

노드 이동성을 고려한 적응형 AODV 알고리즘

(An Adaptive AODV Algorithm for Considering Node Mobility)

홍 윤 식 [†] 홍 준 식 ^{**} 임 화 석 ^{***}
 (Youn-Sik Hong) (Jun-Sik Hong) (Hwa Seok Lim)

요 약 Ad-hoc 네트워크에서 경로 설정을 위한 라우팅 프로토콜 on-demand 방식의 AODV가 가장 대표적이다. 그러나 AODV의 경우 노드 이동성이 커질수록 RREQ, RREP 등과 같은 제어 패킷이 급격히 증가한다. 이러한 네트워크 변동에 따른 제어 메시지 발생 증가는 데이터 패킷 수신율을 저하시킬 뿐만 아니라, 노드의 에너지 소모율도 증가시킨다. 본 논문에서는 네트워크 토폴로지 변동 상황을 감지하여 AODV에서의 RREQ 메시지 발생 빈도를 효과적으로 조절하기 위한 AIAD 방식의 적응형 AODV 알고리즘을 제안하였다. 제안된 프로토콜은 기존 AODV 프로토콜 및 timestamp 방식을 사용한 방식들과 노드 이동 시간, 노드 이동 속도 및 노드 밀집도 등 3가지 평가모델에 대해 각각 그 성능을 비교하였다.

키워드 : 애드-혹 네트워크, AODV, 노드 이동성, Expanding Ring Search(ERS), Node Traversal Time(NTT)

Abstract AODV routing protocol is intended for use by mobile nodes in an ad-hoc network. In AODV nodes create routes on an on-demand basis. As the degree of node mobility becomes high, however, the number of the control packets, RREQ and RREP messages, have increased so rapidly. The unexpected increases in the number of the control packets cause the destination node to decrease the packet receiving rate and also to increase the overall energy consumption of such a network. In this paper, we propose a novel method of adaptively controlling the occurrences of such RREQ messages based on AIAD (additive increase additive decrease) under a consideration of the current network status. We have tested our proposed method with the conventional AODV and the method using timestamp based on the three performance metrics, i.e., how long does node moves, node velocity, and node density, to compare their performance.

Key words : Ad-hoc Network, AODV, Expanding Ring Search (ERS), Node Mobility, Node Traversal Time (NTT)

1. 서 론

Ad-hoc 네트워크란 중계기 역할을 하는 AP(Access

Point) 없이 이동단말(mobile terminal, 노드)들만으로 구성되어, 이들 단말 간에 무선으로 데이터 송수신이 이루어지는 infrastructure less 네트워크를 말한다[1]. Infrastructure-less 네트워크에서 노드(node)는 이동단말의 역할뿐만 아니라 데이터 전송을 중계하는 라우터(router)의 역할도 수행한다. 한편, ad hoc 네트워크에서는 노드들이 자유롭게 불특정 방향으로 움직일 수 있기 때문에 네트워크 토폴로지(topology)가 수시로 변화한다. 따라서 ad-hoc 네트워크에서는 이러한 동적 토폴로지 변화에 적응 가능한 라우팅 프로토콜이 필요하다.

Ad hoc 네트워크의 라우팅 프로토콜은 테이블 구동(table driven), on-demand 및 이 두 가지 방식을 혼합한 방식 등 크게 3가지로 나눌 수 있다[2]. 테이블 구동 라우팅 프로토콜[2]의 경우 각 노드는 자신을 중심으로

[†] 종신회원 : 인천대학교 컴퓨터공학과 교수
 yshong@incheon.ac.kr
^{**} 정 회원 : 청주대학교 전자정보공학부 교수
 jnskhong@cju.ac.kr
^{***} 학생회원 : 인천대학교 컴퓨터공학과
 genocide00@nate.com
 논문접수 : 2008년 1월 2일
 심사완료 : 2008년 7월 30일

Copyright © 2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
 정보과학회논문지: 정보통신 제35권 제6호(2008.12)

도달 가능한 모든 노드들의 경로 정보를 경로 테이블에 유지하고 있다. 이 방식의 장점은 데이터 패킷 전송 시 별도의 경로 획득 절차 없이 경로 테이블에 저장된 경로 정보만을 이용하여 데이터를 즉시 전송할 수 있다는 점이다. 그러나 모든 노드의 경로 테이블을 최신 상태로 유지하기 위해서는 주기적으로 경로 정보를 방송(broadcast)해야 하므로 무선 대역폭 낭비가 심하다. 특히, 노드들이 빈번하게 움직이는 상황 하에서는 경로 방송에 따른 오버헤드(overhead)로 인해 성능이 급격히 저하된다. 경로 방송 오버헤드는 노드 수의 제곱에 비례해 증가하기 때문에, 테이블 구동 방식은 주로 정적 토폴로지를 갖는 소규모 ad-hoc 네트워크의 라우팅 프로토콜로 적합하다.

On-demand 라우팅 프로토콜[3]은 ad-hoc 네트워크의 모든 노드에 대해 경로 정보를 항상 유지하는 것이 아니라 데이터 전송이 필요할 경우에만 별도의 경로 획득 절차를 수행하는 방식이다. 즉 노드 이동성이 빈번한 동적 토폴로지에 적합하도록 제안된 프로토콜으로써, 경로 정보 방송에 따른 오버헤드 및 이에 따른 무선 대역폭 낭비 등의 문제를 효과적으로 개선할 수 있다. 그러나 on-demand 방식은 데이터 패킷 요청이 발생하면 비로소 경로 획득 과정이 시작되기 때문에 경로 획득시간 지연으로 인해 실시간 데이터 송수신에는 부적합하다. 가장 널리 알려진 on-demand 방식 라우팅 프로토콜은 AODV(Ad hoc On demand Distance Vector)이다[4].

