

이동 Ad Hoc 망에 대해 동적 소스 라우팅을 이용하는 단방향 라우팅 프로토콜

곽 승 옥[†] · 이 광 배^{**} · 김 현 옥^{**}

요 약

실제적인 이동 Ad Hoc 망에서는 단말의 비대칭성이나 무선환경 특성으로 인해 단방향 링크가 존재할 수 있다. 그러나 현재 Zone 라우팅 프로토콜이외의 기존의 이동 Ad Hoc 라우팅 프로토콜들은 양방향 링크들만을 지원하도록 구현되어 있다. 따라서 본 논문에서는 가장 대표적인 라우팅 프로토콜인 DSR(Dynamic Source Routing) 프로토콜이 단방향 링크를 포함하는 이동 Ad Hoc 망에서 구동될 수 있도록 동적 소스 라우팅을 이용한 새로운 단방향 라우팅 프로토콜을 제안한다. 단방향 링크 상에서 효율적으로 구동하기 위해서 제안한 단방향 라우팅 프로토콜은 새로운 경로발견 및 경로우치 방법을 사용하였다. 성능평가를 위해서 미국 버클리 대학의 NS-2 시뮬레이터 프로그램에 제안한 내용을 코드로 삽입시켰으며, 성능평가 파라미터를 얻기 위해 카네기멜론 대학에서 제공한 이동 패턴파일과 연결 패턴파일을 조합하여 사용하였다. 성능평가 항목으로 데이터수신율 및 평균경로설정시간을 고려하였으며, 제안된 3가지 경로설정 방식들에 대해 비교·평가하였다.

Unidirectional Routing Protocol using Dynamic Source Routing for Mobile Ad Hoc Networks

Seung-Ug Kwag[†] · Kwang-Bae Lee^{**} · Hyun-Ug Kim^{**}

ABSTRACT

There can exist unidirectional links due to asymmetric property of mobile terminals or current wireless environments in practical mobile Ad Hoc networks. However, at present, the existing mobile Ad Hoc routing protocols except Zone Routing Protocol are implemented to support only bidirectional links. Thus, in this paper, we propose a new unidirectional routing protocol using dynamic source routing in order for the existing Dynamic Source Routing protocol to be run on mobile Ad Hoc networks containing unidirectional links. The proposed unidirectional routing protocol uses new route discovery and route maintenance methods for efficient routing on unidirectional links. For performance evaluation, we insert program codes for the proposed protocol into the NS-2 simulator of U.C. Berkeley. In order to obtain evaluation parameters, we use combinations of mobility pattern files and connection pattern files from Carnegie Mellon University. We consider received data rate and average route discovery time as evaluation criteria, which are compared and evaluated for three proposed route discovery methods.

키워드 : 이동 Ad Hoc 망(Mobile Ad Hoc Network), 동적 소스 라우팅 프로토콜(Dynamic Source Routing Protocol), 단방향 링크(Unidirectional Links), 단방향 라우팅 프로토콜(Unidirectional Routing Protocol)

1. 서 론

최근 이동사용자를 위한 무선통신은 컴퓨터를 포함한 하드웨어 기술력 향상과 무선 데이터통신 기술의 발전에 힘입어 급속한 성장을 하고 있다[1]. 이동호스트 기반의 무선통신은 일반적으로 두 가지의 접근 방법으로 이루어지고 있다.

첫째, 음성뿐만 아니라 데이터를 전송할 수 있는 기존의

셀룰라 기반망의 관점에서 접근하는 것이다.

둘째, 통신을 원하는 각 이동호스트 자체적으로 통신할 수 있는 Ad Hoc 망의 관점에서 접근하는 것이다.

이동 Ad Hoc 망 기술은 개념적으로 고정 기반망이 없는 상황하에서 단지, 이동 호스트만으로 구성된 독자적이고 부분적인 시스템을 가능하게 하는 방식이다[2]. 이는 기반망이 구성되어 있지 않은 군사활동지역, 회의장, 긴급재난지역 등에 응용되어, 신속하고 효율적인 망의 형태를 구성한다. 그러나 이동성에 따른 추가적인 호스트 관리는 무선통신이 갖고 있는 제한적인 통신자원 낭비를 초래하여 데이터 처리율을 저하시키는 문제를 야기시킨다. 따라서 각 호

* 본 논문은 1997년도 한국학술진흥재단의 대학교수 해외파견 연구 지원에 의하여 연구되었음.

† 준 회 원 : 명지전문대학 정보통신과 교수

** 정 회 원 : 명지대학교 전자공학과 교수

논문접수 : 2001년 3월 24일, 심사완료 : 2001년 6월 8일

스트의 이동성으로 인한 경로의 계산과 수정이 효율적으로 진행될 수 있고 무선환경에 따른 단방향 링크를 지원하는 잘 정의된 라우팅 기법이 요구된다[3-7].

이동 Ad Hoc 망에 대한 연구는 인터넷 분야 연구를 활성화시키기 위해 결성된 비영리 조직 IETF(Internet Engineering Task force)의 MANET(Mobile Ad hoc Network) WG(Working Group)에 의해 활발히 지원되고 있다. MANET WG에서 권장하는 Ad Hoc 라우팅 프로토콜에 대한 요구사항은 다음과 같다.

- 첫째, 분산서비스를 지원해야 하며,
- 둘째, 라우팅 루프(loop)가 발생하지 않아야 한다.
- 셋째, 경로설정은 미리 이루어지는 대신에 요구 시에만 수행된다.
- 넷째, 양방향 링크뿐만 아니라 단방향 링크도 지원해야 한다.

현재, MANET WG에 제안된 대표적인 이동 Ad Hoc 망 라우팅 프로토콜들로서 DSR(Dynamic Source Routing), AODV(Ad hoc On demand Distance Vector), TORA(Temporally Ordered Routing Algorithm) 및 ZRP(Zone Routing Protocol)등이 있다. AODV와 TORA는 그 구조상 양방향 링크 상에서만 구동되는 라우팅 프로토콜이고, ZRP는 처음에 양방향 링크 상에서만 구동되도록 설계되었으나 최근에 단방향 링크 상에서도 구동할 수 있도록 확장되었다. 반면에 DSR은 미국 정부의 지원 하에 실제 코드로 구현되어 필드 테스트(Field Test)까지 완료한 뛰어난 라우팅 프로토콜이지만, 그 구조상 양방향 링크뿐만 아니라 단방향 링크 상에서도 구동될 수 있음에도 불구하고 현재 양방향 링크만을 지원하도록 구현되어 있다[8-14]. 앞에서 언급하였듯이 IETF의 MANET WG에서는 실제 이동 Ad Hoc 망 상에 단방향 링크들이 존재할 수 있으므로 단방향 링크도 지원하는 라우팅 프로토콜의 개발을 권장하고 있다. 그러므로 본 논문에서는 가장 뛰어난 이동 Ad Hoc 망 라우팅 프로토콜중 하나인 DSR 라우팅 프로토콜이 단방향 링크들을 포함한 이동 Ad Hoc 망에서 구동할 수 있도록 단방향 라우팅 프로토콜을 제안하였다. 양방향 링크만을 지원하는 기존의 DSR에서는 목적지노드가 경로발견 과정 시 발견된 경로의 역방향 경로를 이용하여 경로응답을 소스노드로 전송한다. 그러나 발견된 경로 상에 단방향 링크를 포함하는 이동 Ad Hoc 망에 대해서는 발견된 경로의 역방향 경로를 이용하여 경로응답을 소스노드로 전송하지 못한다. 그 대신에 경로응답을 소스노드로 보내기 위한 새로운 경로를 발견하는 과정이 필요하다. 본 논문에서는 경로응답에 대한 목적지노드로부터 응답과 중간노드로부터의 응답을 나누어 제시하였으며, 특히 중간노드로부터의 응답시 3가지 다른 경로설정 방법들을 제시하였다. 그 3가지 다른 경로설정 방법들은 이동 Ad Hoc 망의 상황에 따라 각기 다른 장점을

