

# AODV 라우팅 프로토콜에서 재전송률을 낮추기 위한 ERS 알고리즘의 노드순회시간 계산방법

선 창 윤<sup>†</sup> · 강 승 호<sup>\*\*</sup> · 임 형 석<sup>\*\*\*</sup>

## 요 약

에드 혹은 네트워크에 사용되는 라우팅 프로토콜인 AODV(Ad hoc On-demand Distance Vector)는 ERS(Expanding Ring Search) 알고리즘으로 경로설정 과정에서의 라우팅 패킷 재전송을 제어한다. 그러나 기존의 ERS는 라우팅 패킷 재전송의 기준이 되는 NTT(Node Traversal Time) 계산에 이동성이 높은 네트워크 상황을 적절하게 반영하지 못한다. 본 논문은 NTT 계산에 RREP(Route Reply)패킷을 사용하고 출발지 노드와의 인접도에 따라 각 노드의 NTT에 가중치를 달리 적용함으로써 라우팅 패킷의 재전송률을 낮추는 방법을 제안하고 ns2를 이용하여 기존의 ERS와 성능을 비교한다.

키워드 : 에드 혹은 네트워크, AODV(Ad-hoc On-demand Distance Vector), ERS(Expanding Ring Search) 알고리즘, 노드순회시간 (Node Traversal Time)

## Computing Methods of Node Traversal Time of ERS Algorithm to Reduce the Retransmission Rate in AODV Routing Protocol

Chang-Yoon Sun<sup>†</sup> · Seung-Ho Kang<sup>\*\*</sup> · Hyeong-Seok Lim<sup>\*\*\*</sup>

## ABSTRACT

An ad-hoc network routing protocol, AODV(Ad-hoc On-demand Distance Vector) initially uses ERS(Expanding Ring Search) algorithm to control the RREQ(Route Request) retransmission in the route discovery process. Because ERS uses the fixed NTT(Node Traversal Time), it does not reflect accurately ad-hoc network environment. In this paper, we propose an improved ERS algorithm which lowers the retransmission rate. It uses RREP(Route Reply) and gives different weights in NTT of each node in accordance with the adjacency to an originating node. To show the effectiveness of the proposed algorithm, we perform a simulation using ns2. The simulation results showed that the proposed algorithm has better performance than existing algorithms.

Key Words : Ad-hoc Network, AODV(Ad-hoc On-demand Distance Vector), ERS(Expanding Ring Search) Algorithm, NTT(Node Traversal Time)

## 1. 서 론

이동 컴퓨팅 기술의 비약적인 발전과 활발한 연구로 다양한 무선 네트워크 제품과 서비스가 제공 되고 있다. 무선 환경에서는 무선 통신 매체의 특성인 낮은 대역폭, 잦은 지연, 높은 에러율 등을 고려하여 사용자가 이동 중에도 네트워크 서비스를 끊김 없이 제공받을 수 있어야 한다.

무선 네트워크는 2가지로 분류 할 수 있다. 하나는 기지국이나 AP(Access Point)가 관리하는 영역 안에서 형성되

는 네트워크이고 다른 하나는 기존에 설치된 유선망의 도움 없이 이동 단말기기로 구성된 에드 혹은 네트워크이다.

에드 혹은 네트워크는 미리 설치된 장비 없이 임의로 형성된 네트워크로써 기존 유선망을 사용할 수 없는 경우 즉, 화재나 홍수 등의 긴급사태 발생시 긴급구조 등의 목적으로 사용할 수 있고, 전시장, 회의장, 일반적인 사무실 등 네트워크가 일시적으로 요구되는 장소에서도 사용할 수 있다. 이러한 네트워크를 구성하기 위해서는 이동 단말 끼리 데이터를 전송할 수 있어야 하고 이동 단말들은 라우터 기능을 수행해야 한다[1,2].

에드 혹은 라우팅 프로토콜은 테이블 구동(Table-driven)방식과 요구기반(On-demand)방식 그리고 앞의 두 가지 방식을 혼합한 하이브리드(Hybrid)라우팅 방식이 있다. 테이블

<sup>†</sup> 정 회 원 : 알앤비소프트웨어

<sup>\*\*</sup> 준 회 원 : 전남대학교 대학원 전산학과 박사과정

<sup>\*\*\*</sup> 정 회 원 : 전남대학교 전자컴퓨터공학부 교수  
논문접수 : 2005년 5월 3일, 심사완료 : 2006년 7월 5일

구동방식의 라우팅 프로토콜로는 DSDV(Destination Sequence Distance vector)[3], WRP(Wireless Routing Protocol)[4]등이 있고 요구기반 방식 라우팅 프로토콜로는 DSR(Dynamic Source Routing)[5], AODV(Ad hoc On-Demand Distance Vector Routing)[6,7], ABR(Associativity Based Routing)[8], TORA (Temporally Ordered Routing Algorithm)[9]등이 있다. 앞의 두 가지 기법을 혼합한 하이브리드 방식으로는 ZRP(Zone Routing Protocol)[10]등이 있고, 이외에 클러스터 기반 라우팅 프로토콜로써 IMEP(Internet MANET Encapsulation Protocol), CEDAR(Core-Extraction Distributed Ad-hoc Routing), CBRP(Cluster Based Routing Protocol) 등이 있다.

