 탐색에 따른 데이터 전송이 지연되는 것을 줄였다.

경로 유지에 있어서 AODMV나 AODVM와 다른 점은 다중경로가 항상 유지되도록 하는 것이다. 노드의 이동이나 오류가 발생하여 주경로나 보조경로가 단절되는 경우에 경로 유지를 과정을 통해 단절된 경로를 다시 설정하도록 하여 항상 두 개의 경로를 유지하도록 하여 데이터 전송을 안정적으로 할 수 있다.

3.1 AODV의 수정

AODV의 seen table은 각 노드에서 전송받은 RREQ 패킷의 중복을 확인하는데 사용된다. 각 노드는 RREQ 패킷을 받게 되면 그림 1과 같이 RREQ 패킷에 있는 소스노드에 대한 정보와 RREQ ID를 테이블에 저장한다. 노드는 다음에 RREQ 패킷이 왔을 때 소스에 대한 정보와 RREQ ID를

Type	J	R	G	D	U	F	Reserved	Hop Count
RREQ ID								
Destination IP Address								
Destination Sequence Number								
Originator IP Address								
Originator Sequence Number								

그림 1. MP-AODV RREQ 패킷

seen table에 저장되어 있는 정보와 비교하여 이미 받은 RREQ 패킷인지 아닌지를 확인하여 중복되는 RREQ 패킷에 대한 수신을 차단한다. 따라서 각 노드는 한 번의 경로탐색 때마다 한번의 RREQ 패킷을 받게 된다. 그리고 다음 경로탐색 과정 때는 RREQ ID가 1증가되어 오기 때문에 새로운 RREQ 패킷으로 인식하여 RREQ 패킷을 처리하게 된다.

본 논문에서는 MP-AODV에서 주경로와 보조경로가 서로 다른 노드를 사용하도록 하기 위해서 seen table을 사용한다. 주경로 탐색 시, 주경로의 노드가 가지고 있는 seen table을 수정하여 보조경로 탐색 시 발생하는 RREQ 패킷을 이미 받은 패킷으로 인식하도록 함으로써 보조경로 탐색 시에는 주경로의 노드가 경로탐색에 참여하지 못하도록 한다. 이를 통해 주경로의 노드는 보조경로의 노드가 되지 못하도록 하여 노드 비 겹침 다중경로를 생성한다.

MP-AODV에서 주경로와 보조경로를 설정하기 위해 그림 1과 그림 2와 같이 AODV의 RREQ와 RREP 패킷에 1bit 플래그인 'F'를 추가시켰다. 이를 통해 주경로 설정을 위한 패킷(RREQ, RREP)과 보조경로 설정을 위한 패킷(RREQ_2, RREP_2)을 구분한다.

Type	R	A	F	Reserved	Prefix - Size	Hop Count
Destination IP Address						
Destination Sequence Number						
Originator IP Address						
Originator Sequence Number						

그림 2. MP-AODV RREP 패킷

2.2 경로 탐색

본 논문에서 제안하고 있는 MP-AODV의 경로 탐색 과정을 살펴보면, 주경로를 탐색하는 과정은 기존의 AODV와 비슷하다.

그림 3과 같이 기존의 AODV 방법과 동일하게 경로 탐색을 위해 소스 노드에서 RREQ 패킷 (RREQ ID = 1)을 생성하여 브로드캐스트 방식으로 목적지로 패킷을 전송한다. 그림 3의 사각형 안의 값은 RREQ ID를 의미한다. RREQ 패킷을 받은 중간 노드들은 seen table에 RREQ ID와 RREQ를 전송한 소스의 정보를 저장한다.

그림 4와 같이 목적지 노드는 RREQ 패킷이 전송되어 온 경로를 따라서 RREP 패킷을 전송하는데, RREP 패킷을 받은 노드들은 기존 AODV와 달리 seen table의 RREQ ID를 1증가 시킨다. 이렇게 RREQ ID를 1 증가시켜 놓음으로써 보조경로를 찾을 때 주경로에 있는 노드가 겹치지 않도록 하는 것이다.