AODV에서는 경로 획득 단계에서 경로 획득을 위한 제어 패킷 수가 급격히 증가하는 것을 막기 위해 이웃 노드를 중심으로 패킷 전송 범위를 점진적으로 확대해 가는 핸드셰이크(handshake) 방식을 채택하고 있다. 그림 1에 보인 것처럼 AODV에서는 데이터 패킷 송수신은 노드 이동 속도와 거의 무관한 반면 노드 이동 속도가 빨라질수록 경로 획득을 위한 제어 패킷 수가 급격히 증가함을 알 수 있다. 제어 패킷 수 증가는 경로 획득 과정을 지연시킬 뿐만 아니라 무선 대역폭 감소 및 노드의 에너지 소모율 증가 등 부가 영향(side effect)으

로 인해 전체적인 성능 저하를 초래한다. 본 논문에서는 노드 이동성을 고려해 동적 토폴로지에 보다 효과적으로 대처할 수 있는 개선된 AODV 알고리즘을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 AODV에서 사용되는 Expanding Ring Search 알고리즘과 관련 연구에 대해 설명하며, 3장에서는 본 논문에서 제안하는 알고리즘에 대해 상세히 기술한다. 4장에서는 실험 결과를 제시하며 5장에서 결론을 내리고자 한다.

2. Expanding Ring Search(ERS) 알고리즘 및 관련 연구

2.1 Expanding Ring Search 알고리즘

그림 2에 보인 것처럼 출발지 노드 1은 이웃 노드(2, 3, 4)에게 RREQ(route request) 패킷을 전송하고, 일정 시간 동안 RREP(route reply) 패킷을 기다린다. 만약 RREQ 패킷을 수신한 노드가 목적지 노드 8이라면, RREP 패킷을 발생시켜 출발지 노드로 전송한다. 그림 2에서 노드 2는 목적지 노드가 아니므로 RREQ 패킷을 다시 주변 노드인 5로 전달한다. 즉 RREQ 패킷을 수신한 노드는 RREP 패킷을 발생시킬지 아니면 주변 노드로 RREQ 패킷을 중계할지를 결정한다.

AODV 라우팅 프로토콜에서는 불필요한 RREQ 전송을 제한하기 위하여 Expanding Ring Search(ERS) 알고리즘을 사용한다[5]. 이 알고리즘은 전체 네트워크에 플러딩(flooding) 하는 대신 TTL(time to live)값을 이용해 점진적으로 탐색 범위(search ring)를 확장시켜 가며 목적지 노드를 찾는 방법이다. 표 1에 ERS 알고리즘에 사용되는 파라미터들을 요약하였다.

출발지 노드는 RREP 패킷 수신을 위한 타임아웃시간을 RING_TRAVERSAL_TIME으로 설정한다. TTL_VALUE의 초기값은 TTL_START(=1)이며, 타임아웃시간은 80ms가 된다. 이 시간 동안 출발지 노드는 RREP 패킷을 기다린다. 만약 타임아웃시간 경과 후에도 RREP 패킷을 수신하지 못했다면, 출발지 노드는

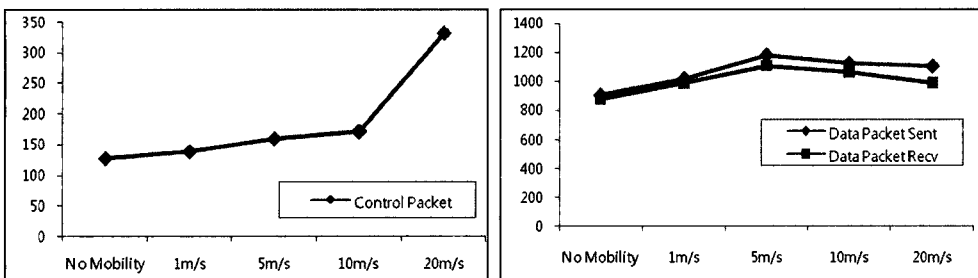


그림 1 노드 이동 속도 증가에 따라 제어 패킷 및 데이터 패킷 변동 (QualNet 시뮬레이터[10] 사용, 노드 수 49개)

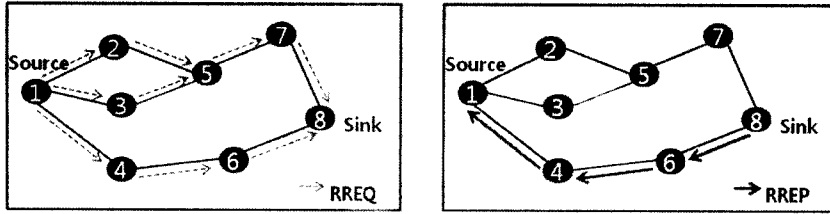


그림 2 RREQ 패킷 전달 경로 및 RREP 패킷 전달 경로

표 1 AODV 파라미터

파라미터	설정 값
NET_DIAMETER	35
NODE_TRAVERSAL_TIME	40ms
NET_TRAVERSAL_TIME	$2 \times \text{NODE_TRAVERSAL_TIME} \times \text{NET_DIAMETER}$
RING_TRAVERSAL_TIME	$2 \times \text{NODE_TRAVERSAL_TIME} \times \text{TTL_VALUE}$
TTL_START	1
TTL_INCREMENT	2
TTL_THRESHOLD	7

TTL_INCREMENT(=2)만큼 증가시킨 TTL_VALUE (=3)를 사용하여 RREQ 패킷을 재전송한다. 이 과정은 TTL_VALUE가 TTL_THRESHOLD(=7)에 도달할 때까지 최대 3회 반복된다. 만약 RREP 패킷을 수신하지 못했다면, 타임아웃시간을 NET_TRAVERSAL_TIME ($2 \times 40 \times 35 = 2800\text{ms}$)으로 설정하고, RREQ 패킷을 재전송한다. 이 시간 경과 후에도 RREP 패킷을 수신하지 못했다면, 목적지 노드가 네트워크 상에 존재하지 않는다고 판단하고 경로 탐색을 중단한다.

