

논문 01-01-06

# 단방향 링크를 지원하는 이동 Ad Hoc 라우팅 프로토콜

## Mobile Ad Hoc Routing Protocol Supporting Unidirectional Links

李光培\*, 金玄郁\*, 鄭根元\*

( Kwang-Bae Lee\*, Hyun-Ug Kim\*, Kun-Won Jung\* )

### 요약

본 논문에서는 단방향 링크를 포함하는 이동 Ad Hoc 망에 대해 비대칭 경로를 지원하는 동적 소스 라우팅 프로토콜을 제안하였다. 기존의 동적 소스 라우팅 프로토콜은 양방향 링크로 구성된 대칭경로만을 지원하도록 구현되어있다. 그러나 실제적인 무선환경에서는 단말의 비대칭성이나 무선환경 특성등에 기인한 단방향 링크가 존재할 수 있다. 따라서 본 논문에서는 좀 더 일반적인 무선환경에 맞는 이동 Ad Hoc 망 라우팅 프로토콜을 구현하기 위하여 단방향 링크를 지원하도록 하였다. 특히 이동성으로 인한 경로 실패를 인식한 경우, 신속한 경로 재설정을 수행하기 위해 다중경로유지 기법을 사용하였다. 성능평가는 평균 경로설정시간 및 평균 데이터수신율에 대해 수행하였으며, 그 값을 얻기 위해 경로 상에 중간노드캐쉬의 사용여부 및 서로 다른 이동성과 연결성을 제공하는 시나리오를 고려하여, 100초 단위로 900초까지 성능평가를 수행하였다.

### Abstract

In this paper, we propose a dynamic source routing protocol supporting asymmetric paths for mobile Ad Hoc networks that contain unidirectional links. At present, the existing dynamic source routing protocol supports only symmetric paths that consist of bidirectional links for routing. However, in fact, there can exist unidirectional links due to asymmetric property of mobile terminals or current wireless environment. Thus, we implement a mobile Ad Hoc routing protocol supporting unidirectional links, which is fit for more general wireless environment. Especially, the proposed protocol uses an improved multipath maintenance method in order to perform rapid route reconfiguration when route error due to mobility is detected. For performance evaluation, we consider the following factors: average route discovery time and average data reception rate. In order to obtain the factors of the performance evaluation, we did simulation for 900 seconds in the step of 100 seconds with the following scenarios: usage of route cache in intermediate nodes on route path, and different mobility/connection patterns.

*Key words* : mobile Ad Hoc networks, dynamic source routing protocol, unidirectional links, route discovery time, data reception rate

\* 明知大學校 電子工學科

(Dep. of Elec. Eng., Myongji Univ.)

接受日: 2001年 3月 5日, 修正完了日: 2001년 6月26日

## I. 서 론

현재까지 개발된 대부분의 네트워크 시스템은 고정된 기지국과 같은 유선 기반망(infrastructure)을 요구해왔다. 그러나 유선 기반망을 설치하기 어려운 산간 지방 또는 빙하지역 같은 오지나 지진, 홍수, 전쟁 등의 재난으로 유선 기반망이 파괴된 지역에서의 신속한 통신을 위해서는 유선 기반망이 필요없는 네트워크 구성이 요구된다.

MANET(mobile Ad Hoc network)이란 고정된 유선 기반망 없이, 라우팅 기능을 가지는 이동 단말들로만 구성된 네트워크를 의미한다. 망 구성 요소는 매우 동적으로 변화하며, 무선환경의 낮은 대역폭과 높은 전송오류, 전송회선의 불안정성 등을 갖기 때문에 기존의 라우팅 프로토콜을 직접 사용할 수 없다. 지금까지 MANET을 위해 개발된 라우팅 프로토콜은 AODV(ad-hoc on-demand distance vector), DSR(dynamic source routing), TORA(temporally ordered routing algorithm), ZRP(zone routing protocol), CBRP(cluster based routing protocol), WRP(wireless routing protocol), LMR(lightweight mobile routing), SSR(signal stability-based adaptive routing), ABR(associativity based routing) 등이 있으며, AODV, DSR 등은 멀티캐스트 및 QoS 기능을 추가하는 연구가 진행되고 있는 상황이다. 또한, HSR(hierarchical state routing), FSR(fisheye state routing), LANMAR(landmark routing) 등의 새로운 라우팅 프로토콜이 제안, 연구되고 있다.<sup>[1][2]</sup>