소스노드가 RREP 패킷을 받으면 주경로가 설정된다. 주경로가 설정되면 소스 노드는 데이터 전송을 시작하고 동시에 보조경로 설정을 위해 그림 5와 같이 RREQ_2 패킷(RREQ ID = 2)을 전송한다. RREQ_2 패킷은 보조경로를 찾기 위한 패킷으로 그림 1에 나오는 F 비트를 1로 설정하여 전송한다. RREQ_2 패킷을 전송받은 중간 노드들은 그림 5와

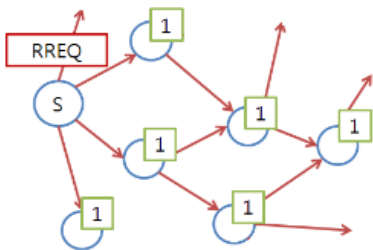


그림 3. 첫 번째 RREQ 전송

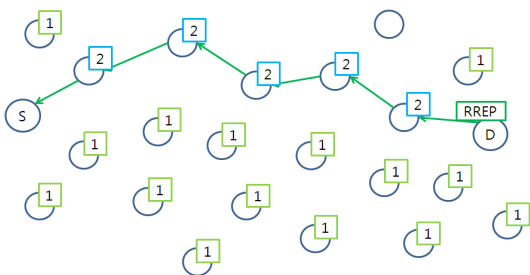


그림 4. 주경로 설정을 위한 RREP 전송

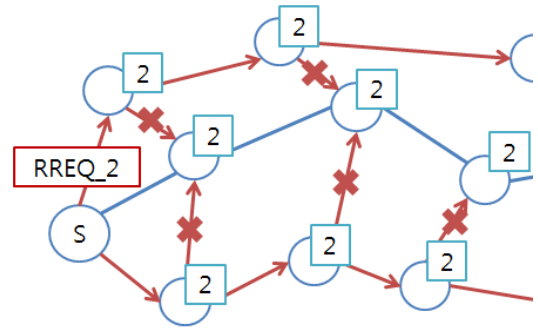


그림 5. 보조경로 설정을 위한 RREQ_2 전송

같이 가지고 있던 seen table의 RREQ ID와 패킷의 RREQ ID를 비교하여 값이 같으면 RREQ_2 패킷을 버리고, 같지 않으면 계속 해서 RREQ_2 패킷을 전송하게 된다. 주경로로 사용되고 있는 노드의 경우에는 이전 경로탐색 과정에서 RREP 패킷을 받을 때, seen table에 있는 RREQ ID를 1증가 시켰기 때문에 RREQ_2 패킷의 RREQ ID와 같은 값을 가지게 된다. 따라서 주경로에 있는 노드들은 RREQ_2 패킷에 대해서는 패킷 처리를 하지 않게 되고, 이를 통해 보조경로와 주경로가 같은 노드를 가지는 것을 방지하여 주경로와 보조경로가 서로 disjoint한 경로를 생성하도록 한다. 목적지 노드가 RREQ_2 패킷을 받으면 RREP_2 패킷(F 비트를 1로 설정)을 RREQ_2 패킷이 온 경로를 따라 전송한다. 소스 노드가 RREP_2 패킷을 받으면 보조경로가 생성 된 것이다.

보조경로를 설정할 때는 그림 6과 같이 seen table의 RREQ ID를 1증가 시키지 않는다. 대신 주경로가 설정되어 있는 경로로 RREP를 다시 보내어 노드들의 seen table의 RREQ ID를 1증가 시킨다. 결과적으로 주경로가 설정된 경로의 노드가 가지고 있는 seen table의 RREQ ID는 3이 되고, 보조경로

