

# 이동 애드혹 네트워크에서의 부하 균등화 및 공평성 지원 방법

안 상 현\* · 유 영 환\*\* · 임 유 진\*\*\*

## 요 약

AODV(Ad Hoc On-Demand Distance Vector)나 DSR(Dynamic Source Routing)과 같은 대부분의 애드혹 라우팅 프로토콜은 네트워크 토폴로지가 변하지 않는 한 새로운 경로 설정을 하지 않는다. 따라서 노드의 이동성이 적은 경우, 트래픽은 몇몇 노드에게 집중될 수 있으며 그 결과로 해당 노드들의 혼잡으로 인한 긴 종단간 지연이 발생할 수 있다. 게다가 몇몇 특정 노드들만 오랫동안 계속해서 사용됨에 따라 그들의 배터리가 빠르게 소모될 것이다. 노드의 배터리 방전은 노드 사이의 연결 단절뿐만 아니라 동시에 새로운 경로 설정을 위하여 많은 route request 패킷(RREQ)을 생성시킨다. 본 논문에서는 각 노드가 자신의 트래픽 부하 상태를 기반으로 RREQ 패킷을 무시하거나 패킷 포워딩을 포기할 수 있게 함으로써 트래픽 집중화 문제를 해결할 수 있는 SLA(Simple Load-balancing Approach)를 제안한다. 그러나 어떤 이기적인 노드들은 자신의 에너지를 절약하기 위하여 고의적으로 패킷 포워딩을 거부할 수도 있다. 이러한 노드들을 패킷 포워딩에 참여시키기 위하여 본 논문에서는 PIFA(Protocol-Independent Fairness Algorithm) 보상 방식을 제안하였다. SLA 성능의 우수함을 증명하기 위하여 SLA를 채택한 AODV 방식과 SLA를 채택하지 않은 AODV 방식을 비교 실험하였고, 이를 통하여 SLA를 채택한 방식이 트래픽 부하 분산에 효율적이며 이로 인하여 전체 애드혹 네트워크 성능 또한 향상시킴을 확인하였다.

## Load-Balancing and Fairness Support Mechanisms in Mobile Ad Hoc Networks

Sanghyun Ahn\* · Younghwan Yoo\*\* · Yujin Lim\*\*\*

### ABSTRACT

Most ad hoc routing protocols such as AODV (Ad Hoc On Demand Distance Vector) and DSR (Dynamic Source Routing) do not try to search for new routes if the network topology does not change. Hence, with low node mobility, traffic may be concentrated on several nodes, which results in long end-to-end delay due to congestion at the nodes. Furthermore, since some specific nodes are continuously used for long duration, their battery power may be rapidly exhausted. Expiration of nodes causes connections traversing the nodes to be disrupted and makes many routing requests be generated at the same time. Therefore, we propose a load balancing approach called Simple Load-balancing Approach (SLA), which resolves the traffic concentration problem by allowing each node to drop RREQ (Route Request Packet) or to give up packet forwarding depending on its own traffic load. Meanwhile, mobile nodes may deliberately give up forwarding packets to save their own energy. To make nodes volunteer in packet forwarding, we also suggest a payment scheme called Protocol-Independent Fairness Algorithm (PIFA) for packet forwarding. To evaluate the performance of SLA, we compare two cases where AODV employs SLA or not. Simulation results show that SLA can distribute traffic load well and improve performance of entire ad-hoc networks.

키워드 : 애드혹(Ad-Hoc), 라우팅(Routing), 부하 균등(Load-Balancing), Credit 기반 지불 방식(Credit-Based Payment Scheme)

### 1. 서 론

MANET(Mobile Ad hoc Network)은 고정된 통신 기반 구조나 기지국을 요구하지 않는 무선 네트워크이다. 애드혹 네트워크 노드들은 자유롭게 이동할 수 있으므로 네트워크 토폴로지는 시간의 흐름에 따라 변화한다. 게다가 제한된 데이터 전송범위, 낮은 대역폭, 높은 에러율, 그리고 제한된 배터리 용량을 가진 무선 채널의 특징들은 애드혹 네트워크 환경에서의 라우팅을 더욱 어렵게 만든다. 가장 대표적인

애드혹 라우팅 프로토콜로는 AODV(Ad Hoc On-Demand Distance Vector)[1]와 DSR(Dynamic Source Routing)[2]을 들 수 있다. AODV는 RREQ(route request) 패킷의 브로드캐스팅을 통하여 새로운 경로를 설정하며 각 목적지 노드에 대하여 오직 하나의 경로만을 유지한다. 반면, DSR은 소스 라우팅 방식을 기반으로 동작하며 각 목적지 노드에 대하여 하나 이상의 경로 정보를 유지한다. 그러므로 현재 사용중인 경로가 더 이상 유효하지 않은 경우, DSR은 새로운 경로 설정을 위한 경로 설정 매커니즘을 시작하는 대신 자신이 유지하고 있던 다른 경로 정보를 사용하여 이를 대체한다.