지니고 있다. 또한 기존의 DSR에서는 목적지노드가 경로요청을 통해 발견한 모든 경로들을 그 발견된 경로들의 역방향 경로들을 통해 즉시 소스노드로 보내주며, 소스노드는 이 경로들을 다중경로로 사용한다. 그러나 단방향 링크를 포함하는 이동 Ad Hoc 망에 대해서는 발견된 경로의 역방향 경로를 이용하여 경로응답을 소스노드로 전송하지 못하므로, 만약 목적지노드가 경로요청을 통해 발견한 모든 경로들을 브로드캐스트로 소스노드에 전달한다면 망상에 엄청난 패킷 트래픽을 유발시켜 현격한 망 시스템 성능 저하를 일으킬 것이다. 그러므로 본 논문에서는 다중경로를 효과적으로 설정할 수 있는 방법을 제시하였다. 또한 양방향 링크만을 지원하는 기존의 DSR에서는 송신노드가 이웃노드로 패킷을 전송할 때 즉시 2 노드간에 링크 에러를 감지할 수 있으나, 단방향 링크를 포함하는 이동 Ad Hoc 망에 대해서는 이 방법을 사용할 수가 없다. 즉 단방향 링크로 연결된 2 노드간에 패킷 전송이 이루어지는 경우 송신노드는 그 이웃 수신노드가 전송한 패킷을 제대로 수신하였는지 즉시 확인할 수 없고 링크 에러도 즉시 감지할 수 없다. 단지 소스노드와 목적지노드 간의 경로 상에 단절된 링크들이 존재하는 지만을 확인할 수 있다. 이것을 구현하는 방법으로는 목적지노드가 소스노드가 보낸 패킷을 받을 때마다 소스노드로 확인(Ack) 패킷을 보내는 방법과 주기적으로 헬로우 메시지를 보내는 방법이 있다. 확인 패킷을 보내는 방법은 이동 무선 환경 하에서 지나친 트래픽 증가를 유발시키므로 본 논문에서는 주기적으로 헬로우 메시지를 보내는 방법을 사용하여 이 문제를 해결하였다. 한편 기존의 DSR에서는 다중경로를 유지하는 방법을 다루지 않았으나, 본 논문에서는 단방향 링크를 포함하는 이동 Ad Hoc 망에 대해 효율적으로 다중경로를 유지하는 방법을 제시하여 신속한 경로 에러 복구를 가능케 하였다.

본 논문의 제2장에서는 Ad Hoc 망의 기본 개념과 기존의 DSR 라우팅 프로토콜을 간단히 설명하였다. 제3장에서는 단방향 링크를 지원하는 본 논문에서 제안한 동적 소스 라우팅을 이용한 단방향 라우팅 프로토콜에 대해 서술하였다. 제4장에서는 여러 시나리오에 따라 성능평가를 수행하며, 마지막 제5장에서 결론을 맺는다.

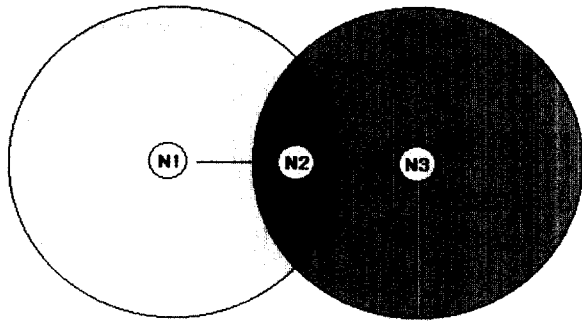
2. Ad-Hoc 망

2.1 기본 개념

이동 Ad Hoc 망은 중앙집중화된 관리나 표준화된 지원 서비스의 도움 없이 임시망을 구성하는 무선 이동호스트들의 집합이다. 다른 이동호스트로의 연결을 제공하기 위해 고정된 액세스 포인트(Access Point) 또는 기지국(Base-Station)을 이용하는 기존의 기반망과는 다르게, 이동 Ad Hoc 망은 각 이동호스트가 라우터로 동작하여 이전 이동호스트로부터 받은 패킷을 다음 이동호스트로 전달한다. 그러

나 이때 이동호스트의 이동에 의해 이미 설정된 경로가 무효화될 수 있기 때문에 통신 연결은 기존의 기반망에 비해 상당히 취약하다. 그러므로 이동호스트는 실제 데이터 처리율을 저하시켜 망 자체가 비효율적이고 비현실적인 시스템이 되지 않도록 이동에 따른 경로의 발견과 재설정에 많은 시간을 소비하지 않아야 한다.

(그림 1)은 이동 Ad Hoc 망에서의 중간노드 N2를 이용하여 두 노드 N1과 N3 간의 연결과정을 나타낸 것이다. 소스노드 N1이 목적지노드 N3에게 데이터 패킷을 보내려고 하는 경우, N3가 N1의 무선통신 범위 내에 위치해 있지 않기 때문에 직접적인 데이터 패킷 전송은 불가능하다. 이때 중간노드 N2가 N1의 무선통신 범위 및 N3의 무선통신 범위 내에 위치하므로 N1과 N3간에 라우터로서 동작한다. 즉 소스노드 N1은 중간노드 N2로 데이터 패킷을 보내고, N2는 그 데이터 패킷을 목적지노드 N3로 전달한다. 그러므로 이동 Ad Hoc 망에서는 상황에 따라 각 노드가 이동호스트 및 라우터로서 동작하게 된다.



(그림 1) 세 개의 무선 이동 호스트를 갖는 Ad Hoc 망

2.2 DSR 라우팅 프로토콜

이동 Ad Hoc 망을 위해 고안된 프로토콜로 임의의 목적지노드에 대해 노드를 가로지르는 소스라우팅을 제공한다. 즉, DSR은 소스노드에서 생성한 패킷의 헤더(header)에 목적지노드까지의 경로를 순차적으로 기록하여 완전한 노드 경로를 생성하는 방식이다.

DSR의 동작은 크게 두 가지 과정으로 구분할 수 있는데, 경로발견 과정과 경로유지 과정으로 구분 할 수 있다. 경로발견 과정은 목적지로 패킷을 전송하기를 원하는 소스노드에서 목적지까지의 경로를 얻기 위한 메커니즘이고 경로유지 과정은 경로발견 과정을 통해 얻어진 경로를 사용하여 패킷을 전송하는 동안 경로의 상태를 감시하여 경로의 에러를 감지하고 이를 처리하기 위한 메커니즘이다.

