

논문 2010-47TC-4-2

# 무선 애드혹 네트워크에서 다중경로 라우팅을 위한 경로 안정성 측정 방법

( A Way to Evaluate Path Stability for Multipath Routing in Wireless  
Ad Hoc Networks )

장 복 전\*, 박 용 진\*\*

( Fu-Quan Zhang and Yong-Jin Park )

## 요 약

무선 애드혹 네트워크에서 노드들의 이동성은 각각 다르다. 각 경로를 구성하는 노드는 라우터로서 동작하고 있고, 따라서 이러한 노드들이 이동하게 되면 빈번하게 새로운 경로를 탐색해야 한다. 경로를 구성하고 있는 노드 중 일 노드가 높은 이동성을 갖을 경우, 이러한 노드는 경로의 단절을 수시로 야기시키므로 경로 선택에서 최대한 회피해야만 한다. 본 논문에서는 무선 애드혹 네트워크에서 다중경로 라우팅을 위한 경로 안정성 평가를 위해 기존에 알려져 있는 다중경로 라우팅 프로토콜인 Ad hoc on-demand multipath distance vector (AOMDV)를 확장한 한 이동성 인식 방법으로, path stability evaluation (PSE) 프로토콜을 제안한다. AOMDV와 PSE 프로토콜을 ns-2를 통해 비교하였으며, 시뮬레이션 결과에서는 PSE가 패킷 전달, 중단 간 딜레이(end-to-end delay), 라우팅 오버헤드(routing overhead) 및 처리율 측면에서 보다 optimistic한 성능을 보여주고 있음을 증명한다.

## Abstract

The mobility of nodes are different in wireless ad hoc networks. Routes have to be refreshed frequently due to the mobility of the nodes acting as routers. Path consists of a few highly mobile nodes should be avoided to forward packets, because even only one highly mobile node may result in the break of path. We develop a mobility aware method to evaluate path stability for multipath routing in wireless Ad Hoc networks. Specifically, we extend it to a well-studied multipath routing protocol known as ad hoc on-demand multipath distance vector (AOMDV). The resulting protocol is referred to as path stability evaluation (PSE). Performance comparison of AOMDV with PSE are studied through ns-2. Simulation shows that PSE has optimistic results in performance metrics such as packet delivery, end-to-end delay, routing overhead and throughput.

**Keywords :** 다중경로, 라우팅, 이동성, 무선 애드혹 네트워크

## I. 서 론

모바일 무선 애드혹의 네트워크(wireless ad hoc network) 토폴로지는 이동 노드들이 자유롭게 돌아다

니거나 네트워크에 참가 및 탈퇴할 수 있어 동적으로 변한다. 이러한 노드의 이동성과 관련한 문제에 대한 연구가 진행되어 왔다. 이런 애드혹 네트워크 상에서 가장 기본적이고 자주 언급되는 부분 중 하나는 이동 호스트가 네트워크 상의 다른 어떠한 호스트로도 데이터의 송수신이 가능하도록 안정적인 경로를 얻어야 한다는 것이다<sup>[1]</sup>.

일반적으로 애드혹 네트워크 상의 각 노드들의 이동성은 제각각 다르다. 이 중, 경로를 설정할 때 낮은 이

\* 학생회원, \*\* 정회원, 한양대학교 전자컴퓨터통신공학부 (Dept. of Electronics and Communications, Hanyang University)

※ 이 논문은 2009년도 두뇌한국21사업에 의하여 지원되었음.

접수일자: 2009년12월22일, 수정완료일: 2010년4월13일

동성을 갖는 노드를 선택하게 되면 경로 재설정 횟수를 줄일 수 있다. 또한 반대로, 높은 이동성을 갖는 노드는 경로의 단절을 야기시키므로 경로 구성 시 최대한 회피해야만 한다.

기존의 이동 애드혹 네트워크 상의 다중 홉 라우팅에서는 일반적으로 Proactive<sup>[2~3]</sup> 혹은 Reactive<sup>[4~6]</sup> 방식의 프로토콜을 사용하여 소스와 목적지 노드 간의 통신에서 단일 활성 경로를 사용하는 방법을 채용해 왔다. Proactive 방식은 On-demand 방식에 비해 사전에 요구되는 라우팅 오버헤드가 크다는 이유로<sup>[7]</sup> 노드의 이동성에 대한 처리 과정에서 불리하다. 그러나 On-demand 프로토콜은 대규모 네트워크와 이동 노드가 많이 밀집해 있는 시나리오<sup>[8, 11]</sup>에서 새로운 경로가 요청되었을 때 상당한 경로 탐색 지연이 발생한다는 단점이 있다. 다중경로 on-demand 프로토콜<sup>[4, 8]</sup>들은 이러한 문제들을 해결한다.

On-demand 프로토콜은 일반적으로 소스 시작의 (Source-initiated) 경로 탐색 절차를 사용한다. 트래픽 소스는 경로가 필요할 때 마다 목적지 노드까지의 경로 설정을 위해 경로 요청 (Route request) 메시지를 Flooding하는 경로 탐색 절차를 시작한 후 경로 응답 (Route reply)을 기다린다. 각 경로 탐색 flood는 상당한 지연 및 오버헤드와 연관이 있다.