Ad Hoc 망에서 완전한 IP 라우터의 기능을 제공하며, 신속하고 부하가 적은 독자적 관리가 가능한 통신망을 지원하기 위해 다음과 같은 요구 조건이 수반된다.<sup>[3][7]</sup> MANET에서 권장하는 Ad Hoc 라우팅 프로토콜에 대한 요구사항은 다음과 같다. 첫째, 분산서비스를 지원해야 하며, 둘째 라우팅 루프가 발생하지 않아야 한다. 셋째, 경로설정은 미리 이루어지는 대신에 요구시만 수행된다. 넷째, 양방향 링크뿐만 아니라 단방향 링크도 지원해야 한다. 그 외에 저전력 소모와 단말간 호환성을 위한 하드웨어적인 기술과 향후 실시간 연결을 위한 QoS와 신뢰할 수 있는 보안기능 등이 요구된다. 가장 대표적인 MANET 라우팅 프로토콜은 AODV와 DSR이다. AODV 프로토콜은 현재 그 구조상 단방향 링크를 지원하는 것이 불가능하다. 반면에 DSR 프로토콜은 그 구조상 양방향 링크와 단방향 링크를 모두 지원할 수 있으나 현재 양방향 링

크만을 지원하도록 설계되어 있다. 그러므로 본 논문에서는 단방향 링크를 지원하는 DSR 프로토콜을 제안하고 NS(Network Simulator) 시뮬레이션 코드로 구현한 후, 그에 대한 성능평가를 수행하였다.

## II. 기존의 DSR

이동 Ad Hoc 네트워크를 위해 고안된 프로토콜로 임의의 목적지노드에 대해 노드를 가로지르는 소스라우팅을 제공한다. 즉, DSR은 소스노드에서 생성한 패킷의 헤더(header)에 목적지노드까지의 경로를 순차적으로 기록하여 완전한 노드경로를 생성하는 방식이다. 중복된 경로와 루프를 제거하기 위해 <소스주소, 요구 식별자> 목록을 이용한다.

경로발견을 위한 패킷 방송 증가는 전체 망의 대역자원을 감소시켜 망의 효율을 급속히 저하시킬 수 있다. 이에 따라 경로설정시의 부하를 최소화하며, 경로의 평균효율을 증대시키기 위한 몇 가지 방식이 제안되고 있다. 첫째, 라우팅 캐쉬를 최대한 이용하는 방식이 있다. RREQ(Route Request)를 받은 노드가 목적지노드는 아니지만, 수신된 노드의 라우팅 캐쉬에 목적지노드까지의 경로가 있을 경우, 목적지노드를 대신해서 RREQ에 대한 RREP(Route Reply)를 소스노드에게 전송하는 방식이다. 둘째, 데이터 전송도중 더 짧은 경로가 발견되면, 그것을 새로운 경로로 반영하는 것이다. 셋째, RREQ 브로드캐스트(broadcast) 시에 데이터 패킷을 피기백(piggy back)시키는 방법이 있다. 마지막으로 RREQ에 대한 RREP가 정해진 시간내에 수신되지 않을 경우, 지수적 회귀(exponential backoff) 알고리즘을 사용하여 RREQ의 재전송을 감소시키는 것으로 이동 등의 무선 환경 변화로 인해 망 내의 연결이 끊어져 노드간이 완전히 분할되었을 때, RREQ의 재전송을 줄임으로써 망효율을 증대시킨다.