2.2.1 경로발견

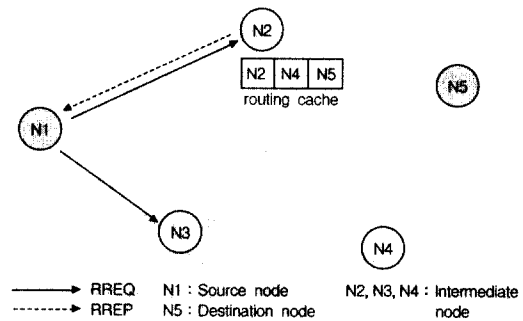
DSR에서의 경로설정엔 두 노드간에 통신의 필요성이 있을 경우에만 동적으로 이루어지게 된다. 경로설정은 소스노드가 RREQ(Route REQuest : 경로설정요구)를 목적지노드로 브로드캐스팅(broadcasting)하면서 시작된다. 이 RREQ에

는 유일한 요구 식별자(request id)가 부여되며, 각 노드는 라우팅 루프의 발생을 방지하기 위해서 최근에 수신된 RREQ를 확인하는 <소스주소, 요구 식별자>를 쌍으로 가지는 요구 식별자 리스트를 관리한다. RREQ는 소스노드에서 목적지노드까지 브로드캐스팅시 발견되는 경로를 기록하기 위한 경로 레코드(route record)를 가지고 있으며, RREQ를 받은 각 노드는 자신의 노드 주소를 RREQ의 경로 레코드에 추가시킨다. 한편, 목적지노드는 RREQ를 받으면 RREQ의 경로 레코드에 저장된 소스노드로부터 목적지노드까지의 경로를 추출하여 RREP(Route REply : 경로 응답)에 복사한 후 소스노드로 전송한다.

RREQ를 브로드캐스트한 노드의 전송영역 내에서 RREQ를 수신한 노드들은 적절한 경로 발견 과정을 수행하기 위해서 다음과 같은 단계를 수행한다.

- ① 수신된 RREQ가 현재노드의 <소스주소, 요구 식별자> 목록에서 발견될 경우는 수신된 RREQ는 이미 수신된 것으로 간주하여 중복된 RREQ로 판단해 폐기한다.
- ② RREQ 상의 경로 레코드에서 자신의 주소가 발견되지 않은 경우, 현재의 노드가 RREQ의 목적지노드일 경우는 RREQ 상의 경로 레코드에 기록된 경로를 RREP에 복사한 후, 그 발견된 경로의 역방향 경로를 이용하여 소스노드로 전송한다.
- ③ 만약 자신이 RREQ의 목적지노드가 아닐 경우, 자신의 주소를 RREQ의 경로 레코드에 추가시키고 RREQ를 다시 브로드캐스트한다.
- ④ 한편 RREP를 수신한 소스노드는 그 RREP에 포함된 경로가 그 자신의 라우팅 캐쉬에 이미 존재하는지 조사해보고 없는 경우에만 라우팅 캐쉬에 추가시킨다.

2.2.2 최적화



(그림 2) 중간노드의 캐쉬에서 목적지노드 발견

경로발견을 위한 패킷 브로드캐스트의 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

를 받은 노드가 목적지노드는 아니지만, 수신된 노드의 라우팅 캐쉬에 목적지노드까지의 경로가 있을 경우, 목적지노드를 대신해서 RREQ에 대한 RREP를 소스노드에게 전송하는 방식으로 (그림 2)는 그 진행 과정을 나타내었다.

둘째, 데이터 전송도중 더 짧은 경로가 발견되면, 그것을 새로운 경로로 반영하는 것이다.

셋째, RREQ 브로드캐스트 시에 데이터 패킷을 피기백(piggy back)시키는 방법이 있다. 그러나 RREQ에 데이터 패킷을 실어보내는 방법은 목적지노드에서 데이터 중복 문제를 야기시킨다.

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, 재전송을 줄임으로써 망효율을 증대시킨다.

2.2.3 경로유지

현재의 DSR은 양방향 링크를 지원하는 IEEE 802.11 MAC layer를 사용하기 때문에 그 다음 노드로 패킷을 전송하는 노드의 MAC layer에서 그 자신 노드와 다음 노드간에 양방향 링크가 유효한 지 무효한지를 자동으로 감지한다. 그러므로 주기적으로 경로유지 정보를 노드들간에 교환하는 대신에 데이터 패킷 전송시 패킷을 전송하는 노드가 경로 상에 그 다음 노드로의 링크가 유효한 지 그 MAC layer를 통해서 검색한다. 만약 그 링크가 유효하다면 데이터 패킷은 그 다음 노드로 전송된다. 그러나 그 링크에 문제가 있다면 그 패킷 전송 노드는 에러 메시지를 즉시 소스노드로 보내 그 사실을 알린다. 이때 에러 메시지를 보낸 노드는 그 자신의 라우팅 캐쉬로부터 문제가 발생한 링크를 포함하고 있는 경로들을 검색하여 제거한다. 마찬가지로 에러 메시지를 소스노드로 전달하는 중간노드들과 소스노드도 그 자신들의 라우팅 캐쉬들로부터 문제가 발생한 링크를 포함하고 있는 경로들을 검색하여 제거한다.

3. 단방향 라우팅 프로토콜

앞의 서론에서 언급했듯이 기존의 DSR 프로토콜은 그 구조상 양방향 링크뿐만 아니라 단방향 링크 상에서도 구동될 수 있음에도 불구하고 현재 양방향 링크만을 지원하도록 구현되어 있다. 그러므로 IETF의 MANET WG에서 권장하는 단방향 링크를 지원하기 위해서 본 논문에서는 단방향 라우팅 프로토콜을 제안한다. 단방향 라우팅 프로토콜에는 단방향 링크를 포함하는 이동 Ad Hoc 망상에서의 기본 경로발견 및 설정 방법뿐만 아니라, 경로 상에 있는 중간노드의 경로 캐쉬 정보를 이용하는 3가지 경로설정 방법들을 제시한다. 또한 사용중인 경로 상에 문제가 발생하여 새로운 경로를 사용할 필요가 있는 경우 새로운 경로발

견 및 설정 과정을 통해 새로운 경로를 발견하는 대신 미리 발견된 다중 경로들 중에서 한 경로를 대체 경로로 사용하는 것이 훨씬 효율적이다. 그러므로 본 논문에서는 단방향 링크를 포함하는 이동 Ad Hoc 망상에서 경로 설정 패킷 트래픽을 줄여주는 효율적인 다중 경로 설정 방법을 제시한다. 그리고 단방향 링크를 포함하는 이동 Ad Hoc 망상에서의 경로 유지 및 경로 에러 발견을 위해 주기적인 헬로우 메시지를 사용하는 방법을 제시하며, 신속한 경로 복구를 위해 순방향-역방향 쌍 경로 관리 방식을 제안한다.

본 논문에서는 요청패킷인 RREQ와 응답패킷인 RREP을 생성한 노드를 정확히 구별하기 위해서 소스노드에서 생성되는 RREQ와 RREP에는 S(Source)를 붙이고, 목적지노드에서 생성하는 RREQ와 RREP에는 D(Destination)를 붙인다. 또한, 경로 상에 있는 중간노드에서 생성하는 RREQ와 RREP에는 I(Intermediate)를 붙여서 표시한다. 또한 소스노드에서 목적지노드까지의 전체 연결을 경로라고 칭하며, 이웃노드간의 연결은 링크로 표현한다. 한편 최대 3개까지의 경로를 다중경로로 설정하며, 설정된 다중경로는 경로 상에 에러를 발견한 경우 다른 경로로의 신속한 복구를 지원하는 경로유지를 위해 사용한다.