애드 혹 네트워크에서는 테이블 구동 방식과 요구기반 방식이 주로 사용된다. 테이블 구동 방식의 모든 노드는 주기적으로 라우팅 정보를 다른 노드들에게 전달하고 라우팅 경로 변경 시는 자신의 라우팅 정보를 방송하여 다른 노드들의 라우팅 테이블 갱신을 유도한다. 테이블 구동 방식의 라우팅 프로토콜은 다른 노드들에 대한 주기적인 라우팅 정보를 유지함으로써 전송 필요시 별도의 경로탐색절차 없이 라우팅 테이블의 정보를 이용하여 전송함으로써 경로탐색지연시간이 짧은 장점이 있다. 그러나 열악한 환경의 무선 대역에서 주기적인 라우팅 정보 방송은 사용할 수 있는 무선대역의 낭비를 가중시킨다. 또한 노드들이 빈번하게 이동하는 애드 혹 네트워크에서 다른 노드들의 라우팅 테이블 갱신을 위한 라우팅 정보 역시 방송해야 하므로 라우팅 패킷으로 인한 오버헤드가 발생하는 단점이 있다.

요구기반 방식의 라우팅 프로토콜은 테이블 구동 방식의 단점을 해결하기 위하여 제안된 프로토콜로서 네트워크 내의 모든 이동 노드에 대한 전체 경로를 상시 유지하는 것이 아니라 데이터 전송 필요시에만 경로탐색절차를 수행한다. 요구기반 방식은 필요시에만 경로탐색절차를 수행하고 주기적인 라우팅 정보방송과 이동시 변경된 라우팅 정보를 방송할 필요가 없으므로 라우팅 패킷 오버헤드를 줄이는 장점이 있다. 이러한 이유로 노드들이 빈번하게 이동하는 애드 혹 네트워크에서 가장 적합한 라우팅 프로토콜로 평가받고 있다[11]. 그러나 데이터 전송 시 경로탐색절차 수행 후 획득된 경로로 데이터를 전송하기 때문에 경로탐색시간이 길어지는 문제점을 가지고 있다.

AODV는 대표적인 요구기반 라우팅 프로토콜로서 경로탐색 방법은 주변노드들에게 라우팅 제어 메시지를 플러딩해서 해당 목적지노드를 찾는다. 그런데 목적지노드가 멀리 있거나 없는 경우엔 목적지 노드를 찾기 위한 라우팅 패킷이 급격하게 증가하고 이로 인해 네트워크의 성능이 크게 저하된다. 이러한 문제점을 해결하기 위해 불필요한 RREQ(Route Request)를 제어하는 ERS 알고리즘을 사용한다. 하지만 유선 멀티 캐스팅에서 유래한 ERS 알고리즘은 애드 혹 네트워크 상황에서 수시로 변하는 NTT(Node Traversal Time)를 적절히 고려하지 못하는 문제가 있다.

본 논문에서는 노드들이 불규칙적으로 이동하는 무선 네

트워크에서 목적지노드로부터 전송되는 RREP(Route Reply)의 소요시간을 중간노드들과 소스노드까지 각각 측정하고 소스노드와의 인접도에 따라 각 노드의 NTT에 가중치를 달리 부여하여 이를 타임아웃에 반영하는 알고리즘을 제안한다. 이렇게 함으로써 라우팅 제어 패킷을 감소시켜 전체 네트워크 성능을 높인다.

본 논문의 구성은 다음과 같다. 2장에는 관련연구로써 AODV 라우팅 프로토콜과 이와 관련된 ERS 알고리즘들을 검토하고, 3장에서는 이를 개선한 새로운 알고리즘을 제안한다. 4장에서는 ns2 시뮬레이터를 이용하여 본 논문에서 제안하는 알고리즘과 기존의 알고리즘을 비교 분석하며, 5장에서 결론 및 향후 연구 등을 기술한다.

## 2. 관련연구

본 장에서는 우선 AODV 라우팅 프로토콜에 대해 알아보고 유선네트워크에서 사용되는 ERS알고리즘과 AODV 라우팅 프로토콜에서 동작하는 ERS 알고리즘, KERS 알고리즘을 설명한다.