DSR은 주기적인 라우팅 정보를 교환하지 않고 경로발견 과정을 통해 얻은 경로를 사용한다. 이때, 중간노드들은 경로유지 과정을 수행하여 경로의 동작상태와 에러를 탐지하고 그 사실을 소스노드에게 알린다. 경로 에러패킷에는 경로에러가 발생한 홉의 양쪽 노드의 주소를 포함한다. 이때, 경로에러를 발견한 노드는 경로 에러패킷을 전송하는 동시에 패킷을 전송할 새로운 경로를 찾기 위해 새로운 RREQ 패킷을 생성하여 경로발견 과정으로 들어간다. 경로 에러패

킷을 수신한 노드들은 자신의 라우팅 캐쉬를 검사하고 에러가 발생한 부분을 포함하는 모든 경로들을 제거한다.

### III. 단방향 링크와 다중경로를 지원하는 DSR

#### 3.1 경로발견

노드간에 통신이 요구되면 DSR 라우팅 기법을 그대로 적용하여 목적지노드까지의 경로를 발견한다. 목적지주소가 자신의 노드일 경우, 소스노드부터 목적지노드까지의 경로를 기록한 패킷을 생성하여 소스노드로 전송한다. 단방향 링크가 존재할 수 있기 때문에 먼저, 목적지노드는 자신의 캐쉬를 검사하고 만약, 소스노드로의 경로를 발견한 경우, 그 경로를 이용하여 경로 정보를 전송하며, 발견하지 못한 경우는 소스 노드로 새로운 경로발견 과정을 시작한다. 이때 소스노드로부터 목적지노드까지의 발견된 경로정보 RREP-D(RREP from Destination)를 RREQ-D(RREQ from Destination) 패킷에 피기백시킨다. RREQ-D를 수신한 소스노드는 목적지노드까지의 최단경로를 선택하여 자신의 라우팅 캐쉬를 갱신한 후, RREP-S(RREP from Source)를 목적지노드까지 발견된 경로를 통하여 유니캐스트 방식으로 전송한다. 이때, RREP-S 패킷에는 역방향 경로를 포함한다. 이 정보를 수신한 목적지노드는 패킷을 검사하여 자신의 노드에 맞는 라우팅 캐쉬를 작성하여 저장한다. 라우팅 경로 검색 시, RREQ 패킷의 브로드캐스트로 인해 목적지노드까지의 최단거리 경로 이외의 여러 우회경로를 발견할 수 있다. 이러한 우회경로들은 사용중인 최단거리 경로가 손실되었을 경우에 신속한 복구를 위해 사용한다. 순방향 경로 탐색 시, 목적지노드는 최초의 RREQ-S(RREQ from Source) 도착 후, 즉시 응답한다. 이후, 목적지노드에 도착한 순방향 경로들은 다중경로 설정을 위해 최대 3개까지 소스노드로 전송한다. 소스노드로의 경로를 발견하지 못한 경우, RREP-D를 포함한 RREQ-D가 순방향 경로 검색 시와 마찬가지로 다중경로로 브로드캐스트되어 소스노드에 도달한다. 수신된 RREP-D의 정보를 바탕으로 소스노드는 목적지노드까지의 다중경로를 설정한다. 또한 소스노드에서 수신한 RREQ-D를 분석하며 수집된 역방향 경로 정

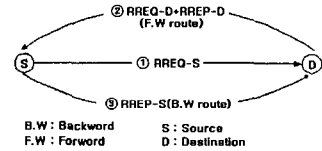


그림 1. 순방향 및 역방향 경로설정

Fig. 1. Forward and backward route discovery

보들은 RREP-S를 통하여 목적지 노드로 전송한다. 마찬가지로 목적지노드는 RREP-S 역방향 다중경로를 설정한다. 그림 1은 순방향 및 역방향 경로발견과 그 설정 과정을 나타낸다.

