

Timestamp를 이용한 AODV 라우팅 알고리즘의 성능 개선

학생회원 이재석*, 박혜련**, 정회원 이기현**, 석정봉*

Performance Improvement of AODV Routing Algorithm Using Timestamp

JaeSuk Lee*, Hae-Ryeon Park**, *Student Members*,
Kee-Hyun Lee**, Jung-Bong Suk*, *Regular Members*

요 약

RFC3561로 공식 승인된 현재의 AODV 라우팅 방식은 노드 간 패킷 전달시간으로 40ms의 고정 값으로 설정되어 있다. 이 값과 TTL 값을 이용하여 RREQ 메시지가 목적 노드에 도착 한 후, RREP 메시지가 소스 노드에 되돌아오는 시간을 계산하게 되는데, 여기서 고정된 값을 사용하기 때문에 망의 효율이 저하되는 문제가 발생한다. 본 논문에서는 이러한 문제점을 개선하기 위하여 현재의 AODV 방식에서 RREQ 메시지에 timestamp 필드를 설정하여 패킷 발생 시간을 기록하여 전송하고, 이것을 받은 노드는 현재 시간과 패킷 내 timestamp 값을 이용하여 노드간의 패킷 전송시간을 망의 트래픽 상황에 따라 동적으로 계산하는 방식을 제안한다. 제안 방식의 정당성을 입증하기 위해, 제안 방식을 현재 방식과 함께 컴퓨터 모의실험을 수행하였으며, 그 결과 제안방식이 현재 방식보다, 특히 노드 이동성과 트래픽 부하가 클 경우, 우수한 성능을 나타냄을 증명하였다.

Key Words : ad hoc 라우팅, AODV, timestamp

ABSTRACT

One of the seminal operation parameters of AODV is the node-to-node travel time of data, which is currently set to fixed value of 40 ms in the RFC3561 and widely used to compute other parameters such as the expected round-trip time of the connection setup message, etc. We have naturally thought the network performance could be improved by dynamically varying the node-to-node travel time with respect to the traffic condition in the network, rather than using the fixed value, which motivates this work. To this end, we apply the idea of using timestamp; every node places the current time in the message before sending it out, and the receiver node computes the node travel time based on the moving average algorithm by considering not only the current value but also the previous ones in an accumulated and exponentially decreasing fashion with time. We evaluate the performance of the proposed scheme in respect of the number of RREQ messages generated, throughput, and delay as a function of traffic load and node mobility, and compare the result with the original AODV scheme. The results show that the proposed scheme presents noticeable performance improvements, especially under the condition of high node mobility and high traffic load.

* 본 연구는 2002년도 연세대학교 학술연구비 및 한국산업기술재단 지역혁신인력양성사업 (05-TG10) 지원 연구결과의 일부임.

* 연세대학교 전산학과 정보통신망 연구실 (ljsjjang@hanmail.net, jbsuk@yonsei.ac.kr),

** 명지대학교 컴퓨터공학과 (haerin20@hanmail.net, khlee@mju.ac.kr)

논문번호 : KICS2006-02-078, 접수일자 : 2006년 2월 14일, 최종논문접수일자 : 2006년 8월 2일

I. 서론

이동 ad-hoc 네트워크는 기지국 혹은 AP에 의한 중앙 집중화된 관리나 표준화 된 지원 서비스 없이 임시 망을 구성하는 무선 이동 단말들의 집합이다. 각 노드는 이동 노드와 네트워크 내의 다른 노드를 대신하여 데이터를 전송해주는 라우터의 역할을 동시에 수행한다. 즉 단말들이 코어와 에지의 구분이 없이 필요에 따라서 라우터 역할을 하기도 하고 일반 단말의 역할도 동시에 수행한다. Ad-hoc 네트워크의 단말은 대부분이 이동성을 갖고 있어, 네트워크의 형태가 매우 동적으로 변한다^[1].

Ad hoc 네트워크의 라우팅 알고리즘은 1990년 말경 많은 연구 및 표준기고를 통해 어느 정도 성숙단계에 도달하여, 현재는 새로운 라우팅 알고리즘의 개발 보다는 이미 개발된 라우팅 알고리즘의 단점을 보완하여 보다 향상된 성능을 위한 달성하기 위한 연구가 많이 진행되고 있다^[2,3]. Ad-hoc 라우팅 프로토콜은 보통 라우팅 테이블을 유지하고 업데이트 하는 방법에 따라 테이블 기반 (table-driven) 방식과 요구 기반 (on-demand) 방식으로 구분된다. 그 중 Ad hoc On-demand Distance Vector (AODV) 라우팅 프로토콜은 대표적인 요구 기반 방식으로, 2003년 7월 IETF RFC 정식 문서로 채택되었다^[4].