대부분의 기존 On-demand 애드혹 라우팅 프로토콜은 이 경로 탐색 절차 후 단일경로를 설정하게 된다. 다중경로 애드혹 라우팅 프로토콜은 한번의 경로 탐색 절차를 통해 복수의 Disjoint한 경로를 설정한다. 단일경로 프로토콜은 소스로부터 목적지까지의 경로가 단 하나 뿐이므로, 해당 경로에 이상이 생길 때마다 새로이 경로 탐색 절차를 수행해야만 한다. 반면 다중경로 프로토콜의 경우, 다중경로 모두가 끊길 때에만 경로 탐색 절차가 재요구된다. 따라서 On-demand 다중경로 프로토콜은 경로가 끊길 때 단일경로 프로토콜에 비해 어플리케이션 데이터 트래픽에 주는 방해 (interruption)가 더 적다. 또한 다중경로 프로토콜은 경로 탐색을 처리해야 하는 횟수가 더 적으므로 잠재적으로 더 낮은 라우팅 오버헤드를 갖는다.

이러한 부분들은 다중경로 애드혹 라우팅 프로토콜이 많은 장점을 가지고 있음을 보여준다. 다중경로 프로토콜은 단일경로 프로토콜과 비교할 때 일반적으로 더 낮은 오버헤드와 더 낮은 패킷 손실률, 그리고 향상된 신뢰성을 갖는다<sup>[8, 10~12]</sup>.

본 논문에서는 무선 애드혹 네트워크에서 다중경로 라우팅을 위한 경로 안정성 평가를 위해 기존에 알려져 있는 다중경로 라우팅 프로토콜인 Ad hoc on-demand multipath distance vector (AOMDV)를 확장한 한 이동성 인식 방법으로, path stability evaluation (PSE) 프로토콜을 제안한다. AOMDV와 PSE 프로토콜을 ns-2를 통해 비교하였으며, 시뮬레이션 결과에서는 PSE가 패킷 전달, 종단 간 딜레이, 라우팅 오버헤드 및 처리율 측면에서 보다 낙관적 (optimistic)인 성능을 보여주고 있음을 증명한다.

서론에 이어 II장에서는 본 논문의 배경 및 관련 연구를 설명한다. III장에서는 제안하는 PSE 프로토콜에 관해 구체적으로 설명한다. 또한 III장 1절에서는 경로 탐색 절차를 설명하고, III장 2절에서는 새로운 동적 이동성 정보 수집 방법에 대해 설명한다. 3절에서는 경로 유지 절차에 대해 설명한다. 4절에서는 경로 선택 및 패킷 배분 방법에 관해 서술한다. IV장에서는 AOMDV와 PSE의 성능을 시뮬레이션을 통해 비교한다. V장에서는 결론을 기술한다.

## II. 배경 및 관련 연구

### 1. 다중경로 라우팅

다중경로 라우팅 프로토콜은 한 쌍의 소스-목적지를 위해 복수의 경로를 이용한다는 특징이 있다. 이론적으로, 다중경로 라우팅은 그림 1에 나타난 바와 같이 Node-disjoint, Link-disjoint 및 Non-disjoint 방식으로 분류할 수 있다. Node-disjoint 다중경로 방식에서는 소스 노드와 목적지 노드를 제외한 나머지 노드들은 하나의 경로에서만 사용될 수 있다.

Link-disjoint 다중경로 방식은 각 노드들이 복수의 경로에서 공유될 수 있는 방식이다. 그러나 가장자리에 위치한 노드는 단 하나의 경로에서만 사용될 수 있다.

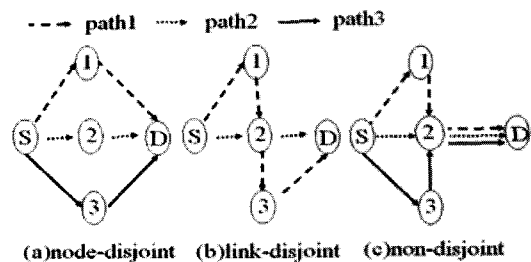


그림 1. 다중경로 분류  
Fig. 1. Multipath categories.

Non-disjoint 다중경로 방식은 노드 및 링크를 모두 공유한다. 이들 중 Node-disjoint 방식은 완전 독립 경로를 제공하므로 부하 조절 측면에서 다른 방식보다 중요하다. 따라서, 이 방식은 본 논문에서 제안한 방법에 적용시키기에 가장 알맞은 방식이므로, 본 논문에서는 이 방법을 적용한다.

### 2. AOMDV

3장의 제안 내용에 대한 이해를 돕기 위해 AOMDV의 주요 특징을 간략히 서술한다.

ADMODV<sup>[8]</sup>은 잘 알려진 단일경로 라우팅 프로토콜인 AODV<sup>[5~6]</sup>를 확장한 것이며, 노드가 경로 탐색 절차 수행 동안 소스 노드와 목적지 노드 사이에 복수의 loop-free한 경로를 찾도록 한다. 이러한 복수의 경로를 찾기 위해, 중개 노드들의 라우팅 Entry는 목적지 노드로 향하는 다음 홉(Next-hop) 노드들의 리스트과 그에 상응하는 홉 카운트 정보를 가지고 있다.