#### 3.2 최적화

순방향 경로발견 시에 RREQ-S 패킷이 목적지노드까지의 라우팅 정보를 가지고 있는 중간노드를 만나면 브로드캐스트를 중지하고 그 정보를 사용하여 경로를 설정한다. 즉, RREQ-S의 브로드캐스트가 중단되며, 이 노드의 라우팅 정보를 이용하여 완성된 순방

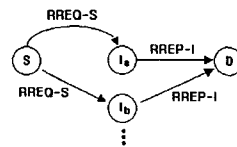


그림 2. 중간노드를 이용한 최적화

Fig. 2. Optimization using an intermediate node

향 경로정보를 만든 후, 목적지노드까지 RREP-I(RREP from Intermediate)를 유니캐스트(unicast)로 보낸다. 역방향 경로발견도 순방향 경로발견 시와 동일한 방법으로 수행된다. 그림 2는 중간노드를 이용한 최적화 과정을 보여준다.

#### 3.3 경로 유지

유선 망과는 달리 Ad Hoc 망에서의 모든 노드들은 무선 환경에 의한 요인과 이동성에 기인하여 설정된 경로를 사용하지 못할 경우가 발생한다. 이러한 경로에러에 대해 양방향 링크만이 존재하는 네트워크 상에서는 문제를 일으키는 링크 상에 패킷을 전송할 때, 그 에러가 한 홉(hop) 후에 즉시 발견되어 처리될 수 있으나 단방향 링크가 존재하는 네트워크 상에서

는 그렇지 못하기 때문에 그러한 네트워크에 적합한 경로유지 방법을 제시하였다. 즉, 소스노드는 목적지 노드로 주기적으로 경로 유효성 확인 헬로우 메시지를 다중경로 각각에 대해 전송하며, 목적지노드는 그 헬로우 메시지(hello message)를 받으면 그에 대응하는 역방향 경로를 통해 소스노드로 헬로우 응답메시지를 보낸다. 소스노드는 헬로우 응답메시지를 받으면 현재 시간과 함께 그 경로가 유효하다고 재마크(remark)한다. 해당 경로의 사용여부는 데이터 패킷 전송시 확인된다. 데이터 패킷 전송시 소스노드는 사용하고 자 하는 경로가 유효하다고 재마크되어 있는지 또는 주기적인 경로 유효성 확인중 인지를 점검한다. 그 결과, 경로에러로 판별된 경우에는 여분의 다른 경로를 이용하며, 만약 모든 경로가 에러인 경우에는 다시 새로운 경로를 발견하기 위한 경로발견 과정을 수행한다.

## IV. 성능평가

### 4.1 실험환경

본 논문에서 제시한 단방향 링크 지원 Ad Hoc 라우팅 프로토콜에 대한 성능을 평가하기 위해서 미국 버클리 대학, 남가주 대학 및 카네기 멜론 대학에서 개발한 네트워크 시뮬레이터인 NS-2 및 CMU 확장 프로그램을 사용했으며, CMU 확장 프로그램에 본 논문에서 제시한 내용을 코드로 구현한 후, 성능평가를 수행하였다.<sup>[8],[9]</sup> 한편 성능평가를 위해 500MHz 펜티엄II, 64MB RAM 및 13GB 하드디스크를 갖춘 PC가 사용되었다.

### 4.2 결과 및 고찰

#### 4.2.1 시나리오 파일 생성

NS에 사용되는 입력은 노드의 이동성을 나타내는 이동패턴파일(movement pattern file), 네트워크 계층에서 제공하는 패킷의 작업부하를 나타내는 통신패턴파

일(communication pattern file), 및 사용된 라우팅 프로토콜을 정의한 라우터설정파일로 구성된다. 여기서 이동패턴파일과 통신패턴파일을 이용하여 적절한 시뮬레이션 시나리오를 작성한다. 다음은 DSR 프로토콜을 사용하여 NS를 구성하는 시나리오를 보여준다.

위의 시나리오는 1500m×300m의 범위에서 정지시간 0인 50개의 노드가 최대 속도 20m/s로 움직이며, 10개의 연결을 갖는다. 각 소스는 초당 512 바이트 패킷 4개를 보내며, 900초 후에 성능평가가 완료됨을 나타낸다. 출력은 out.tr 파일로 생성된다.

