

# Ad Hoc 네트워크에서 Hop 수에 따른 Timestamp 적용을 이용한 AODV 프로토콜 성능 향상 기법

최재형, 신종원, 조환규  
통신구조연구실  
부산대학교 전자계산학과  
e-mail:jaehchoi@pusan.ac.kr

## Performance Improvement Technique of AODV Protocol Using Timestamp per Hop Count in Ad Hoc Networks

Jae-Hyung Choi, Jong-Won Shin, Hwan-Gue Cho  
Communication Architecture Lab  
Department of Computer Science, Pusan National University

### 요 약

Ad Hoc 네트워크에서 on demand 라우팅 프로토콜로 대표적인 AODV(Ad Hoc On Demand Distance Vector Routing)는 table driven 기반의 라우팅 프로토콜과 비교해 active route만을 라우팅 테이블에 저장하기 때문에 오버헤드가 작고 경로단절시 경로복구를 통해 경로 재설정할 수 있는 장점을 가지고 있으나 경로복구시 대역폭 낭비가 심하고 복구시간이 길다는 drawback을 가지고 있다. 이러한 단점은 많은 라우팅 패킷의 발생으로 기인하는 점이다. 본 논문에서는 Expanding Ring Search 알고리즘에서 time out을 timestamp를 이용하여 네트워크 상황에 맞게 가변적으로 적용하는 알고리즘을 개선하여 hop 수에 따라 node traversal time을 설정하도록 제안하였다. 제안한 알고리즘은 시뮬레이션을 통하여 검증하였으며 라우팅 패킷의 발생을 줄이고 throughput에서 향상을 보였다.

### 1. 서론

Ad Hoc 네트워크는 Infrastructure없이 임시 망을 구성하는 mobile node들의 집합으로 주로 군사용이나 대체 네트워크로 활용할수 있는 방향으로 연구가 진행되어 왔다[1]. 현재로는 mobile IP와 같이 IP에 이동성을 지원하기 위한 연구와 Ad Hoc네트워크에 관한 연구가 각광받고 있으며 국제기구인 IETF에 MANET WG(Mobile Ad Hoc Network Working Group)이 결성되어 다수의 라우팅 프로토콜이 draft로 제출, 표준화가 진행중이다.

Ad Hoc 환경에서의 라우팅 프로토콜로 초기에 유선환경에서의 Table Driven 기반의 알고리즘이 대두되었으나 많은 단점으로 인해 On Demand방식의 라우팅 프로토콜에 대한 연구로 방향이 바뀌게 되었

다.

본 논문에서는 Ad Hoc 네트워크와 Routing Protocol에 대해 간략하게 알아보고 On Demand 라우팅 프로토콜인 AODV(Ad Hoc On Demand Distance Vector)에 대해서 알아보고 Expanding Ring Search 알고리즘과 이에 대한 개선연구들과 함께 hop 수에 따른 가변적인 Time Out 적용을 제안하는 새로운 알고리즘을 제시한다.

### 2. 관련연구

#### 2.1 Ad Hoc Network

일반적으로 무선 네트워킹은 Infrastructure Network와 Infrastructure가 없는 Ad Hoc Network로 분류된다.

*Ad Hoc Network*는 Infrastructure가 없으므로 인해 별다른 관리가 없이도 자발적으로 네트워크를 형성하고 다른 노드들의 존재를 파악하여 다른 노드들과의 통신과 정보, 서비스의 공유를 위한 상호연동이 필요하다[1]. 각 이동 노드들이 라우팅 기능을 포함하여야 하며 변화하는 네트워크 토폴로지, 노드들의 배터리수준, 무선링크의 대역폭 제한, 비대칭링크 등에 대한 제한점이 있다.

## 2.2 Routing Protocol in *Ad Hoc Network*

*Ad Hoc* 네트워크에서의 라우팅 프로토콜은 노드들의 이동성을 지원, 관리하여야 하며 유선환경에 비해 부족한 대역폭을 효과적으로 사용할 수 있도록 하여야 한다.

현재까지 연구되고 있는 라우팅 프로토콜로는 초기에 사용된 유선환경에서 사용되고 있는 Table Driven 방식의 문제점을 해결한 On Demand 방식의 라우팅 프로토콜이 주로 연구되고 있으며 또한 이 두가지 방식의 프로토콜들의 장단점을 결합한 ZRP(Zone Routing Protocol)프로토콜[2]과 같은 Hybrid방식도 제안되었다.

