

애드 혹 네트워크에서 부분적인 경로 수정을 이용한 라우팅 프로토콜 설계

김지성[○] 서효중

가톨릭대학교 컴퓨터공학화

kingut80@gmail.com, hjsuh@catholic.ac.kr

Design of Routing Protocol for using Local Path Modify in Ad-hoc Network

Jiseong Kim[○] Hyojoong Suh

Dept of Computer Science and Engineering, The Catholic University of Korea

요 약

본 논문에서는 한정된 에너지를 사용하는 애드 혹 네트워크에서 부분적인 경로 수정을 통해 에너지 사용을 분산함으로써 네트워크의 수명을 증가시키는 Local Path Modify Routing (LPMR) 프로토콜을 제안한다. 애드 혹 네트워크에서 노드의 한정된 에너지는 전체 네트워크 수명에 큰 영향을 끼친다. 따라서 최근 에너지를 효율적으로 사용하기 위한 On-demand 방식의 프로토콜들이 많이 제안되고 있다. 그 중 Power-aware source Routing (PSR)은 에너지의 사용량에 따라 경로를 주기적으로 변경해 줌으로써 네트워크의 수명을 늘린다. 하지만 잦은 플래딩으로 인해 라우팅의 오버헤드의 증가와 이 오버헤드로 인한 에너지의 소모가 증가하는 문제점이 있다. 본 논문에서는 에너지의 사용이 많은 노드를 선별하여 각 노드의 주변 노드 정보를 통해 부분적으로 경로를 변경해 줌으로써 잦은 플래딩 없이 경로를 변경함으로써 라우팅 오버헤드가 감소하게 된다. 또한 에너지를 많이 소모하는 노드의 에너지를 분산해 줌으로써 각 노드들이 에너지를 고르게 사용함으로써 네트워크의 수명도 증가하게 된다. 본 논문에서는 NS-2(Network Simulator-2)를 이용하여 LPMR의 성능을 검증한다.

1. 서 론

애드 혹 네트워크는 무선통신 능력을 갖춘 노드들이 직접적인 통신 또는 멀티 홉 방식을 통해 각 노드들이 서로 패킷을 주고받는 형태의 네트워크이다. 애드 혹 네트워크를 구성하는 노드들은 한정된 용량의 배터리를 사용하기 때문에 배터리의 용량이 모두 소진되게 되면 정상적인 라우팅 동작을 할 수 없게 되어 네트워크에서 탈퇴하게 되고 이는 전체 네트워크의 성능저하로 이어지게 된다. 따라서 본 논문에서는 이러한 제한된 배터리의 용량을 사용하는 노드들로 구성된 애드 혹 네트워크를 최대한 효율적으로 동작시키기 위해서 각 노드들이 에너지를 균등하게 소모하여 각각의 노드의 수명을 늘리는 방식의 라우팅 프로토콜을 제안한다.

애드 혹 라우팅 프로토콜은 Proactive (Table-Driven)[1] 프로토콜 방식과 Reactive(On-Demand)[1] 프로토콜 방식, 그리고 둘을 혼합한 Hybrid[1] 방식이 있다. Proactive 프로토콜 방식은 유선네트워크와 비슷하다. 네트워크가 처음 시작 될 때 각 노드가 하나 혹은 그 이상의 테이블에 라우팅 정보를 유지하여 이러한 정보를 이용해 바로 네트워크 통신이 가능하여 딜레이가 없다는 장점이 있지만 네트워크의 크기가 클 경우 테이블을 유지하기 위한 비용이 많이 든다는 단점이 있다.

Reactive 프로토콜은 라우팅 테이블을 유지하는 대신 플래딩을 통해 목적지까지의 경로를 찾아 데이터를 주고받는 방식을 취하고 있다. 하지만 Reactive 프로토콜 경우 노드의 경로를 찾기 위한 플래딩 발생으로 인해 라우팅 오버헤드증가와 이로 인한 딜레이가 발생한다는 단점을 가지고 있다. Hybrid 프로토콜은 Proactive와 Reactive 프로토콜의 장점만을 취하기 위해 전체 네트워크의 소규모의 그룹으로 나누어 부분적으로 테이블을 유지하는 방식이다. 하지만 Hybrid 네트워크에서만 효율적으로 운영되어지는 단점을 가지고 있다.