AODV와 DSR의 성능비교 결과[3]는 노드 이동성이 감소함에 따라 패킷 전송 비율은 증가하고 라우팅 오버헤드는 감소함을 보이고 있다. 흥미로운 사실은 노드 이동성이 감

\* 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음.  
(KRF 2003 041 D00501)

† 정 회 원 : 서울시립대학교 컴퓨터과학부 교수(corresponding author)

\*\* 성 회 원 : University of Cincinnati, ECECS Department, Post Doc.

\*\*\* 성 회 원 : 수원대학교 정보미디어학과 교수

논문접수 : 2004년 10월 25일, 심사완료 : 2004년 11월 15일

소함에 따라 패킷 전송 지연이 증가한다는 것이다. AODV나 DSR은 현재 사용중인 경로가 유효하다면 새로운 경로 설정을 하지 않으므로 노드의 이동성이 적은 환경에서는 현재 사용중인 경로가 오랜 기간 동안 사용될 확률이 높다. 이는 몇몇 특정 노드로의 트래픽 집중을 야기시킬 수 있으며 이로 인하여 높은 전송 지연시간뿐만 아니라 해당 노드들의 배터리를 빠르게 고갈시키는 결과를 초래할 수 있다. 이러한 배터리 고갈은 다른 노드들의 전송 파워를 증가시키고 결과적으로 트래픽이 잘 분산된 경우에 비하여 애드혹 네트워크의 수명을 상대적으로 짧게 만든다.

본 논문은 애드혹 네트워크 환경에서의 부하 균등을 고려한 새로운 프로토콜인 SLA(Simple Load-balancing Approach)를 제안한다. SLA에서의 부하 균등의 개념은 기존 유선 망에서의 그것과는 다르다. 유선망에서의 부하 균등은 전체 네트워크 성능향상을 위한 혼잡상태 해결이 목적인 반면, SLA는 노드의 이동성이 낮은 환경에서 몇몇 노드로 트래픽이 집중되는 것을 방지함으로써 노드의 배터리 수명을 향상시키고자 한다. AODV나 DSR은 현재 사용 중인 경로가 유효한 경우 새로운 경로 설정을 하지 않으므로 노드의 이동성이 낮은 환경에서의 이러한 특성은 현재 사용 중인 경로 노드들을 혼잡상태에 빠지게 할 수 있다. 그러므로 SLA는 각 노드가 자신의 부하 상태를 기반으로 패킷 포워딩을 명시적으로 포기하거나 또는 단순히 RREQ 패킷을 무시할 수 있으며, 이를 통하여 전체 네트워크에 트래픽을 균등하게 분산시킬 수 있으며 모든 MANET 노드들이 자신들의 에너지를 공평하게 소비하게 함으로써 네트워크 전체 수명을 연장시킬 수 있다.

SLA는 새롭게 제안된 독립적인 라우팅 프로토콜이 아니라 AODV나 DSR과 같은 기존의 라우팅 프로토콜과 함께 동작하는 프로토콜이다. SLA는 모든 MANET 노드들이 다른 노드의 패킷을 성실하게 포워딩 시켜주는 안전한 애드혹 네트워크를 가정한다. 그러나 사실상 어떤 이기적인 노드들은 적당한 보상이 주어지지 않는 한 자신의 에너지를 절약하기 위하여 의도적으로 패킷 포워딩을 거부할 수도 있다. 따라서 본 논문은 노드들을 패킷 포워딩에 자발적으로 참여시키기 위한 PIFA(Protocol-Independent Fairness Algorithm)를 추가로 제안하였다. 비록 PPM이나 PTM[4], 그리고 Sprite[5]가 같은 목적을 위하여 제안된 방식이라 하더라도, 이러한 방식들은 소스에서 목적지 노드까지의 전체 경로 정보를 요구하므로 오직 DSR 프로토콜과 함께 사용될 수 있다. 그러나 PIFA 방식은 모든 종류의 라우팅 프로토콜과 함께 사용될 수 있는 장점을 가진다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 부하 균등 방식과 균등한 배터리 소모 방식에 대하여 살펴본다. 3장과 4장에서는 SLA와 PIFA 방식을 설명하고, 5장은 SLA 방식의 성능 평가 결과를 보인다. 마지막으로 6장에서 결론을 내린다.