### 2.2.1 Table Driven 방식 라우팅 프로토콜

Table Driven 방식은 초기에 *Ad Hoc* 네트워크에서 쓰인 라우팅 알고리즘으로 유선방식의 네트워크에서 사용되는 Bellman-Ford 방식을 *Ad Hoc*에 적용한 것이다. Table Driven 방식의 라우팅 프로토콜에서는 *Ad Hoc* 네트워크내의 모든 이동단말은 네트워크내의 모든 이동단말에 대한 라우팅 정보를 라우팅 테이블에 지속적으로 유지한다. 또한 라우팅 정보의 변경이 있을시 변경되는 자신의 라우팅 정보를 알리게 되는에 이러한 방식의 장점으로는 상시 네트워크내의 이동노드들에 대한 라우팅 정보를 유지함으로써 경로 획득 지연시간이 짧다. 그러나 주기적으로 라우팅 정보를 알리게 됨으로써 한정된 무선 대역폭을 라우팅 패킷의 잦은 발생으로 인하여 낭비할수 있다. 이를 구현한 라우팅 프로토콜로는 DSDV(Destination Sequence Distance Vector)프로토콜[3]과 라우팅 정보를 전체 이동 노드에 전달하지 않고 이웃한 이동 노드들에게만 전달하여 대역폭의 낭비를 줄이는 WRP(Wireless Routing Protocol) 프로토콜이 있으며 OLSR(Optimized Link State Routing)프로토콜, TBRPF(Topology Broadcast Based on Reverse-Path Forwarding)프로토콜과 이

동단말의 라우팅정보를 계층적으로 분류한 CGSR(Clusterhead Gateway Switch Routing)프로토콜이 연구되고 있다.

이러한 Table Driven 방식은 소규모의 *Ad Hoc* 네트워크에 유리하며 규모가 커지는 네트워크에서는 불리하다.

### 2.2.2 On Demand 방식 라우팅 프로토콜

이 방식은 Table Driven방식의 단점을 개선하기 위해 제안된 알고리즘으로 전체 노드에 대한 라우팅 테이블을 지속적으로 유지하는 것이 아니라 데이터 전송이 필요한 때에만 필요한 경로를 획득하기 위한 라우팅 경로 탐색을 수행하게 된다. 그러므로 Table Driven 방식에 비해 경로 획득시간이 길어지게 되는 문제점이 있으나 현재로서는 *Ad Hoc* 네트워크에 가장 적합한 라우팅 프로토콜로 연구되고 있다. 이를 구현한 라우팅 프로토콜로는 DSR(Dynamic Source Routing)[4], AODV(Ad Hoc On Demand Distance Vector Routing)[5], TORA(Temporally Ordered Routing Algorithm)등의 프로토콜이 있다.

## 2.3 AODV Routing Protocol

대표적인 On Demand 라우팅 프로토콜로써 AODV는 DSR과 비슷하나 DSR이 라우트 캐쉬를 유지하며 지정된 경로수 이상으로 경로정보가 들어올때만 새롭게 경신되는 점을 개선하여 유효한 시간을 설정하여 경로를 관리하며 또한 DSDV가 네트워크내의 모든 단말에 대해 경로정보를 유지하는데 반해 AODV는 경로획득절차에 따라 얻어진 경로만을 보관한다. 즉 AODV는 DSDV와 DSR의 장점을 결합하고 문제점을 해결한 프로토콜이라 할 수 있다.

AODV 프로토콜에서는 4가지 라우팅 패킷이 있는데 RREQ(Route Request), RREP(Route Reply), RERR(Route Error), RREP-ACK(Route Reply Acknowledgement)가 그것이다[5].

소스 노드는 자신의 라우팅 테이블에서 목적노드를 찾을 수 없을 경우 RREQ 패킷을 발생시키게 되는데 RREQ 패킷을 발생시키는 노드는 소스노드와 경로단절시 경로복구를 위해 작업하는 경로상의 중간노드가 될 수 있다.

RREQ 패킷을 수신한 노드는 목적노드가 자신이거나 자신의 라우팅 테이블에 목적노드가 존재할 때에는 RREP 메시지를 이용하여 소스노드에게 Unicast로 경로정보를 전달하게 되며 그렇지 않을

경우 RREQ소스노드까지의 역경로(reverse route)를 라우팅 테이블에 저장한 후 받은 RREQ패킷을 전달(forwarding)한다. 이렇게 패킷이 퍼져나가는 것을 flooding이라고 한다.

RREP-ACK는 RREQ에 대한 소스노드의 응답을 위한 패킷이다.

