

## Central Node를 이용한 MANET 라우팅 프로토콜에 관한 연구

김희수<sup>†</sup>

### A Study on Routing Protocol using Central Node for Ad hoc Network

Hee-Su Kim<sup>†</sup>

#### Abstract

Ad hoc network는 무선 노드들의 집합으로서 어떤 인프라스트럭처 도움 없이 그들 서로가 multi-hop 경로를 통해 통신한다. 본 논문에서는 ad hoc network에서 사용되는 proactive 라우팅 프로토콜과 on-demand 라우팅 프로토콜의 혼합인 hybrid 라우팅 프로토콜에 대해 제안하였다. 본 논문에서는 기존의 hybrid 라우팅 프로토콜인 ZRP와는 달리 Ad hoc network를 구성하는 노드들 중에 네트워크 서비스를 제공해주는 특별한 노드를 설정하여 라우팅 하는 방법을 제안한다. 이러한 역할을 해주는 특별한 노드를 본 논문에서는 C-Node라 부른다. C-Node를 이용한 라우팅으로 기존의 라우팅 프로토콜보다 경로 설정 시간과 flooding 시간을 줄이므로서 효율적인 라우팅을 수행할 수 있게된다.

#### Key words :

### 1. 서 론

Ad hoc network에서 네트워크 안에 존재하는 노드들은 제한된 무선 전송 범위 때문에 서로에게 패킷 전송을 도와준다. 어떤 송신 노드로부터 목적 노드까지의 네트워크 경로는 송신자부터 목적지까지의 multi-hop 경로를 생성하기 위해 패킷을 전달해주는 많은 중간 노드가 필요하다. 그리고 배터리를 사용하므로 에너지의 공급이 일정치 않다는 특성을 갖는다<sup>[1,2]</sup>. Ad hoc network는 중앙식 관리나 기지국이나 access point 같은 고정된 네트워크 인프라스트럭처가 필요하지 않고 필요할 때 빠르고 낮은 비용으로 구성할 수 있다. 이러한 특징 때문에 ad hoc network는 긴급 구조 상황, 전쟁 상황, 법 집행 과정, 상업용과 교육용 그리고 유선 컴퓨팅이 불가능한 경우에 이용되고 있다<sup>[3]</sup>.

Ad hoc network의 형태가 수시로 변화하는 특성 때문에 최적화된 경로를 찾고 유지하는 것은 매우 중요하며 어려운 일이다. 그래서 ad hoc 네트워크에 사용하는 라우팅 알고리즘은 네트워크의 형태 변화에 민첩하게 대응할 수 있어야만 한다. 그리고 두 단말간의 통

신을 위해서는 중간 단말을 통해 중계기의 역할을 수행하는 멀티 홉 경로 설정이 필요하다. 현재 ad hoc 네트워크에서 사용되는 라우팅 프로토콜은 크게 proactive 프로토콜과 on-demand 프로토콜이 사용된다<sup>[4, 5]</sup>.

Proactive 프로토콜(table-driven)은 각각의 이동 노드가 네트워크 내의 모든 경로를 유지하고 있기 때문에 경로의 요구시 최적의 경로를 설정 할 수 있다. 그러나 최신의 라우팅 정보를 유지하기 위해 제어 패킷을 통해 주기적으로 정보를 갱신해야 하므로 실제 전달하는 데이터 외에 많은 양의 트래픽을 유발하는 단점을 가지고 있다<sup>[8]</sup>. Proactive 프로토콜의 단점을 보완하기 위해 on-demand 프로토콜에서는 특정 목적지에 대한 경로를 요구하였을 때만 경로 설정을 수행한다. 설정된 경로는 목적지 노드에게 더 이상의 요구 패킷이 없거나, 목적지까지의 경로를 사용할 수 없을 때까지 경로 유지는 계속 수행된다. 이 프로토콜의 단점은 경로 설정 과정에서 지연이 발생을 한다는 점과 홉 거리에 따른 관점에서 볼 때 최적의 경로를 보장하지 못한다는 점이다<sup>[6, 7]</sup>. 이러한 table-driven 프로토콜과 on-demand 프로토콜의 혼합 프로토콜인 Zone Routing Protocol(ZRP)이 있다. ZRP는 전체 네트워크를 몇 개의 zone으로 구성하여 이것을 기반으로 한 계층적 라우팅 방식을 이용하여 table-driven 절차의 범위를 이웃 노드 범위로 제한한다. 한편 전역적인 검색은 모든 노드들에 질의하는