Ad-hoc 네트워크의 lifetime을 늘리기 위한 방법은 대부분 On-Demand 방식을 적용하여 연구되고 있으며 경로 설정 시 배터리의 잔량을 고려하여 경로를 선택하고 주기적인 모니터링을 통하여 배터리가 많은 쪽으로 경로를 수정하는 방식을 취하고 있다.

본 논문에서는 On-Demand 방식을 적용하여 부분적인 경로 수정을 통해 이 방식의 PSR의 단점으로 지적된 잦은 플래딩으로 인한 라우팅 오버헤드를 줄이고 각 경로가 집중된 노드의 에너지의 사용을 분산함으로써 에너지 효율적인 LPMR(Local Path Modify Routing) 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구를 통해 기존에 연구되었던 On-demand 방식을 사용한 프로토콜들의 단점을 분석하고, 3장에서 LPMR의 동작 및 특성에 대해 자세히 살펴본다. 그리고 4장에서는 NS-2[5]를 통해 LPMR에서의 네트워크 수명 및 라우팅 오버헤드를 기존 On-demand 방식의 프로토콜들과 비교한다. 5장에서는 결론 및 향후 연구 방향에 대하여 논한다.

2. 관련 연구

본 장에서는 On-demand 형태를 취하는 LPMR 프로토콜의 이해를 돕기 위해 기존의 프로토콜들을 살펴보고 그들의 단점을 분석한다.

2.1 Dynamic Source Routing

DSR[2]은 On-demand 형태 중 가장 간단한 대표적인 방식이다. source에서 destination 까지 경로를 찾기 위해 source의 주소를 담아 route request(RREQ) 패킷을 브로드 캐스트 하고 이후 목적지 node가 RREQ 패킷을 받게 되면 route reply(RREP) 패킷을 통해 경로의 생성을 source 노드에 알리게 된다. 이때 각 경로의 node들은 cache형태의 table에 경로를 저장하여 지정된 경로를 통해 데이터 패킷을 전달하는 방식이다. DSR의 경우 각 노드가 주기적으로 링크의 단절을 감시하여 단절이 발생할 경우 이를 source 노드에 알려 경로를 재설정하게 된다. DSR의 경우 가장 큰 단점은 링크가 단절이 된 후에야 경로를 변경한다는 것과 경로를 설정하기 위한 route request 패킷에 대한 아무런 제약이 없으므로 경로를 찾는 오버헤드가 크다는 것이다.

2.2 Ad hoc On-demand Distance Vector Routing

AODV[3]는 DSR의 라우팅 오버헤드 및 보다 더 효율적인 경로 설정을 위해 source 와 destination간의 최단 경로를 찾는 라우팅 프로토콜이다. AODV의 경우 라우팅 비용에 거리를 계산하여 RREQ 패킷에를 추가하고 목적지가 아닌 중간의 노드가 받았을 경우 만약 새로 받은 RREQ가 이전의 받은 RREQ보다 더 긴 경로를 가지고 있다 새로 받은 패킷을 무시해 버림으로써 라우팅 오버헤드를 감소시켰다. 하지만 AODV도 링크가 단절된 후에 경로를 재설정하는 단점을 가지고 있다.