### 2.1 AODV 라우팅 프로토콜

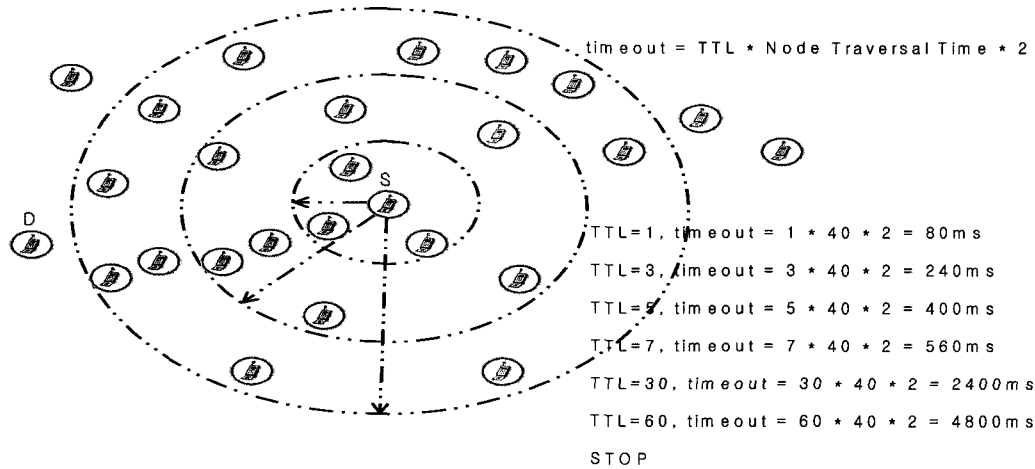
AODV는 1999년 C. Perkins가 제안한 것으로 애드 혹 네트워크에서 대표적인 요구 기반 라우팅 프로토콜이다. 애드 혹 네트워크 내의 모든 노드들은 데이터 전송이 있는 라우팅 경로정보만을 라우팅 테이블에 유지, 관리하게 된다. 데이터 전송이 필요한 소스노드는 요구기반 방식으로 목적지노드까지의 최단경로를 라우팅 경로 설정 방법을 통하여 찾아낸다.

AODV가 경로 설정을 위해 사용하는 메시지는 RREQ(Route Request)와 RREP(Route Reply)가 있고 경로 설정과정은 다음과 같다.

우선 소스노드는 목적지노드로의 경로를 생성하기 위해 RREQ를 네트워크 내에 플러딩하게 된다. RREQ를 수신한 노드가 목적지노드도 아니고 목적지노드까지의 경로도 모르면 상위노드를 라우팅테이블에 저장한 후 수신한 RREQ를 이웃노드들에게 전송하게 된다. RREQ를 수신한 노드가 자신이 목적지노드이거나 목적지노드까지의 라우팅 경로를 가지고 있는 중간노드라면 RREP를 이용하여 응답하게 된다. RREP는 RREQ 전송시 형성된 단일 경로를 따라 소스노드에 전송된다. 만일 일정시간이 경과 했는데도 RREP가 전송되어 오지 않으면 소스노드는 ERS 알고리즘에 의해 RREQ를 재전송하게 된다.

### 2.2 ERS(Expanding Ring Search) 알고리즘

ERS 알고리즘은 처음 유선 멀티 캐스팅에서 사용되었다. ERS의 중요한 목적은 RREQ가 처음부터 넓은 네트워크에 플러딩 되는 것을 방지하고, 점진적으로 범위를 넓게 확장하여 목적지 노드를 찾는데 발생하는 불필요한 RREQ(Route Request)를 제한하고자 함이다.



(그림 1) ERS 알고리즘

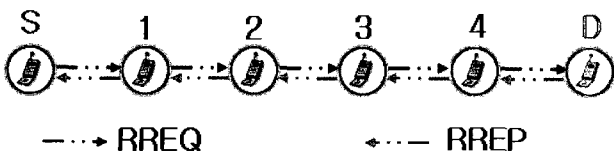
(그림 1)에 도시한 바와 같이 ERS는 처음 TTL(Time To Live)를 1로 하여 인접노드에게 RREQ를 플래딩한다.  $TTL \times Node\_Traversal\_Time \times 2$ 로 계산되는 타임아웃시간 이내에 RREP(Route Reply)가 오지 않으면 TTL을 2만큼 증가시켜 다시 플래딩을 하고 역시 응답이 없으면 TTL을 2의 배수로 8까지 증가시켜 가며 재전송을 시도한다. 그래도 응답이 없으면 TTL을 30과 60으로 전송을 시도하고 역시 응답이 없으면 목적지 노드가 존재 하지 않는 것으로 판단하고 재전송을 포기한다. 여기서 NTT(Node Traversal Time)는 한 홉 간의 실제 패킷 전송 소요시간을 의미 하는데 고정 값인 40ms가 사용된다[13].

2.3 AODV 라우팅 프로토콜의 ERS 알고리즘

(그림 2)는 경로설정과정에서의 라우팅 패킷들을 나타낸다. 우선 소스노드(S)는 RREQ에 소스노드의 현재 시간을 담아 플래딩한다. 목적지노드(D)나 목적지까지의 경로를 알고 있는 중간 노드는 RREP에 RREQ의 출발시간을 복사하여 단일 경로를 따라 소스노드에 전송한다. RREP를 수신한 소스노드는 식(1)과 같이 소스노드의 현재시간과 RREP에서 얻은 RREQ를 전송할 당시의 시간차를 구하고 이를 홉 수로 나눈 평균값을 ERS 알고리즘의 NTT로 사용한다.