#### 4.2.2. 평가방식 및 파라미터

성능평가를 위해 이동패턴과 통신패턴을 정의한 4개의 시나리오 파일을 사용하였다. 제안한 라우팅 알고리즘의 성능을 평가하기 위해 다음과 같이 시나리오를 설정하였다. 첫째, 다중경로는 3개까지 사용한다. 둘째, 각각의 다중경로가 유효한지를 점검하기 위해 사용되는 헬로우 메시지의 시간간격은 20초로 고정하였다. 셋째, 소스노드와 목적지노드의 캐쉬에만 경로를 기록한 경우와 경로 상의 모든 중간노드의 캐쉬에 경로를 기록한 경우를 나누어 성능평가를 수행하였다. 비교 파라미터는 소스노드에서 발생된 경로요청패킷이 목적지노드까지의 경로를 발견하는 평균 경로설정 시간과 수신된 데이터량에 대한 백분율을 사용하였다. 성능평가 완료시간은 100초를 단위로 900초까지 수행하였다.

#### A. 데이터 수신율 분석

##### 가.노드의 이동성에 따른 분석

노드의 정지시간이 작다는 것은 전체적으로 이동성이 크다는 것을 의미한다. 성능평가 시간이 증가함에 따라 이동노드간의 충돌이 빈번히 발생하므로 데이터수신율은 감소하였다. 비연속적인 이동성을 갖는 경우, 연결이 많은 시나리오에 대해 중간노드 캐쉬를 사용할 때, 데이터수신율이 높았다. 반면에 연결이 적은 시나리오에 대해서는 중간노드 캐쉬를 사용하지 않을 때, 데이터수신율이 높았다. 연속적인 이동성을 갖는 경우, 연결이 적은 시나리오에 대해 중간노드 캐쉬를 사용할 때, 데이터수신율이 높았고, 연결이 많은 시나리오에 대해서는 중간캐쉬를 사용하지 않았을 때, 높은 데이터수신율을 나타내었다. 그림 3과 4는 노드의 이동성에 따른 데이터수신율을 보여주고 있다. 그림에서 c10 cache는 10개의 연결에 중간노드 캐

```
./ns cmu/scripts/run.tcl -rp cmu/dsr/dsr.tcl -x 1500 -y 300 \
-cp scen/cbr-50-10-4-512 -sc scen/scen-1500x300-50-0-20-1\
-stop 900 -tr out.tr
```

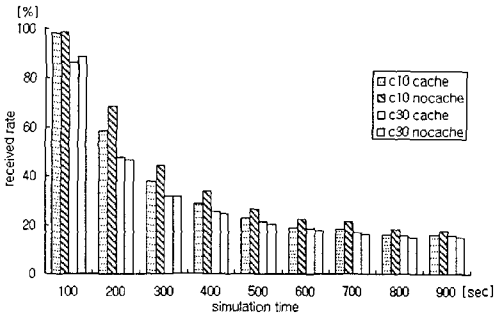


그림 3. 정지시간이 120인 경우, 데이터수신율  
Fig. 3. Data reception rate when pause time=120sec

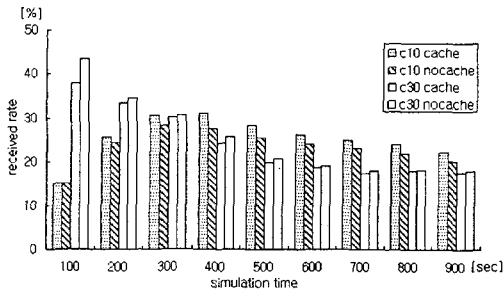


그림 4. 정지시간이 0인 경우, 데이터수신율  
Fig. 4. Data reception rate when pause time=0sec

쉬를 사용한 경우를 나타내고, c10 nocache는 10개의 연결에 중간노드 캐쉬를 사용하지 않은 경우를 나타낸다.