2.3 Power-aware Source Routing

PSR[4]은 위에서 제시한 프로토콜들의 단점인 링크의 단절을 극복하기 위해 RREQ 패킷에 각 노드의 에너지 지 비용을 추가하였다.

$$\text{where } C_i(t) = p_i \left(\frac{F_i}{R_i} \right)^a \quad (1)$$

p_i : transmit power of node i

F_i : full-charge battery capacity of node i

R_i : remaining battery capacity of node i at time t
 a : a positive battery weight factor

그리고 이렇게 산출된 라우팅 비용의 합(2)으로 전체 경로로 라우팅 비용을 계산 하여 이 값이 가장 작은 경로를 선택하게 된다.

$$C(r, t) = \sum_{i \in r} C_i(t) \quad (2)$$

최초 경로 설정 시 설정된 라우팅 비용과 주기적으로 모니터링 통해서 얻은 경로 값을 비교하여 일정한 값을 초과 하였을 경우에 경로를 재탐색 하여 네트워크의 수명을 높인다. PSR의 경우 Source에서 Destination까지 주기적인 메시지를 보내 라우팅 비용을 모니터링 하기 때문에 많은 flooding 발생으로 인한 라우팅 오버헤드가 발생하게 된다. 또한 설정 된 모든 노드의 라우팅 비용을 합하여 비용이 낮은 값을 선택하기 때문에 비용이 갑자기 높아진 특정한 하나의 노드가 발생하였을 경우 모든 경로를 다시 재탐색하여 바꾸어 주어야 하기 때문에 라우팅 오버헤드가 발생하게 된다.

3. Local Path Modify Routing(LMPR) 프로토콜

LMPR은 경로 설정 단계, 부분적인 경로 변경 단계, 전체 경로 변경 단계의 세 부분으로 구성된다. 이 절에서는 먼저 LMPR이 사용한 routing table 및 neighbor 테이블의 저장 형태를 언급하고 각각의 단계별로 상세히 살펴본다.

3.1 routing table 및 neighbor table

네트워크내의 각각의 노드는 routing table과 neighbor table을 가지게 된다. 우선 routing table은 표1과 같은 필드로 구성되어 있다.

표1 Routing table

dst	next	energy	hop count	b_id	refresh
-----	------	--------	-----------	------	---------

dst는 경로의 목적지 이며 next는 목적지 까지 가기 위한 다음 경로를 나타낸다. energy, hop_count, broadcast id는 경로를 설정하기 위해 필요한 필드 이고 refresh는 전체 경로 변경 시 필요한 필드 이므로 경로 설정 단계와 전체 경로 변경 단계에서 자세히 언급한다.

neighbor table은 자신의 주변 노드를 저장하는 공간으로 네트워크가 시작할 때 HELLO 패킷을 통해 자신의 주변에 어떠한 노드가 있는지를 파악하여 주소를 저장하게 된다.

3.2 경로설정 단계

source 노드에서 데이터를 목적지 까지 보내고자 한다면 source 노드 표2의 Header를 갖는 RREQ 패킷을 브로드캐스트 하게 된다.

표2 RREQ Header

hop count	지나온 경로 수
broadcast id	각 RREQ 패킷의 고유번호
destination	목적지 주소
source	출발지 주소
energy	각 노드의 energy잔량 합
now id	RREQ를 받은 노드 id 혹은 source id

목적지가 아닌 노드가 처음 RREQ 패킷을 수신하게 되면 다음과 같은 형태로 routing table에 저장한다.

- table : dst ← RREQ packet : source
- table : next ← RREQ packet : now id
- table : energy ← RREQ packet : energy
- table : b_id ← RREQ packet : broadcast id
- table : hop count ← RREQ packet : hop_count

그리고 만약 자신의 routing table을 검사하여 이전에 동일한 source 와 동일한 broadcast id를 가지는 RREQ 패킷을 받았는지를 검사하고 만일 같은 RREQ 패킷을 받은 적이 있다면 hop count와 energy를 비교하여 hop count 가 작거나 같고 energy가 더 많을 때만 table을 갱신한다. 만약 위의 조건에 맞지 않는다면 받은 RREQ 패킷은 무시하게 된다. RREQ를 처음 받거나 table 갱신되었다면 now id에 자신의 주소와 energy에는 자신의 에너지 잔량을 더하여 2로 나누어준 값을 변경한 후 다시 브로드캐스트 하게 된다. 이때 에너지 잔량의 계산은 PSR의 $C_i(t)$ 로 계산한다.