광주보건대학 병원전산과(Dept. of Hospital Information Management, Gwangju Health College)

<sup>†</sup>Corresponding author: hskim@ghe.ac.kr

(Received : August 12, 2008, Accepted : December 6, 2008)

대신에 선택된 노드에게만 질의함으로써 효율적으로 수행된다. 따라서 네트워크 내에 새로운 링크가 생성되면 table-driven 프로토콜은 위상 변화가 네트워크 전체에 영향을 미치지만 ZRP는 변화가 발생한 이웃 노드들에게만 영향을 미친다.

본 논문에서는 네트워크 영역 또는 존을 기반으로 하기보다는 특정 노드를 중심으로 한 라우팅 방법을 제안하였다. 본 논문에서 네트워크를 구성하는 일반 노드들을 위해 서비스를 해주는 노드를 C-Node라 부른다. 이러한 C-Node와 일반 노드들 사이의 경로는 양방향으로 항상 유지되고, 일반 노드들 사이의 경로는 on-demand 방식으로 설정된다. 그리고 C-Node에서는 경로 설정 및 유지를 위해 DSR을 이용하게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 라우팅 프로토콜에 대하여 알아보고, 3장에서는 제안한 라우팅 방법에 대하여 설명한다. 4장에서는 제안한 라우팅 프로토콜의 성능을 평가하고, 마지막으로 5장에서는 결론을 맺는다.

## 2. 관련 연구

### 2.1. Table-driven 라우팅 프로토콜

Ad hoc network를 구성하는 모든 노드는 주기적으로 라우팅 정보를 다른 노드들에게 전달하고, 라우팅 경로 변경시는 자신의 라우팅 정보를 방송하여 다른 노드들의 라우팅 테이블 갱신을 유도한다. 이 프로토콜은 다른 노드들에 대한 주기적인 라우팅 정보를 유지함으로써 전송 필요시 별도의 경로 획득 절차 없이 라우팅 테이블의 정보를 이용하여 전송하므로 경로 획득 지연시간이 짧은 장점이 있다. 그러나 노드들이 빈번하게 이동하는 ad hoc network에서 다른 노드들의 라우팅 테이블 갱신을 위한 라우팅 정보를 방송해야 하므로 라우팅 패킷으로 인한 오버헤드가 발생하여 성능저하에 많은 영향을 미치는 단점이 있다. 따라서 이 방식은 노드의 숫자가 적은 소규모 ad hoc 네트워크에 적합하나, 중대형 네트워크에서는 많은 단점을 가지고 있다. 이러한 라우팅 프로토콜로는 DSDV(Destination-Sequence Distance Vector) 라우팅 프로토콜, 라우팅 정보를 이웃한 이웃 노드에게만 전파하여 오버헤드를 감소시킨 WRP(Wireless Routing Protocol)과 DSDV 라우팅 프로토콜에서 라우팅 정보를 감소시키기 위하여 이동 노드들을 계층적으로 분류한 CGSR(Clusterhead Gateway Switch Routing) 프로토콜이 있다. 그러나 Table-driven 방식은 ad hoc 환경에 적용하기에는 많은 문제점이 존재하여 자체로는 각광 받지 못하였고 이를

개선한 알고리즘으로 발전하게 되었다.