ERS 알고리즘에서 사용하는 NODE_TRAVERSAL_TIME(NTT)은 홉(hop)당 최대 전송지연시간을 의미한다. ERS 알고리즘은 유선 네트워크에서 RREQ 메시지 발생 개수를 제한하기 위하여 고안된 것이며, NTT는 40ms로 고정되어 있다. 그러나, 이처럼 고정된 값을 사용하면 네트워크 토폴로지가 수시로 바뀌는 ad-hoc 망에서는 불필요한 RREQ 패킷을 발생시킬 가능성이 커진다. 예를 들면, 이웃 노드가 출발지 노드로부터 여전히 전송 범위 내에 있지만 조금 더 멀리 이동했다면, 출발지 노드의 RREQ 메시지 전송에 따른 응답시간이 길어지게 된다. 이러한 상황 변화를 전혀 모르는 상태라면, 출발지 노드는 정해진 재전송 대기시간 경과 후에 RREQ 메시지를 재전송하게 된다 즉, 토폴로지 변동에 대한 느린 적응성으로 인해 제어 패킷 발생 빈도가 증가함은 물론 데이터 패킷 수신율은 오히려 감소하고, 반면 에너지 소모율은 증가한다. 만약 노드 이동에 따른 토폴로지 변동 상황을 적시에 반영할 수 있다면, RREQ 메시지 재전송을 피할 수 있게 되어 불필요한 패킷 발생을 억제할 수 있을 것이다.

2.2 관련 연구

ERS 알고리즘의 성능을 좌우하는 파라미터로는 NTT와 탐색 횟수(search threshold)가 있다. 탐색 횟수에 관해서는 토폴로지에 무관하게 3~4회가 적절하다는 연구 결과가 제시된 바 있다[6]. 이미 표 1에 보인 것처럼 TTL_THRESHOLD가 7로 지정되어 있어, ERS 역시 최대 3회로 제한하고 있음을 알 수 있다.

반면 NTT는 RREP 수신을 위한 최대대기시간인 RING_TRAVERSAL_TIME을 구할 때나 식 (1)의 실제 추정 대기시간을 구할 때 매우 비중 있는 역할을 하고 있음을 알 수 있다.

$$WT = 2 \times H \times \text{NODE_TRAVERSAL_TIME} \quad (1)$$

WT(waiting time): 출발지 노드가 RREP 패킷을 수신하기까지 대기시간

H: 목적지 노드에서 출발지 노드까지의 홉(hop) 수

Tripathi[7]등은 RREQ 패킷에 timestamp 필드를 추가함으로써 시간 정보를 라우팅 경로 설정 과정에 도입하였다. 예를 들면, 출발지 노드에서 RREQ 패킷 생성 시각과 목적지 노드에서 RREQ 패킷 수신 시각 정보를 알 수 있다면 해당 경로에서의 홉당 평균 전달지연시간을 추정할 수 있다. 홉당 평균 전달지연시간은 망의 현재 상태를 반영하고 있으므로 NTT값 설정에 사용될 수 있다. 경로 설정 과정에서 홉당 평균 전달지연시간을 추정하는 방법은 RREQ 패킷을 사용한 방식[8,9]과 RREP 패킷을 사용한 방식[10]이 있다. RREQ 패킷을 이용한 연구로는 식 (2)와 같이 n 번째 시점에서 측정된 값과 가중 누적된 이전 값을 고려하여 계산된 이동 평균값을 계산하는 방식이 있다(단, $0 \leq a \leq 1$)[8]. 이 식은 TCP 재전송 타이머에서 사용되는 SRTT(Smoothed Round Trip Time) 공식을 차용한 것이다.

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

RREP 패킷을 사용한 관련 연구로는 목적지 노드가 자신과의 인접도에 따라 경로에 포함된 노드들에 지수적 또는 선형적으로 가중치를 부여하여 그 평균값을 NTT로 설정하는 방식이 제안되었다[9]. 예를 들면, 목적지 노드에서 모두 4개의 중간 노드를 거쳐 출발지 노드에 도달하는 경우 식 (3(a))는 지수적 가중치 배분 방식일 때, 식 (3(b))는 선형적 가중치 배분 방식일 때 각각 NTT를 계산하는 식이다. 식 (3)에서 t_i 는 $i-1$ 번째 노드와 i 번째 노드간 전송시간을 나타낸다.

$$NTT = \frac{t_1 + t_2 + 2t_3 + 4t_4 + 8t_5}{16} \quad (a)$$

$$NTT = \frac{t_1 + 2t_2 + 3t_3 + 4t_4 + 5t_5}{15} \quad (b) \quad (3)$$

그러나 위에서 언급한 방식들처럼 실시간으로 NTT 값을 변경할 경우 네트워크 변동 상황을 즉각 반영할 수 있지만, 급격한 NTT 값 변경으로 인해 망의 안정성은 상대적으로 저하된다. 또한 이들 논문[8,9]의 실험 결과에 제시된 것처럼 노드의 이동성이 낮거나 전송 지연이 전혀 발생하지 않을 경우 기존 AODV 프로토콜보다 오히려 성능이 저하된다.