AODV와 관련된 연구 중 timestamp를 이용한 기존의 연구를 살펴보면 node_traversal_time을 측정하기 위해 따로 제어패킷을 만들어 전송하는 방법이 제안하였으나 정식으로 승인된 버전에서는 생략되었다^[5]. 또한, timestamp를 이용하여 시간차를 계산하여 확보된 거리정보를 라우팅에 사용한 연구^[6]와, 라우팅 과정의 시간차를 소스와 목적지 사이에서 구하고 그 값의 평균을 node_traversal_time 값 계산에 이용한 연구도 진행되었다^[7].

RFC3561로 공식 승인된 기존의 AODV 알고리즘은 각 노드에서 발생하는 패킷이 다음 노드에 전송되는 시간이 40ms로 고정되어 있다. 이 설정과 TTL 값을 바탕으로 RREQ 메시지가 목적지에 도착을 하여 RREP가 소스에 다시 전송되는 시간을 계산한다. 이 방식은 고정된 값을 사용하기 때문에 망의 상황에 적절히 대응하지 못하는 문제점이 있다.

본 논문에서는 이러한 문제점을 개선하기 위하여 기존의 AODV 방식에서 RREQ 메시지에 timestamp 필드를 사용하여 패킷 발생 시간을 기록하여 전송하고, 이것을 받은 노드는 현재 시간과 패킷 내

timestamp 값의 차이를 계산함으로써, 노드간의 패킷 전송시간을 구하고 이를 바탕으로 패킷이 목적지에 도착하고 응답을 기다리는 시간을 동적으로 구하는 방식을 제안하고, 컴퓨터 모의실험을 통해 제안방식이 성능과 정당성을 확인하도록 한다.

2장에서는 AODV 방식의 제어 패킷 구조 및 동작에 대하여 알아보고, 3장에서는 각각 기존 AODV 방식의 문제점을 기술하고, 이를 개선하기 위한 제안한 node_traversal_time 값이 망의 트래픽 상황에 따라 동적으로 계산하는 알고리즘을 소개한다. 4장에서는 제안방식의 성능과 정당성을 입증하기 위해 설정한 시뮬레이션 환경과 기존 방식과 함께 수행한 모의실험 결과를 기술한 후 5장에서 본 논문의 결론을 맺는다.

II. AODV 라우팅 프로토콜의 문제점 및 제안

AODV는 DSDV와 같이 목적지 순차 번호를 사용하여 라우팅 루트를 방지하며, DSR과 유사한 루트 탐색 절차를 사용한다. 루트 탐색이 필요한 경우 RREQ 메시지가 생성되어 이웃 노드로 브로드 캐스팅되며, 목적 노드로의 루트 정보를 가진 중간 노드 또는 목적 노드가 RREQ 메시지를 수신하면 RREP 메시지로써 응답한다. 중간 노드가 목적 노드로의 루트 정보를 가지고 있지 않을 경우 RREQ 메시지를 이웃 노드로 브로드캐스팅 한다. RREP 메시지는 RREQ 메시지가 전달된 루트의 반대 방향으로 유니캐스팅 된다. RREQ 메시지를 수신한 노드는 역방향 루트 정보를 생성하여 저장하며 RREP 메시지를 수신한 노드는 순방향 루트 정보를 생성하여 저장한다. 하나의 노드가 동일한 RREQ 메시지를 중복적으로 수신한 경우 최초로 수신된 것만 사용한다. 루트 내의 특정 링크에서 오류가 발생한 경우 지역적인 루트 재탐색 절차를 수행하거나, 또는 RERR 메시지가 생성 소스 노드로 전달하여 소스 노드로 하여금 루트 재탐색 절차를 시작하게 한다. RERR을 수신한 노드는 오류가 발생한 링크와 관련된 루트 정보를 삭제한다. RREQ, RREP 및 RERR 패킷의 형태 및 각 필드의 기능은 IETF RFC 문서^[4]에 나타나 있다.