### 2.2. On-demand 라우팅 프로토콜

이 라우팅 프로토콜은 table-driven의 단점을 해결하기 위하여 모든 노드가 빈번하게 이동하는 ad hoc network 환경에 적합하도록 제안된 프로토콜로서 네트워크 내의 모든 이동 노드에 대한 전체 경로를 항상 유지하는 것이 아니라 데이터 전송 필요시에만 경로 획득 절차를 수행한다. 따라서 주기적인 라우팅 정보 방송과 이동시 변경된 라우팅 정보를 방송할 필요가 없으므로 라우팅 패킷 오버헤드를 줄이는 장점이 있다. 그러나 데이터 전송시 경로 획득시간이 길어져 실시간 통신에 부적합한 문제점을 가지고 있다.

#### 2.2.1. AODV

AODV는 망을 구성하는 노드가 빈번히 이동함으로써 링크의 위상이 자주 바뀌는 환경에서 사용되기 위한 라우팅 프로토콜로 네트워크에 참여하는 각 노드에 의해 운용되어지며, 경로 배정에 대한 요구가 없을 때에는 동작하지 않는다. 따라서 링크의 변화를 적절한 시간 내에 감지할 수 있는 메커니즘이 제공되어야 한다[9]. Distant-Vector routing 프로토콜은 어느 한 노드의 링크가 끊겨졌을 때 여러 노드가 이 정보를 잘못 전달하여 해당 경로에 대한 무한 루프가 발생하는 문제점을 가지고 있다<sup>[10]</sup>. 이를 보완하기 위해 AODV는 Destination sequence number 필드를 추가하였다. Destination sequence number는 경로 정보가 어떤 것이 최신인지 구별하기 위한 필드로 목적지의 노드가 하나의 경로 요청 메시지를 받을 때마다 이에 대한 응답 메시지에 그 순서 번호를 증가시켜 기입하는 것이다. 이에 의해 목적지에 대한 경로를 요청한 노드와 중간 노드들은 경로 정보간에 어떤 것이 가장 최신 것인지 판단할 수 있게 된다.

AODV는 Route Request(RREQ)와 Route Reply(RREP) 두 가지가 있다. 이 메시지는 일반적인 IP 헤더를 사용하여 UDP 기반으로 처리되며 RREQ의 전송 범위는 IP 헤더의 TTL에 의해 지정된다. 목적지로의 경로를 찾기 위해 소스는 RREQ를 전파한다. RREQ는 목적지에 도착하거나 목적지에 관한 최근의 경로 정보를 가지고 있는 중간 노드에 도달 할 때까지 계속 전파된다. AODV는 RREP가 RREQ가 전달되어 온 경로를 거꾸로 이용하기 때문에 양방향 링크들만을 사용한다. RREP가 소스 노드로 되돌아갈 때 경로상의 노드들은 자신들의 테이블에 경로를 넣는다. 발견된 경로 정보를 테이블에 유지할 때 AODV는 타이머를 사용한다.

그림 1은 AODV의 경로 설정 방법을 보여준다. 경로를 초기에 요구하는 노드가 A일 때 RREQ를 방송하게 되고, 이에 대한 응답을 할 수 없으면 이 메시지는 계속 전달되게 된다. 목적지 노드 또는 목적지까지의 경로를 알고 있는 노드에 의해 RREP가 unicast back되면 그 경로를 알려지게 된다.

2.2.2. DSR

DSR은 데이터를 보내는 노드가 경로 요청을 할 때, 그 노드는 소스 라우팅을 행하여 목적지까지 hop-by-hop 경로 정보를 완벽하게 알고 있다는 것이다. 또한 이 정보가 경로 캐쉬에 저장되어 다음 전송 요구 시 사용되며 데이터 패킷은 패킷 헤더에 소스 경로를 가지고 있다. 즉, 소스 라우팅을 사용한다는 것이다. 그리고 DSR은 모든 노드가 자기 자신을 root로 하는 shortest tree 형태를 유지한다. DSR은 경로 탐색과 경로 유지의 두 단계로 이루어진다. DSR의 경로 설정 방법은 그림 2와 같다.