3.1 기본 동작

3.1.1 경로발견

소스노드가 목적지노드로 데이터 패킷을 보내려고 할 때, 먼저 그 자신의 캐쉬를 검사하고 목적지노드까지의 순방향 경로가 있는 경우에는 그 경로를 이용하여 데이터 패킷을 전송한다. 순방향 경로가 발견되지 않은 경우에는 기존 DSR 라우팅 기법과 같이 소스노드로부터 목적지노드까지의 경로를 발견하기 위해 경로요청패킷 RREQ-S를 브로드캐스트시킨다. 목적지노드가 RREQ-S 패킷을 받았을 때, 목적지노드는 자신의 캐쉬를 검사하고, 만약 소스노드와의 역방향 경로를 발견한 경우, 그 경로를 이용하여 RREQ-S 패킷으로부터 얻은 순방향 경로정보를 RREP-D 패킷으로 전송한다. 반면에 소스노드와의 역방향 경로를 발견하지 못한 경우는 소스 노드로 새로운 경로발견 과정을 시작한다. 즉, 순방향 경로정보를 포함하는 RREP-D 패킷을 새로운 역방향 경로정보를 요청하는 RREQ-D 패킷에 포함시켜 소스노드로 브로드캐스트시킨다. 소스노드는 도착한 RREP-D 패킷을 통해서 소스노드로부터 목적지노드까지의 순방향 경로를 알 수 있을 뿐만 아니라, RREQ-D 패킷을 통해서 목적지노드로부터 소스노드까지의 역방향 경로도 알 수 있다.

RREP-D를 수신한 소스노드는 먼저 RREP-D를 통해 얻은 목적지노드까지의 순방향 경로가 자신의 라우팅 캐쉬정보로 사용될 수 있는지 점검하고 그러한 경우 자신의 캐쉬에 순방향 경로정보를 추가시킨다. 만약, 소스노드가 RREP-D와 함께 RREQ-D를 수신했다면, RREQ-D에 포함된 역방

향 경로를 RREP-S 패킷에 복사시킨 후, RREP-S를 목적지 노드까지 유니캐스트 방식으로 전송한다. RREP-S 패킷을 수신한 목적지노드는 그 패킷으로부터 목적지노드로부터 소스노드까지의 역방향 경로정보를 추출하여 목적지노드 자신의 캐쉬에 추가시킨다.

3.1.2 다중경로 설정

순방향 경로 발견 과정 시, RREQ-S 패킷의 브로드캐스트로 인해 목적지노드까지의 최단거리 경로 이외의 여러 우회경로를 발견할 수 있다. 이러한 우회경로들은 사용중인 최단거리 경로가 손실되었을 경우, 신속한 복구를 위해 사용한다. 순방향 경로 탐색 시, 목적지노드는 최초의 RREQ-S의 도착 후, 즉시 응답한다. 이후, 목적지노드에 도착한 순방향 경로들은 다중경로 설정을 위해 최대 3개까지 소스노드로 나중에 발견된 역방향 경로를 통해서 전송한다. 한편 목적지노드로부터 소스노드까지 역방향 경로는 목적지노드가 소스노드로 순방향 경로 정보를 전달하는 과정에서 발견되며 다중경로 설정을 위해 최대 3개까지 소스노드에서 선택된다. 이 역방향 경로 정보는 최단 순방향 경로를 통해 소스노드로부터 목적지노드로 전달된다.

3.1.3 경로유지

양방향 링크만으로 구성된 Ad Hoc 망과는 다르게 단방향 링크가 존재하는 Ad Hoc 망에서는 경로에러가 데이터 전송시 즉시 발견되지 못한다. 그러므로 경로에러를 발견하기 위해서 소스노드는 주기적으로 헬로우 메시지를 순방향 경로를 통해서 목적지노드로 전송하고 목적지노드는 헬로우 메시지를 받으면 헬로우 응답 메시지를 역방향 경로를 통해서 소스노드로 즉시 전송한다. 이때, 소스노드는 헬로우 메시지를 보낸 후, 지정시간 내에 헬로우 응답 메시지를 받지 못하면 해당 경로 상에 에러가 발생했음을 인지한다. 본 논문에서는 경로에러 발견시 신속한 경로복구를 위해 최대 3개의 다중경로를 사용하고 있다. 즉, 3개의 순방향 경로와 대응되는 3개의 역방향 경로를 경로발견 과정시에 설정한다. 그러므로 주기적인 경로에러 발견과정은 설정된 모든 다중 경로들에 대해서 수행된다. 그러나 경로에러 발견시, 그 에러가 순방향 경로 상에서 발생하였는지 역방향 경로 상에서 발생하였는지 알 수가 없다. 신속히 경로를 복구하기 위해서 본 논문에서는 한 순방향 경로에 대해서 그에 대응하는 역방향 경로를 쌍으로 이루어 설정하였다. 즉, 3쌍의 순방향-역방향 경로가 설정된다. 그러므로 에러가 발견된 순방향-역방향 경로 쌍은 소스 및 목적지노드의 캐쉬로부터 제거된다. 데이터 패킷 전송시에 현재 유효한 최단 순방향 경로가 사용되며, 만약 설정된 모든 경로가 무효하여 캐쉬로부터 제거되었다면, 다시 새로운 경로발견 과정을 수행해야 한다.

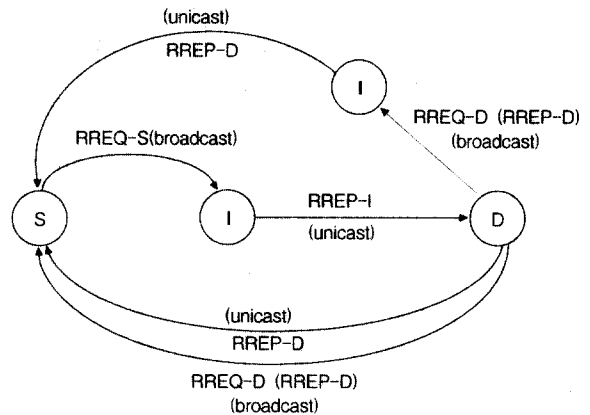
3.2 최적화

순방향 경로발견 시에 RREQ-S 패킷이 목적지노드까지

의 라우팅 정보를 가지고 있는 중간노드를 만나면 브로드캐스트를 중지하고 그 정보를 사용하여 경로를 설정한다. 본문에서는 중간노드가 목적지노드까지의 경로를 자신의 캐쉬에 갖고 있는 경우, 다음과 같은 3가지 경로설정 방법들을 제시하였다.

3.2.1 경로설정 방법 1

이 방법은 경로설정시 발생하는 전체 브로드캐스트 RREQ-S 패킷의 수를 줄이며, 노드의 이동으로 인한 거짓 경로 정보를 사용할 가능성을 줄여준다. 또한 설정되는 최대 3개까지의 순방향 경로들은 모두 목적지노드에 의해 결정된다.



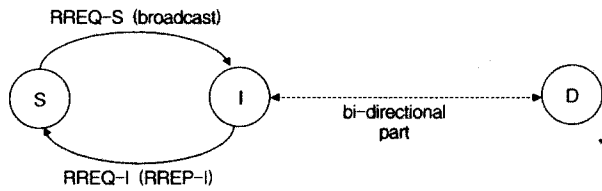
(그림 3) 경로설정 방법 1

(그림 3)과 같이 소스노드의 RREQ-S에 대해서 중간노드가 목적지노드까지 단방향 또는 양방향 경로를 캐쉬에 가지고 있을 경우, 목적지노드로 유니캐스트하고 목적지노드에서 발견된 순방향 경로 정보를 소스노드로 전달한다. 즉, 중간노드에서는 어떠한 응답패킷도 소스노드로 전송하지 않는다. 순방향 경로 정보를 포함하는 RREP-D를 소스노드로 전달하는 과정에서, 소스노드까지의 경로를 갖고 있는 중간노드를 만나게 될 경우, 그 중간노드는 RREP-D를 유니캐스트 방식으로 소스노드까지 전송한다.