AOMDV에서 트래픽 소스(traffic source)가 목적지로의 경로를 필요로 할 때, 그 소스는 RREQ 메시지를 Flooding함으로써 경로 탐색 절차를 시작한다. 중개 노드는 서로 다른 곳에서 온 RREQ들을 버리지 않는데, 이는 이러한 RREQ에서 얻어진 정보를 통해 잠재적인 대체 역경로를 찾을 수 있을 지도 모르기 때문이다. 만약 새로운 RREQ 메시지가 Loop-free한 상태를 보장하고 다른 마지막 홉(Last-hop) 노드에서 온 것이라면, 소스로 향하는 새로운 역경로가 중개 노드에 기록된다.

만약 중개 노드가 하나 혹은 그 이상의 목적지로의 유효한 전달 경로를 알고 있다면, RREP 패킷은 그 역경로를 따라 소스로 전달된다. 목적지 노드에서는 복수의 Link-disjoint경로를 얻기 위해 하나 이상의 RREQ를 받을 때, 각 RREQ에 대한 복수의 RREP 메시지를 전송하게 된다.

AOMDV는 각 중개 노드에서 이용 가능한 경로 상의 마지막 홉 및 다음 홉 정보를 담은 유일 경로 식별자를 검사함으로써, Link-disjoint 및 Node-disjoint 방식의 다중경로를 제공한다.

본 논문에서는 노드의 이동성에 초점을 맞추고 있기 때문에, Node-disjoint 다중경로 방식이 보다 유용하다. AOMDV에 관한 보다 자세한 내용은 [8~9]에서 찾을 수 있다.

## III. PSE

### 1. 경로 탐색

경로 탐색 (Route Discovery) 절차는 대부분 AOMDV와 유사하다. 차이는 다음과 같다: RREQ 메시지가 목적지 노드에 도달하거나 중개 노드가 목적지 노드로의 유효한 어느 경로라도 알고 있을 때, RREP 패킷이 생성되고 그 경로의 역경로를 따라서 소스로 전달된다.

표 1에 나타난 바와 같이, PSE에서는 RREP가 각 경로 상의 모든 노드들의 최대 속도 안정성을 나타내는 Stability라는 확장 필드를 운반한다.

경로 안정성을 나타내기 위해 노드의 속도 값만을 사용하는 이유를 설명한다. 먼저 이동 노드들은 실제로는 단순히 등속 직선 운동만 하지는 않기 때문에, 시뮬레이션 상황에서도 이런 한정된 이동성을 적용시킬 수 없다<sup>[13~14]</sup>. 모든 이동 모델에서, 이동 노드가 자신의 현재 위치에서 새로운 위치로 이동을 할 때, 방향과 속도가 새로이 선택될 수 있다. 새롭게 선택된 방향 및 속도는 무작위로 혹은 계획적으로 선택된다. 일반적으로 사용되는 경로 안정성 계산법에서는 각 이동 모델의 속도와 방향 모두에 의존하기 때문에, 본 논문에서는 계산을 줄이기 위해 속도 값만을 사용한다. 즉, 만약 한 경로가 다른 경로보다 더 높은 속도 값을 갖는다면, 그것은 그 경로가 더 낮은 안정성을 갖고 있다는 의미이다.

그림 2는 경로 안정성 갱신과 비교 과정을 나타내고 있다. RREP가 목적지 노드로부터 역경로를 따라서 전달되고 있을 때, RREP는 속도 값을 Piggyback한다. 현재 노드의 속도와 RREP 패킷의 속도 값을 각 중개

표 1. AOMDV와 PSE의 라우팅 테이블 entry의 경로 리스트 구조

Table 1. Route-list structure of routing table entries for AOMDV and PSE.

AOMDV	PSE
destination	destination
route-list	route-list
{	{
(next-hop1, hopcount1,	(next-hop1, hopcount1,
last-hop1),	last-hop1, stability1),
(next-hop2, hopcount2,	(next-hop2, hopcount2,
last-hop2),	last-hop2, stability2),
...	...
}	}

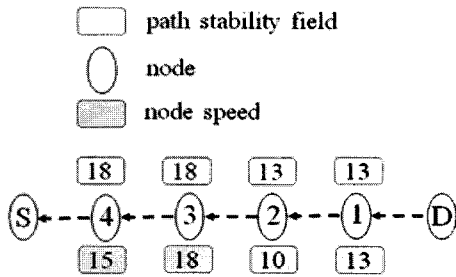


그림 2. 경로 안정성 갱신 및 비교 과정  
Fig. 2. The procedure of comparison and path stability update.

노드에서 비교하고, 더 큰 값이 RREP 패킷 필드에 갱신된다. 역경로 상의 중개 노드들에서 측정된 최대 속도 값은 RREP 패킷이 소스 노드에 도달함으로써 획득된다.

2. 동적 이동성 정보 수집

이 절에서는 새로운 동적 이동성 정보 수집 방법을 설명한다. AOMDV는 이웃 노드의 캐시를 갱신하기 위해 HELLO 메시지를 브로드캐스트(broadcast)한다. AOMDV에서 사용되는 HELLO 메시지는 이 메시지를 생성한 호스트의 주소만을 유지하고 있다.

