

네트워크 토폴로지 변화를 고려한 적응형 AODV 알고리즘

김지홍, 김용현, 이수용, 임화석, 오명근, 홍윤식
인천대학교 컴퓨터공학과

e-mail : {riot999, yh-kim, bluegenii} @incheon.ac.kr
{genocide00, omkjuly}@nate.com, yshong@incheon.ac.kr

An Adaptive AODV Algorithm for Considering the Changes In The Network Topology

Ji-Hong Kim, Yong-Hyun Kim, Su-Yong Lee, Hwa-Seok Lim, Myung-Keun Oh, and Youn-Sik Hong
Dept. of Computer Science and Engineering, University of Incheon

요 약

AODV에서는 RREQ 메시지 전송을 통해 라우팅 경로를 설정한다. Ad-hoc 네트워크에서 노드가 자주 이동하거나 전송 지연 시간이 클 경우 RREQ 메시지 발생이 증가한다. 이러한 네트워크 변동에 따른 RREQ 메시지 발생 증가는 결과적으로 데이터 패킷 수신율을 저하시킬 뿐만 아니라, 노드의 에너지 소모율도 증가시킨다. 본 논문에서는 네트워크 토폴로지 변동 상황을 감지하여 AODV에서의 RREQ 메시지 발생 빈도를 효과적으로 조절하는 알고리즘을 제안하였다. 제안된 알고리즘을 50개의 노드가 10m/s 이하의 속도로 무작위로 이동하는 Ad-hoc 네트워크에 적용한 결과, 기존 AODV 알고리즘에 비해 RREQ 메시지 발생 빈도가 25% 감소하였다. 뿐만 아니라 RREP 패킷과 RERR 패킷 역시 각각 26% 및 31%씩 감소하였다. 모든 종류의 메시지 발생 빈도 수가 감소함에 따라 데이터 패킷 수신율은 3% 증가했으며, 에너지 소모율 역시 13% 감소하였다.

1. 서론

Ad-hoc 네트워크란 기지국이나 AP(Access Point) 없이 이동단말(mobile terminal, 노드)들로만 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 함께 수행한다. Ad-hoc 네트워크의 노드들은 이동성을 갖고 있으므로 네트워크 토폴로지(topology)가 동적으로 변화한다. 이와 같은 네트워크 환경에 대해 기존 유선 네트워크의 라우팅 프로토콜을 그대로 적용한다면, 토폴로지 변동에 대한 느린 적응성으로 인해 제어 패킷 발생 빈도가 증가함은 물론 데이터 패킷 수신율은 오히려 감소하고, 반면 에너지 소모율은 증가한다.

Ad-hoc 네트워크 라우팅 알고리즘은 1990년 말부터 제안되기 시작하여, 현재는 기존에 제안된 알고리즘의 단점을 보완한 연구 결과들이 제시되고 있다[2][3]. Ad-hoc 네트워크의 라우팅 프로토콜 중 가장 대표적인 것은 on-demand 방식의 AODV(Ad hoc On demand Distance Vector)이다. AODV에서는 패킷 발생율이 급격히 증가하는 것을 막기 위해 이웃 노드를 중심으로 패킷 전송 범위를 점진적으로 확대해가는 핸드셰이크(handshake) 방식을 채택하고

있다.

즉, 출발지노드는 이웃 노드에게 RREQ(request) 메시지를 전송하고, 일정 시간 동안 RREP(reply) 메시지를 기다린다. 일정 시간 경과 후에도 RREP 메시지를 수신하지 못했을 경우 RREQ 메시지를 재전송하게 된다. 그러나 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 재전송하고 만다. 만약 노드 이동에 따른 토폴로지 변동 상황을 적시에 반영할 수 있다면, RREQ 메시지 재전송을 피할 수 있게 되어 불필요한 패킷 발생을 억제할 수 있다.

본 논문에서는 RREQ 발생 빈도를 네트워크 상황에 맞게 효과적으로 조절하는 알고리즘을 제안하고자 한다. RREQ와 같은 제어 패킷 발생 빈도가 감소하면 에너지 소모율이 줄어들 뿐만 아니라 데이터 패킷 수신율도 개선할 수 있기 때문이다.