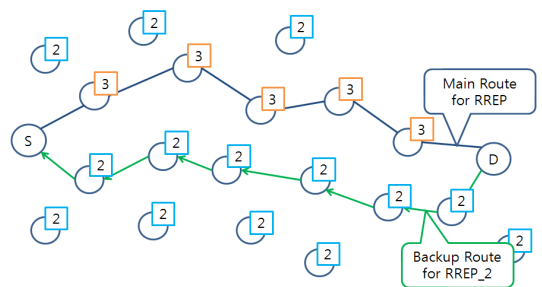


그림 6. 보조경로 설정을 위해 RREP_2 전송 및 주경로의 seen table을 갱신하기 위한 RREP 전송

가 설정된 경로의 노드가 가지고 있는 seen table의 RREQ ID는 2가 된다. 즉, 주경로의 RREQ ID는 보조경로의 RREQ ID보다 항상 1이 큰 값을 가지게 되는 것이다. 이는 보조경로가 끊어진 경우 다시 노드 disjoint한 보조경로를 찾기 위함이다.

그림 7은 이러한 경로 탐색 과정을 알고리즘으로 나타낸 것이다. 위 쪽 부분은 RREQ 패킷과 RREQ_2 패킷을 받았을 때 중간 노드와 목적지 노드가 해당 패킷을 처리하는 과정이며, 아래 쪽 부분은 RREP 패킷과 RREP_2 패킷을 중간노드와 목적지 노드가 받았을 때 해당 패킷을 처리하는 과정이다.

3.3 경로 유지

본 논문에서 제안하는 MP-AODV는 경로 유지를 위해 두 가지 방법을 제시한다. 첫 번째로 주경로가 단절 되었을 경우 소스 노드는 보조경로로 데이터 전송을 시작하고, 주경로를 다시 찾기 위해서 경로 탐색을 재실행한다. 이때 기존의 AODV에서 RREQ 패킷의 RREQ ID를 1증가 시키는 것과 달리 RREQ ID를 2증가 시켜 전송한다. 이는 기존의 주경로가 항상 1이 증가된 RREQ ID를 가지고 있기 때문이다. 두 번째로 보조경로가 단절 될 경우에는 보조경로만 다시 찾는다. 이를 위해 경로 탐색에서 설명했듯이 보조경로 설정 시에 주경로로 설정된 노드들의 RREQ ID를 1증가 시킨 것이다.

MP-AODV의 소스와 목적지 노드들에서 주경로와 보조경로를 구분하기 위해서 라우팅 테이블에 route_flag라고 하는 1bit 플래그를 추가하였다. 소스와 목적지 노드는 route_flag가 0으로 세팅될 경우 주경로로 인식하고 1로 세팅될 경우 보조경로로 인식한다. 이를 통해 경로가 끊어져서 RERR 패킷이 전송되어 올 경우, RERR 패킷을 전달한 이전 노드를 라우팅 테이블을 통해 비교하여 어떤 경로가 끊어졌는지 판단 한다.

또한, 각 노드들의 라우팅 테이블에 소스노드 정보를 유지하도록 하여 경로가 겹치게 될 경우 경로가 바뀌는 것을 방지하도록 한다. 이를 자세히 살펴 보면, 그림 8과 같은 환경에서 여러 개의 다른 소스에서 하나의 목적지로 경로를 요청하는 경우가 발생할 수 있다.

이 때, 소스와 목적지 간의 경로를 다른 소스와 목적지간의 경로로 설정하게 될 경우 그림 9와 같이 기존 소스의 경로가 삭제되고, 다른 소스의 경로로 설정되게 된다. 이때, 기존 소스가 가지고 있던 주경로와 보조경로의 노드 비 겹침이 깨지게 된다.