3.2.2 경로설정 방법 2

이 방법은 위의 경로설정 방법 1에 비해 경로발견 시간을 단축할 수 있으나, 중간노드로부터 목적지노드사이 노드의 이동으로 인한 거짓 경로 정보를 사용할 가능성을 높여준다. 응답하는 중간노드는 중간노드로부터 목적지노드까지의 단 하나의 최단 경로만을 소스노드에 전달한다. 설정되는 순방향 경로들은 모두 소스노드에 의해 결정된다. 소스노드의 RREQ-S에 대해서 중간노드가 목적지노드 간에 양방향 경로를 캐쉬에 가지고 있고 중간노드는 소스노드까지 역방향 경로를 갖고 있다면, (그림 4)에서와 같이 중간노드는 순방향 경로와 목적지노드로부터 중간노드까지의 역방향 경로를 RREP-I에 실어 소스노드로 유니캐스트 방식으로 전송한다. 그 외의 경우는 경로설정 방법 1에서와

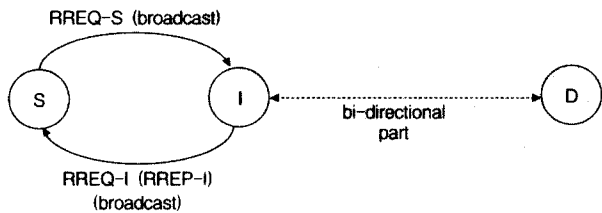
같이 중간노드가 목적지노드에게 유니캐스트로 RREQ-S를 전달한다.



(그림 4) 경로설정 방법 2

3.2.3 경로설정 방법 3

이 방법은 위의 경로설정 방법 2에 비해 중간노드가 소스노드에게 즉시 응답하게끔 하지만, 중간노드로부터 소스노드로 응답 패킷을 브로드캐스트하므로 이동 Ad Hoc 망상에서의 트래픽을 증가시키는 단점을 가진다. 이때 설정되는 순방향 경로는 모두 소스노드에 의해 결정된다. 소스노드의 RREQ-S에 대해서 중간노드가 목적지노드 간에 양방향 경로를 캐쉬에 가지고 있고 중간노드는 소스노드까지의 역방향 경로를 갖고 있지 않다면, (그림 5)와 같이 중간노드는 순방향 경로정보와 목적지노드로부터 중간노드까지의 역방향 경로정보를 RREQ-I에 실어 브로드캐스트로 소스노드에게 전송한다. 그 외의 경우는 경로설정 방법 2와 같이 수행한다.



(그림 5) 경로설정 방법 3

4. 성능평가

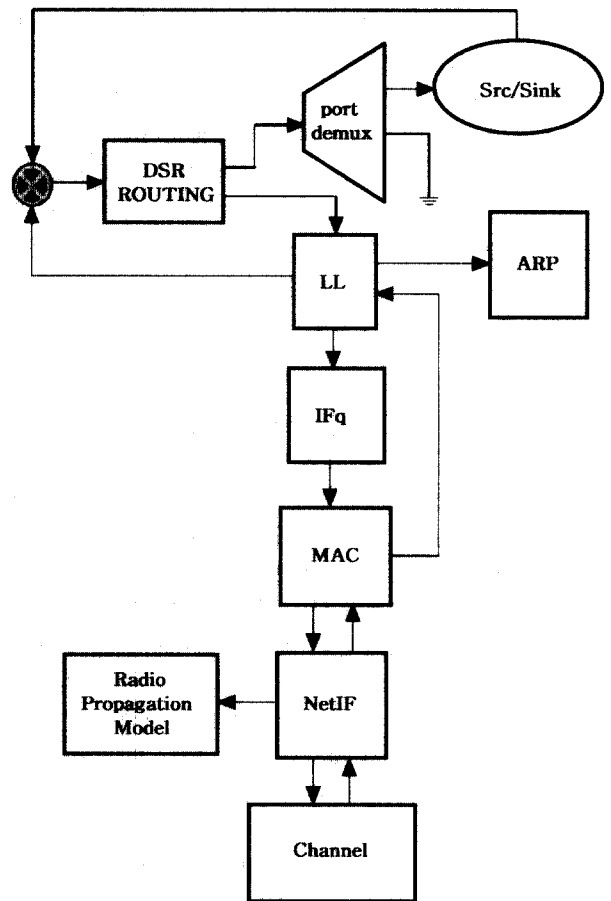
4.1 실험환경

본 논문에서는 성능평가를 위해 미국 정부의 지원하에 버클리 대학, 남가주 대학 그리고 카네기멜론 대학에서 개발한 NS-2 네트워크 시뮬레이터에 본 논문에서 제시한 내용을 코드로 구현하여 추가시켰다[15-17]. 한편 성능평가를 위해 MAC layer로서는 대칭형 양방향 링크만을 지원하는 IEEE 802.11 대신에 단방향 링크도 지원하는 웨이브랜(WaveLAN) 프로토콜을 사용하였다. 그리고 성능평가를 위한 여러 시나리오에 NS-2에 있는 이동 패턴파일과 연결 패턴파일을 서로 조합하여 만들었다. 하드웨어로는 500MHz 펜티엄 II, 64MB RAM, 13GB 하드디스크를 갖춘 PC를 사용하였다.

4.2 구성 및 평가방식

성능평가에서는 NS-2 네트워크 시뮬레이터에서 제공하

는 DSR 라우팅을 사용하는 이동노드 구조를 사용하였으며 (그림 6)에 도시하였다. 그 그림은 한 이동노드에서의 데이터 흐름을 나타내며, 'Src/Sink'는 소스노드/목적지노드의 application layer를 의미하며, 'LL'은 link layer, 'ARP'는 address resolution protocol을 의미한다. 또한 'IFq'는 interface queue를, 'MAC'은 medium access control layer를, 'NetIF'는 network interface (card)를, 그리고 'Channel'은 physical layer를 의미한다. 본 논문에서는 (그림 6)의 'DSR routing'에 대해 기존의 양방향 프로토콜 대신 단방향 프로토콜을 수행하도록 대체하였다.



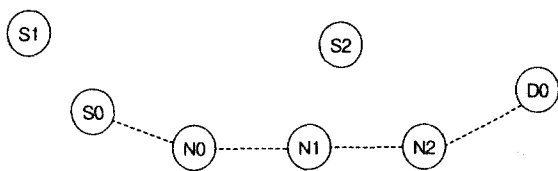
(그림 6) NS-2 시뮬레이터 하에서 DSR 라우팅을 사용하는 이동노드의 구조

성능평가를 위해 NS-2 네트워크 시뮬레이터에 포함되어 있는 2개의 이동 패턴파일과 2개의 연결 패턴파일을 조합하여 4개의 시나리오를 만들었다. 이동패턴은 노드들이 최대 20m/sec 속도로 연속적으로 움직이는 경우와 일정시간 (평균 120초) 동안 정지하였다가 움직이는 경우를 고려하였으며, 연결패턴은 데이터를 보내기 위해 서로 연결된 소스노드와 목적지노드가 많은 경우와 적은 경우를 고려하였다. 다중경로는 3개까지 허용하였으며, 다중경로를 유지하기 위해 사용하는 헬로우 메시지 시간간격은 20초로 하였다. 또한 소스노드와 목적지노드의 캐쉬에만 경로를 기록한 경우

와 경로상의 모든 중간노드의 캐쉬에 경로를 기록한 경우를 나누어 성능평가를 수행하였다. 비교 항목으로는 수신된 데이터율과 소스노드에서 발생된 경로요청패킷이 목적지노드까지의 경로를 발견하는 평균 경로설정 시간을 사용하였다. 또한 성능평가를 위해 시뮬레이터를 100초 단위로 900초까지 구동시켰다.