(1)에서 *Current\_time*은 RREP를 수신한 출발지 노드의 현재 시간을 의미하고 *h*는 수신한 RREP에 담긴 RREQ 전

$$NTT = \frac{Current\_time - h}{HopCount * 2} \quad (1)$$



(그림 2) 경로설정과정에서의 라우팅 패킷들

송 당시의 시간을 의미한다[7].

2.4 KERS(Kim's Expanding Ring Search) 알고리즘

KERS[12]의 기본 아이디어는 NTT로 고정 값을 사용하지 않고 실제 소요시간을 측정하여 사용하는데 있다. 우선 소스노드는 RREQ에 현재시간을 담아 플래딩한다. RREQ를 수신한 각 노드는 식(2)와 같이 수신시간과 RREQ에 저장된 시간의 차를 구하고 이 차를 소스노드로부터 현재노드까지의 홉 수로 나눈 실제 소요시간의 평균값을 NTT로 적용한다. 그리고 RREP를 전송하는 노드에서 계산된 NTT가 소스노드의 NTT가 된다.

$$NTT = \frac{RCT - RTS}{RH} + a \quad (2)$$

식(2)에서 *RCT*(RREQ Current Time)는 수신한 노드의 현재시간을 *RTS*(RREQ Timestamp)는 RREQ에 담아 보내진 시간을 그리고 *RH*는 현재노드까지의 홉 수를 각각 나타낸다. *a*는 NTT를 효과적으로 운용하기 위해 측정된 NTT에 약 10%정도의 여유시간으로 RFC 793[14]에서 TCP retransmission timer를 사용한다[12].

3. 제안하는 방법

애드 혹 네트워크에서는 단말기나 매체의 특성, 이동성 등의 이유로 인하여 노드 간 NTT의 가변성이 매우 크다. 그런데 이와 같은 가변성을 고려하지 못하고 고정된 NTT를 사용하면 경로 설정 시 불필요한 RREQ의 재전송으로 네트워크에 큰 오버헤드를 초래한다. 특히 네트워크 규모가 커지고 노드들의 이동 속도가 높으면 높을수록 그 영향은 더욱 크다. AODV[7]나 KERS[12]도 유선네트워크에서 사용되는 ERS가 애드 혹 네트워크에서 흔히 발생하는 NTT의

가변성 문제를 적절하게 고려하지 못한다는 사실을 인식하고 고정된 NTT 대신 실제 네트워크에서의 패킷 전송에 소요 되는 시간을 고려한 NTT를 사용한다.

그러나 NTT의 가변성과 관련하여 RREQ보다는 재전송과 시간적으로 보다 가까운 RREP의 전송시간을 이용하는 것이 적절하다. 왜냐하면 RREQ가 목적지 노드로 전송 중이거나 도착한 이후에도 노드들이 이동하여 NTT에 변화를 가져올 가능성이 크기 때문이다.

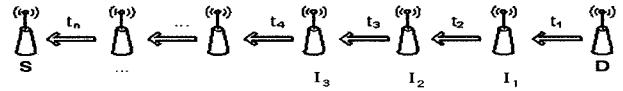
또한 NTT의 계산에 있어서 소스노드와 목적지노드 사이에 소요된 시간을 홉 수로 단순 평균하는 방법도 전송시의 현재성을 확보하는 것이 NTT 계산에 고려해야 할 중요한 요소라는 점에 비추어 볼 때 적절하지 못하다. NTT를 계산하는 목적은 소스노드가 목적지노드에 새로운 연결을 시도할 때 사용하려는 것이다. 따라서 전송시간의 현재성이란 소스 노드의 관점에서 고려되어야 한다. 소스노드의 관점에서 패킷 전송에 소요되는 시간으로 가장 현재적이며 안정적인 것은 최 인접 노드와의 전송시간이고 그 다음은 소스노드로부터 2홉 거리에 있는 노드와의 전송시간이다. 이처럼 소스노드와의 인접도를 고려하여 NTT를 계산한다면 실제 네트워크의 현재 상황을 적극 반영하게 되고 결국 예측하지 못한 지연시간으로 인해 발생하는 불필요한 RREQ의 재전송을 줄일 수 있다.

본 장에서는 RREQ대신 RREP를 사용하고 각 노드로 이동할 때 마다 시간을 측정해서 출발지 노드와의 인접성에 따라 가중치를 달리 부여하는 방법들을 제시한다.