소스 노드가 패킷을 목적지에 보낼 경우 먼저 캐쉬에 경로가 저장되어 있는지를 확인한다. 만약 경로가 존재한다면, 그 경로를 사용하여 패킷을 전송하고, 그렇지 않다면 RREQ를 망에 전송하므로써 경로 탐색 절차를 실행시킨다. RREQ는 소스와 목적지의 주소 그리고 독자적인 숫자를 포함하고 있다. 경로 탐색 절차 중 각 중간 노드는 목적지로의 경로를 알고 있는지 확인하고 이를 알지 못한다면 RREQ의 route record 항에 이를 저장한다. 만약 RREQ가 목적지에 도달하게 되면 이 패킷은 route record를 가지는 RREP로 바뀌게 된다. 그와 달리 RREQ가 목적지로의 경로 정보를 가지는 중간 노드에 도착하게 되면 그 노드는 자신의 경로 캐쉬에서 목적지까지의 경로를 찾아 RREP의 route record에 추가한다. RREP를 다시 소스에게 보내기 위해서는 RREP 생성 노드는 반드시 스스로의 경로를 가지고 있어야만 하고 만약 캐쉬에 소스까지의 경로가 저장되어 있다면 캐쉬에 저장되어 있는 경로를 사용한다.

3. 제안한 방법

본 논문에서는 C-Node에 on-demand 라우팅 프로토콜을 채택하기 위해 다음과 같은 내용을 수정하였다. 첫째로 C-Node는 새로운 이웃과 경로를 설정하기 위해 Hello 메시지를 보내어 자신의 존재를 알리고 이웃 노드들은 링크 계층 통보없이 C-Node에 여전히 도달할 수 있는지를 찾을 수 있다. 둘째로 C-Node에

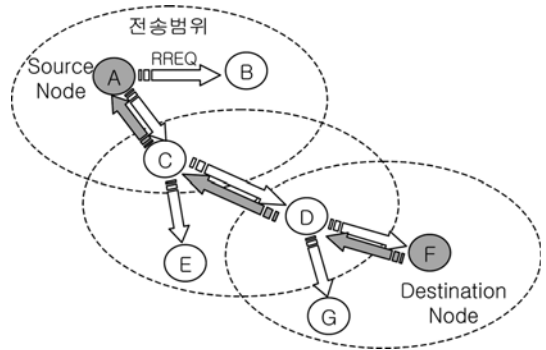
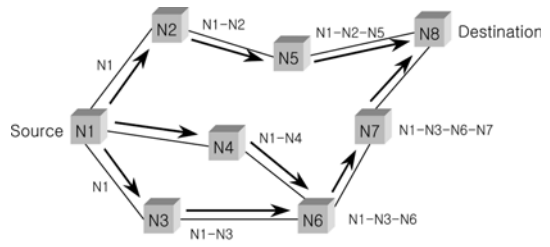
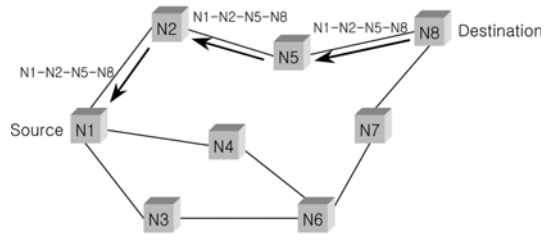


그림 1. AODV의 경로 설정 방법



(a) 경로 설정을 위한 라우트 레코드 생성



(b) 라우트 레코드를 갖는 라우트 응답 전달

그림 2. DSR의 경로 설정 방법

경로를 추가하기 위해서 이웃 노드들에게 updates 메시지를 송신하므로써 새로운 경로를 설정할 수 있다. 셋째로는 RERR과 RREQ는 C-Node의 트래픽 존재 여부와 상관없이 C-Node에서 생성할 수 있다. 그림 3은 C-Node가 하나인 ad hoc network를 보여주고 있다.