2.1 AODV 라우팅 프로토콜의 문제점

AODV 라우팅 프로토콜은 불필요한 RREQ의 전송을 제한하기 위해 Expanding Ring Search (ERS) 알고리즘을 사용한다. 즉, 헤더의 TTL 필드를 이용

하여 출발지 노드로부터 가까운 지역으로부터 점차로 넓은 지역으로 확대해가면서 RREQ를 전송한다.

출발 노드는 RREQ 패킷 헤더에 TTL을 1로 하고 $2 * TTL * node_traversal_time(2 * 1 * 40)ms$ 을 타임아웃 값으로 하여 RREP를 기다린다. 이 시간동안 RREP를 수신하지 못하여 타임아웃되었을 때, 출발지 노드는 $TTL_increment(=2)$ 만큼 증가시킨 TTL을 이용하여 RREQ를 재방송한다. 이 과정은 RREQ에 TTL이 $TTL_threshold(=8)$ 될 때까지 계속된다. TTL이 $TTL_threshold(=8)$ 까지도 RREP를 수신하지 못하면, TTL을 $TTL = net_diameter(=35)$ 로 하여 재방송한다.

이렇게 $TTL=net_diameter(=35)$ 를 사용하여 RREQ를 발생시키는 횟수는 $RREQ_retries(=2)$ 까지 재방송하는데, 그래도 RREP를 수신하지 못하면 출발지 노드는 이 네트워크에 목적지 노드가 존재하지 않는다고 판단하여 더 이상 RREQ를 발생시키지 않는다.

ERS 알고리즘에서 사용하는 $node_traversal_time(=40ms)$ 은 ad-hoc 네트워크에서 패킷을 전달하기 위한 노드 간 한 홉의 평균 지연시간이다. 이 값에는 큐잉 지연, 노드의 장비의 성능, 패킷 전달시간과 무선사용으로 인한 무선 사용시간 지연 등의 여러 가지 가변적인 값이 포함되어 있다. Ad-hoc 망에 참가하는 단말의 성능은 각기 상이하고, 또한 무선자원은 시간에 따라 매우 민감하고도 빈번한 변화를 일으키게 되는데, 이러한 요인으로 인하여 노드간의 지연시간은 상당히 다양한 값을 가질 수 있다. 그러나 현재 ERS 알고리즘에서 사용하는 $node_traversal_time$ 값은 40ms의 고정 값을 사용하고 있다. 이것은 불필요한 RREQ의 재전송을 유발시켜 성능저하를 야기하게 된다.

2.2 제안한 AODV 라우팅 프로토콜

AODV에서는 라우팅 패킷마다 확장필드를 가지고 있다. 특히 RREQ에 확장필드에는 timestamp를 이용할 수 있는 공간이 존재한다. 본 논문에서는 이 timestamp를 이용하여 노드 간 한 홉의 평균 패킷 지연시간인 가변적인 $node_traversal_time$ 을 측정하는 새로운 알고리즘을 제안한다.

우선 RREQ를 발생시키는 출발지 노드는 RREQ를 발생시킬 때마다 timestamp 공간에 RREQ 발생시간을 기록하고 RREQ를 주변노드로 전달한다. RREQ를 수신한 노드는 식(1)과 같이 현재시간과 RREQ의 timestamp 값의 차를 계산한다. 이 값은 바로 이전

노드에서 RREQ를 다음 노드까지의 RREQ가 전달된 시간이 된다.

$$\text{현재시간} - \text{RREQ timestamp 값} \quad (1)$$

또한 RREQ가 중계될 때 마다 계속 새로운 timestamp를 찍어서 인근 노드에게 보내게 된다. 노드들은 RREQ를 받을 때마다 노드 전송시간을 계산하며 중복되는 RREQ에 대해서도 계산한다. 이렇게 되면 바로 인근 노드가 보낸 중복 RREQ는 전송 시간이 매우 작게 나와서 실제 사용되는 것과는 차이가 생기게 된다. 이러한 중복성을 방지하기 위해 RREQ 전송 시간을 계산할 때는 Sequence Number를 확인하여 최신의 RREQ만 대해서 전송 시간을 계산한다. n 번째 시점에서 $node_traversal_time$ 값은 식(2)와 같이 현 시점(즉 n번째 시점)에서 측정된 값과 가중 누적된 지난 값을 고려하여 계산된 이동 평균값 (moving average) 을 사용한다.