3. Control Packet Minimized AODV 프로토콜

본 논문에서 제안하는 라우팅 프로토콜은 네트워크 상황에 맞게 NTT 값을 가변적으로 조절함으로써 불필요한 RREQ 메시지 발생 빈도를 감소시켜 에너지 소모율을 감소시키고, 목적지에서 수신하는 데이터 패킷의 수신율을 증가시키고자 하는 것이다.

그림 2에 보인 것처럼, 출발지 노드에서 발생시킨 RREQ 패킷이 목적지 노드까지 도달할 수 있는 경로는 여러 개이지만, 목적지 노드가 발생시킨 RREP 패킷이 출발지 노드까지 도달할 수 있는 경로는 하나뿐이다. RREP 패킷은 단일 경로를 통해 목적지 노드에서 출발지 노드까지 전송되므로 비교적 안정적으로 NTT 값을 예측할 수 있다.

AODV에서는 라우팅 패킷마다 확장필드를 갖고 있으며, 이 확장필드에 해당 패킷의 발생시간(timestamp)을 저장할 수 있다. 따라서, 출발지 노드에서의 RREP 패킷 수신시간(DT)에서 해당 RREP 패킷에 저장된 timestamp, 즉 목적지 노드에서의 RREP 발생시간(ST)과의

차를 구하면 RREP 패킷 전달시간이 된다. 무선 링크(link)간 평균 전달지연시간(또는 홉당 평균 전달지연시간) MT는 식 (4)와 같이 RREP 패킷 전달시간을 전체 홉 수로 나누면 된다.

$$MT = (DT - ST) / H \quad (4)$$

MT(mean time): 노드간 평균 전달지연시간

ST: (목적지 노드에서의)RREP 패킷 발생 시간,

DT: (출발지 노드에서의)RREP 패킷 수신 시간,

식 (4)의 MT는 노드간 평균 전송지연시간을 의미하지만, NTT는 모든 링크에 적용되는 홉당 최대 전달지연시간이다. 예를 들어 그림 2와 같이 출발지 노드에서 목적지 노드까지 3홉 거리만큼 떨어져 있고 각각 40ms, 30ms, 20ms의 전달지연시간이 발생했다면, MT는 30ms이지만 NTT는 적어도 40ms이상으로 설정되어야만 한다. MT는 NTT를 설정함에 있어 상대적인 기준을 제공한다. 따라서, 표 2에 나타난 것처럼 MT와 NTT간 차에 따라 NTT를 재설정하는 것이 필요하다.

실험 결과 NTT를 40ms이하로 설정할 경우 응답대기시간이 단축됨에 따라 RREQ 재전송 발생 빈도가 급증해 오히려 성능 저하가 발생하였다. 그림 3에 나타난 것처럼 NTT가 증가함에 따라 RREQ 패킷 생성 횟수가 감소함을 알 수 있다. NTT가 40ms일 경우에 비해 20ms로 감소시키면 RREQ 생성 횟수는 48% 증가(52개 → 77개)한다. 따라서, NTT의 최소값은 40ms(lower_bound) 이상으로 설정해야만 한다. 그러나 지나치게 NTT를 증가시키면 출발지 노드에서 목적지 노드까지 종단간 전달지연시간이 불필요하게 지연되는 문제를 초래할 수 있다.

적용형 NTT 값 설정을 위한 Control Packet Minimized AODV(CP_AODV) 알고리즘은 다음과 같다.

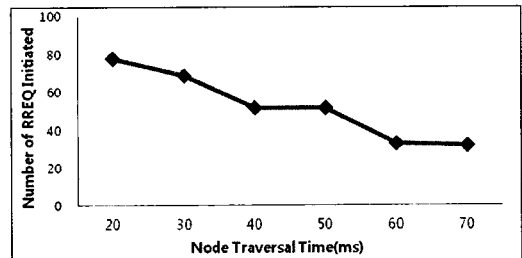


그림 3 NTT값에 따른 RREQ 생성 횟수 변동

표 2 링크 전달지연시간 변동에 따른 NTT 재설정 (단위: ms)

각 링크 별 전달지연시간			MT	Current NTT	NTT - MT	New NTT
link 1	link 2	link 3				
46	36	26	36	40	4	45
34	24	14	24	40	16	35

NTT값의 급격한 변경은 오히려 탐색 범위를 넓히지 못하거나 지나치게 확대할 가능성이 있기 때문에 NTT 값은 AIAD(additive increase additive decrease) 방식으로 변경된다.

```

IF (min_threshold > NTT - MT)
{
    NTT = NTT + increment;
}
ELSE IF (max_threshold < NTT - MT)
{
    NTT = NTT - increment;
    IF (NTT < lower_bound)
    {
        NTT = lower_bound;
    }
}
    
```

위 알고리즘에서 *min_threshold*, *max_threshold*, *increment*는 네트워크의 상태에 따라 조절 가능한 제어 파라미터들이며, *max_threshold* > *min_threshold*이다. NTT와 MT의 차가 *min_threshold* 보다 작아졌다는 것은 링크간 평균 전달지연시간이 그만큼 증가한 셈이므로, 해당 최대 전송지연시간인 NTT를 이전보다 늘려야 할 상황이 발생한 것으로 판단한다. 즉 이전 NTT 값을 *increment* 만큼 증가시킨다. 반대로 NTT와 MT의 차가 *max_threshold* 보다 크다는 것은 현재 네트워크 상황에서는 링크간 평균 전달지연시간이 NTT에 비해 훨씬 빠른 것으로 해석할 수 있다. 즉 응답 패킷 수신을 위한 최대 대기시간인 NTT를 단축시킬 수 있는 상황으로 판단한다. 즉 이전 NTT 값을 *increment* 만큼 감소시킨다. 제어 파라미터 *increment*의 경우 노드 이동이 빈번하거나 노드 간 전송지연시간 변동 폭이 클 경우 그 값을 상대적으로 크게 조절할 필요가 있다. 만약 NTT와 MT의 차가 *min_threshold* < NTT - MT < *max_threshold* 범위 내에 있을 경우 네트워크 상태가 상대적으로 안정화되어 있는 것으로 간주하여 이전 NTT를 그대로 유지한다.