## 2. 관련 연구

LBAR(Load-Balanced Ad hoc Routing)[6]과 DLAR(Dynamic

Load-Aware Routing)[7]은 MANET 환경에서의 부하 균등을 고려한 방식이다. 이 방식들은 네트워크 트래픽 상태를 확인하여 최소의 부하를 가지는 경로를 찾아낸다. LBAR은 각 노드를 관통하는 경로의 수와 각 이웃 노드들의 활동성의 합계, 즉, 노드 활동성과 트래픽 간섭을 고려하여 최소 부하 경로를 설정한다. DLAR은 중간 노드들에 버퍼링 되어 있는 패킷 수를 경로 선택의 기준으로 삼는다. LBAR은 소스 노드가 설정 메시지를 브로드캐스팅하고 중간 노드들이 자신의 노드 활동성을 기반으로 이를 업데이트 함으로써 경로 비용을 계산한다. 이와 비슷하게 DLAR도 목적지 노드로 RREQ 메시지가 진행되면서 버퍼링 되어 있는 패킷의 수나 부하 정보를 수집한다.

이와 같은 방식들은 공동의 목표를 위하여 모든 MANET 노드들이 잘 협력한다는 가정을 하고 있다. 그러나 어떤 연구는 이러한 가정에 의문을 제시하였다. 이기적인 노드들은 자신의 에너지 사용을 절약하기 위하여 다른 노드의 패킷을 포워딩 하지 않을 수도 있다는 것이다. 이러한 가능성을 제거하기 위하여 PPM(Packet Purse Model)과 PTM(Packet Trade Model)[4]은 *nugget*라는 가상 화폐를 사용하여 패킷 포워딩을 보상한다. Sprite(a Simple, cheat-Proof, cRedIT-based systEm)[5] 또한 패킷을 포워딩 한 노드에게 *credit*으로 보상한다. 조작할 수 없는 하드웨어를 기반으로 하는 PPM이나 PTM과 달리 Sprite는 CCS(Credit Clearance Service)라 불리는 *credit* 관리 서버를 사용하여 속임수를 방지하는 안전 알고리즘을 제안하였다. 그러나 Sprite는 모든 메시지의 수신을 CCS에게 알려야 하기 때문에 MANET 환경에서는 많은 오버헤드를 발생시킨다. 또한 PPM, PTM, 그리고 Sprite 모두 소스에서 목적지 노드까지의 전체 경로 정보를 알아야 하기 때문에 오직 DSR 프로토콜에만 적용 가능하다.

## 3. Simple Load-balancing Approach(SLA)

AODV나 DSR은 네트워크 토폴로지가 변할 때마다 새로운 경로를 설정하므로 MANET 노드가 빠르게 이동한다면 자동적으로 부하 균등의 효과를 얻을 수 있다. 반면 노드 이동성이 낮은 환경에서는 AODV나 DSR이 새로운 경로를 설정할 필요가 없으므로 오랫동안 같은 경로가 사용될 것이며 이는 해당 경로에 심각한 혼잡상태를 유발시키고 결과적으로 네트워크 성능을 저하시키게 된다.

이러한 문제를 해결하기 위하여 본 논문에서는 기존 라우팅 프로토콜에 추가적으로 구현될 수 있는 SLA(Simple Load-balancing Approach)를 제안하였다. 각 노드는 독립적으로 트래픽의 집중 정도를 판단하게 된다. 각 노드는 정기적으로 전체 네트워크 트래픽에 대한 자신의 트래픽 비율이 두 개의 경계치( $\tau_l, \tau_h, \tau_l \leq \tau_h$ )를 초과하는지 여부를 확인한 후, 그 결과에 따라 다음과 같은 3가지 상태 중 하나가 된다(여기서 경계치는 임의의 값으로 설정된다). 정상 상태, 수동 부하 균등 상태, 능동 부하 균등 상태. 정상 상태 노드는 RREQ를 브로드캐스팅하고 데이터 패킷을 포워딩하는

일반적인 MANET 노드와 똑같이 동작한다. 반면 수동 또는 능동 부하 균등 상태의 노드는 자신의 트래픽 부하를 분산 시키기 위해 노력한다.