나. 노드의 연결 개수에 따른 분석

노드의 연결이 적다는 것은 전체적으로 패킷충돌이 적어진다는 것을 의미한다. 모든 연결 상태에 대해 초기에는 비연속적인 이동성을 갖는 경우에 더 높은 데이터수신율을 보이다가 점차 시간이 경과됨에 따라 연속적인 이동성을 갖는 경우에 더 높은 데이터수신율을 나타내었다. 연결이 많고 연속적인 이동성을 갖는 경우에는 중간노드 캐쉬를 사용하지 않았을 때, 더 높은 데이터수신율을 나타내었고, 반면에 비연속적인 이동성을 갖는 경우에는 중간노드 캐쉬를 사용했을 때 더 높은 데이터수신율을 나타내었다. 한편 연결이 적고 연속적인 이동성을 갖는 경우에는 중간노드 캐쉬를 사용하였을 때, 더 높은 데이터수신율을 나타내었고, 반면에 비연속적인 이동성을 갖는 경우

에는 중간 캐쉬를 사용하지 않았을 때 데이터수신율이 더 높았다. 그림 5와 6은 노드의 연결개수에 따른 데이터수신율을 보여준다. 그림에서 s0 cache는 노드

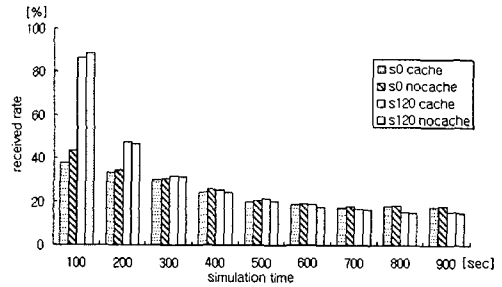


그림 5. 연결이 30인 경우, 데이터수신율  
Fig. 5. Data reception rate when the number of connections=30

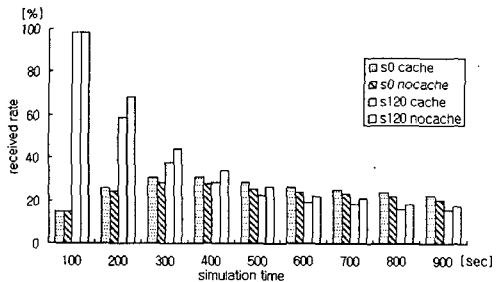


그림 6. 연결이 10인 경우, 데이터수신율  
Fig. 6. Data reception rate when the number of connections=10

의 정지시간이 0초이고, 중간노드 캐쉬를 사용하는 경우를 나타내며, s0 nocache는 노드의 정지시간이 0초이고 중간노드 캐쉬를 사용하지 않는 경우를 나타낸다.

B. 평균 경로설정시간 분석

경로설정시간은 짧으면 짧을수록 더 좋은 성능을 의미한다. 연결이 많고, 연속적인 이동성을 갖는 경우 가장 짧은 경로설정시간을 나타내었다. 한편, 비연속적인 이동성을 갖는 경우에는 연결이 적을 때, 더 짧은 경로설정시간을 나타내었다. 그리고 비연속적인 이동성을 갖는 경우에 중간노드 캐쉬를 사용하지 않을 때, 경로설정시간을 단축시켰다. 연속적인 이동성을 갖는 경우에는 중간노드 캐쉬정보를 사용하는 것