본 논문의 구성은 다음과 같다. 본론의 1 장에서는 본 논문에서 제안하는 알고리즘의 기반이 되는 AODV에 대해서 기술하고자 한다. 2 장에서는 관련연구 및 제안하고자 하는 알고리즘과 시뮬레이션 결과

에 대해서 분석하고 마지막 장에서는 결론에 대해서 논하고자 한다.

2. 본론

1 Expanding Ring Search(ERS) 알고리즘

AODV 라우팅 프로토콜에서는 불필요한 RREQ 전송을 제한하기 위하여 Expanding Ring Search(ERS) 알고리즘을 사용한다. 이 알고리즘은 전체 네트워크에 플러딩(flooding) 하는 대신 RREQ 메시지를 이용해 점진적으로 범위를 넓게 확장시켜 가며 목적지 노드를 찾는 방법이다.

출발지 노드는 RREQ 패킷의 IP 헤더에 TTL을 $TTL_START(=1)$ 로 초기화시키고, 타임아웃시간을 $2 * TTL * NODE_TRAVERSAL_TIME (=2*1*40)ms$ 설정하여, 이 시간 동안 RREP 패킷을 기다린다. 만약 타임아웃시간 경과 후에도 RREP 패킷을 수신하지 못했다면, 출발지 노드는 $TTL_INCREMENT(=2)$ 만큼 증가시킨 TTL을 사용하여 RREQ를 재방송한다. 이 과정은 $TTL_THRESHOLD(=8)$ 에 도달할 때까지 반복된다. 만약 이렇게 해도 패킷을 수신하지 못했다면, TTL을 $NET_DIAMETER(=35)$ 에 이를 때까지 2회 재방송한다. 그래도 RREP 패킷을 수신하지 못했다면 이 경우 이는 목적지 노드가 네트워크 상에 존재하지 않는다고 판단하여 더 이상의 경로 탐색을 실시하지 않는다[4].

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 사용하기 때문에 Ad-hoc 네트워크에서는 불필요한 RREQ 패킷을 발생시킨다. 예를 들면, RREP 패킷 전송 시간이 40ms보다 클 경우, 이웃 노드에서는 RREP 패킷을 보내 응답했음에도 불구하고 NTT 초과로 출발지 노드는 RREQ 패킷을 재전송하게 된다.

2 관련연구

AODV의 ERS 알고리즘에서 사용하는 NTT값을 측정하기 위해 따로 제어 패킷을 만들어 전송하는 방법이 제안되었으나 정식으로 승인된 버전에서는 생략되었다. 이와 관련된 연구가 계속적으로 진행되어 왔으며, Timestamp값을 이용하여 시간차를 계산하여 확보된 거리 정보를 라우팅에 적용 하는 방법[5], 라우팅 과정의 시간차를 출발지와 목적지 노드 사이에서 구해 그 평균을 NTT값 계산에 이용한 연구[6]등이 있었다. 또한 RREQ 패킷에 Timestamp 필드를 설정하여 패킷 발생 시간을 기록하여 전송하고, 이것을 받은 노드는 현재 시간과 패킷 내 Timestamp 값을 이용하여 노드 간의 패킷 전송시간을 동적으로 계산하는 방식의 연구도 진행되었다[7].

하지만 기존에 제시된 방법들과 같이 실시간으로

NTT의 값을 변경할 경우 네트워크 상황에 즉각적으로 대응할 수 있는 장점이 있지만, 네트워크의 안정성은 상대적으로 저하될 수밖에 없다. 또한 노드의 이동성이 낮을 경우나 전송지연이 발생하지 않을 경우 기존 AODV 프로토콜보다 오히려 RREQ 패킷 전송 빈도가 증가하여 전체적인 성능 저하를 가져오는 단점이 있다.

3 제안하는 알고리즘 및 성능 측정

3.1 Control Packet 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. 네트워크 상황 변동을 고려한 실시간 NTT 값을 구하기 위해 아래와 같은 수식을 사용하였다.