3.1. C-Node 탐색

C-Node에서는 자신의 존재를 이웃 노드들에게 통보하기 위해 가장 높은 sequence number를 갖는 Hello 메시지를 보낸다. 노드들은 Hello 메시지를 수신하였을 때, 자신의 이웃에 C-Node가 존재함을 알 수 있게된다. 만약 노드가 일정한 시간동안에 Hello 메시지 패킷을

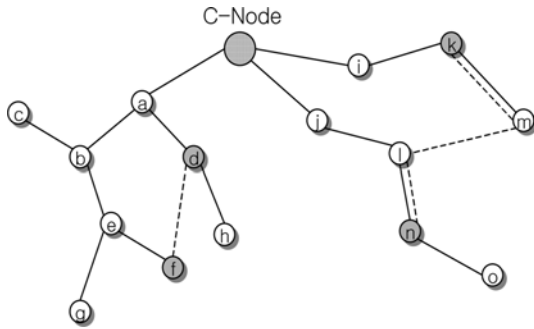


그림 3. C-Node가 하나인 ad hoc 네트워크

수신하지 못한다면 라우팅 계층에서는 이웃노드와의 연결이 끊어진 것으로 선언한다. MAC 계층은 전달한 데이터 패킷을 수신할 수 없을 때 라우팅 계층에게 연결 실패를 통보한다. 이러한 데이터 트래픽에서 이와 같은 메커니즘은 연결 실패를 보다 빠르게 검출할 수 있다.

### 3.2. 경로 설정 및 유지

그림 4에서는 경로의 설정을 보여주고 있다. 그림 4에서 노드 c가 C-Node를 알고 있을 때 C-Node에게 방송을 한다. 그러므로 노드 b는 C-Node에 대해 알게 되고 역시 이웃 노드 e에 대해서도 알게 된다. 노드 b가 재방송을 하게 될 때 노드 a는 노드 b와 노드 c 그리고 C-Node의 경로에 대해 알게 될 것이다. 이와 같은 방법으로 노드 b는 노드 d로부터 교체된 경로를 알 것이다. 그러나 b는 가장 짧은 경로인 노드 c를 통한 경로를 선택할 것이다. 만약 노드 b와 노드 c의 연결이 실패한다면 노드 b는 노드 d를 통한 C-Node로 경로를 바꿀 것이다. 이동 노드로부터 C-Node를 향하는 아래 쪽 노드는 모든 위쪽 노드에 관해서 알지 못하지만 부모 노드에 대해서는 알 것이다.

노드 b가 방송할 때, 노드 a가 중요한 목적지 노드가 아니기 때문에 노드 a에 연결하기 위해 방송하지는 않을 것이다. 이와 같은 경우에 노드 c는 단지 노드 b에 관해서만 알 것이다. 비슷한 방법으로 노드 a와 노드 b에 대해서는 모르지만 노드 c에 관해서만 알 것이다. 노드 a에게 패킷을 보내기 위하여 C-Node는 노드 a를 위해 query를 송신할 것이다. 네트워크 내의 모든 이동 노드를 위해 C-Node로부터 query가 초기화되는 것을 막기 위해 다음의 메커니즘은 어떤 외부 제어 메시지 없이 역경로의 설정을 채택하였다. 노드로부터 C-Node를 향해 데이터 패킷을 송신할 때, C-Node를 향