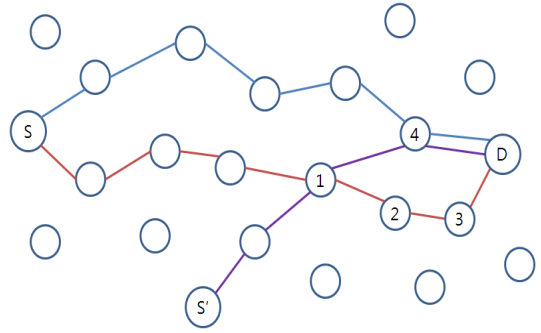


그림 8. 서로 다른 소스가 하나의 목적지 로 경로를 설정

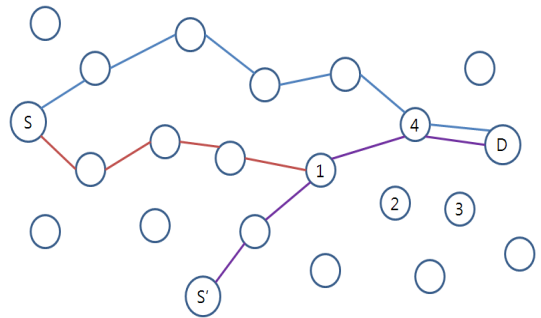


그림 9. 서로 다른 소스로 인한 라우팅 정보의 변경

이를 방지하기 위해 MP-AODV는 그림 10과 같이 각 노드의 라우팅 테이블에 필드하나를 추가하여 소스 노드의 정보를 포함시키고, 소스가 다를 경우에 라우팅 테이블에 각각의 경로를 따로 가지고 있도록 하였다.

기존의 다중경로 라우팅 프로토콜인 AODMV, AODVM 그리고 AODV-BR의 경우 한번 다중경로가 생성된 후, 어느 하나의 경로가 단절 되면 하나의 경로를 따로 탐색하지 못하고 전체 경로가 단절 되어야지만 새로운 다중경로를 탐색할 수 있지만 MP-AODV의 경우는 주경로와 보조경로가 단절 될 경우 이를 따로 복구 할 수 있다. 이를 통해 네트워크 변화에 따른 경로 단절에 보다 능동적으로 대처할 수 있다.

Destination	Next Hop	Source	...
D	2	S	
...			

그림 10. 기존의 라우팅 테이블에 소스정보 포함

IV. 시뮬레이션 실험 및 성능 평가

4.1 실험 환경

본 논문에서 제안하는 MP-AODV의 성능을 평가하기 위해 Qualnet을 이용하여 시뮬레이션 하였다^[6]. MP-AODV는 Qualnet의 AODV 소스를 기반으로 작성하였다. 표 1의 실험 환경을 살펴보면, 네트워크의 전체 크기는 1500m x 1500m이고, 전체 노드의 수는 50개이다. 각 노드의 분포는 Uniform분포를 통해 랜덤하게 배치되었고, 시뮬레이션이 진행된다면 노드는 random way point 방식을 통해 최대 5m/s의 속도로 이동하게 된다. 이 방식은 노드가 0~5m/s의 속도로 자신이 이동하고자 하는 임의의 장소로 이동하게 되고, 해당 위치에 도달하게 되면 설정되어 있는 멈춤 시간(Pause time) 동안 해당 자리에 있다가 이후 다시 임의의 장소로 이동을 시작하게 된다. 노드의 멈춤 시간이 적을수록 노드의 이동이 잦아지게 되고 노드가 오래 멈춰 있을수록 노드의 이동은 적게 되어 진다. 이를 통해 MANET 환경에서의 노드 이동성을 보였다^{[3][8][9]}.

총 시뮬레이션 시간은 500초이고 데이터 전송은 5초부터 495초까지 전송되도록 수행하였다. 데이터 전송은 임의의 소스와 목적지를 지정하고 소스에서 목적지로 512바이트의 크기를 가지는 패킷을 매 초당 5개씩 보는 CBR(Constant Bit Rate)을 사용하였다. 각 시뮬레이션은 AODV와 AODV-Local Repair, MP-AODV에 대해서 20번씩 수행하여 평균을 계산하였다.