$$node_traversal_time(n) = node_traversal_time(n-1) * (1-a) + \text{측정된 전송시간}(n) * a \quad (2)$$

여기서 a는 0과 1사이의 값을 갖으며, a가 커지면 망의 변화에 빠르게, a가 작아지면 더디게 대응한다. 이렇게 측정된 $node_traversal_time$ 을 어떤 식으로 운용하는가에 따라 네트워크에 성능에 많은 영향을 미칠 수 있다. 측정된 값을 노드가 하나만 운용할 것인가, 아니면 네트워크가 운용되는 처음부터 계속적으로 운용할 것인가, 최근에 몇 개만 가지고 운용할 것인가에 문제는 네트워크에 성능향상에 큰 영향을 미칠 수 있다.

III. 모의실험 및 성능 평가

모의실험은 Scalable Network Technologies^[8]사의 QualNet 3.8을 이용하였으며, 구현된 AODV 라우팅 프로토콜은 draft-ietf-m anet-aodv-08 버전을 기반으로 하였다. 실험 환경으로 노드의 이동성에 따라 AODV 라우팅 프로토콜의 성능이 어떠한 변화를 보이는지 알아보기 위하여 pause time을 매 실험마다 변경하였다. 트래픽 특성은 다양한 트래픽에서 AODV 라우팅 프로토콜의 성능을 평가하기 위하여 CBR, VBR과 랜덤 트래픽을 사용하였다.

표 1과 같이 기본 망 크기 및 노드 수는 각각 $1500 \times 1500m^2$, 50, 전송속도는 최대 2MB/s, 전파 도달 거리는 250m, 노드 최대 이동속도는 10m/s로

표 1. 실험 시나리오

	시나리오1	시나리오2	시나리오3
노드 수	50	50	20
연결 수	30	30	10
네트워크 크기(m)	1500x1500	1500x1500	2000x2000
traffic type	CBR 10, VBR 10 TRAFFIC-GEN 10	CBR 30	CBR 10
pause time(s)	0~600 까지 50씩 증가	300	300
mean interval(s)	0.1, 0.5, 1	0.1~1 까지 0.1씩 증가	0.5

하였다. Pause time은 0에서 600초까지 변화시켜 노드 이동성에 따라 성능 변화를 관찰하였다.

실험 시나리오는 표 1에 나타난 바와 같이 노드의 이동성, 부하의 변동 및 노드의 밀집도에 따른 성능변화를 측정하기 위해 3가지 경우에 대해, 기존 방식과 제안 방식을 비교하였다. 제안 방식의 경우 식(2)의 a값은 모든 경우 0.125로 고정시켰다.

1) 노드 이동성에 대한 실험 결과

그림 1은 pause time 변화에 따른 데이터 전송 경로를 찾기 위해 발생하는 RREQ의 생성 횟수를 나타내는 것으로, 이동성이 클 경우 제안방식이 기존 AODV 방식보다 많은 RREQ가 생성되는데, 이

것은 이동성 증가에 따라 링크 변동이 커져 새로운 경로를 자주 찾기 때문이다. 여기서, RREQ을 보낸 후 경로를 찾지 못해 RREQ의 TTL을 변경하여 재전송하는 횟수는 포함하지 않는다. 그러나, pause time이 200s 이상이 되어 노드 이동속도가 일정 수준 이하로 떨어지면, 제안 방식과 기존 AODV와 거의 동등한 성능을 나타냄을 보이고 있다.

생성되는 RREQ 패킷 양은 제안방식이 기존방식보다 많았지만, 망 상에서 전달되는 RREQ 패킷의 양은 그림 2에서 기존의 AODV 방식이 훨씬 많음을 보이고 있다. 이것은 한번 제어 패킷을 보낸 후 오랜 시간 동안 기다리게 되면서 TTL 값을 변경하여 계속 RREQ를 발생하기 때문이다. 즉, 제안 방식은 처음 보내는 제어 패킷은 많게 되지만, 목적지를 찾기까지 발생하는 제어 패킷의 양은 적기 때문이다. 특히, 노드 이동이 활발한 경우 기존 방식보다 훨씬 적은 RREQ 패킷이 전송됨을 보이고 있다.