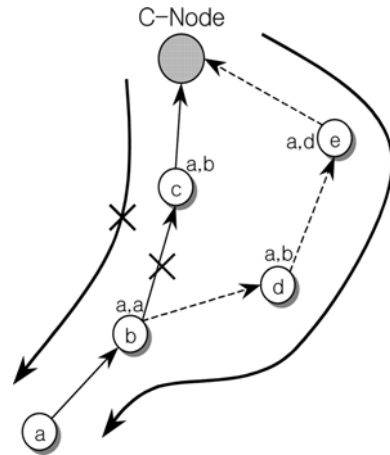


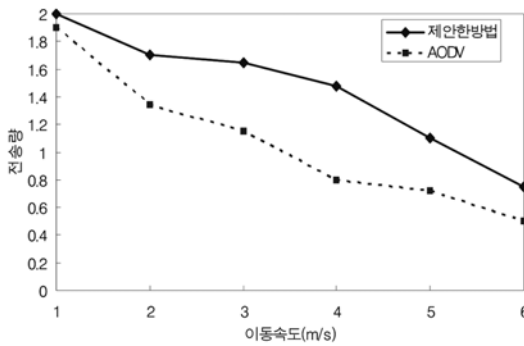
그림 4. C-Node와 일반 노드와의 경로 설정

한 경로에 있는 중간 노드들은 데이터 패킷의 소스 노드를 향해 경로를 설정할 수 있다.

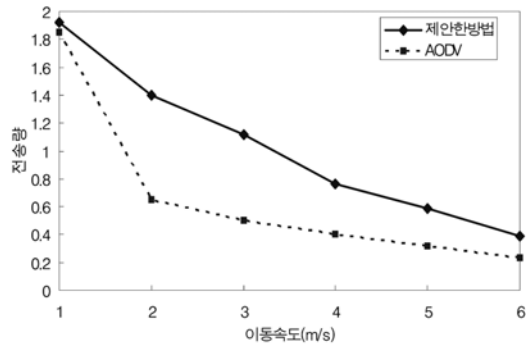
## 4. 실험 및 결과

이 장에서는 본 논문에서 제안한 C-Node를 이용한 hybrid 라우팅 프로토콜에 대해 ns2 시뮬레이터를 이용하여 성능을 평가하였다. 성능 평가의 목표는 한정된 망의 크기와 노드의 수 사이에서 전송 부하에 따른 성능 평가이다.

30개의 노드가 400m×400m 크기의 지역 내에서 랜덤하게 지역을 이동하게 하며, 노드의 이동 속도는 0m/s~10m/s이고 pause time은 30초로 하였다. 각 프로토콜과 네트워크 크기 그리고 노드의 속도에 대해 10번의 실험을 반복하였고, 매 실험마다 서로 다른 초기 네트워크 설정을 하였다. 그리고 시뮬레이션에는 300초의 시간이 주어졌다. 데이터 전송률은 2Mb/sec 이고 패킷의 크기는 64 bytes로 고정하였다. 그림 5에서는 본 논문에서 제안한 방법과 AODV와 노드 수에 따른 패킷의 전송량을 보여준다. 그림에서도 알수 있듯이 기존의 라우팅 프로토콜과 본 논문에서 제안한 방법의 차이는 노드의 수가 많고 노드들의 이동 속도가 증가함에 따라 더욱더 심해짐을 볼 수 있다. 그림 6은 DSR과 제안한 방법의 패킷 전송량을 비교하여 보여주고 있다. 그림 6도 그림 5와 마찬가지로 제안한 방법이 패킷 전송률이 우수함을 보여주고 있고 DSR보다는 AODV가 나은 성능을 보인 것을 확인할 수 있었다. 그림 7에서는 제안한 방법과 DSR 그리고 AODV의 전송 평균 지연 시간을 보여주고 있다.