4. 실험 및 결과분석

제안된 CP_AODV 프로토콜의 성능을 평가하기 위한 도구로 QualNet 4.0 시뮬레이터[11]를 사용하였다. 제안된 프로토콜은 기존 AODV 프로토콜 및 관련 연구에서 언급한 timestamp를 이용한 프로토콜[10] 등과 노드 이동시간, 노드 이동속도 및 노드 밀집도 등 3가지 평가 모델에 대해 각각 성능을 비교하였다. 시뮬레이션을 위

표 3 시뮬레이션 파라미터

파라미터	설정 값
네트워크 크기	1,500m × 1,500m
대역폭	2MB
노드 수	49개
배치 방식	GRID
무선 전송범위	250m
노드 이동속도	0 ~ 10m/s
시뮬레이션 시간	600초

해 설정된 파라미터는 표 3과 같다. 트래픽은 512byte 패킷 크기의 CBR(constant bit ratio)을 사용하였다. VBR로 설정했을 경우 CBR과 유의미한 차가 없었기 때문에, 트래픽 타입은 CBR을 적용하였다. 한편, CP_AODV 프로토콜에 사용된 제어 파라미터들은 실험을 통해 얻어진 값을 사용하였다(*min_threshold* = 5, *max_threshold* = 10, *increment* = 5).

4.1 이동시간에 따른 실험 결과 및 분석

노드 이동에 따른 네트워크 토폴로지 변동을 시뮬레이션 과정에 반영하기 위해 정지시간(pause time)을 0~600초 범위 내에서 조절하며 실험하였다. 정지시간이란 전체 시뮬레이션 시간 중 노드가 이동하지 않고 멈춰있는 시간을 말한다. 정지시간이 0초이면 전체 시뮬레이션 시간 내내 노드가 이동할 수 있기 때문에 노드 이동성이 가장 높다. 반면, 정지시간이 600초이면 노드 이동은 전혀 발생하지 않으며, 결과적으로 노드 이동성이 가장 낮다.

RREQ 메시지는 이 메시지를 생성한 노드에 따라 2가지로 구분할 수 있다. 출발지 노드가 경로를 찾기 위해 생성한 RREQ 메시지(RREQ initiated)와 라우터 역할을 담당하는 중간 노드가 메시지 전달을 위해 생성한 RREQ 전달 메시지(RREQ forwarded)가 있다.

CP_AODV의 경우 AODV에 비해서는 RREQ 생성 횟수(그림 4) 및 RREQ 전달 횟수(그림 5)에서 각각 33% 및 28% 감소하였으며, timestamp에 비해서는 각각 12% 및 21% 감소하였다. 그림 4에서 알 수 있는 것처럼, 정지 시간이 400초 미만일 경우 제안한 방식이

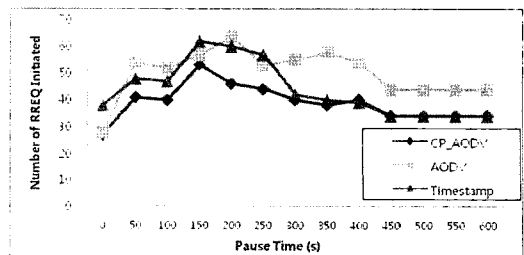


그림 4 RREQ 생성 횟수

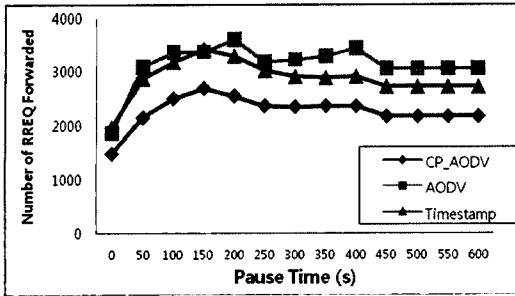


그림 5 RREQ 전달 횟수

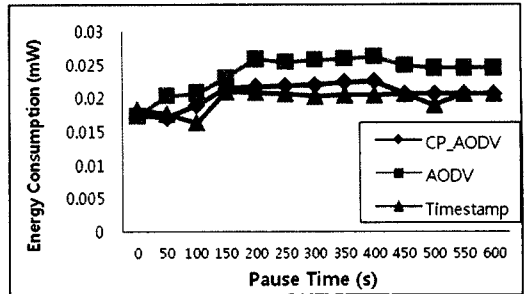


그림 8 에너지 소모율

timestamp에 비해 RREQ생성 횟수가 뚜렷이 감소함을 알 수 있다. ad-hoc 망을 돌아다니는 RREQ 메시지 수가 감소함에 따라 RREP 및 RERR 메시지 발생 빈도에도 직접적인 영향을 미침을 알 수 있다. 제안한 방식이 RREP 발생 횟수(그림 6)와 RERR 발생 횟수(그림 7)에서도 AODV에 비해서는 각각 25% 및 32% 감소하였으며, timestamp에 비해서는 각각 22% 및 33% 감소하였다. RREP 및 RERR 패킷 발생 횟수는 RREQ 패킷 발생 횟수에 거의 비례적으로 변화함을 알 수 있다.