그림 3, 4 및 5는 단말 이동에 따른 각 트래픽의 throughput 성능을 비교한 결과이다. 제안방식의 경우, 경로를 찾고 관리하는 시간이 짧기 때문에 노드 이동이 활발한 경우 괄목할 성능 향상을 보이고 있다. VBR 트래픽인 그림 3 보다 CBR 트래픽인 그림 4가 이동성 변화에 더 무관한 모습을 보이고 있는데, 이는 CBR 트래픽일 경우 일정한 시간 간격으로 같은 양의 패킷이 발생함에 기인하는 것으로 파악된다.

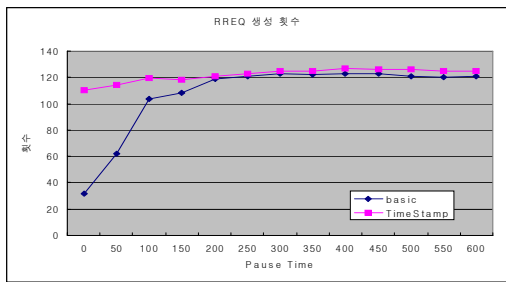


그림 1. RREQ 생성 횟수

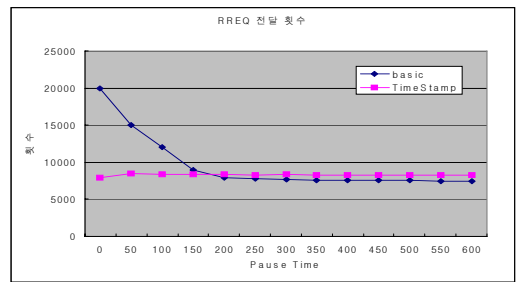


그림 2. RREQ 전달 횟수

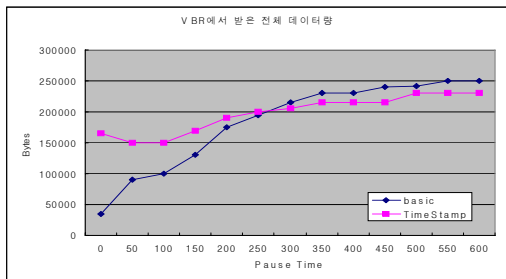


그림 3. VBR 트래픽 throughput

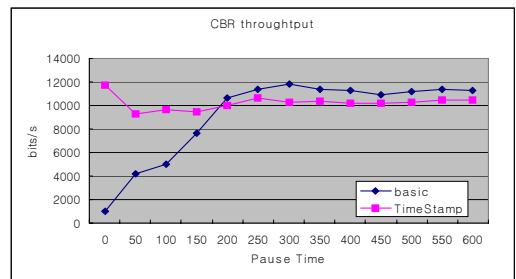


그림 4. CBR 트래픽 throughput

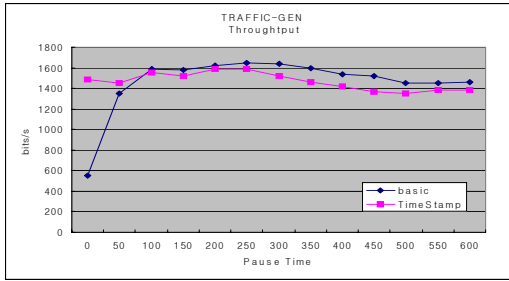


그림 5. TRAFFIC-GEN 트래픽 throughput

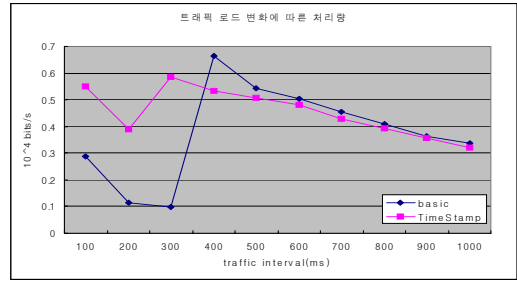


그림 6. 시나리오 2의 트래픽 부하 변화에 따른 처리량

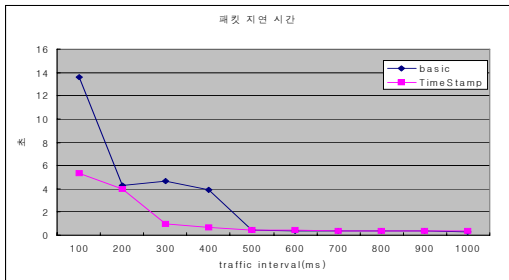


그림 7. 시나리오 2의 패킷 지연 시간

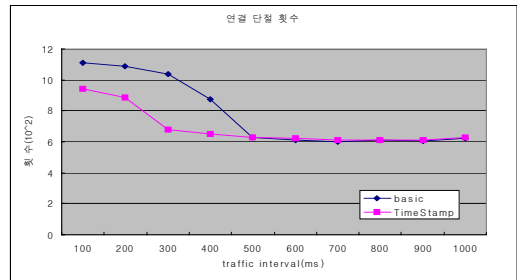


그림 8. 트래픽 부하에 따른 연결 단절 횟수

2) 네트워크 부하에 따른 실험 결과

그림 6은 트래픽 부하 변화에 따른 데이터 처리량을 나타낸 그림으로, 트래픽 부하가 큰 상황에서 제안 방식의 성능이 일반 기존 방식보다 크게는 6 배 정도 월등히 우수함을 보이고 있다. 트래픽 부하가 적은 경우 기존 방식보다 성능이 저하됨을 보이고 있으나, 그 차이는 미미함을 나타내고 있다.

그림 7는 트래픽 부하 변화에 따른 데이터 패킷의 지연시간을 나타낸다. 전체적으로 제안방식이 기존방식보다 지연시간이 적음을 알 수 있고, 특히 트래픽 부하가 큰 경우 우수한 특성을 보이고 있다.