3.1 지수적 가중치 배분을 통한 NTT 계산방법

본 논문에서 제안하는 방법들은 AODV나 KERS와는 달리 (그림 4)에서 보듯이 목적지노드(D)에서 RREP를 출발지노드(S)로 전송할 때 홉 간의 시간차를 이용해 NTT를 계산한다. 우선 D는 RREP에 현재시각을 기록하여 다음 노드에 전송한다. 다음 노드(I<sub>1</sub>)는 RREP를 수신한 시각과 RREP에 기록된 시각과의 차(t<sub>1</sub>)를 계산하고 현재 시각과 함께 RREP에 기록하여 다음 노드로 전송한다. 세 번째 노드(I<sub>2</sub>)는 수신한 RREP에 기록된 이전 노드간의 전송시간(t<sub>1</sub>)과 자신과 바로 이전노드 간의 측정된 전송시간(t<sub>2</sub>)과의 평균을 구하여 현재 시각과 함께 다음 노드에 전송한다. 네 번째 노드부터는 RREP에 기록된 이전 노드들의 평균 전송시간과 RREP를 전송한 직전 노드와의 전송시간을 평균하여 다음 노드에 전송한다. 출발지노드(S)는 이렇게 구해진 NTT를 동일 목적지노드에 대하여 순차적으로 3개까지 연속해 저장해가며 그 평균값을 ERS의 NTT로 사용한다.

출발지노드로부터 목적지노드까지 7개의 노드가 있는 경우에 대하여 각 노드에서 계산되는 NTT를 예시하면 <표 1>과 같고 이를 수식(3)으로 일반화 할 수 있다. 여기서 t<sub>i</sub>는 i-1번째 노드와 i번째 노드간의 전송시간을 나타낸다. RREP를 전송하는 각 노드들이 NTT를 이와 같이 계산하면 출발지노드는 자신과의 인접도에 따라 지수적으로 가중치를 부여하게 된다.



(그림 3) RREP를 이용한 방법

<표 1> 알고리즘 1에 의한 7개 노드로 구성된 경로의 NTT 계산

| 노드        | NTT   |
|-----------|---|
| 목적지노드 (D) | —   |
| I1        | t <sub>1</sub>  |
| I2        | $\frac{t_1 + t_2}{2}$   |
| I3        | $\frac{\frac{t_1 + t_2}{2} + t_3}{2} = \frac{t_1 + t_2 + 2t_3}{4}$  |
| I4        | $\frac{\frac{t_1 + t_2 + 2t_3}{4} + t_4}{2} = \frac{t_1 + t_2 + 2t_3 + 4t_4}{8}$                                |
| I5        | $\frac{\frac{t_1 + t_2 + 2t_3 + 4t_4}{8} + t_5}{2} = \frac{t_1 + t_2 + 2t_3 + 4t_4 + 8t_5}{16}$                 |
| 출발지노드 (S) | $\frac{\frac{t_1 + t_2 + 2t_3 + 4t_4 + 8t_5}{16} + t_6}{2} = \frac{t_1 + t_2 + 2t_3 + 4t_4 + 8t_5 + 16t_6}{32}$ |

$$NTT = \frac{t_1 + \sum_{i=1}^{n-2} 2^{i-1} t_{i+1}}{2^{n-2}} \tag{3}$$

3.2 선형적 가중치 배분을 통한 NTT 계산방법

선형적 가중치 배분을 통한 계산방법은 지수적 가중치 배분을 통한 계산방법과 마찬가지로 RREP를 사용하나 가중치가 출발지 노드와의 인접도에 따라 선형적으로 증가하는 형태를 취한다. 목적지노드를 제외한 출발지노드까지의 모든 노드는 목적지노드로부터의 홉 수를 직전 노드와의 전송시간에 곱하고 이와 같이 계산된 이전 노드까지의 전체 전송시간을 더해 이를 다시  $\sum_{i=1}^{hp} i$  (hp: 홉 수)로 나눈 값을 자신의 목적지노드까지의 NTT로 사용한다. 그리고  $\sum_{i=1}^{hp} i$ 로 나누기 전의 전체 전송시간과 홉 수, 현재 시각을 RREP에 저장하여 다음 노드로 전송한다. 예를 들어 노드 I<sub>3</sub>은 노드 I<sub>2</sub>로부터 t<sub>1</sub>+2t<sub>2</sub>와 홉 수 2, RREP 전송시각을 전송받는다. 이로부터 t<sub>3</sub>을 구하고 여기에 홉 수 3을 곱한 다음 t<sub>1</sub>+2t<sub>2</sub>에 더하고 이를  $\sum_{i=1}^3 i$ 로 나눈 값을 자신의 NTT로 사용한다. 마찬가지로 t<sub>1</sub>+2t<sub>2</sub>+3t<sub>3</sub>과 홉 수 3, RREP 전송시간을 I<sub>4</sub>에게 보낸다. 7개 노드에 대하여 예시하면 <표 2>와 같고 수식(4)로 일반화 할 수 있다.

$$NTT = \frac{2 \sum_{i=1}^n i t_i}{n(n+1)} \tag{4}$$

〈표 2〉 알고리즘 2에 의한 7개 노드로 구성된 경로의 NTT 계산

| 노드    | NTT   |
|-------|---|
| 목적지노드 | —   |
| I1    | $t_1$   |
| I2    | $\frac{t_1 + 2t_2}{3}$                              |
| I3    | $\frac{t_1 + 2t_2 + 3t_3}{6}$                       |
| I4    | $\frac{t_1 + 2t_2 + 3t_3 + 4t_4}{10}$               |
| I5    | $\frac{t_1 + 2t_2 + 3t_3 + 4t_4 + 5t_5}{15}$        |
| 출발지노드 | $\frac{t_1 + 2t_2 + 3t_3 + 4t_4 + 5t_5 + 6t_6}{21}$ |