표 1. 시뮬레이션 설정값

변수	설정값	
네트워크 크기	1500m X 1500m	
MAC	IEEE 802.11	
라우팅 프로토콜	AODV, MP-AODV	
노드	수	50개
	이동성	Random Way Point
	속도	0~5m/s
	멈춤시간	100,200,300,400,500s
어플리케이션	종류	CBR
	시간	5~495s
	크기	512bytes
	개수	5 packet/s
수행 시간	500초	
수행 회수	20번	

4.2 결과 및 분석

본 논문에서는 MANET 환경에서 제안한 MP-AODV와 AODV 그리고 중간노드에 직접 단절된 경로를 찾는 AODV-Local repair를 적용하여 결과를 비교하였다. 실험에서 측정한 값은 소스가 목적지로 데이터 패킷을 성공적으로 전송하는 비율과 데이터 패킷이 소스에서 목적지로 전송되는 데 걸리는 지연시간, 그리고 라우팅 정보를 생성하고 유지하기 위해서 사용되는 컨트롤 패킷에 대한 오버헤드이다. 실험결과는 이러한 값들을 멈춤 시간에 따라 그래프를 통해 나타내었다.

그림 11에서 제시하고 있는 패킷 전송 성공률은 소스 노드가 목적지 노드로 보낸 패킷에 대해 도착한 패킷 수로 측정이 가능하며 식 (1)에 의해 구하였다. 그림 12에서 제시되어 있는 중단 간 지연시간은 패킷이 전송되는데 걸리는 시간이다.

$$\text{패킷 전송률} = \frac{\text{도착한 패킷 수}}{\text{총 전송한 데이터 패킷}} \times 100 \quad (1)$$

결과를 살펴보면 본 논문에서 제시한 MP-AODV가 기존의 AODV와 AODV-Local Repair보다 높은 데이터 전송 성공률과 낮은 중단 간 지연시간을 가

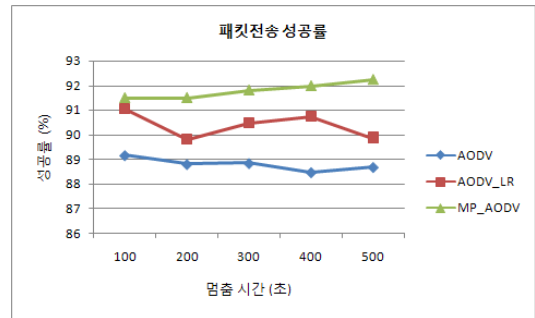


그림 11. 패킷 전송 성공률

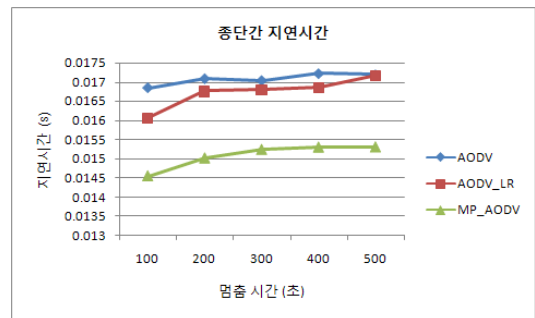


그림 12. 중단 간 지연시간

지는 것을 알 수 있다. AODV의 경우에는 경로가 단절 되었을 때 경로를 재탐색하고, AODV-Local Repair의 경우에는 단절이 발생한 직전 노드로부터 목적지까지의 경로를 재탐색 하게 된다. 이에 비해 MP-AODV는 두 개의 경로를 가지고 있으므로 주 경로가 끊어짐과 동시에 보조경로를 통해 데이터 전송이 가능하기 때문이다. 그렇기 때문에 경로를 재탐색하는 동안 발생하게 되는 패킷의 손실이 적어 AODV나 AODV-Local repair에 비해 패킷 전송률이 2~3% 향상되고, 경로를 재탐색을 하는 동안 대기를 하지 않아도 되기 때문에 소스와 목적지간의 지연시간도 10%정도 줄어들었다.

표 2는 주경로 단절 되었을 경우, 보조경로가 존재하여 보조경로를 통해서 데이터 전송을 계속한 경우와 보조경로가 존재하지 않아서 다시 경로탐색을 한 경우를 나타낸다. 결과를 살펴보면 주경로가 단절되었을 때 62.48%만큼 보조경로가 존재하여 데이터 전송의 지연 없이 보조경로를 통해 데이터 전송을 계속하고, 노드의 위치에 따라 주경로와 노드 비 겹침한 보조경로가 없을 경우에는 다시 주경로 부터 경로탐색을 시작한다. 이 경우, 새로운 주경로와 보조경로를 탐색하므로 그에 따른 오버헤드가 발생할 수 있다.