ST : (출발지 노드에서의)RREQ 발생 시간

RT : (목적지 노드에서의)RREQ 수신 시간

H : 출발지 노드에서 목적지 노드까지의 홉 수

NTT (Node Traversal Time) : 홉당 최대 전송지연시간

CT : 노드간 평균 전달 시간

$$RREQ\text{전달시간} = RT - ST \quad (1)$$

$$RREQ\text{가 전달된 홉 수} = H + 1 \quad (2)$$

$$CT = (RT - ST) / (H + 1) \quad (3)$$

AODV에서는 라우팅 패킷마다 확장필드를 갖고 있다. 이 확장필드에는 해당 패킷의 발생시간을 알 수 있는 Timestamp 정보가 저장되어 있다. 따라서, RREQ 전달시간은 목적지 노드에서의 RREQ 수신시간(RT)에서 RREQ 패킷에 저장된 Timestamp 값, 즉 RREQ 발생시간(ST)과의 차를 구하면 된다(식(1)). RREQ 패킷이 중간 노드를 경유할 때마다 홉 수가 1씩 증가하기 때문에, RREQ가 전달된 전체 홉 수는, 식 (2)와 같이, 목적지 노드가 수신한 RREQ 패킷에 포함되어 있는 홉 수(H)에 1을 더하면 된다. 따라서 노드 간 평균전달시간 CT는 식(1)에서 구한 RREQ 전달시간을 식(2)에서 구한 전체 홉 수로 나누면 된다[8].

이렇게 구한 CT는 노드간 평균전송시간이지만, NTT는 모든 링크(link)에 적용되는 최대지연시간이다. 예를 들어 출발지 노드에서 목적지 노드까지 3홉 거리만큼 떨어져 있고 각각 30ms, 20ms, 10ms 의 전달지연시간이 발생했다면, $CT=20ms$ 이지만 NTT는 적어도 30ms이상으로 지정되어야만 한다. 즉 CT는 NTT를 구함에 있어 상대적인 기준을 제공한다. 특히 40ms이하로 NTT를 설정할 경우 RREQ 패킷 발생 빈도가 급격히 늘어나 오히려 성능 저하가 발생한다. 따라서, NTT는 최소값(lower_bound) 이상으로 설정해야만 한다.