본 논문에서는 이 HELLO 메시지에 2개의 새로운 필드를 추가하여 확장한다. 첫 번째 필드는 경로 안정성을 기록한다. 이 값은 주기적으로 갱신되고 노드가 HELLO 메시지를 브로드캐스트(broadcast)할 때 piggyback하여 브로드캐스트된다. 두 번째 필드는 경로의 마지막 홉을 기록한다. 한 경로의 마지막 홉 정보는 다른 소스-목적지 쌍을 갖는 각 경로가 공유하고 있는 노드에게 그 경로 각각에 대한 안정성을 측정할 수 있는 정보를 제공하는 유일 경로 식별자의 역할을 한다. 즉, 한 경로가 서로 다른 두 경로에 의해 공유되고 있을 때, 그 각각의 경로에 대한 속도 값 계산을 수행할 수 있도록 한다. 한 중개 노드에서 특정 경로의 마지막 홉을 안다면, 그 경로를 따라 갱신된 안정성 값을 중계 및 비교할 수 있다.

HELLO 패킷의 TTL 값은 단 하나이므로 AOMDV에서는 이것에 정보를 piggyback할 수 없다. 제안된 방법은 HELLO 패킷을 확장하여 AOMDV와는 대조적으로 모든 경로를 따라 정보를 중계할 수 있게 한다.

제안된 방법에서는 기존의 AOMDV에서 라우팅 경로 유지를 위해 사용되는 HELLO 메시지를 확장하여 사용하므로, 추가적인 제어 오버헤드는 발생하지 않는다.

한 경로를 따라 각 노드가 HELLO 메시지를 주기적으로 브로드캐스트할 때, 이동성 정보를 뿌리기 위해 아래의 절차가 수행된다.

Procedure SendHello:

- 1: Foreachentry in Routing Table do
- 2: HelloStability = MAX(Rtablestability,Currentspeed);
- 3: HelloLast-hop = last-hop;
- 4: End
- 5: DoSendRoutines();

HelloStability는 HELLO 메시지에서 현재 경로의 안정성을 나타내기 위해 확장된 첫 번째 필드이다. III장 1절에서 설명하였듯이, 경로 안정성은 간단히 Rtablestability와 Currentspeed 중 큰 속도 값으로 나타난다. Rtablestability는 표 1에서 보여진 바와 같이 라우팅 테이블에서 확장된 필드이다. Currentspeed는 현재 노드가 HELLO 메시지를 보내려 할 때의 현재 노드의 속도값을 의미한다. HelloLast-hop는 사용 중인 경로의 마지막 홉을 기록하기 위해 확장된 두 번째 필드이다. DoSendRoutines()는 AOMDV의 HELLO 메시지 수신 과정에서 수행될 필요가 있는 모든 루틴을 나타내고 있다.

각 노드가 HELLO 메시지를 받을 때, HELLO 메시지가 담고 있는 속도 정보와 현재 노드의 속도 정보를 비교하고, 큰 값이 라우팅 테이블로 업데이트된다.

비록 어느 경로가 선택되었더라도, 만약 소스 노드 혹은 목적지 노드가 최대 속도 값을 갖고 있다면 최종 속도 값이 항상 이 값으로 나오게 되므로, 목적지 노드와 소스 노드에서 얻어진 속도는 수집 대상에 포함시키지 않는다. 따라서 특히 고려되어야 하는 점은, 소스 노드와 목적지 노드는 비교 대상이 되지 않게 해야 한다는 것이다. 불필요한 계산을 줄이기 위해, HELLO 메시지가 라우팅 테이블에 존재하는 다음 홉 노드로부터 온 경우에만 비교하도록 한다.

한 HELLO 메시지를 수신하면, 다음의 절차가 수행된다.

Procedure RecvHello:

- 1: ForeachHELLOmessage do
- 2: If HELLOsource is intermediate node do
- 3: If exists RtableNextHop is HELLOsource do

```

4:   If exists Rtablelast-hop equals Hellolast-hop do
5:   Rtablestability= MAX(Helloostability,Currentspeed);
6:   End if
7:   End if
8:   End if
9: End
10: DoRecvRoutines();
    
```

HELLO<sub>source</sub>는 HELLO 메시지를 보내고 있는 노드이다. 2번째 줄에서 소스 노드와 목적지 노드가 경로 안정성 계산에서 제외되도록 하고 있다. 3-4번째 줄은 만약 HELLO<sub>source</sub>가 현재 노드의 다음 홉이고, 둘 모두 같은 마지막 홉 값을 갖는다면, 이는 경로 안정성 값이 같은 경로를 따라 중계하고 있음을 나타낸다.

5번째 줄은 들어오는 HELLO 메시지의 Helloostability가 경로 안정성 (속도 값)을 갖고 있음을 나타낸다. Currentspeed는 현재 노드가 HELLO 메시지를 받고 있을 때의 노드의 속도 값을 의미한다.

그 값들의 비교 후, 가장 큰 값은 표 1에 나타난 라우팅 테이블에서 확장된 필드인 Rtablestability에 갱신된다. Rtablestability는 마지막 홉에서 현재 노드까지 노드들 중 최대 속도를 갖고 있다. 다시 말해, 경로 안정성 값은 HELLO 메시지에 의해 중계된다.

DoRecvRoutines() 함수는 본래의 AOMDV HELLO 메시지 수신 과정에서 행해져야 할 필요가 있는 모든 루틴들을 포함하고 있다.