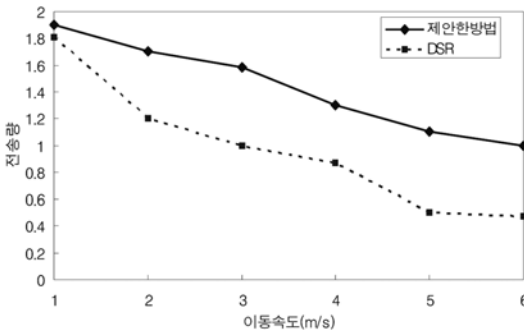


(a) 노드가 10개인 경우

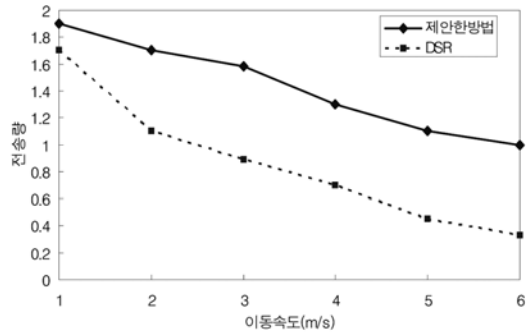


(b) 노드가 30개인 경우

그림 5. AODV와 제안한 방법의 노드 수에 따른 패킷 전송량 비교



(a) 노드가 10개인 경우



(b) 노드가 30개인 경우

그림 6. DSR과 제안한 방법의 노드 수에 따른 패킷 전송량 비교

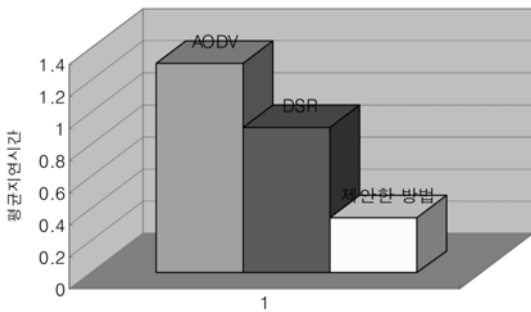


그림 7. 전송 평균 지연 시간 비교

C-Node를 이용하여 라우팅을 하므로써 경로 설정 시간과 데이터 전송률과 지연 시간을 줄이므로써 효율적인 라우팅을 수행할 수 있었다. 그리고 여러 개의 C-Node 사이의 경로 연결 및 유지는 anycast query를 보냄으로써 보다 효율적으로 경로를 설정하고 유지할 수 있다.

향후 연구로는 여러 개의 C-Node로 구성된 대형 네트워크에서 C-Node 사이의 향상된 anycast 기술이 연구되어야 할 것이다.

### 5. 결 론

본 논문에서는 기존의 hybrid 라우팅 알고리즘과는 다른 ad hoc 네트워크를 구성하는 노드들 중에 네트워크 서비스를 제공하는 특별한 노드를 선택하여 그 노드를 중심으로 라우팅하는 방법을 제안하였다. 이러한

### 참고문헌

- [1] Charles E. Perkins, Ad hoc Networking, Addison Wesley, 2001.
- [2] J. Broch, D. A. Maltz, D. B. Johnson, Y-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proceedings of the 4th International Conference on Mobile Computing and Networking, pp 85-97,

- October 1998.
- [3] C. K. Toh, "Ad Hoc Network Wireless Networks", Protocols and System, Prentice Hall PTR, 2002.
  - [4] J. Broch et. al., "A performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", in proc. ACM Mobicom98, Dallas, TX, October 1998.
  - [5] S. Basagni, I. Chlamtac, and A. Farago, "A Generalized Clustering Algorithm for Peer-to-Peer Networks", Workshop on Algorithmic Aspects of Communication ICALP Bologna, Italy, July 1997.
  - [6] P. Tsuchiya, "The Landmark Hierarchy: A New Hierarchy for Routing in Very Network", in ACM Sigcomm, 1988.
  - [7] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks", Mobile Computing, 1994.
  - [8] S. Roy and J.J. Gracia Luna Aceves, "Using Minimal Source Trees for On-Demand Routing in Ad Hoc Networks", in IEEE Infocom, Anchorage, Alaska, 2001.
  - [9] J. Broch et. al., "A performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", in proc. ACM Mobicom98, Dallas, TX, October 1998.
  - [10] Charles E. Perkins, E. M. Royer, "Ad hoc On-Demand Distance Vector Routing".