목적지 노드가 RREQ 패킷을 처음 수신하게 되면 받은 패킷의 energy 와 source 그리고 now id를 저장한 후 일정시간을 기다리게 된다. 일정시간 안에 도달된 패킷들의 energy를 비교하여 경로의 energy 가장 많은 쪽을 선택하여 출발지를 다시 목적지로 삼아 RREP 패킷을 보내게 된다. RREP 패킷을 받은 노드들은 routing table의 dst 와 next를 저장하고 목적지를 확인해 다음 경로로 RREP 패킷을 전송해 줌으로써 경로설정을 완료 되게 된다.

3.3 부분적인 경로 변경 단계

네트워크의 중심에 있는 노드들은 경로에 많이 포함되게 되어 에너지를 많이 소비하게 될 것이다. 따라서 본 논문에서는 이러한 과부하가 걸리는 노드를 찾기 위해 각 노드는 일정시간 동안 노드의 에너지 변화를 주기적으로 측정하고 수신 받은 유효한 데이터 패킷을 source 와 destination별로 count 하여 가장 패킷을 많이 보낸 source to destination의 경로를 부분적으로 변경하였다.

$$C_i(t) - C_i(t_0) > d$$

$t - t_0$: 에너지 변화 측정 시간

d : threshold value

부분적인 경로 변경 과정은 neighbor table을 통한 변경과 Local RREQ 패킷을 통한 변경으로 나눌 수 있다. 처음 경로 변경은 neighbor table을 통해 경로를 변경하고 이 방법이 실패할 경우 Local RREQ 패킷을 이용하여 경로를 변경한다.

3.3.1 neighbor table을 통한 경로 변경

과부하가 걸리는 노드로 선정이 되면 변경해야 할 경로의 source 노드를 가지고 routing table의 검색을 통해 이전 노드를 검색하여 이전 노드에게 변경해야 목적지, 출발지 주소와 목적지까지 가기위한 Next 주소 그리고 자신의 id 와 $C_i(t)$ 를 담은 REPAIR 패킷을 보내주게 된다. REPAIR 패킷을 받은 노드는 먼저 Neighbor Search 패킷을 브로드캐스트 하여 주변 노드의 neighbor table에 과부하가 걸리는 노드로부터 받은 Next의 주소가 있는 지를 찾아보게 된다. 이때 만약 과부하가 걸리는 노드가 Neighbor Search 패킷을 받으면 패킷을 무시된다.

Neighbor Search 패킷을 받은 노드들 중 자신의 주변에 Next의 주소를 가진 노드가 있다면 출발지와 목적지 노드의 주소를 통해 routing table을 갱신하고 자신의 energy를 담아 ACK 패킷을 보내게 된다. ACK 패킷을 받은 노드는 패킷 중 가장 에너지가 많은 노드를 선택하여 자신의 routing table을 갱신하게 되는데 노드 선택시 과부하가 걸리는 노드보다 energy가 작은 노드는 선택에서 제외되고 ACK 패킷을 보낸 모든 노드가 과부하가 걸리는 노드보다 에너지 적다면 경로는 변경되지 않는다. 노드를 선택 했다면 과부하가 걸리는 노드에게 새로 설정된 노드의 id를 담은 RESPOND 패킷을 보낸다. RESPOND 패킷을 받은 노드는 자신의 routing table에서 변경된 경로를 삭제하고 next 노드에게 경로가 변경되었다고 알려 줌으로써 Neighbor Search를 통한 부분적인 경로 변경이 완료 된다.

그림 1과 2는 Neighbor Search를 통한 경로 변경 과정을 보여주는 그림이다.