### 3. 경로 유지

PSE에서의 경로 유지 (Route Maintenance)는 AOMDV의 경로 유지를 확장한다. AOMDV처럼 PSE 또한 RERR 패킷을 사용한다. 노드는 목적지 노드로의 마지막 경로가 끊겼을 때 RERR을 목적지 노드로 생성 및 전송한다. 이 때 경로 단절로 인해 소실되어야 하는 패킷들이 발생하는데, 이 패킷들은 다른 대체 경로를 통해서 새로이 재전달된다. 다중 경로 프로토콜에서는 stale된 경로의 발생 가능성이 더 높다. 그러나 그러한 stale된 경로는 HELLO 메시지를 사용해서 순항적으로 제거될 수 있다.

### 4. 경로 선택 및 패킷 분배

AOMDV는 한번에 단 하나의 경로만을 사용하고 현재 이용 중인 경로에 이상이 생기지 않는 한 결코 다른

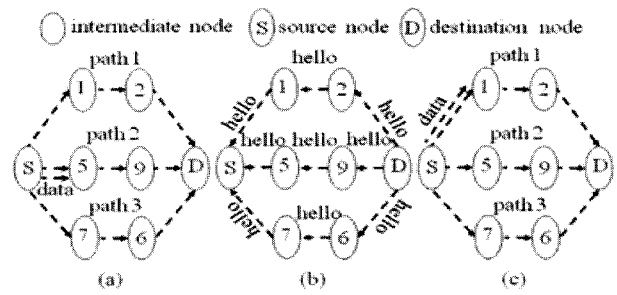


그림 3. 경로 최적화 및 패킷 할당 예  
Fig. 3. An example of path optimization and packet allocation.

사용 가능한 경로로 변경하지 않는다.

PSE는 경로 유지 과정 동안 브로드캐스트된 HELLO 메시지를 이용함으로써 주기적으로 가장 최신의 이동성 정보를 수집한다. 더 합리적인 패킷 배분은 이동성 정보를 통해 수행될 수 있는데, 본 논문에서는 네트워크의 변경 상황에 빠른 응답(quick response)을 이용하는 방법을 제안한다.

본 논문에서는 최소 속도 값을 갖는 경로에게 패킷 우선 순위를 부여하는 방법을 제안한다. 동적인 이동성 정보 갱신으로, 현재 사용 중인 경로에서는 속도 값이 높아질 경우 트래픽을 다른 경로들로 넘겨줄 수 있으며, 따라서 발생할 수 있는 경로 단절과 한 경로에서의 높은 부하를 피하게 된다.

그림 3에서는 경로 최적화 및 패킷 배분의 예를 보여 주고 있다. 우선, 그림 3a와 같이 다중경로가 설정되고 패킷들은 경로 탐색 단계에서 수집된 최소 이동성을 갖는 경로로 분배된다.

데이터 전송이 시작되고, 각 노드의 이동성 값은 변한다. 패킷 배분은 각 변경을 조정하기 위해 재정리된다. 본 제안에서, 이 재정리 처리는 추가적인 오버헤드의 사용을 방지하기 위해 HELLO 메시지를 이용해 경로 유지 단계에서 수행된다(그림 3b).

Upstream 노드로부터 HELLO 메시지를 수신한 후, 각 경로의 이동성 정보는 최신 값으로 갱신된다(그림 3c). 또한 갱신된 이동성 정보에 따라 새로운 패킷 할당 결정이 이루어진다.

## IV. 시뮬레이션 및 결론

Node-disjoint AOMDV와 PSE의 효율성을 평가 및 비교하기 위해, 본 논문에서는 NS-2.32<sup>[17]</sup>을 이용하여 시뮬레이션(simulation)하였다. 각 시뮬레이션은 다른

이동성과 트래픽 패턴을 갖고 있으며 총 25회의 시뮬레이션 결과들을 통해 평균 값을 계산하였다.

1. 시뮬레이션 파라미터

표 2는 시뮬레이션 파라미터를 정리한 것이다.

표 2. 시뮬레이션 파라미터  
Table 2. simulation parameters.

Parameters Name	Parameters Value
Protocol	AOMDV and PSE
Simulation time	600 seconds
Simulation area	100m x 300m
Nodes number	50
Propagation model	Two-ray Ground Reflection
Transmission range	250 m
Max. speed (m/s)	10
Min. speed (m/s)	1,2,3,4,5
Pause time	NO
Packet type	UDP, CBR
Max. connections	10
Packet size	1000 bytes
CBR rate	512kb/s

2. 성능 평가 요소

시뮬레이션의 성능 평가 요소는 다음과 같다:

- 패킷 전달률은 소스 노드의 어플리케이션 레이어에 의해 송신된 패킷수와 목적지 노드의 어플리케이션 레이어에 의해 수신된 패킷수 사이의 비율이다.
- 종단간 딜레이는 데이터 패킷의 종단간 평균 지연으로 측정된다.
- 라우팅 오버헤드는 시뮬레이션 시간 동안 수신된 데이터 패킷의 총 수에서 생성된 제어 패킷 (RREQ, RREP, HELLO 등)의 총 수 사이의 비율이다.
- 처리율(throughput)은 통신에서 네트워크 상의 어떤 노드나 터미널로 부터 또 다른 터미널로 전달되는 단위 시간당 데이터 전송으로 처리하는 양을 말한다.