그림 8는 트래픽 부하 변화에 따른 연결 단절이 얼마나 일어나는지를 보여주는 그림이다. 트래픽 부하가 클 경우 상대적으로 많은 연결 단절이 일어나는 것을 볼 수 있는데, 이는 많은 데이터가 발생하여 네트워크 부하가 걸리고 이에 따라 정상적으로 데이터를 전송하지 못하기 때문이다. 그러나, 이 경우에도 제안방식이 트래픽 부하가 큰 영역에서 링크 단절이 기존 방식보다 적게 나타난 것을 볼 수 있다.

3) 노드의 밀집성에 대한 실험 결과

노드의 밀집성에 대한 실험 결과는 표 2에 정리되어 있다. 이 표로부터 발생된 총 제어패킷의 수 (RREQ, RREP, RERR 패킷 수의 합)는 timestamp를

표 2. 시나리오3 결과

	AODV	Timestamp
RREQ 생성	1618	1316
RREP 생성	502	491
RERR 생성	574	530
합	2694	2337
데이터 생성	2346	2427
제어패킷과 데이터 비율	1.148338	0.962917

사용한 제안 방식이 기존 방식보다 15.2% 적고, 데이터 처리량은 오히려 3.45% 높음을 알 수 있다. 이로 부터 대형 망 환경에서도 timestamp 값을 이용하여 망의 환경에 따라 node_traversal_time 값을 동적으로 변화시키는 제안 방식이 고정 값을 사용하는 기존 방식보다 유리함을 알 수 있다. 또한 제어패킷과 데이터 패킷의 비율은 기존 방식의 경우 약 1.15, 제안 방식의 경우 약 0.96로서, 제안 방식을 사용할 경우 제어 패킷의 오버헤드가 20% 정도 상당히 감소함을 나타내고 있다.

IV. 결론

본 논문에서는 이동 ad-hoc 망의 AODV 라우팅 방식의 성능을 개선하기 위해, AODV 파라미터 중 node_traversal_time 값을 고정 값이 아닌 망의 상

황에 따라 동적으로 변경하는 방식을 제안하고, 노드의 이동성, 네트워크 부하량, 노드의 밀집도 변화에 따른 성능의 우수성을 컴퓨터 모의실험을 통하여 기존방식과 비교하였다. 제안 방식에서는 RREQ 메시지 내의 timestamp 값을 이용하여, moving average 알고리즘 기반으로 node_traversal_time 값을 망 환경에 따라 동적으로 계산 변화시킨다.

노드의 이동성이 큰 경우, 중복되는 제어패킷의 양과 전체 전달되어진 제어패킷의 양은 기존 방식에서 더 많았으며, 데이터 처리량도 제안방식이 높음을 확인하였다. 이동성이 적은 경우, 제어패킷의 양은 기존 방식에서 더 적게 나타났으나, 제안 방식과의 차이는 크지 않음을 보였다.