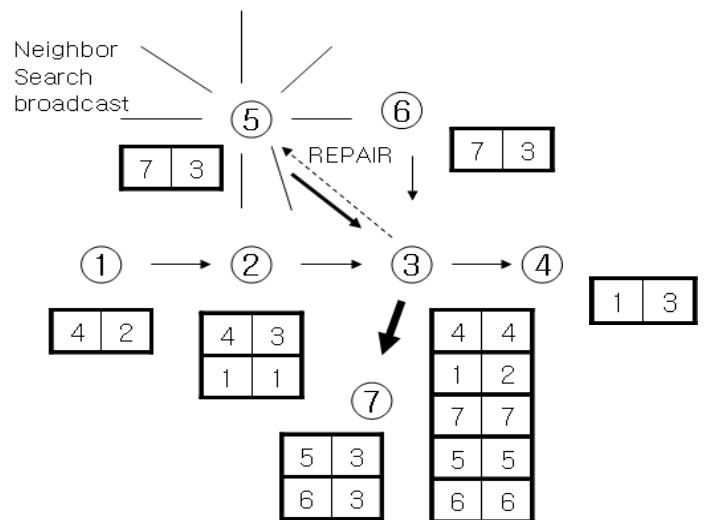


그림 1 Neighbor search를 통한 경로 재설정

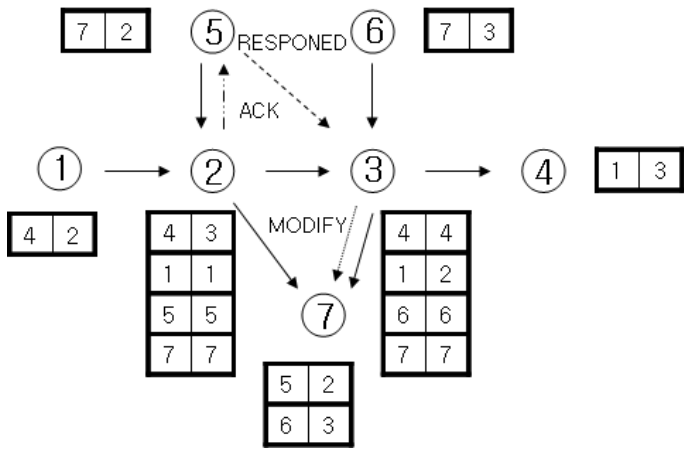


그림 2 경로 재설정 후의 테이블

3.3.2 Local RREQ를 통한 경로 변경

만약 neighbor search를 통한 경로 변경이 일정 시간 내에 이루어지 않을 경우 REPAIR 패킷을 받은 노드는 기존 RREQ Header에 real destination 필드와 avoid 필드를 추가 한 Local RREQ 패킷을 브로드캐스트 하여 자신부터 Next까지의 경로를 재설정하게 된다. 이때 Local RREQ 패킷에는 표3과 같이 정보를 담아 보낸다.

표3 Local RREQ Header

hop count	1
broadcast id	각 RREQ 패킷의 고유번호
destination	Next id
source	REPAIR 패킷의 출발지 주소
energy	현재 노드 에너지 잔량
now id	현재 노드 id
real destination	REPAIR 패킷의 목적지 주소
avoid	과부하가 걸리는 노드의 id

Local RREQ를 통한 경로 변경 과정은 경로 설정 과정과 동일하며 RREP 받은 노드는 real destination을 이용하여 routing table을 갱신하게 된다. 그리고 만약 일정 시간 안에 RREP 패킷을 받지 못할 경우 경로는 변경되지 않는다.

3.4 전체 경로 변경 단계

부분적으로 경로를 변경할 경우 발생할 수 있는 문제점은 주변 2 hop 혹은 3 hop 안에서 경로를 계속적으로 변경하기 때문에 네트워크의 밀집도가 낮을 경우 PSR의 방식보다 에너지 효율이 떨어질 수 있다는 단점을 가지고 있다. 따라서 본 논문에서 제시하는 LMPR 프로토콜은 각 노드의 에너지의 잔량이 threshold 값 밑으로 떨어지게 되면 이를 source 노드에 알려 전체적인 경로를 재수정하는 방법을 제시한다. 라우팅 오버헤드를 줄이기