3. 이동 모델

랜덤 웨이 포인트 이동 모델 (random way-point mobility model)은 많은 저명한 애드혹 네트워크 프로토콜들의 시뮬레이션 연구에서 사용되고 있다. 이 모델에서는 사람이 회의장이나 박물관에서 실제 이동할 때와 같은 상황을 모델링한 것이다<sup>[11, 14~15]</sup>.

본 논문에서는 랜덤 웨이 포인트 이동 모델을 사용하고, 노드는 이동 후 약간의 시간 동안 (pause time) 중지하고 다시 무작위로 속도 및 방향을 선택하여 움직인다. 이 때 선택되는 속도는 정해진 최소 속도와 최대 속도 값 사이에서 결정된다.

4. 결과 및 분석

이동성 조사에서, 그림 4는 두 프로토콜 상에서 노드 속도가 끼치는 영향을 나타낸다. 그림 4에 나타난 바와 같이 전달률의 감소는 경로 변경 횟수가 증가함에 따라, 즉 이동성이 증가함에 따라 더욱 심해진다. 이동성이 증가하면, 더 많은 수의 경로가 invalid하게 되고 새로운 request가 요구된다. 이 request 메시지들이 새로운 경로 탐색을 위해 네트워크로 전달될 동안 버퍼가 가득 차게 되고, 패킷들은 버려진다.

PSE의 패킷 전달률은 네트워크 이동성이 증가함에 따라 차차 AOMDV에 접근한다.

이것은 AOMDV가 항상 데이터 패킷을 최소 경로를 통해 전달하기 때문이다. AOMDV와는 달리 PSE에서는 항상 데이터 패킷을 안정성 있는 경로를 통해 전달한다.

최소 홉 카운트를 갖는 경로, 즉 최단 경로는 처리량 (Throughput)을 반드시 최대화시키지는 않는다<sup>[18]</sup>. 일반적으로 낮은 이동성을 갖을 때 중개 노드에서의 패킷 드롭은 더 긴 길이를 갖는 경로에서 더 자주 발생한다. 다시 말해, 패킷 드롭은 낮은 이동성 상황에서 경로 길이에 민감하고 빈번한 링크 단절로 인해 네트워크 이동성의 증가에도 점차 민감해지게 된다. 따라서, PSE의 패킷 전달률은 네트워크 이동성이 증가하는 AOMDV에 근접하게 된다.

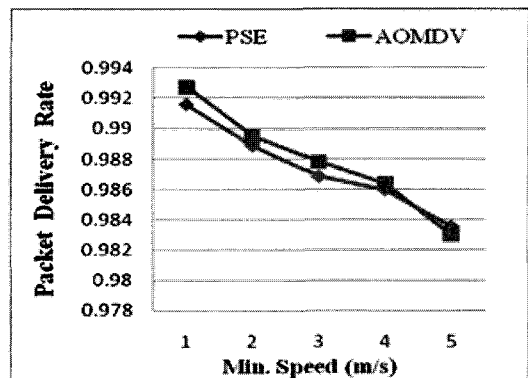


그림 4. 패킷 전달률  
Fig. 4. Packet delivery rate.

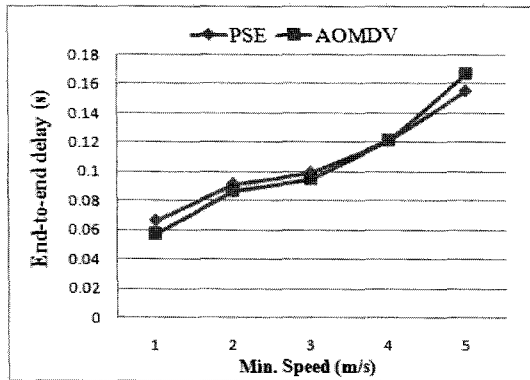


그림 5. 종단 간 딜레이  
Fig. 5. End-to-end delay.

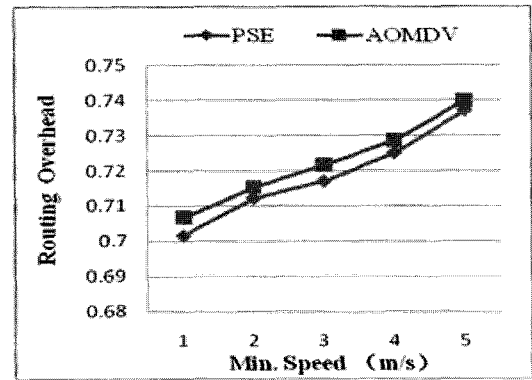


그림 6. 라우팅 오버헤드  
Fig. 6. Routing overhead.

다음으로, 그림 5는 종단 간 딜레이를 나타내고 있다. 그림 5에서 알 수 있듯이, 네트워크에 이동성이 증가하는 동안 두 프로토콜 모두 지연 시간은 증가한다. 어떤 대체 경로도 소스 노드에서 이용 가능하지 않은 경우, 경로 변경 및 링크 단절의 가능성은 더 높아진다. RREQ 메시지는 소스 노드로부터 생성된다. 이 메시지는 경로 탐색을 재수행하기 위해 모든 노드로 전송될 것이다. 그 결과, 많은 양의 RREQ 메시지가 전송된다. 이러한 메시지들과 경로 재설정은 트래픽 지연을 유발한다.

네트워크에 이동성이 더욱 커짐에 따라, 두 프로토콜 간의 지연 차는 점차 작아진다.