4.3 결과 및 고찰

성능평가 결과에 대한 분석을 위하여 수신된 데이터율 비교 항목에 대해 노드의 연결 개수와 노드의 이동성을 나누어 비교·분석하였으며, 평균경로설정시간에 대해서는 노드의 연결 개수를 고려하여 분석하였다. 분석과 관련된 그림들에서 'no'는 최적화의 경로설정 방법 1을, 'yes/no-b'는 경로설정 방법 2를 그리고 'yes/b'는 경로설정 방법 3을 나타낸다. 또한 's0 cache'에서 's'는 이동패턴을 의미하고, '0'은 노드들이 정지함이 없이 연속적으로 움직이는 것을 의미하며, 'cache'는 소스노드와 목적지노드 뿐만 아니라 경로상에 중간노드들도 경로 정보를 가지고 있는 경우를 나타낸다. 's120 nocache'는 '120'은 최대 20m/sec 속도로 이동하던 노드들이 평균 120초 동안 정지하였다가 다시 이동함을 의미하며, 'nocache'는 소스노드와 목적지노드에만 경로 정보를 가지고 있고 경로 상에 중간노드들은 경로 정보를 가지고 있지 않은 경우를 나타낸다. 'cache' 즉 중간노드의 라우팅 캐쉬를 사용하는 경우와 'nocache' 즉 중간노드의 라우팅 캐쉬를 사용하지 않는 경우에 대한 예를 (그림 7)을 통해 설명한다. 현재 소스노드 'S0'과 목적지노드 'D0'간에는 이미 경로가 설정되어 있고, 이때 새로운 노드 'S1'과 'S2'가 목적지노드 'D0'로의 경로 발견을 요청중이라고 가정하며 여기서는 순방향 경로 정보에 대해서만 언급한다. 중간노드의 라우팅 캐쉬를 사용하는 경우, 'S0'와 'D0' 간에 순방향 경로 정보는 'S0', 'N0', 'N1', 'N2'의 라우팅 캐쉬에 있다. 그러므로 'S1'와 'S2'에 의한 RREQ 경로 발견 요청은 노드 'S0', 'N0', 'N1', 'N2'나 목적지노드인 'D0'에서 처리된다. 이때 'S0', 'N0', 'N1', 'N2'는 'S1'나 'S2'의 경로 발견 요청시 중간노드가 된다. 반면에 중간노드의 라우팅 캐쉬를 사용하지 않는 경우, 'S0'와 'D0' 간에 순방향 경로 정보는 'S0'의 라우팅 캐쉬에만 있으므로, 'S1'와 'S2'에 의한 RREQ 경로 발견 요청은 노드 'S0'나 목적지노드인 'D0'에서 처리된다. 이때 'S0'는 'S1'나 'S2'의 경로 발견 요청시 중간노드가 된다.



(그림 7) 중간노드들의 라우팅 캐쉬를 사용하는 경우와 사용하지 않는 경우에 대한 예

4.3.1 데이터 수신율 분석

1) 연결 개수가 적은 경우

(그림 8)을 살펴보면 연속적인 이동성을 갖는 경우, 전체적으로 중간노드의 캐쉬정보를 사용하는 것이 더 좋은 데이터수신율을 나타내었으며 그 중에서도 경로설정 방법 3을 사용하였을 때 가장 높은 데이터수신율을 얻었다. 시간이 경과함에 따라 데이터수신율은 전체적으로 서서히 증가하다가 다시 감소하면서 비교적 일정한 값을 가졌다. 비연속적인 이동성을 갖는 경우, 전체적으로 데이터수신율이 초기에 급격히 감소하다가 400초 이후에서는 서서히 감소한 후, 일정한 값을 가졌다. 초기에는 중간노드의 캐쉬정보를 사용하지 않는 경로설정 방법 1이 가장 높은 데이터수신율을 나타내었으나, 300초 이후부터는 중간노드의 캐쉬정보를 이용하는 경로설정 방법 2가 가장 높은 데이터수신율을 제공했다. 경로설정 방법 1에 대해서 중간노드의 캐쉬정보를 이용하지 않는 경우 더 높은 데이터수신율을 보여주었고, 반면에 경로설정 방법 2에 대해서 100초 이외에는 중간노드 캐쉬정보를 이용한 경우 더 높은 데이터수신율을 보였다. 경로설정 방법 3에 대해서 500초까지는 중간노드의 캐쉬정보를 이용하지 않는 경우가 그 이후에는 캐쉬정보를 이용한 경우가 더 높은 데이터수신율을 제공했다.

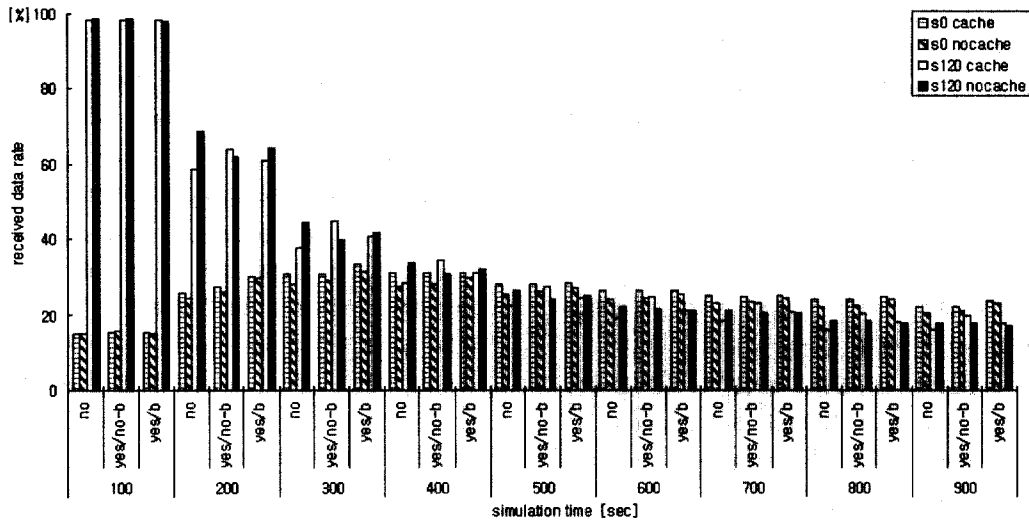
2) 연결 개수가 많은 경우

(그림 9)을 살펴보면 연속적인 이동성을 갖는 경우, 초기에는 중간노드의 캐쉬정보를 사용하지 않는 경로설정 방법 1이 가장 높은 데이터수신율을 보였으나, 그 이후에는 중간노드의 캐쉬정보를 사용하지 않는 경로설정 방법 1과 중간노드의 캐쉬정보를 사용하는 경로설정 방법 3의 데이터수신율이 가장 높았다. 경로설정 방법 1의 경우에는 중간노드의 캐쉬정보를 이용하지 않을 때 더 높은 데이터수신율을 보였으며, 경로설정 방법 2와 3의 경우에는 거의 대부분의 경우 중간노드의 캐쉬정보를 이용하였을 때 더 높은 데이터수신율을 보였다. 시뮬레이션 시간이 지남에 따라 데이터수신율이 전체적으로 점차 감소하다가 나중에 비교적 안정된 값을 가졌다. 비연속적인 이동성을 갖는 경우, 초기에는 중간노드의 캐쉬정보를 이용하는 경로설정 방법 2의 데이터수신율이 가장 높았으나, 그 이후에는 세 가지 경로설정 방법들이 거의 비슷한 데이터수신율을 제공하였다. 경로설정 방법 1의 경우에는 200초 이후부터는 중간노드의 캐쉬정보를 이용할 때 더 높은 데이터수신율을 보였다. 반면에 경로설정 방법 2와 3의 경우, 처음에는 중간노드 캐쉬정보를 이용할 때, 더 높은 데이터수신율을 보였으나, 나중에는 중간노드의 캐쉬정보를 사용하지 않았을 때 더 높은 데이터수신율을 보였다.