노드  $i$ 의 트래픽 부하 비율  $L_i$ 는 다음과 같은 3가지 방식 (M1, M2, M3) 중 하나에 의해 결정될 수 있다.  $E_i$ 와  $P_i$ 는 노드  $i$ 의 현재 에너지와 버퍼링된 패킷 수를 나타내며,  $E$ 와  $P$ 는 전체 네트워크에 대한 값이다. 전체 네트워크의  $E$ 와  $P$ 값을 계산하는 방법은 부록에서 자세히 설명한다.

- M1 : 큐에 버퍼링된 패킷 수만 고려

$$L_i = \frac{P_i}{P}$$

- M2 : 남은 배터리 양에 대한 버퍼링된 패킷 수의 비율을 고려

$$L_i = \frac{P_i/E_i}{P/E}$$

- M3 : 버퍼링된 패킷을 모두 포워딩 한 후 남은 배터리 양을 고려

$$L_i = \frac{R}{R_i} = \frac{E - f \times P}{E_i - f \times P_i}$$

$f$ 는 한 패킷을 포워딩 하는데 요구되는 에너지

M1은 매우 간단하지만 노드의 에너지 정보를 전혀 고려하지 않는 반면, M2와 M3는 노드의 에너지 정보를 고려한다. M2가 M3보다 간단하지만, M3가 <표 1>에서 보는 바와 같이 부하 균등 측면에서 보다 효율적이다. 비록 노드 A가 노드 B보다 2배 많은 패킷을 가지고 있다 하더라도 노드 A의 에너지가 5배 많으므로, 노드의 에너지 고갈 속도의 균형을 맞추기 위해서는 노드 A가 좀더 많은 패킷을 처리하는 것이 바람직하다.  $L_i$ (M2)와  $L_i$ (M3)를 비교해보면, M3가 보다 확실하게 이러한 점을 반영하는 것을 확인할 수 있다. 반대로 M1은 노드의 에너지 상태를 고려하지 않으므로 노드 B가 노드 A보다 많은 패킷을 처리해야 한다는 잘못된 판단을 내리고 있는 것을 볼 수 있다.

<표 1> M1, M2, M3에 의한 트래픽 부하 비율

	$P_i$	$E_i$	$R_i$	$L_i$ (M1)	$L_i$ (M2)	$L_i$ (M3)
node A	100	1000	800	1.33	0.8	0.56
node B	50	200	100	0.67	2	4.5
average	75	600	450			

(Required energy to forward one packet,  $f=2$ )

### 3.1 IGN\_REQ(Ignore Route Requests)

평균 네트워크 트래픽 부하에 대한 노드 자신의 트래픽 부하 비율이  $\tau$ 에서  $\tau_1$ 사이일 때, 노드  $i$ 는 IGN\_REQ라 불리는 수동 부하 균등 상태가 된다. 이 상태하에서는  $L_i$ 가  $\tau_1$  이하 일 때까지 다른 노드로부터의 RREQ 패킷은 무시된다. 결과적으로 노드  $i$ 를 통과하는 새로운 경로는 설정되지 않으므로써 노드  $i$ 를 통과하는 트래픽은 결국 감소하게 된다.

### 3.2 GIVE\_UP(Give Up Packet Forwarding)

노드의 트래픽 부하가 평균보다  $\tau_2$ 이상으로 과부하 되었을 때, 해당 노드는 자신의 상태를 GIVE\_UP이라 불리는 능동 부하 균등 상태로 바뀐다. 이 상태하에서의 노드는 단순히 RREQ 패킷을 무시하는 것뿐만 아니라 일부 패킷의 포워딩을 포기한다. GIVE\_UP 상태에서의 노드  $i$ 는 자신의 상태가 GIVE\_UP으로 변경된 이후 자신을 통과하는 첫 번째 패킷의 소스 노드에게 GIVE\_UP 메시지를 전송한다. GIVE\_UP 메시지에는 소스 노드 ID와 노드  $i$ 를 지나가는 패킷들의 목적지 목록이 포함된다. 소스 노드가 해당 메시지를 수신하면, 소스 노드는 노드  $i$ 를 우회하는 새로운 경로 설정을 위하여 목적지 노드들에 대한 새로운 경로 설정 매커니즘을 시작한다.