네트워크의 부하에 대한 성능 실험 결과는, 트래픽 부하가 클 경우 크게는 6배 정도의 제안 방식이 기존 방식보다 월등한 성능을 보였다. 반대로 트래픽 부하가 적을 경우, 제안 방식이 기존 방식보다 성능이 떨어지긴 했으나 그 차이는 크지 않음을 보였다. 지연 시간 면에 있어서도 트래픽 부하 변화에 따라 제안방식이 기존방식보다 트래픽 부하가 클 경우 특히 우수한 특성을 보임을 확인하였다. 서비스 반경을 넓혀 설정한 대규모 망 환경에서도, 제안 방식의 우수성에 대한 유사한 실험결과를 얻었다.

즉, 본 논문을 통해 노드간의 패킷 전달 시간을 40ms 로 고정시킨 현재의 AODV 방식보다, 망의 트래픽 상황을 반영하여 이 값을 동적으로 변경시킨 제안 방식이 우수한 특성을 나타냄을 여러 모의 실험을 통해 관찰하였고, 그 우수성은 특히 노드의 이동성 및 트래픽 부하가 큰 환경에서 더욱 두드러짐을 확인하였다.

참 고 문 헌

[1] 김중천, 김영용, "Ad Hoc 통신망 프로토콜 개발 동향," *Telecommunication Review*, 12(3), pp. 298-310, 2002.

[2] C. K. Toh, "Ad-hoc Mobile Wireless Networks : Protocols and Systems," Prentice Hall PTR, 2002.

[3] C. E. Perkins, "Performance Comparison of Two On-Demand Routing Protocols for Ad-hoc Networks," *Proc. of the IEEE Infocom*, pp.3-12, March 2000.

[4] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector(AODV) Routing," RFC 3561, July 2003.

[5] <http://www3.ietf.org/proceedings/02mar/I-D/draft-ietf-manet-aodv-10.txt>.

[6] S. M. Tripathi, and R. K. Ghosh, "An Implementational Study of Certain Heuristics for the Performance Enhancements of AODV," *Proc. of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pp.395-402, Jan., 2002.

[7] H. C. Kim, J. E. Kang, D. M. Kim, and J. Y. Lee "A Study of Optimization with Timestamp for AODV Protocol in Ad-hoc Network," *Proc. of the ISCIT 2001*, Vol.1, pp.280-285, Nov., 2001.

[8] <http://www.scalable-networks.com/>

이 재 석 (JaeSuk Lee)

학생회원



2004년 2월 연세대 전산학과 학사
 2006년 2월 연세대 전산학과 석사
 2006년 3월 (주)유비플러스 근무
 <관심분야> Ad hoc 네트워킹,
 홈 네트워킹, 임베디드시스템

박 혜 련 (Hae-Ryeon Park)

학생회원

1982년 2월 이화여대 수학과 학사
 1989년 5월 U. Massachusetts, Amherst 수학 및 통계학과 석사
 2006년 8월 명지대 컴퓨터공학과 박사
 1982년 3월 ~ 1984년 6월 한국
 전기통신공사 사업지원본부
 연구원
 <관심분야> 트래픽제어, QoS, 라우팅프로토콜, 이동통신

이 기 현 (Kee-Hyun Lee)

정회원

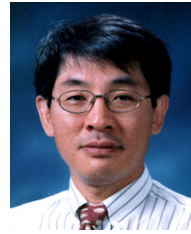


1965년 2월 성균관대 경제과 졸업
1972년 2월 성균관대 경영정보
학 석사
1986년 2월 광운대학교 전산학
과 석사
1993년 2월 광운대학교 전산학
과 석사

1982년 3월~현재 명지대학교 컴퓨터공학과 교수
<관심분야> 컴퓨터망, 데이터통신, 시스템분석·설계,
IT활용분야

석 정 봉 (Jung-Bong Suk)

정회원



1979년 2월 연세대 전자공학과
학사
1981년 2월 한국과학기술원 전
기 및 전자공학과 석사
1989년 8월 U. Massachusetts,
Amherst 전기 및 컴퓨터공학
과 박사

1981년 3월~1993년 1월 ETRI 책임연구원
1982년 3월~현재 연세대학교 컴퓨터정보통신공학과
교수
<관심분야> 무선 LAN 프로토콜, Ad hoc 네트워킹