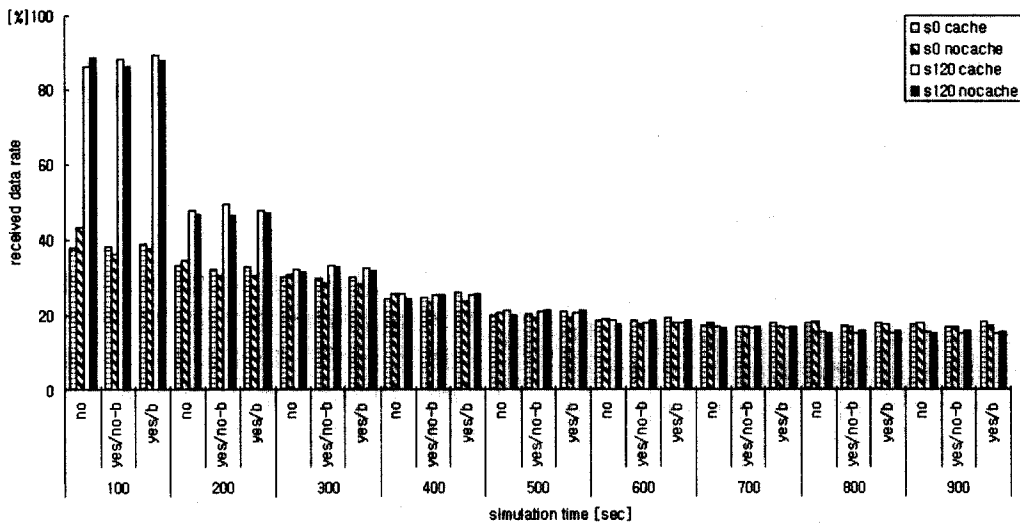
4.3.2 평균 경로설정시간 분석

1) 연결 개수가 적은 경우

(그림 10)에서 알 수 있듯이 연결노드의 개수가 적고 연속적인 이동성을 갖는 경우, 시뮬레이션 시간이 200초 이후



(그림 8) 연결이 10인 경우 이동성에 따른 데이터수신률



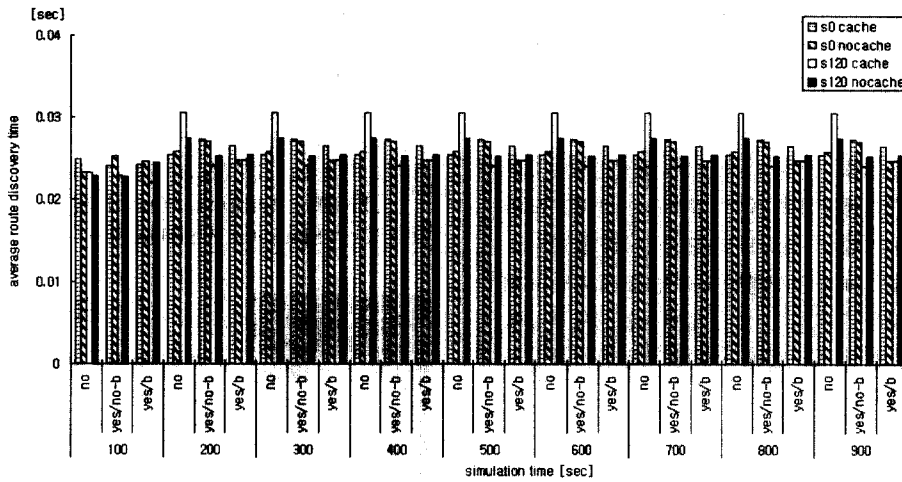
(그림 9) 연결이 30인 경우 이동성에 따른 데이터수신률

에서 각 경로설정 방법들에 대해 거의 일정한 값으로 안정되었고, 중간노드의 캐쉬정보를 사용하지 않는 경로설정 방법 3이 가장 짧은 경로설정시간을 나타내었다. 경로설정 방법 1의 경우에는 중간노드의 캐쉬정보를 사용할 때 경로설정시간이 더 짧아졌으나, 경로설정 방법 2와 3의 경우에는 중간노드의 캐쉬정보를 사용하지 않을 때 경로설정시간이 더 짧아졌다. 반면에, 연결노드의 개수가 적고 비연속적인 이동성을 갖는 경우, 시뮬레이션 시간이 200초 이후에서 각 경로설정 방법들에 대해 거의 일정한 값으로 안정되었고, 중간노드의 캐쉬정보를 사용하는 경로설정 방법 2가 가장 짧은 경로설정시간을 제공하였다. 연속적인 이동성을 갖는 경우와는 반대로, 경로설정 방법 1의 경우에는 중간노드의 캐쉬정보를 사용하지 않을 때 경로설정시간이 더 짧아졌으나, 경로설정 방법 2와 3의 경우에는

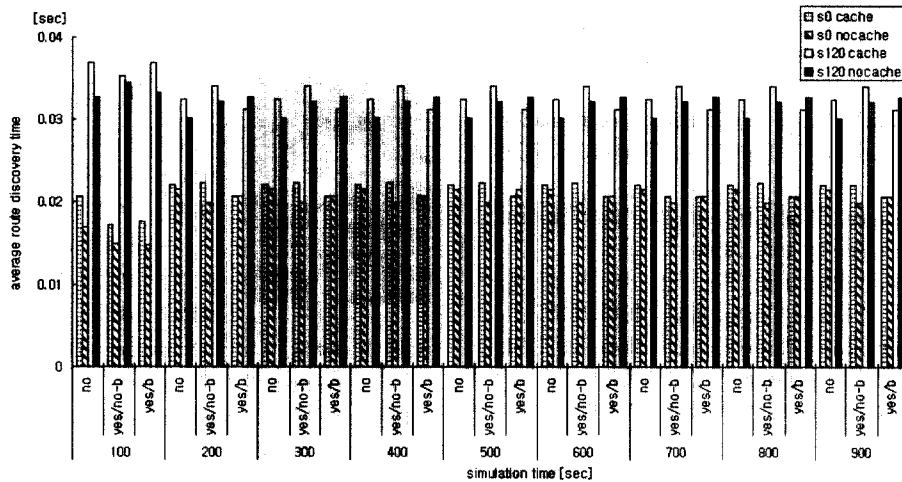
중간노드의 캐쉬정보를 사용할 때 경로설정시간이 더 짧아졌다. 한편, 경로설정 방법 1의 경우에는 연속적인 이동성을 갖는 경우가 비연속적인 이동성을 갖는 경우보다 더 짧은 경로설정시간을 나타내었다. 경로설정 방법 2의 경우에는 연속적인 이동성을 갖는 경우가 비연속적인 이동성을 갖는 경우보다 더 긴 경로설정시간을 나타내었다. 반면에 경로설정 방법 3의 경우에는 연속적인 이동성을 갖는 경우 중간노드의 캐쉬정보를 사용하지 않을 때 경로설정시간이 더 짧아졌지만, 비연속적인 이동성을 갖는 경우 중간노드의 캐쉬정보를 사용할 때 경로설정시간이 더 짧아졌다.