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

```

IF (min_threshold > NTT - CT)
{
    NTT = NTT + increment;
}
ELSE IF (max_threshold < NTT - CT)
{
    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 CT의 차가 *min_threshold* 보다 작다는 것은 CT가 현재 네트워크 상황에서 노드간 평균 전달시간임을 감안하면, 홀당 최대 전송지연시간인 NTT를 이전보다 늘려야 할 상황이 발생한 것으로 판단할 수 있다. 즉 이전 NTT 값을 *increment* 만큼 증가시킨다. 반대로 NTT와 CT의 차가 *max_threshold* (*max_threshold* > *min_threshold*)보다 크다는 것은 현재 네트워크 상황에서는 노드간 평균 전달시간이 NTT에 비해 훨씬 빠른 것으로 해석할 수 있다. 즉 RREP 패킷 수신을 위한 최대 대기시간인 NTT를 줄여도 되는 상황으로 판단할 수 있다. 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.

한편, NTT가 40ms 이하로 감소할 경우에는 40ms 로 설정한다. 지나치게 작은 NTT는 대기시간을 단축시켜 RREQ 재전송 발생 빈도를 높여 네트워크 안정화에 불리하게 작용할 수 있기 때문이다. 만약 NTT와 CT의 차가 *min_threshold* < NTT - CT < *max_threshold* 범위 내에 있을 경우 네트워크 상태가 상대적으로 안정화되어 있는 것으로 간주하여 이전 NTT를 그대로 유지한다.

2.4 시뮬레이션 결과

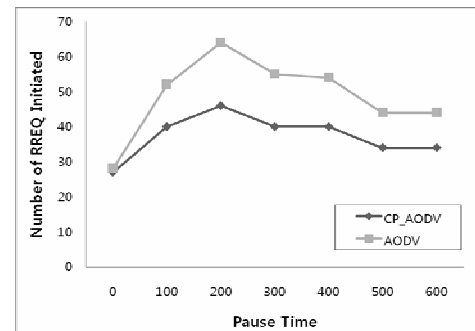
CP_AODV 알고리즘의 성능을 평가하기 위한 도구로 QualNet 4.0 시뮬레이터를 사용하였다[9]. 이 알고리즘에 사용된 제어 파라미터들은 실험을 통해 그 값을 지정하였으며(*min_threshold* = 5, *max_threshold* = 15, *increment* = 5), 기존 AODV 프로토콜과 성능을 비교하였다. 시뮬레이션에 사용된 파라미터는 <표 1>과 같다. 트래픽은 512byte 패킷 크기의 CBR(constant bit ratio)을 사용하였다. 한편, 네트워크 토폴로지 변동 상황을 반영하기 위하여 정지 시간(pause time)을 0~600초 범위 내에서 조절하며 시뮬레이션 하였다. 정지 시간이란 시뮬레이션 시간

동안 노드의 움직임이 멈춰 있는 총 시간을 말한다. 이를 통해 노드의 이동 시간에 따른 CP_AODV 알고리즘의 성능을 측정할 수 있다.

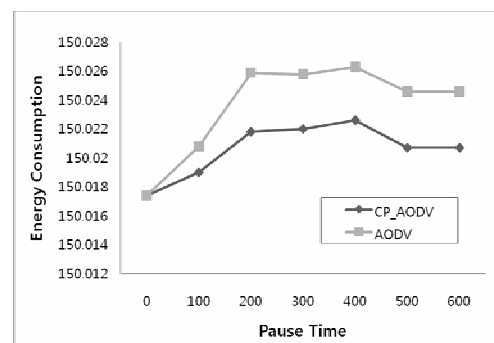
RREQ 메시지는 이 메시지를 발생시킨 노드 종류에 따라 2가지로 구분할 수 있다. 즉, 출발지 노드(source node)가 라우팅 경로를 찾기 위해 발생시킨 RREQ 생성 메시지(RREQ initiated)와 라우터 역할을 담당하는 중간 노드가 메시지 전달을 위해 발생시킨 RREQ 전달 메시지(RREQ forwarded) 등이다. (그림 1)에서 알 수 있는 것처럼, 정지 시간이 0인 경우(모든 노드가 이동할 가능성이 최대)를 제외하면 CP_AODV가 AODV에 비해 RREQ 생성 메시지 발생 빈도가 감소함을 알 수 있다.

<표 1> 시뮬레이션 파라미터 설정

파라미터	값
네트워크 크기	1,500m x 1,500m
대역폭	2MByte
노드 수	49
배치 방식	GRID
전송거리	250m
이동속도	0 ~ 10(ms)
실험 시간	600초



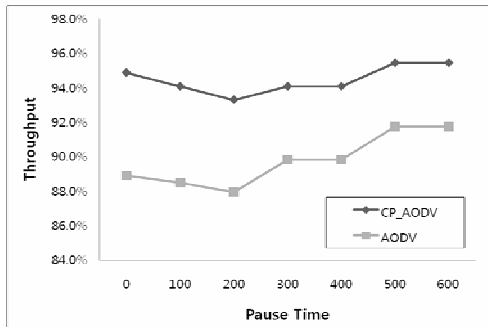
(그림 1) RREQ 생성 횟수



(그림 2) 에너지 소모율

CP_AODV의 경우 AODV에 비해 RREQ 생성 메시지 발생 빈도 및 RREQ 전달 메시지 발생 빈도 모두 각각 33% 및 28% 감소함을 알 수 있다(<표 2>). 전체적인 RREQ 메시지 발생 빈도가 감소함에 따라 자연히 목적지 노드에 도달하는 RREQ 메시지 수도 줄게 되며, 이는 RREP 패킷 및 RERR 패킷 발생에도 직간접적인 영향을 미친다. RREP 패킷 및 RERR 패킷