위해 한 중간 경로에 있는 노드가 source 노드의 대해서 한번씩만 전체경로를 변경하였다. 따라서 에너지가 threshold 값 밑으로 떨어지게 되면 데이터 패킷을 받을 때 source 노드의 주소를 사용해 routing table을 검색하고 그 source 노드에 대해 refresh 필드를 체크한 후 전체 경로를 수정하기 위한 REFRESH 패킷을 sources 노드에 보내게 된다. 이 때 REFRESH 패킷을 받은 노드들은 자신이 source가 아니라면 source id를 이용해 routing table의 refresh 필드를 체크하면서 source 노드로 패킷을 전달하게 된다. 만약 경로를 재설정할 때 REFRESH 패킷을 보낸 노드가 포함되어도 경로는 바뀌지 않고 새로 설정된 경로를 통해 데이터 패킷을 계속적으로 송수신 한다.

위의 각 단계에서 살펴 본 바와 같이 LPMR 프로토콜의 방식을 사용할 경우 PSR의 패킷 플래딩을 통해 모니터링을 하는 대신 노드 자체의 모니터링을 통해 재설정 시기를 결정함으로써 경로 수정을 위한 플래딩 오버헤드가 발생하지 않으며 부분적인 경로 수정을 통해 경로 재설정에 대한 오버헤드도 줄어들게 된다. 또한 과부하가 걸리는 노드의 에너지 소비를 주기적으로 분산해 줌으로써 전체 네트워크가 고르게 에너지를 쓰게 됨으로 네트워크의 수명도 늘어나게 된다.

4. 성능 평가

4.1 실험 환경

본 논문에서 제안하는 라우팅 프로토콜의 성능을 평가하기 위해 ns-2 시뮬레이터를 사용하여 라우팅 프로토콜을 작성하여 ns-2 에 삽입 후 실험을 수행하였다. 표4는 본 논문에서 사용한 실험 환경을 보여주고 있다.

표4 ns-2 실험 환경

전파도달 범위	250m
전송계층 프로토콜	UDP
응용계층 Traffic	CBR(230byte)
Transmission decrement	패킷 길이 $\times 2 \times 10^{-4}$
Receive Decrement	패킷 길이 $\times 2 \times 10^{-4}$
Initial Energy	100

전체 네트워크의 트래픽은 지정된 10개의 source 노드에서 동시에 랜덤하게 설정된 목적지 노드로 각각 200s 동안 0.05s 마다 패킷을 계속적으로 보내게 하여

source node의 에너지가 고갈될 때까지 수행하였다. 본 실험은 라우팅 오버헤드와 네트워크의 수명에 대해서 수행하였으며 라우팅 오버헤드는 라우팅 트래픽의 총량을 수신된 데이터의 총량으로 나누어 계산하였으며 네트워크의 수명은 최초로 탈퇴되는 노드가 발생한 시간으로 정하였다. 표5와 표6은 네트워크의 노드 수와 밀도에 따라 구성된 네트워크 환경을 보여준다.

표 5 노드 수에 따른 네트워크 환경

노드 수	네트워크면적	노드 배치
20	2500m ²	uniform(0,350)
30	2500m ²	uniform(0,400)
40	2500m ²	uniform(0,450)
50	2500m ²	uniform(0,500)

표 6 밀도에 따른 네트워크 환경

노드 수	네트워크면적	노드 배치
50	2500m ²	uniform(0,500)
50	3600m ²	uniform(0,600)
50	4900m ²	uniform(0,700)
50	6400m ²	uniform(0,800)

4.2 성능 평가

본 논문에서 제시하는 라우팅 프로토콜의 성능을 평가하기 위해 2장에서 언급한 DSR, AODV, PSR 과 함께 실험을 실행하였다. 그림 3, 4, 5, 6은 ns-2를 통해 수와 밀도에 따른 네트워크의 수명과 라우팅 오버헤드를 나타낸 그래프이다.