2) 연결 개수가 많은 경우

(그림 11)에서 알 수 있듯이 연결노드의 개수가 많고 연속적인 이동성을 갖는 경우, 시뮬레이션 시간이 200초 이후에서



(그림 10) 연결이 10인 경우 평균경로설정시간



(그림 11) 연결이 30인 경우 평균경로설정시간

각 경로설정 방법들에 대해 거의 일정한 값으로 안정되었고, 중간노드의 캐쉬정보를 사용하지 않는 경로설정 방법 2가 가장 짧은 경로설정시간을 보여주었다. 경로설정 방법 1과 2의 경우에는 중간노드의 캐쉬정보를 사용하지 않을 때 경로설정시간이 더 짧아졌으나, 경로설정 방법 3의 경우에는 중간노드의 캐쉬정보를 사용할 때와 사용하지 않을 때 거의 같은 경로설정시간을 나타내었다. 반면에 연결노드의 개수가 많고 비연속적인 이동성을 갖는 경우, 시뮬레이션 시간이 200초 이후에서 각 경로설정 방법들에 대해 거의 일정한 값으로 안정되었고, 중간노드의 캐쉬정보를 사용하지 않는 경로설정 방법 1이 가장 짧은 경로설정시간을 제공하였다. 연속적인 이동성을 갖는 경우와 마찬가지로, 경로설정 방법 1과 2의 경우에는 중간노드의 캐쉬정보를 사용하지 않을 때 경로설정시간이 더 짧아졌다. 그러나 경로설정 방법 3의 경우에는 중간노드의 캐쉬정보를 사용할 때 경로설정시간이 더 짧아졌다. 한편, 경로설정 방법 1, 2 그리고 3 모든 경우에서 연속적인 이동성

을 갖는 경우가 비연속적인 이동성을 갖는 경우보다 더 짧은 경로설정시간을 나타내었다.

5. 결 론

본 논문에서는 단방향 링크가 존재하는 실제적인 이동 Ad Hoc 망에서 가장 대표적인 DSR 라우팅 프로토콜이 구동할 수 있는 단방향 라우팅 프로토콜을 제안하고, 미국 버클리 대학의 NS-2 네트워크 시뮬레이터에 코드로 구현한 후, 성능평가를 수행하였다. 성능평가 항목으로는 데이터수신율과 평균경로설정시간을 고려하였으며, 카네기멜론 대학에서 제공하는 이동 패킷파일과 연결 패킷파일을 조합하여 4가지 시나리오 파일을 만들었다. 이 시나리오 파일을 이용하여 성능평가 항목에 대해 제안한 알고리즘들을 비교·평가하여 다음과 같은 결론을 얻었다. 데이터 수신율의 경우 연결노드의 개수에 상관없이 시뮬레이션의 초기에는 비연속적인 이동성을 갖는 경우가

연속적인 이동성을 갖는 경우보다 훨씬 더 높은 데이터수신율을 보였으나, 중반 이후부터는 연속적인 이동성을 갖는 경우가 오히려 약간 더 높은 데이터수신율을 제공하였다. 한편, 평균 경로설정시간의 경우 전체적으로 연결노드의 개수가 많고 연속적인 이동성을 갖는 경우 가장 짧은 경로설정시간을 나타내었다. 비연속적인 이동성을 갖는 경우에는 연결노드의 개수가 적을 때 더 가장 짧은 경로설정시간을 나타내었다.

이상으로 본 논문에서 제안한 단방향 라우팅 프로토콜이 향후 연구될 비대칭 경로를 포함하는 보다 일반적인 무선환경에서 동작하는 MANET 통신 시스템을 구현하기 위한 기초로서 사용될 수 있으리라 사료된다.

참 고 문 헌

[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] R. Pandya, and et al., "IMT-2000 standard : Network aspects," IEEE Pers. Commun., pp.20-29, Aug. 1997.

[3] A. Ephremides and T. Truong, "Scheduling Algorithms for Multi-hop Radio Networks," IEEE Trans. Comput. Vol.38, 1989.

[4] Andrew S. Tanenbaum, *Computer Networks*, Third Edition, Prentice-Hall International Inc., 1996.

[5] C. E. Perkins and P. Bhagwat, "Routing over Multi-hop Wireless Network of Mobile Computers," SIGCOMM'94 : Computer Communications Review, pp.234-244, Oct. 1994.

[6] D. D. Clark and D. L. Tennenhouse, "Architectural Consideration for a New Generation of Protocols," ACM SIGCOMM '90, Aug. 1990.

[7] Shree Murphy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Algorithm for Wireless Networks," MONET, Vol. 1(2), pp.183-197, Oct. 1996.

[8] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," T. Imielinski and H. Korth, editors, Mobile Computing, Kluwer Academic Publishers, 1996.

[9] C. Perkins and E. Royer, "Ad Hoc On-Demand Distance Vector Routing," 2nd IEEE Workshop on Selected Areas in Communication, pp.90-100, Feb. 1999.

[10] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," IEEE INFOCOM'97, Apr. 1997.

[11] M. R. Pearman and Z. J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," IEEE Journal on Selected Area in Communication, pp.90-100, Feb. 1999.

[12] P. Sinha, S. V. Krishnamurthy and Son Dao, "Scalable Unidirectional Routing with Zone Routing Protocol (ZRP) Extensions for Mobile Ad-Hoc Networks," IEEE WCNC'2000, pp.1329-1339, 2000.

[13] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," ACM SIGCOMM'97, 1997.

[14] J. Broch, D. A. Maltz, and et. al., "A Performance Comparison

of Multi-Hop Wireless Ad-Hoc Network Routing Protocol," The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, Oct. 1998.

[15] Bjarne Stroustrup, *The C++ Programming Language*, Third Edition, Addison-Wesley, 1997.

[16] Clif Flynt, *Tcl/TK for Real Programmers*, AP PROFESSIONAL, Academic Press, 1994.

[17] NS-2 Simulator Web Site, <http://www.isi.edu/nsnam/ns/>.



박 승 옥

e-mail : kswmj@mail.mjc.ac.kr

1996년 명지대학교 전자공학과 졸업(공학사)

1998년 명지대학교 전자공학과 졸업
(공학석사)

2001년 명지대학교 전자공학과 졸업
(공학박사)

2001년~현재 명지전문대학 정보통신과 조교수

관심분야 : 무선 네트워크, 보안, 웹프로그래밍, 멀티미디어



이 광 배

e-mail : kblee@mju.ac.kr

1979년 고려대학교 전자공학과 졸업(공학사)

1981년 고려대학교 전자공학과 졸업
(공학석사)

1981년~1982년 삼성반도체

1982년~1983년 금성사

1984년~1986년 Univ. of Southern California, Computer Engineering 전공(공학석사)

1986년~1991년 Arizona state Univ., Computer Engineering 전공
(공학박사)

1992년~1998년 명지대학교 전자공학과 교수

1998년~1999년 Worchestor Polytechnic Institute 대학 WINLAB 연구실 방문 연구 교수

1999년~2000년 Columbia 대학 COMET연구실 방문 연구 교수

2000년~현재 명지대학교 전자공학과 교수

관심분야 : 이동 무선 인터넷망, 멀티미디어, 컴퓨터 구조



김 현 옥

e-mail : acalab@mju.ac.kr

1978년 고려대학교 전자공학과 졸업(공학사)

1980년 고려대학교 전자공학과 졸업
(공학석사)

1987년 고려대학교 전자공학과 졸업
(공학박사)

1980년~1981년 동양공업전문대학 전자과 전임강사

1981년~1988년 명지대학교 전자공학과 교수

1988년~1990년 Dept. of Computer Science of Arizona State University Adjunct Faculty

1990년~현재 명지대학교 전자공학과 교수

관심분야 : 이동 무선 인터넷망, 멀티미디어, 컴퓨터 구조, 보안