RERR 패킷은 경로관리를 위한 것으로 이동노드의 고장등으로 연결이 불가능하게 되었을 때나 어떤 경로의 사용이 불가할 때 해당경로에 대한 경로단절을 이웃노드들에게 알리게 된다.

### 2.4 기존 Expanding Ring Search 알고리즘 연구

Expanding Ring Search알고리즘은 유선인터넷에서 처음 사용되었으며 RREQ의 전송을 처음부터 네트워크전지역에 전송하는 것이 아니라 RREQ의 TTL을 점진적으로 증가시킴으로써 불필요한 RREQ의 발생을 방지하여 네트워크 대역폭의 낭비를 막는 것이다. MANET에서는 TTL\_START 값이 1, TTL\_INCREMENT 값이 2로 설정되어 있어 RREQ 패킷의 TTL이 1, 3, 5, 7, 35, 70으로 증가하게 된다. 이때 RREQ에 대한 RREP를 기다리는 time out은 식 (1)으로써 계산된다[5].

식 (1)

$$timeout = 2 * TTL * NODE\_TRAVERSAL\_TIME$$

여기서 NODE\_TRAVERSAL\_TIME은 40ms로 설정되는데 이러한 고정적인 한 홉 평균지연시간의 적용은 불필요한 RREQ 패킷의 발생을 야기시킬 수 있다. [6]에서는 timesatmp를 이용하여 가변적인 node traversal time을 적용함으로써 이러한 문제를 해결하려하였다. [6]에서는 RREQ 패킷의 수신시 timestamp를 이용하여 아래 식 (2)와 같이 계산한 값으로 node traversal time을 설정하였다.

식 (2)

$$\frac{RREQ_{ct} - RREQ_{st}}{RREQ_{hop} + 1}$$

RREQ<sub>ct</sub> = RREQ를 수신한 노드의 현재시간  
 RREQ<sub>st</sub> = RREQ의 timestamp의 값  
 RREQ<sub>hop</sub> = RREQ의 Hop Count값

또한 실험환경을 제공하는 ns-2의 aodv 라이브러리에 포함되는 CMU/MONARCH group에서 개발한 코드에서는 node traversal time을 30ms로 설정하고

있으며 가변적인 node traversal time을 적용하기 위하여 한 노드가 발생시킨 RREQ 패킷에 대한 RREP 패킷을 수신하였을 때 timestamp를 이용하여 node traversal time을 적용하였다.

### 3. AODV 성능향상 제안

본 논문에서 제안하는 알고리즘은 기존의 timestamp를 이용한 가변적 node traversal time적용에 확장하여 Expanding Ring Search 알고리즘에서 node traversal time의 수집을 hop 단위로 하여 보다 세밀하게 timeout을 적용하는 것을 제시한다.

기존의 timestamp를 이용한 가변적인 node traversal time의 적용에서의 문제점으로 아래그림과 같이 노드간 traversal time이 설정되어지는 경우 특정홉이내의 RREQ 패킷이 timeout되는 경우가 발생하게된다.

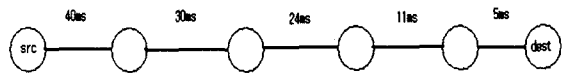


그림 (1)기존 Expanding Ring Search 알고리즘 drawback example

그림 (1)에서는 src에서 dest까지 평균 홉을 구하면 110/5=22ms가 된다. 그러나 이 node traversal time을 Expanding Ring Search에 적용시 1홉과 3홉에서 RREQ time out이 발생하게 되어 불필요한 RREQ의 재전송이 야기된다.

이러한 drawback을 제거하기위해 node traversal time을 홉수에 따라 분리하여 저장하여 Expanding Ring Search 알고리즘에서 TTL에 따라 각기 네트워크상황을 반영하는 node traversal time을 적용할 수 있게한다.

식 (3)

$$timeout(X) = 2 * TTL * NODE\_TRAVERSAL\_TIME(X)$$

본 논문에서 제안하는 timeout알고리즘은 식 (3)과 같다. NODE\_TRAVERSAL\_TIME(X)는 hop 수에 따라 node\_traversal\_time을 계산한다.

### 4. Experiment

시뮬레이션은 ns-2[7]를 이용하였고 AODV는 CMU/MONARCH group에서 개발한 코드를 사용하였다. 시뮬레이션 모델은 500m\*500m과 1000m\*1000m의 3개의 정방형 랜덤이동모델로 각각

10 node, 20 node, 30 node로 구분하였으며 802.11b 표준사양을 따른다. 시뮬레이션시간은 200초이며 각 노드의 최고속도는 10m/sec으로 설정하고, pause 시간은 0으로 설정하여 tcp연결에 대해 실험하였다.