제안한 방식이 AODV와 timestamp에 비해 모든 종류의 제어 패킷 발생 횟수를 줄일 수 있음을 확인하였다. 이러한 제어 패킷 감소로 인해 ad-hoc 망에서의 전체 에너지 소모율도 자연히 줄어들 것으로 예측할 수 있다. 실제 시뮬레이션 결과 역시 제안한 방식이 AODV

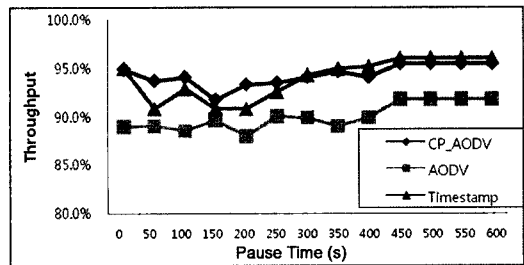


그림 9 데이터 패킷 수신률

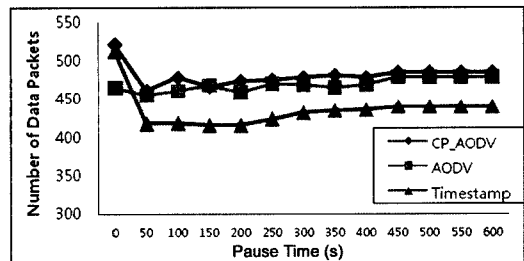


그림 10 데이터 패킷 수신 횟수

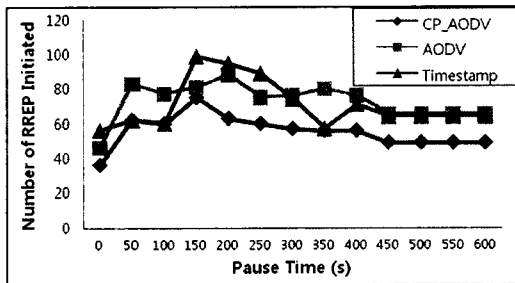


그림 6 RREP 발생 횟수

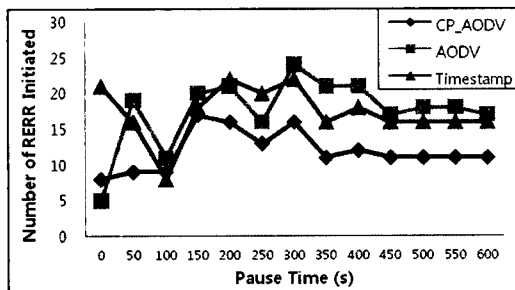


그림 7 RERR 발생 횟수

에 비해 에너지 소모율이 13% 감소함을 알 수 있다(그림 8). 에너지 소모율 변동 곡선이 그림 4~그림 7의 RREQ, RREP 및 RERR 제어 패킷 변동 곡선과 비교하지 않는 이유는 제어 패킷 외에 데이터 패킷 전송 시 에너지 소모율이 함께 포함되었기 때문이다.

한편, 그림 8에서 제안한 방식이 timestamp에 비해 에너지 소모율이 오히려 6% 증가함을 알 수 있다. 그 이유는 목적지 노드가 최종적으로 수신하는 데이터 패킷 개수에서 차이가 나기 때문이다. 그림 9와 그림 10에 나타난 것처럼, timestamp방식은 제안한 방식과 비교해 데이터 패킷 수신률은 엇비슷하지만, 데이터 패킷 수신 개수는 약 10% 적음을 알 수 있다. 다시 말해 timestamp 방식이 제안한 방식에 비해 상대적으로 적은 양의 데이터 패킷을 송수신하기 때문에, 에너지 소모율이 상대적으로 낮게 측정된 것이다.

제안한 방식이 기존 AODV와 비교해 데이터 패킷 수

신율은 평균 5%, 데이터 패킷 수신 개수는 평균 3% 증가함을 알 수 있다(그림 9, 10). Ad-hoc 망에서 각 노드가 전달해야 하는 제어 패킷 수가 줄어들어 네트워크 트래픽이 감소한 만큼 데이터 패킷 수신률도 증가하는 것으로 보인다. 또한 노드 이동에 따라 네트워크 토폴로지가 바뀌면 제어 패킷 생성을 통해 경로를 재설정해야 하는데, 이러한 경로 재설정 작업이 제어 패킷 감소 등으로 상대적으로 원활하게 이뤄지기 때문에 보다 안정적으로 데이터 패킷을 수신할 수 있는 것으로 판단된다.

4.2 이동속도에 따른 실험 결과 및 분석

노드의 이동속도를 1 m/s단위로 변화시켜 가며 RREQ 발생 횟수(그림 11), 에너지 소모율(그림 12) 및 데이터 패킷 수신율(그림 13)을 각각 측정하였다. 예를 들어 이동속도를 5m/s로 설정했을 경우, 각 노드는 4~5m/s의 이동 속도를 갖게 된다. 4.1절에서 이동속도를 0~10m/s로 지정했을 경우와는 달리 모든 노드는 거의 일정한 속도로 움직이게 된다. 또한, 정지 시간은 100초로 설정하였으며, 다른 파라미터는 표 3의 값을 사용하였다. 그림 11에서 알 수 있는 것처럼 제안한 방식의 평균 RREQ 발생 횟수는 58.7회로써, 기존 AODV(67.9회)나 timestamp(66.1회)에 비해 낮게 측정되었다. 또한, RREQ 발생 횟수에 대한 표준 편차도 가장 작아 노드 이동속도 변화에 대해 가장 안정적인 상태를 유지함을 알 수 있다.