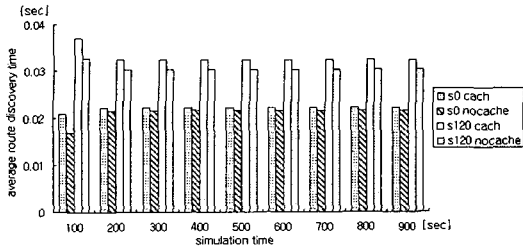


그림 8. 연결이 30인 경우, 평균경로설정시간

Fig. 7. Average route discovery time when the number of connections=30

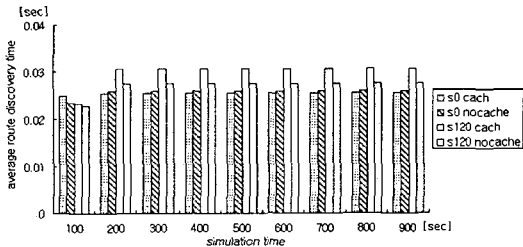


그림 7. 연결이 10인 경우, 평균경로설정시간

Fig. 8. Average route discovery time when the number of connections=10

이 경로설정시간을 단독시켰다. 그림 7과 8에서는 노드의 연결개수에 따른 평균 경로설정시간을 각각 도시하였다.

## V. 결 론

일반적인 무선환경에서 동작하는 MANET 라우팅 프로토콜을 연구하기 위하여 본 논문에서는 단방향 링크를 지원하는 DSR 라우팅 알고리즘을 구현하였다. 제안한 알고리즘은 리눅스 환경하에서 C++과 Otel로 구현하였고, NS를 사용하여 네 가지의 환경변수를 변화시키면서 두 가지의 성능분석요소를 기준으로 제안한 알고리즘을 비교·평가하여 다음과 같은 결론을 얻었다. 성능평가 시간이 충분히 지난 후에는 모든 성능평가 시나리오들에 대해 20% 대의 거의 비슷한 데이터수신율을 보였다. 정지시간이 120초이고 10개의 연결이 있는 경우 즉 비연속적인 이동성과 연결 노드의 개수가 적은 시나리오의 경우, 초기에는 98% 이상의 데이터수신율을 가졌으나 시간이 지나감

에 따라 급격히 감소하여 성능평가 시간이 500초 이후에서는 20% 대의 데이터수신율을 보였다. 중간노드 캐쉬를 사용할 때보다 중간노드 캐쉬를 사용하지 않을 때 더 높은 데이터수신율을 제공했다. 정지시간이 120초이고 30개의 연결이 있는 경우 즉 비연속적인 이동성과 연결 노드의 개수가 많은 시나리오의 경우, 초기에는 80% 대의 데이터수신율을 가졌으나 시간이 지나감에 따라 급격히 감소하여 성능평가 시간이 400초 이후에서는 20% 대의 데이터수신율을 보였다. 성능평가 초기이외에는 중간노드 캐쉬를 사용하지 않을 때보다 중간노드 캐쉬를 사용할 때 약간 더 높은 데이터수신율을 제공했다. 정지시간이 0초이고 10개의 연결이 있는 경우 즉 연속적인 이동성과 연결 노드의 개수가 적은 시나리오의 경우, 초기에는 15% 정도의 데이터수신율을 가졌으나 시간이 지나감에 따라 데이터수신율이 30% 정도로 급격히 증가하다가 20% 대로 서서히 감소하였다. 중간노드 캐쉬를 사용하지 않을 때보다 중간노드 캐쉬를 사용할 때 약간 더 높은 데이터수신율을 제공했다. 한편 평균 경로설정 시간은 대부분의 시나리오에서 0.02-0.03초 정도의 시간이 걸렸고 초기이외에는 성능평가 시간에 대해 거의 일정한 값을 가졌다. 연속적인 이동성과 연결노드의 개수가 많은 시나리오에서 0.02초 정도로 가장 짧은 경로설정시간을 나타내었으며, 비연속적인 이동성과 연결노드의 개수가 많은 시나리오에서 0.03초 이상으로 가장 긴 경로설정시간을 나타내었다. 연속적으로 이동하고 연결 노드의 개수가 적은 경우 이외에는 모두 중간노드 캐쉬를 사용하지 않았을 때 더 짧은 경로설정시간을 가졌다.

이상으로 본 논문에서 제안한 Ad Hoc 라우팅 프로토콜이 단방향 링크를 포함하는 보다 일반적인 무선환경에서 동작하는 MANET 라우팅 프로토콜에 대한 연구의 기초가 될 수 있으리라 사료된다.