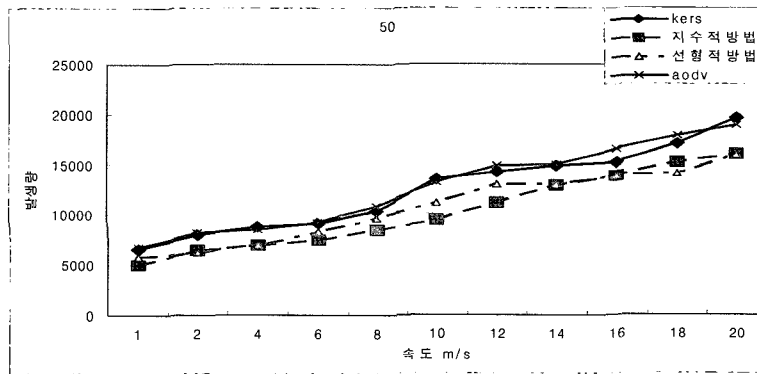
선형적 가중치 배분을 통한 계산방법은 지수적 가중치 배분을 통한 계산방법에 비해 출발지노드와의 인접도에 따른 가중치를 보다 분산해서 적용하므로 지수적 가중치 배분을 통한 계산방법이 갖는 전송시간 반영의 지나친 편중성을 방지할 수 있다.

#### 4. 실험 및 분석

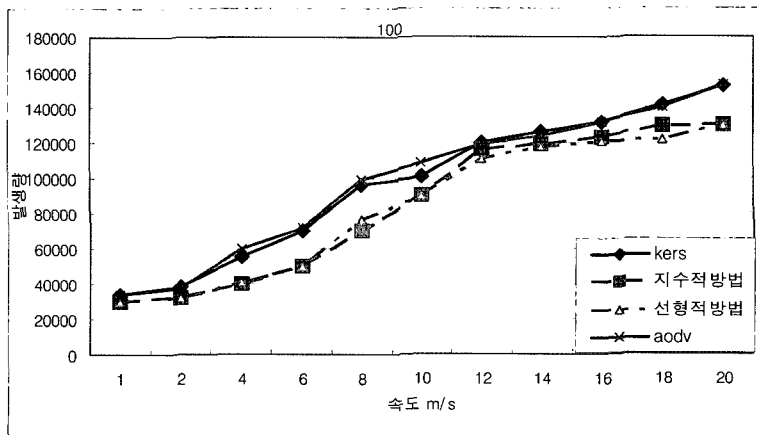
본 장에서는 제안된 알고리즘의 성능 평가를 위해 설정한 실험모델을 설명하고 실험을 통하여 측정된 결과를 분석한다.

##### 4.1 실험 모델

본 논문에서는 AODV의 ERS와 KERS, 제안한 알고리즘을 각각 실험하여 전체 패킷 발생량, 에러 발생량 등을 비교 분석하였다. 실험은 버클리(Berkeley) 대학의 Network simulator 2.26[16]를 사용하였다. 실험환경은 무선 애드 혹 망을 기본으로 하고 최대 20m/s의 속도의 이동성을 가지는 50개의 노드들이 최대 3000m x 3000m의 지역 내에서 이동하는 경우와 100개의 노드들이 최대 3000m x 3000m의 지역 내에서 이동하는 경우로 하였다. 각 노드의 속도는 1m/s에서 20m/s까지 변화를 주었고 소스노드와 목적지노드는 임의로 선택되어 100개의 연결이 만들어진다. 각 노드는 임의적으로 움직이지만 중간노드 역할도 하게 된다. 일단 소스노드와 목적지노드가 연결되어지면 ftp로 1Mbytes를 전송하고 연결을 해제한다. 실험에 사용된 MAC layer는 ns-2.26에 포함되어 있는 IEEE 802.11을 사용하였다.



(그림 4) 노드가 50개일 때의 전체 패킷 발생량



(그림 5) 노드가 100개일 때의 전체 패킷 발생량

<표 3> 실험 환경 변수

| 환경변수     | 변수 값          |               |
|----------|---------------|---------------|
|          | 노드 수          | 50개           |
| 크기       | 3000m x 3000m | 3000m x 3000m |
| 시뮬레이션 시간 | 200초          | 200초          |
| 연결 수     | 100개          | 100개          |
| 데이터 량    | 1Mbytes       | 1Mbytes       |