발생 빈도 역시 26% 및 31%씩 각각 감소하였다.



(그림 3) 전송 성능

<표 2> 시뮬레이션 결과 (평균)

비교 파라미터	AODV	CP_AODV
Number of RREQ Initiated	48	32
Number of RREQ Forwarded	3085	2234
Number of RREP Initiated as Destination	70	52
Number of RERR Initiated	16	11
Energy Consumption(mW)	0.0236	0.0206
Number of Data Packets Received	468	485
Throughput	89.8%	94.5%

전체적인 패킷 개수가 감소함에 따라 이에 비례해 네트워크의 에너지 소모율도 줄어들 것으로 예상했지만, 시뮬레이션 결과 에너지 소모율은 패킷 감소율보다 다소 적은 비율(13%)로 감소하였다 (그림2). 이는 라우팅 경로 설정을 위한 RREQ, RREP 등과 같은 제어 패킷 크기에 비해 데이터 패킷이 훨씬 크기 때문에, 컨트롤 패킷 발생 빈도 감소에 따른 에너지 소모율 저하 효과가 크게 반영되지 않은 것으로 판단된다.

그러나 전체적인 패킷 발생 빈도 감소는 무선 채널 확보 경쟁 등에 따른 무선 네트워크 부하를 경감시킴에 따라 전송 성능 개선에 유리하게 작용함을 알 수 있다. 즉, CP_AODV가 기존 AODV에 비해 데이터 패킷 수신률은 3% 개선되었으며, 이에 따라 전체적인 전송 성능(throughput)도 5% 향상되었다 (그림 3).

3. 결론

본 논문에서는 노드의 이동성으로 인해 네트워크 토폴로지가 수시로 변경되는 무선 Ad-hoc 네트워크에 적합한 라우팅 프로토콜인 CP_AODV 알고리즘을 제안하였다. 즉 AODV를 기반으로 하며, ERS 알고리즘에서 홉당 최대 전송지연시간을 의미하는 NTT를 네트워크 상황 변화에 맞게 조정하는 적응형 알고리즘을 제안하였다. 기존 연구에서는 NTT를 네트워크 상황에 맞게 조절하기 위해 유선 네트워크에서 사용되던 RTT 기반의 홉당 전송

지연시간 산정 등의 기법을 도입하였다. 그러나 본 논문에서는 홉당 평균전송시간인 CT를 구해, CT와 NTT간 차를 토대로 NTT의 증감 여부를 결정하였다. NTT 증감 범위는 10ms 범위 이내로 설정하였다. 특히 NTT는 40ms 이하로 설정할 경우 오히려 RREQ를 급격히 증가시키게 되므로, NTT의 하한값(lower bound)을 40ms로 지정하였다.

네트워크 토폴로지 변동에 따라 NTT, 즉 재전송 대기시간을 조절하면 불필요한 RREQ 패킷 발생을 막을 수 있다. RREQ 패킷 발생 빈도 조절은 RREP 및 RERR 등과 같은 제어 패킷 발생 빈도에 직간접적인 영향을 미친다. 특히 RREQ 패킷 발생 빈도를 감소시키면 에너지 소모율 저하는 물론 무선 채널 확보 경쟁 감소에 따른 수신율 개선 등 부가적인 효과를 얻을 수 있음을 실험을 통해 확인하였다.

제안된 알고리즘의 성능을 측정하기 위해 50개 미만의 노드에 대해 10m/s 의 이동성을 부여하여 실험하였다.

앞으로 인접노드간 거리 및 노드 밀집도가 CP_AODV 알고리즘 성능에 어떤 영향을 미치는지 연구해 보고자 한다.

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] 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
- [5] S.M.Tripathi 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, January 2002
- [6] 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, November., 2001
- [7] 이재석, 박혜련, 이기현, 석정봉, "Timestamp를 이용한 AODV 라우팅 알고리즘의 성능 개선", 한국통신학회논문지 Vol.31 No8A, August 2006
- [8] 김현창, 정석문, "상태공간 측정을 통한 AD HOC 네트워크의 최적화 연구", 한국군사과학기술학회지 제8권 제5호, December 2005.
- [9] <http://www.scalable-networks.com>