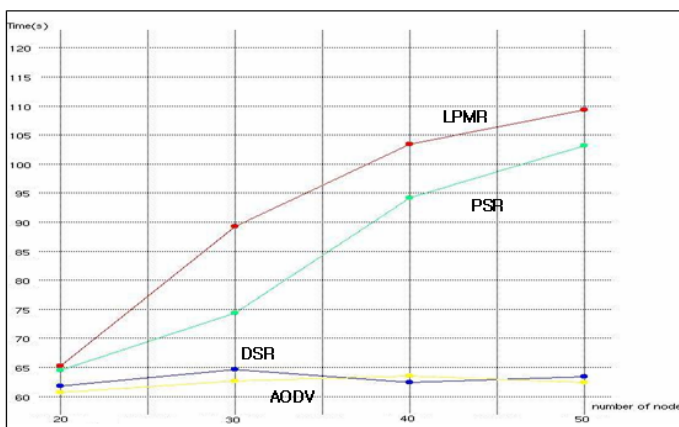


그림 3 노드 개수에 따른 네트워크 수명

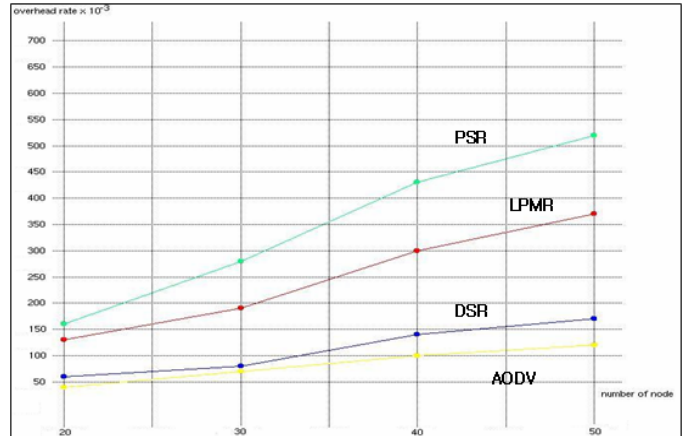


그림 4 노드 개수에 따른 라우팅 오버헤드

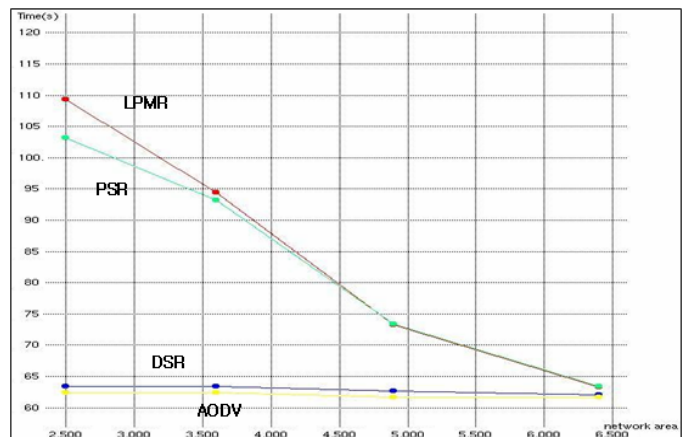


그림 5 노드 밀도에 따른 네트워크 수명

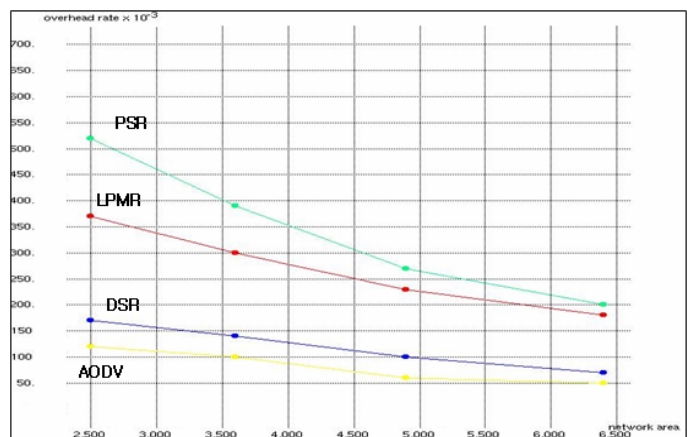


그림 6 노드 밀도에 따른 라우팅 오버헤드

네트워크의 수명 측면에서 살펴볼 때 노드 수가 많을수록 그리고 밀도가 높을수록 source에서 destination까지 가는 경로의 가지 수가 많아짐으로 경로의 변경이 용이하여 보다 높은 네트워크 life time의 얻을 수 있었으며 부분적으로 경로를 수정하는 LPMR의 경우 PSR 보다 부분적으로 더 자주 경로를 변경하기 때문에 더 높은 네트워크 수명을 가질 수 있었다. 하지만 밀도의 경우 네트워크의 크기가 커질수록 과부하가 거리는 노드가 적어짐

과 동시에 부분적이 경로 수정이 용의하지 않아 밀도가 작아질수록 PSR과 거의 동일한 수명시간을 나타내는 것으로 분석되었다. 그리고 한번 설정된 경로를 계속 유지하는 DSR과 AODV의 경우 노드의 수나 밀도에 관계없이 거의 일정한 수명을 나타내는 것을 볼 수 있다.

네트워크의 라우팅 오버헤드 측면에서 살펴보면 전체 경로 수정이 아닌 부분적인 경로수정을 통해 경로를 유지한 LPMR이 더 작은 라우팅 오버헤드를 나타내는 것을 볼 수 있으며 노드의 수나 밀도에서도 거의 비슷한 라우팅 오버헤드를 관찰할 수 있다. 그리고 경로가 단절 될 때 경로를 변경하는 DSR과 AODV의 경우 경로를 주기적으로 변경하는 PSR과 LPMR보다 훨씬 적은 라우팅 오버헤드를 나타내는 것을 볼 수 있다.

5. 결론 및 향후 연구

본 논문에서는 라우팅 오버헤드 감소 및 네트워크의 수명을 늘리기 위한 라우팅 프로토콜을 제안하였다.