PSE와 AOMDV의 지연 차는 우선적으로 경로 스위칭으로 인한 패킷 드롭으로 인해 결정된다. 이동성이 증가하면 더 많은 수의 경로가 이용할 수 없게 되고, 대체 경로로의 빈번한 스위칭이 요구된다. 소스 노드가 새로운 경로로 변경하는 동안, 이전 경로에 있는 중개 노드들의 버퍼는 곧 가득 차게 될 것이고, 이로 인해 패킷이 버려지게 된다.

PSE는 항상 안정된 경로로 데이터 패킷을 전달한다. 그러나 AOMDV는 PSE와는 대조적으로 항상 최단 경로를 통해 데이터 패킷을 전송한다. 이 최단 경로는 안정된 경로가 아닐 수 있다.

낮은 이동성을 갖고 있을 때 패킷 드롭은 경로 길이에 민감하다. 이 때 이동성이 증가하게 되면, 이로 인해 링크 단절이 빈번하게 발생하기 때문에 경로 스위칭으로 인한, 즉 이동성 증가로 인한 패킷 드롭률이 점차 높아지게 된다. 따라서, PSE의 지연은 네트워크의 이동성이 증가함에 따라 AOMDV에 근접하게 된다.

그림 6에서는 이동성이 오버헤드에 끼치는 영향을

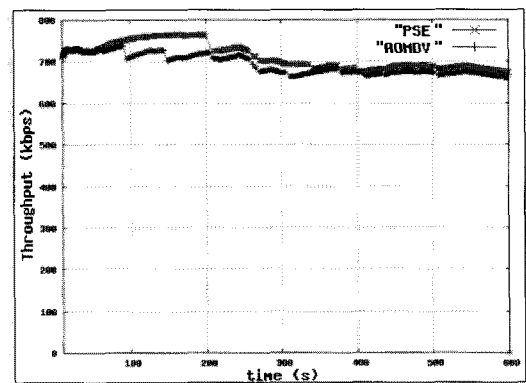


그림 7. 최소 2m/s시의 안정된 처리율의 예  
Fig. 7. Example of stable throughput in minimal speed 2m/s.

나타낸다. 오버헤드의 증가는 경로 탐색을 위한 제어 패킷의 전송이 요구되는 경로 재설정으로 인한 것이다. 각 프로토콜들의 라우팅 오버헤드는 네트워크의 이동성이 증가함에 따라 빈번한 링크 단절이 발생하게 되어 점차 증가한다.

AOMDV는 한 경로가 단절될 때까지 이 경로를 유지하고, 단절한 후에 다른 대체 경로로 변경한다. 반면 AOMDV와는 대조적으로, PSE는 경로들의 안정성을 동시에 평가 및 유지한다. 데이터 패킷은 높은 안정성을 가진 다수의 이용 가능한 경로들로 분배된다. 네트워크 이용도는 이러한 부하 조절로 인해 향상된다. 이것은 라우팅 오버헤드를 최소화하는 데에 도움이 된다.

그림 7 및 그림 8에서는 두 프로토콜에서 최소 속도가 2m/s일 때와 4m/s일 때의 경우에 대한 처리율의 예를 보여준다. 본 논문에서 제안한 프로토콜은 대체로 보다 높고 안정된 처리율을 갖는다. 그림 9에서는 AOMDV와는 달리 PSE가 최대 16%만큼 증가된 처리량을 갖는 것을 나타내고 있다.

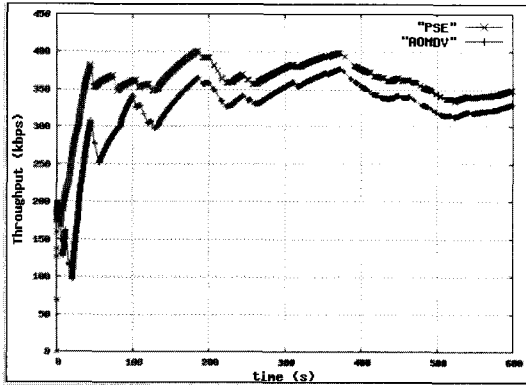


그림 8. 최소 4m/s시의 더 높아진 처리율의 예  
 Fig. 8. Example of higher throughput in minimal speed 4m/s.

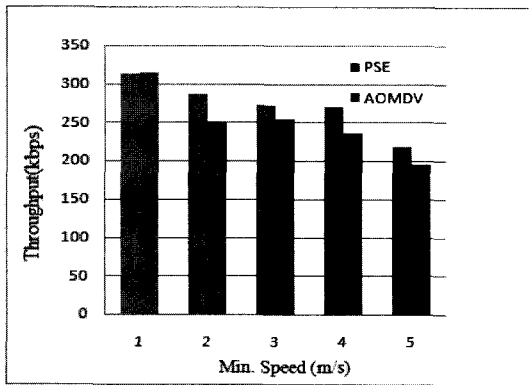


그림 9. 두 프로토콜의 처리율 비교  
 Fig. 9. Throughput of both protocols.

IV. 결론 및 미래연구

본 논문에서는 이동성 인식을 다중경로 라우팅을 위한 PSE에 통합시켰다. 특히, 다중경로 라우팅 프로토콜로 잘 알려진 AOMDV를 확장시켰다.