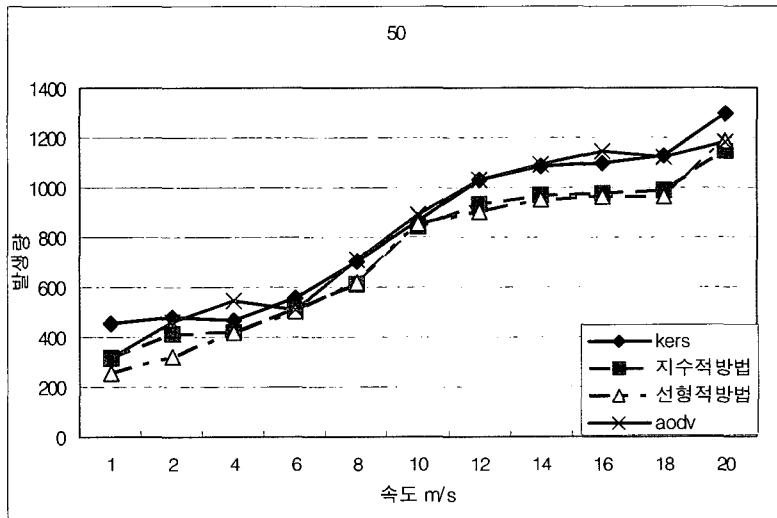
<표 3>과 같이 실험 환경 변수를 바탕으로 노드의 이동 속도를 변화 시키며 실험하였다.

4.2 실험결과 및 분석

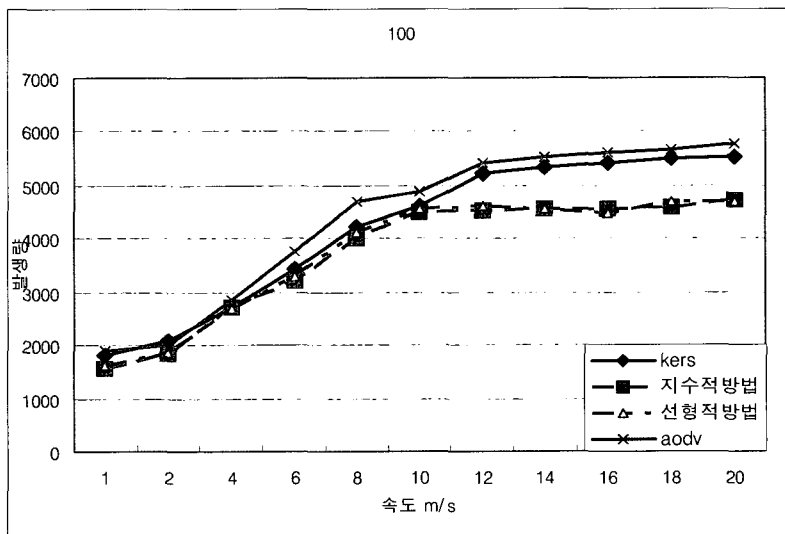
본 실험에서는 노드 50개와 100개를 가지고 AODV의

ERS와 KERS 그리고 제안한 알고리즘들을 속도를 증가시켜 가며 전체 패킷 발생량과 에러 발생량을 측정하였다. 실험 결과는 속도를 1m/s에서 20m/s까지 증가 시키면서 각 속도에 대해 20번씩 실행 시켜 평균치를 구한 것이다.

(그림 5, 6)은 전체 패킷 발생량을 (그림 7, 8)은 에러 발생량을 나타낸 것이다. (그림 5, 7)에서 알 수 있듯이 노드가 50개일 때의 전체 패킷 발생량이나 에러 발생량을 보면 제안한 알고리즘들이 AODV의 ERS나 KERS 보다 약 5% 정도의 개선이 있는 것으로 나타났다. (그림 6, 8)은 노드가 100개일 때의 패킷 발생량과 에러 발생량을 나타낸 그림인데 노드 50개일 때와 비슷한 수준의 성능 향상을 보였다. 그러나 지수적 가중치에 의한 방법과 선형적 가중치에 의한 방법간에는 유의할 만한 성능상의 차이를 보이지는 않았다. 이는 노드들의 패킷 전송거리가 250m인 점과 이동 속도에



(그림 6) 노드가 50개일 때의 에러 발생량



(그림 7) 노드가 100개일 때의 에러 발생량

비해 노드들의 이동지역이 상대적으로 작기 때문인 것으로 보인다.

### 5. 결론 및 향후 연구

본 논문에서는 AODV 라우팅 프로토콜의 RREP를 이용해 목적지노드에서 소스노드까지의 시간차를 구하고 이에 소스노드와의 인접도를 고려해 가중치를 달리 부여하여 타임아웃에 반영하는 개선된 ERS알고리즘을 제시하였다. 네트워크의 현재 상황을 적극 반영하여 경로설정 과정상의 RREQ의 재전송률을 낮추어 네트워크 대역폭의 낭비를 줄일 수 있다. 특히 AODV 프로토콜은 같은 요구기반 방식의 프로토콜인 DSR에 비해 이동성이 보다 높은 환경에서 중단 간 지연이나 처리량에서 성능이 나옴에도 불구하고 라우팅 패킷의 발생량은 많은 걸로 알려져 있다[15]. 따라서 라우팅 패킷인 RREQ의 재전송률을 낮추는 방법은 AODV 라우팅 프로토콜에게는 중요한 의미를 갖는다. 실험은 AODV의 ERS와 KERS 그리고 제안한 방법들을 전체 패킷 발생량과 에러 발생량에 대해 실행하였다. 실험 결과는 AODV의 ERS나 KERS 보다 제안한 알고리즘들이 패킷 발생량이나 에러 발생량 측면에서 성능향상을 보였다.