에너지를 라우팅 비용으로 하여 에너지가 많은 노드들을 경로로 설정하고 일정시간 후 과부하 생기는 노드에 대해서 부분적으로 경로를 수정함으로 라우팅 오버헤드 감소 방안을 제안하였다. 또한 각 노드의 에너지 사용을 균등하게 함으로 보다 네트워크 전체의 에너지를 효율적으로 활용함으로써 네트워크의 수명을 높였다. 그리고 ns-2 시뮬레이션을 이용한 실험을 통해 네트워크 수명 연장이라는 동일한 목적을 가지는 PSR 보다 라우팅 오버헤드 및 네트워크 수명 측면에서 LPMR이 보다 효율적이라는 것을 증명하였다.

본 논문에서는 라우팅 비용 산정 시 거리에 따른 전송 파워를 고려하지 않았기 때문에 모든 노드가 에너지를 주고를 받을 때 패킷의 길이에 따라서만 에너지를 소모하였다. 따라서 앞으로의 연구에서는 거리에 따라서 달라지는 전송파워를 고려하여 보다 현실적이 라우팅 알고리즘을 위해 연구할 것이다.

6. 참고 문헌

- [1] A review of current routing protocols for ad hoc mobile wireless networks Royer, E.M.; ChaiKeo-ng Toh; Personal Communications, IEEE [see also IEEE Wireless Communications] Volume 6, Issue 2, April 1999 Page(s):46 - 55
- [2] D.B. Johnson and D.A Maltz, "Mobile Computing," Kluwer Academic Publishers, 1999.
- [3] C. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing", Proceedings of the 2nd IEEE workshop on mobile computing systems and applications, pp90-100, Feb. 1999.
- [4] M. Malekei, K. Dantu, and M. Pedram, "Power-aware source routing protocol for mobile ad hoc network", Proceedings of the IEEE international symposium on Modeling and Optimizaion in Mobile, Ad Hoc, and Wireless Network, pp.305-310, April 2005.
- [5] ns-2 simulator, <http://www.isi.edu/nsnam/ns>