에너지 소모율(그림 12)과 데이터 패킷 수신율(그림 13) 역시 제안한 방식이 AODV나 timestamp 방식에

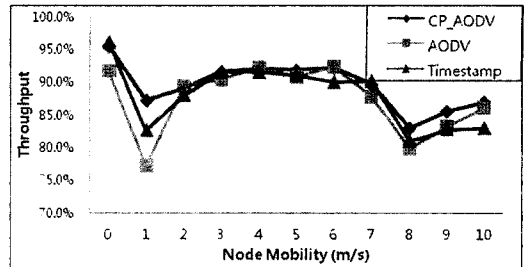


그림 13 데이터 패킷 수신률

비해 각각 11~14% 및 1.7~2.4% 개선된 결과를 얻었다. 이 실험 결과에서 주목할 점은 노드의 이동속도가 느리다고 해서 이동속도가 빠를 때 비해 항상 유리하지는 않다는 점이다. 노드의 이동 방향에 따라서는 이동 거리가 짧더라도 인접 노드로부터 전송 범위를 벗어날 가능성이 일정하지 않기 때문이다.

4.3 노드 밀집도에 따른 실험 결과 및 분석

36, 49, 64, 81, 100, 121개로 노드 개수를 변화시켜 가며, RREQ 발생 횟수(그림 14), 에너지 소모율(그림 15) 및 데이터 패킷 수신율(그림 16)을 각각 측정하였다. 노드 개수가 증가함에 따라 노드 간 거리는 50m단위로 단축시켰다. 즉, 노드 수가 64개 일 때 노드 간 거리는 200m이지만, 81개 일 때는 노드 간 거리는 150m로 50m줄어든다. 정지 시간은 100초로 설정하였으며, 다른 파라미터는 표 3의 값을 사용하였다. 또한 timestamp방식에 대한 실험 결과는 4.1절 및 4.2절 결과와 유사하기 때문에, 기존 AODV방식과 직접 비교를 위해 실험 결과에 포함시키지 않았다.

정해진 네트워크 크기에서 노드 수가 증가하면 인접 노드간 거리는 단축된다. 바꿔 말하면, 노드 밀집도가 높아지면, 노드의 전송 범위 내에 놓인 인접 노드 수가 늘어나므로, 출발지 노드의 RREQ 패킷에 대해 응답할 수 있는 이웃 노드 수도 함께 증가한다. 따라서 그림 14 및 그림 16에서 알 수 있는 것처럼 노드 밀집도 증가에 따라 RREQ 발생 빈도는 급격히 감소하는 반면 데이터 패킷 수신율은 오히려 증가함을 알 수 있다. 노드 밀집

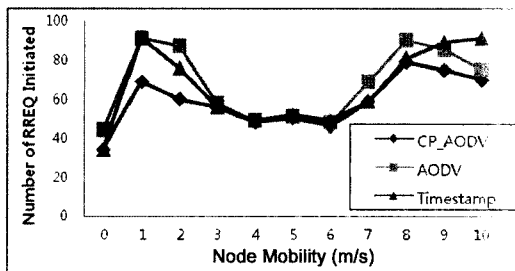


그림 11 RREQ 발생 횟수

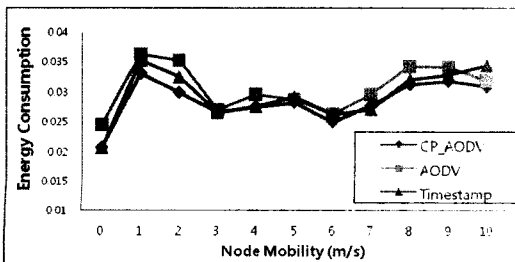


그림 12 에너지 소모율

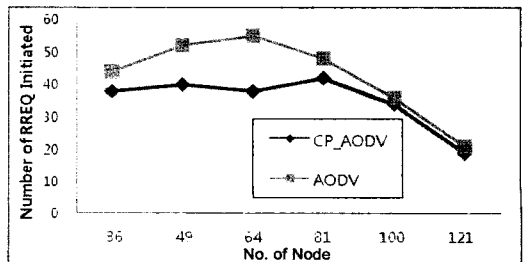


그림 14 RREQ 발생 횟수

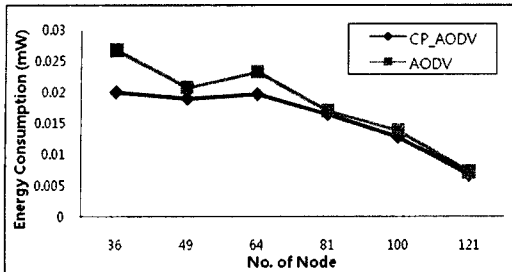


그림 15 에너지 소모율

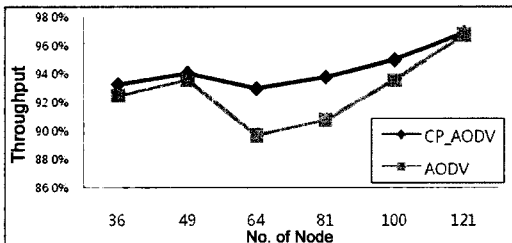


그림 16 데이터 패킷 수신률

도가 높다면, 일부 노드가 전송 범위를 벗어나더라도 전송 범위 내에 있는 또 다른 인접 노드를 찾을 가능성이 매우 높기 때문이다. 따라서 경로 재설정을 위한 제어 패킷 발생 횟수도 상대적으로 작아진다. 이로 인해 그림 15에 보인 것처럼 에너지 소모율도 RREQ 발생 횟수 변동에 비례해 감소함을 알 수 있다.