향후 연구로는 지수적 또는 선형적 가중치 부여 방법 이외에 피보나치 순열 등 여러 가지 가중치 부여 방법을 고안하여 각각의 방법들이 어떤 네트워크 환경에 보다 나은 성능을 보이는 지를 측정하는 것이다. 또한 이러한 가중치 부여 방법과 RREP의 선택이 각각 어느 정도 성능향상에 기여하는지도 분석이 필요할 것으로 생각된다.

### 참 고 문 헌

[1] E. M. Royer and C. K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," IEEE Personal Communications, pp.46-55, Apr., 1998.  
 [2] C. K. Toh, "Ad Hoc Mobile Wireless Networks Protocols and System," Prentice Hall PTR, pp.13-25, 2002.  
 [3] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers," Computer Communication, pp.234-244, Oct., 1994.  
 [4] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," ACM Mobile Networks and Applications Journal, Special issue on Routing in Mobile Communication Networks, pp.183-197, Oct. 1996.  
 [5] D. Johnson and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF MANET Working Group, Internet Draft, [\[dmaltz/internet-drafts/draft-ietf-manet-dsr-09.txt\]\(http://www-2.cs.cmu.edu/~dmaltz/internet-drafts/draft-ietf-manet-dsr-09.txt\), 15. Apr., 2003.  
 \[6\] C. E. Perkins and E. M. Royer, "Ad-hoc On-demand Distance Vector Routing," Mobile Computing Systems and Applications, Proceedings WMCSA '99. Second IEEE Workshop on, pp.90-100, Feb., 1999.  
 \[7\] RFC 3561 - Ad hoc On-demand Distance Vector\(AODV\) Routing, <http://www.faqs.org/rfcs/rfc3561.html>  
 \[8\] C. K. Toh, "Associativity-Based Routing For Ad-Hoc Mobile Networks," University of Cambridge Computer Laboratory Cambridge CB2 3QG, United Kingdom, Prentice Hall PTR, pp.79-95, Nov., 1999.  
 \[9\] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," IEEE INFOCOM '97 Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE, Vol.3, pp.1405-1413, Apr., 1997.  
 \[10\] Zone Routing Protocol \(ZRP\), <http://www.tct.hut.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf>  
 \[11\] 김종천, 김영용, "Ad Hoc 통신망 프로토콜 개발동향", Telecommunication Review 제12권3호, pp.298-311, 2002.  
 \[12\] 김현창, "Ad Hoc 네트워크에서 Timestamp를 이용한 AODV 프로토콜의 최적화", 연세대학교 대학원 전기전자공학과, \[http://nasla.yonsei.ac.kr/publications/hpmn\\\_papapers/ms/2002/AdHoc 네트워크에서 Timestamp를 이용한 AODV프로토콜의최적화.pdf\]\(http://nasla.yonsei.ac.kr/publications/hpmn\_papapers/ms/2002/AdHoc%20네트워크에서%20Timestamp를%20이용한%20AODV프로토콜의최적화.pdf\), 2002.  
 \[13\] K. Carlberg and J. Crowcroft, "Building Shared Trees Using a One-to-Many Joining Mechanism," ACM Computer Communication Review, Vol.27, No.1, pp.5-11, 1997.  
 \[14\] RFC 793-TCP\(Transmission Control Protocol\), <http://www.faqs.org/rfcs/rfc793.html>  
 \[15\] S. Das, C. E. Perkins, and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," In Proc. of INFOCOM-2000, pp.3-12, Mar., 2000.  
 \[16\] The networks simulator - ns2, <http://www.isi.edu/nsnam/ns>](http://www-2.cs.cmu.edu/~</a></p>
</div>
<div data-bbox=)



### 선 창 윤

e-mail : cyssun@gmail.com

2002년 광주대학교 중국어학(문학사)

2005년 전남대학교 대학원 전산학과(이학 석사)

2005년~현재 알앤비소프트웨어

관심분야: 알고리즘, 애드 혹 네트워크, 소프트웨어공학



**강 승 호**

e-mail : kinston@natural.chonnam.ac.kr  
1994년 전남대학교 전산학과(이학사)  
2003년 전남대학교 대학원 전산학과(이학석사)  
2003년~현재 전남대학교 대학원 전산학과 박사과정

관심분야: 알고리즘, 애드 혹 네트워크, 생물정보학



**임 형 석**

e-mail : hslim@chonnam.ac.kr  
1983년 서울대학교 컴퓨터공학과(공학사)  
1985년 한국과학기술원 전산학과(이학석사)  
1993년 한국과학기술원 전산학과(공학박사)  
1996년~1997년 미국 Purdue 대학 방문 교수

1987년~현재 전남대학교 전자컴퓨터공학부 교수  
관심분야: 알고리즘, 병렬 및 분산처리, 생물정보학