시뮬레이션에서는 PSE가 네트워크의 이동성이 점차 증가할 동안 라우팅 오버헤드를 줄인다는 점, 패킷전달률 및 종단 간 딜레이는 AOMDV에 근접해 간다는 점을 증명하였다.

향후에는 구체적인 이동 모델을 갖는 응용을 위해 이동 노드의 속도 및 방향을 이용하여 경로 안정성 평가를 적용시켜 볼 것이다.

References

[1] Piyush Gupta and P. R. Kumar "A system and traffic dependent adaptive routing algorithm for ad hoc networks ", Proceedings of the 36th

IEEE Conference on Decision and Control, pp. 2375-2380, San Diego, Dec. 1997.

[2] E.M. Belding-Royer, "Routing approaches in mobile ad hoc networks" in: Mobile Ad Hoc Networking (S. Basagni, M. Conti, S. Giordano, I. Stojmenovic, eds.), IEEE/Wiley, 2004, pp.275-300.

[3] C.E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for Mobile Computers." in: Proc. ACM SIGCOMM'94, London, UK, Sept. 1994.

[4] D. G. S. Mueller, "Multipath routing in mobile ad hoc networks: issues and challenges" Performance Tools and Application to Networked Systems, ser. Lecture Notes in Computer Science, M. C. Calzarossa and E. Gelenbe, Eds., vol. 2965, pp.209 - 234, Apr 2004.

[5] C.E. Perkins, E.M. Royer, S. DAS. "Ad hoc on-demand distance vector (AODV) routing" RFC3561[S], July 2003.

[6] C. Perkins "Ad hoc On-Demand Distance Vector (AODV) Routing" RFC July 2003

[7] Brown, T.X, Doshi, S., Zhang, Q., "Optimal power aware routing in a wireless ad hoc network," IEEE LANMAN 2001 Workshop Proceedings, pp. 102 - 105.

[8] M. Marina, S. Das. "On-demand multipath distance vector routing in ad hoc networks". Proceedings of the International Conference on Network Protocols (ICNP), Riverside, CA, November, 2001.

[9] M. Marina, S. Das. "Ad hoc on-demand multipath distance vector routing" Wireless Communication and mobile computing. 2006 Published online in Wiley InterScience

[10] Philipp Hurni , Torsten Braun "Energy-Efficient Multi-path Routing in Wireless Sensor Networks" LNCS, Volume 5198/2008, Feb, 2008.

[11] Yumei Liu, Lili Guo, "Energy Efficient on-demand Multipath Routing Protocol for Multi-hop Ad Hoc "Networks2008 IEEE

[12] Chen Ni, Hyeoncheol Zin, "Dynamic Packet Balancing Agent in MANETs based on AOMDV" 2009 First Asian Conference on Intelligent Information and Database Systems, IEEE

[13] Tracy Camp , Jeff Boleng, Vanessa Davies, "A Survey of Mobility Models for Ad Hoc Network Research" NSF; Grant Number: ANI-9996156 and ANI-0073699.

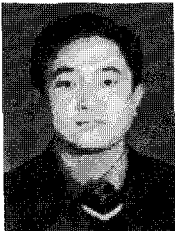
[14] Atulya Mahajan, Niranjan Potnis, Kartik Gopalan, and An-I A. Wang, "Evaluation of



Mobility Models For Vehicular Ad-Hoc Network Simulations”, Dept. of Computer Science, Florida State University.

- [15] F. Bai, A. Helmy, “A Survey of Mobility Modeling and Analysis in Wireless Ad-hoc Networks”, Chapter in the book “Wireless Ad-Hoc and Sensor Networks”, Kluwer Academic Publishers, June 2004.
- [16] William Su, Sung-Ju Lee, Mario Gerla, “Mobility prediction and routing in ad hoc wireless networks”, International Journal of Network Management 11(1): 3-30, 2001.
- [17] “The Network Simulator,”  
<http://www.isi.edu/nsnam/ns/>
- [18] Richard Draves, “Comparison of Routing Metrics for Static Multi-Hop Wireless Networks”, Microsoft Research

저 자 소 개



**장복전**(학생회원)  
 2000년 NorthEast Univ.  
 전자공학과 학사 졸업.  
 2005년 ShenYangLiGong Univ  
 컴퓨터공학과 석사 졸업.  
 2006년~현재 한양대학교 전자  
 통신컴퓨터공학과  
 박사 재학.

<주관심분야 : 통신, 컴퓨터, 신호처리, 반도체>



**박용진**(정회원)  
 1969년 와세다대학교 전자통신  
 공학과 학사 졸업.  
 1971년 와세다대학교 전자통신  
 공학과 석사 졸업.  
 1978년 와세다대학교 전자통신  
 공학과 박사 졸업.

1979년~현재 한양대학교 전자통신컴퓨터공학부  
 교수

2001년 4월~2006년 와세다대학교 대학원 국제  
 정보통신대학원 강사

2002년 10월~2006년 10월 한국인터넷협회 이사

2003년 2월~현재 한국첨단망협회  
 (국내 연구개발망 추진기관) 이사

2005년 1월~현재 IEEE Asia-Pacific Region  
 이사

2005년 2월~현재 IEICE 한국 지부 대표

2009년~현재 IEEE 아시아·태평양지역 회장

<주관심분야 : 컴퓨터통신, 이동 데이터 통신>