그림 13은 라우팅 오버헤드를 나타내며, 식 (2)을 이용해서 측정할 수 있다.

$$\text{라우팅 오버헤드} = \frac{\text{컨트롤 패킷}}{\text{컨트롤 패킷} + \text{데이터 패킷}} \times 100 \quad (8)$$

표 2. 주경로 단절시 보조경로 확보율

	주경로 단절 횟수	보조경로로 전송	경로탐색
평균횟수	34.25	21.4	12.85
평균비율 (%)	100	62.48	37.52

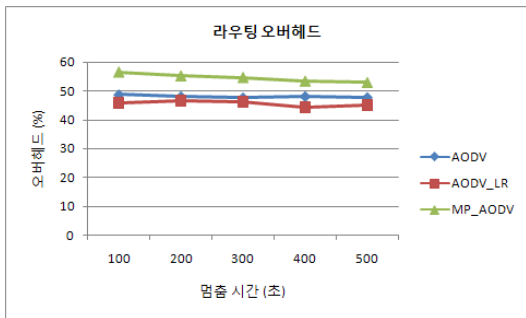


그림 13. 라우팅 오버헤드를 비교한 그래프

컨트롤 패킷은 RREQ나 RREP 패킷과 같은 경로를 설정하기 위해 전송되는 패킷을 의미하고, 데이터 패킷은 실제로 전송되는 데이터에 대한 패킷을 의미한다. 그림 13을 살펴보면 AODV나 AODV-Local Repair에 비해 본 논문에서 제안한 MP-AODV가 더 큰 오버헤드를 가지는 것을 볼 수 있다. 이는 MP-AODV가 두 개의 경로를 검색하기 위해 두 번의 RREQ 패킷 flooding과 추가적인 RREP 패킷 전송을 하고 경로가 끊어졌을 때 다시 새로운 경로를 찾기 위한 컨트롤 패킷의 전송이 발생하기 때문이다.

V. 결론

본 논문에서 제안한 라우팅 프로토콜은 MANET 환경에서 AODV을 기반으로 2개의 노드 비 겹침 경로를 찾는 프로토콜로서 기존의 AODV를 최대한 활용하여 구현하였으며, 비교적 크지 않는 라우팅 오버헤드로 노드 비 겹침 다중경로를 찾을 수 있는 방법이다. 주경로와 보조경로가 겹치지 않는 노드 비 겹침 다중 경로를 사용함으로써 주경로와 보조 경로가 동시에 단절되지 않도록 하였다. 또한 다중 경로를 찾을 때, 주경로와 보조경로의 탐색을 따로 하여 주경로 생성 후 바로 데이터 전송을 시작함으로써 다중경로를 찾을 때 생기는 데이터 전송의 지연을 줄였으며, 한번 다중경로를 찾은 후에 경로유지 알고리즘을 통해 계속해서 2개의 경로를 유지하도록 함으로써 하나의 경로가 끊어져도 다른 경로를 통해 데이터 패킷을 보낼 수 있도록 하였다.

이러한 경로탐색과 경로유지를 통해 데이터 패킷의 성공적인 전송을 보장하고, 경로가 단절되었을 때 다시 경로를 찾는 시간을 줄여 소스와 목적지간의 지연시간을 줄일 수 있도록 하였다.

앞으로 고려해 볼 수 있는 사항으로는 AODV를 기반으로 하는 다른 다중경로 라우팅 프로토콜인 AOMDV, AODVM, AODV-BR 과의 비교 실험을 통해 각각의 성능을 비교하는 것이다. 이러한 비교 실험을 하기 위해서는 각 라우팅 프로토콜을 시뮬레이션 상에 구현해야 하는 어려움이 있어서 다음 연구 주제로 남겨놓았다.

참고 문헌

- [1] Stephen Mueller, Rose P. Tsang, and Dipak Ghosal, "Multipath Routing in Mobile Ad Hoc