그림 14에서 알 수 있는 것처럼 제안한 방식의 평균 RREQ 발생 횟수는 35.2회로써, 기존 AODV(42.7회)에 비해 낮게 측정되었다. 또한, 제안한 방식이 AODV에 비해 에너지 소모율 및 데이터 패킷 수신율 측면에서 각각 평균 13% 및 2% 향상된 것으로 나타났다. 그러나 노드 밀집도가 높아질수록 그 차는 점점 줄어들며, 노드 개수가 100개 이상(노드 간 거리 100m 이내)일 경우에는 데이터 패킷 수신율을 제외하면 거의 성능 차가 없는 것으로 측정되었다. 제안한 방식에서는 토폴로지 변동에 따른 제어 패킷 발생 빈도를 최소화시킴으로써 데이터 패킷 수신율을 증가 및 에너지 소모를 감소 등의 부가 효과를 얻을 수 있었다. 그러나, 노드 밀집도가 높아지면 제어 패킷 발생 빈도에 거의 변화를 줄 수 없게 되어 기존 AODV와 유사한 성능을 갖게 된다.

5. 결론

본 논문에서는 노드의 이동성으로 인해 네트워크 토폴로지가 수시로 변경되는 무선 ad-hoc 네트워크 환경에 보다 효과적으로 대처할 수 있는 CP_AODV 알고리즘을 제안하였다. Ad-hoc 네트워크에서 노드 이동성이 높아질수록 데이터 패킷 수는 거의 변화가 없지만 경로

획득을 위한 제어 패킷 수는 급격히 증가한다. 이러한 제어 패킷 수 증가를 최소화하기 위해, 홉당 최대 전송 지연시간인 NTT를 네트워크 상황에 맞게 AIAD방식으로 유연하게 조절하는 방안을 제안하였다. CP_AODV가 AODV에 비해서는 RREQ 생성 횟수 및 RREQ 전달 횟수에서 각각 33% 및 28% 감소하였으며, timestamp에 비해서는 각각 12% 및 21% 감소하였다. 또한, CP_AODV가 AODV와 timestamp에 비해 RREQ외에 RREP, RERR 등 모든 종류의 제어 패킷 발생 빈도를 뚜렷이 감소시킴을 확인하였다. 뿐만 아니라, CP_AODV가 기존 AODV에 비해 에너지 소모율 및 데이터 패킷 수신율 측면에서 각각 평균 13% 및 2% 향상된 것으로 나타났다. 다시 말해 CP_AODV를 적용할 경우 토폴로지 변동에 따른 제어 패킷 발생 빈도를 최소화시킴으로써 데이터 패킷 수신율을 증가 및 에너지 소모를 감소 등의 부가 효과를 얻을 수 있었다. 그러나, 노드 밀집도가 지나치게 높아지면 제어 패킷 발생 빈도에 거의 변화를 줄 수 없게 되어 기존 AODV와 유사한 성능을 갖게 된다.

참고문헌

- [1] Corson, Scott S. Macker, J., "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, January 1999.
- [2] C. K. Toh, "Ad-hoc Mobile Wireless Networks: Protocols and Systems," Prentice Hall PTR, 2002.
- [3] C. Perkins, "Performance Comparison of Two On-Demand Routing Protocols for Ad-hoc Networks," Proc. of the IEEE Infocom, pp.3-12, March 2000.
- [4] <http://www.faqs.org/rfcs/rfc3561.html>, Ad hoc On-Demand Distance Vector (AODV) Routing (RFC 3561).
- [5] 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.
- [6] Hassan, J.and Jha, S., "Optimizing expanding ring search for multi-hop wireless networks," GLOBECOM, pp1061-1065, 2004.
- [7] S.M.Tripathi R. K. Ghosh, "An Implementational Study of Certain Heuristics for the Performance Enhancements of AODV," Proc. Of the 10th Euro-Micro Workshop on Parallel, Distributed and Network-based Processing, pp.395-402, January 2002.
- [8] 이재석, 박혜련, 이기현, 석정봉, "Timestamp를 이용한 AODV 라우팅 알고리즘의 성능 개선", 한국통신학회논문지 Vol. 31, No.8, August 2006.
- [9] Park, Incheon and Pu, Ida, "Energy Efficient Expanding Ring Search," 1st ASIA Int. Conf. on Modeling & Simulation(AMS), pp. 198-199, 2007.
- [10] 신창윤, 강승호, 임형석, "AODV라우팅 프로토콜에서 재전송률을 낮추기 위한 ERS알고리즘의 노드순회시간

계산방법”, 정보처리학회논문지 제13-C권 제4호, pp. 447-454, August 2006.

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



홍 윤 식

1983년 한양대학교 전자공학과 졸업(학사). 1985년 한국과학기술원 전기 및 전자공학과(공학석사). 1989년 한국과학기술원 전기 및 전자공학과(공학박사). 1991년~현재 인천대학교 컴퓨터공학과 교수
관심분야는 모바일 컴퓨팅, USN



홍 준 식

1986년 명지대학교 전기공학과 졸업(학사). 1993년 연세대학교 전기공학과(공학석사). 2002년 충북대학교 전기공학과(공학박사). 2007년~현재 청주대학교 전자정보공학부 전임강사. 관심분야는 영상처리, 패턴인식 등



임 화 식

2008년 인천대학교 컴퓨터 공학과 졸업(학사). 2008년~인천대학교 대학원 컴퓨터공학과 재학. 관심분야는 ad-hoc 및 USN 네트워크