4.1 실험 결과

실험은 ns-2 의 CMU/MONARCH group의 AODV 모듈과 본 논문에서 제안하는 알고리즘으로 각각 시뮬레이션 시나리오에서의 데이터 패킷의 처리율(throughput)과 각 라우팅 패킷의 발생을 비교하였다.

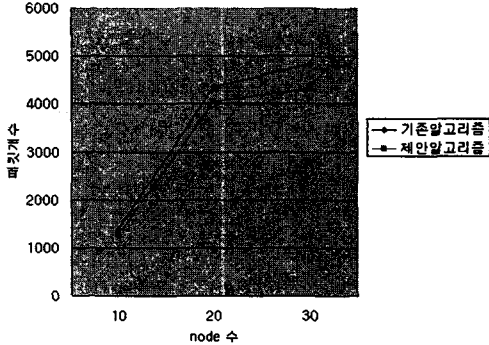


그림 (2) 라우팅 패킷 발생 개수

그림 (2)에서 기존알고리즘에 비해 라우팅 패킷의 개수는 10%내외로 줄어들음을 알 수 있다. 특히 노드 수가 증가할수록 라우팅 패킷의 발생감소 비율이 커짐을 알 수 있었다.

반면 처리율에서는 예상과는 달리 3%내외의 편차를 보여 크게 차이가 나지 않았다.

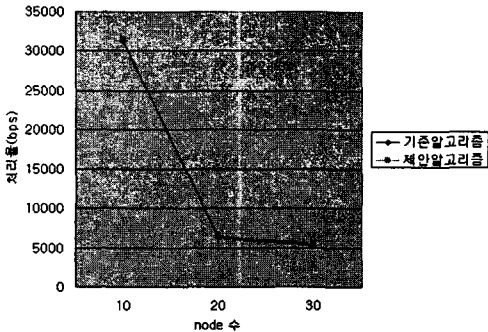


그림 (3) 처리율

5. Conclusion

본 논문에서는 AODV에서 node traversal time을 함수에 따라 가변적으로 적용하는 방법을 제안하였다. 기존의 방법은 node traversal time을 40ms로

고정시켜 사용하거나 RREQ 패킷과 RREP 패킷의 수신시 timestamp를 이용하여 평균홉당 지연시간을 가변적으로 적용시켰다. 여기서 발생하는 평균홉당 지연시간이 부적절하게 적용되는 경우를 해소하고자 홉수에 따라 node traversal time을 적용하는 방법을 적용하여 RREQ 패킷의 time out 시간을 함수에 따라 달리 적용할 수 있어 라우팅 패킷의 발생을 줄이고 throughput에서 향상을 보였다.

향후 연구되어야 할 사항을 정리하면 가변적인 timeout의 적용과 노드의 이동성과의 상관관계에 대한 연구와 실제 라우팅 패킷이 줄어들었으나 처리율이 그만큼 향상되지 않았던 점에 대해서는 고찰할 필요가 있겠다.

참고문헌

[1]C.-K. Toh, "Ad Hoc Mobile Wireless Networks Protocols and Systems", Prentice Hall PTR, 2002, pp.13-25, pp.39-56  
 [2]Z. J. Haas and M. R. Perlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", Internet Draft,IETF MANET Working Group, draft-ietf-manet-zone-03.txt, March 2000.  
 [3]C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV)for Mobile Computers", Computer Communication, October 1994, pp.234-244.  
 [4]D. B. Johnson, D. A. Maltz, Yih-Chun Hu , "The Dynamic Source Routing Protocol for Mobile AdHoc Networks (DSR)", Internet Draft, IETF MANET Working Group,draft-ietf-manet-dsr-09.txt,4, 2002.  
 [5]Samir R.Das & Elizabeth M.Belding-Royer &C.E.Perkins "Ad hoc on-demand Distance Vector (AODV)Routing for Mobile Ad HocNetworks",Internet Draft,IETF MANET Working Group,draft-ietf-manet-aodv-13.txt,IETF Internet draft,2,2003,Work in progress  
 [6] Kim Hyun Chang, "Optimization of AODV Protocol with Timestamp in Ad Hoc Networks",12,2001  
 [7] The Network Simulator-NS2, <http://www.isi.edu/nsnam/ns>