참고문헌

- [1] Tony Larsson and Nicklas Hedman, "Routing Protocols in Wireless Ad-hoc Networks : A Simulation Study," Master's thesis, Lulea University of Technology, Stockholm, 1998.
- [2] Elizabeth M. Royer and et. al., "A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks," IEEE Personal Communication, pp.46-55, Apr. 1999.
- [3] David B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," IEEE Workshop on Mobile Computing Systems and Application, Dec. 1994.
- [4] J. J. Garcia Luna Aceves, "Loop-Free Routing Using Diffusing Computations," IEEE Trans. on Networking, vol.1(1), pp.130-141, Feb. 1993.
- [5] Lidong Zhou and et. al., "Securing Ad Hoc Networks," IEEE Network, pp.24-30, Dec. 1999.
- [6] S. Murthy and J. J. Garcia Luna Aceves, "Loop-Free Internet Routing Using Hierarchical Routing Trees," INFOCOM'97, Apr. 1997.
- [7] J. Ioannidis and G.Q.M., "The Design and Implementation of Mobile Internetworking Architecture," USENIX Conference, pp.495-502, Jan. 1993.
- [8] Bjarne Stroustrup, The C++ Programming Language, Addison-Wesley, Third Edition, 1997.
- [9] Clif Flynt, Tcl/TK for Real Programmers, AP PROFESSIONAL, Academic Press, 1994.

저 자 소 개

李 光 培 (正會員)



1979년 2월 고려대학교 전자공학과 졸업(공학사)  
 1981년 2월 고려대학교 전자공학과 졸업(공학석사)  
 1981년 3월~1982년 3월 삼성반도체  
 1982년3월~1983년 4월 금성사  
 1984년~1986년 Univ. of

Southern California, Computer Engineering 전공(공학석사)  
 1986년~1991년 Arizona state Univ., Computer Engineering 전공(공학박사)  
 1992년 3월~1998년 1월 명지대학교 전자공학과 교수  
 1998년 2월~1999년 1월 Worchester Polytechnic Institute 대학 WINLAB연구실 방문 연구교수  
 1999년 2월~2000년 1월 Columbia 대학 COMET연구실 방문 연구 교수  
 2000년 2월~현재 명지대학교 전자공학과 교수  
 관심 분야 : 이동 무선 인터넷망, 멀티미디어, 컴퓨터 구조  
 e-mail : kblee@mju.ac.kr

金 玄 郁 (正會員)



1978년 2월 고려대학교 전자공학과 졸업(공학사)  
 1980년 2월 고려대학교 전자공학과 졸업(공학석사)  
 1987년 2월 고려대학교 전자공학과 졸업(공학박사)  
 1980년 3월~1981년 2월 동양공업전문대학 전자과 전임강사

1981년 3월~1988년 1월 명지대학교 전자공학과 교수  
 1988년 2월~1990년 1월 Dept. of Computer Science of Arizona State University Adjunct Faculty  
 1990년 3월~현재 명지대학교 전자공학과 교수  
 관심 분야 : 이동 무선 인터넷망, 멀티미디어, 컴퓨터 구조, 보안  
 e-mail : acalab@mju.ac.kr

## 鄭 根 元 (正會員)



1991년 8월 명지대학교 전자  
공학과 졸업(공학사)

1993년 8월 명지대학교 전자  
공학과 졸업(공학석사)

1996년 3월 명지대학교 전자  
공학과 입학(박사학위과정)

1994년 3월~1997년 8월 여주  
대학교 전자과 강사

1995년 3월~1998년 2월 경원전문대학교 사무자동화  
과, 전자과 강사

1996년 4월~1997년 3월 명지대학교 산업기술연구소  
전임연구원

1998년 3월~1999년 8월 성덕대학 겸임교수

1998년 9월~1999년 8월 강원 전문대학 정보통신과  
강사

2001년 3월~현재 명지대학교 전자 정보통신공학부  
강사

2001년 3월~현재 (주) 컴섹 선임연구원

e-mail : acajgw@hanmail.net