그럼에도 불구하고 GIVE\_UP 상태하의 노드  $i$ 는 스스로부터의 데이터 패킷이 수신되는 한 해당 라우팅 테이블 엔트리를 삭제하지 않는다. 이는 수신된 데이터 패킷으로 인하여 해당 엔트리의 타이머가 만료되지 않기 때문이다. 결과적으로 소스가 새로운 우회 경로 설정에 실패한다 하더라도 노드  $i$ 는 자신을 통과하는 경로를 따라서 데이터 패킷 포워딩을 계속한다. 그러나 새로운 경로가 설정된다면 더 이상의 데이터 패킷이 노드  $i$ 에 도착하지 않으므로 해당 라우팅 테이블 엔트리에 대한 타이머가 만료되고 따라서 엔트리는 결국 삭제될 것이다. IGN\_REQ 상태에서의와 같이 GIVE\_UP 상태 하의 노드  $i$ 도 어떠한 경로 설정 매커니즘에도 참여하지 않는다. 노드  $i$ 는 트래픽 부하 비율  $L_i$ 이  $\tau_2$  또는  $\tau_1$ 이하 일 때 각기 IGN\_REQ 상태 또는 정상 상태로 되돌아간다.

SLA는 각 노드가 자신의 혼잡 상태를 스스로 해결하게 함으로써 이동 노드의 제한된 자원이 가능한 한 공평하게 사용될 수 있도록 하였다. 5절에서 보이는 바와 같이 SLA는 노드 이동성 정도에 상관없이 트래픽을 분산시키는데 효율적으로 동작함을 확인할 수 있다. 또한 SLA는 기존의 어떠한 종류의 애드혹 라우팅 프로토콜에도 쉽게 적용될 수 있는 방법이며, 이들과 독립적으로 존재한다.

## 4. Protocol-Independent Fairness Algorithm(PIFA)

이전 절에서 노드들은 패킷 포워딩에 있어서 정직하다고 가정하였다. 다시 말해서 자신의 상태가 IGN\_REQ 인지 또는 GIVE\_UP인지 결정함에 있어서 정직하다고 가정하였다. 그러나 일부 이기적인 노드들은 다른 노드들과 협력하지 않고 자신의 에너지를 절약하기 위하여 다른 노드의 패킷을 고의적으로 포워딩 하지 않을 수도 있다. 이러한 이기적인 노드들을 격리시키거나 또는 자발적으로 패킷 포워딩에 참여시키기 위하여, 본 논문에서는 *credit* 기반 지불 방식을 제안한다. 기존의 PPM/PTM[4]이나 Sprite[5] 방식 또한 *credit* 기반 방식이지만, 이 방식들은 오직 DSR과 같은 소스 라우팅 방식과 함께 사용할 수 있다. 그러나 본 논문에서 제안한 PIFA(Protocol Independent Fairness Algorithm) 방식은 라

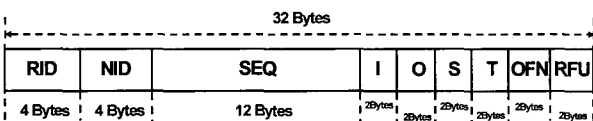
우딩 프로토콜의 종류에 상관없이 사용할 수 있다는 장점을 가진다.

PIFA를 사용하는 MANET 노드는 자신이 충분한 *credit* 을 가지고 있을 때만 패킷을 전송할 수 있으며, 노드들은 다른 노드의 패킷을 포워딩 해줌으로써 *credit*을 모을 수 있다. PIFA는 더 많은 *credit*을 모으기 위하여 자신이 실제로 포워딩 한 패킷 수를 속이려는 노드를 발견하고 격리할 수 있다. PIFA는 둘 이상의 노드 사이에 부당한 결탁은 없으며, AODV에서와 같이 소스와 목적지 노드 사이의 전체 경로 정보가 제공되지 않는다고 가정하였다. 또한 연속된 두 노드 사이의 1 홉 전송은 늘 성공한다고 가정하였다.

PIFA는 노드들의 CDB(Credit Database)를 관리하기 위한 CM(Credit Manager) 서버 노드를 사용한다. MANET 노드들은 정기적으로 자신이 포워딩 한 패킷의 수를 CM에게 보고하며, CM은 이러한 보고의 신뢰성을 확인한 후 현재의 MANET 토폴로지를 추론한다.

(그림 1)은 MANET 노드가 자신의 이웃 노드 각각에 대하여 CM에게 전송하는 report 메시지 형식을 보인 것이다. 각 필드의 의미는 다음과 같다.

- RID : 보고하는 노드의 ID
- NID : 이웃 노드의 ID
- SEQ : report 메시지의 동기화를 위하여 사용되는 순서 번호
- I : 이웃 노드로부터 수신된 패킷의 수
- O : 이웃 노드로 전송한 패킷의 수
- S : 해당 노드에서 시작하여 이웃노드들 중 한 노드로 전송한 패킷의 수
- T : 이웃 노드로부터 수신하여 해당 노드에서 종결된 패킷의 수
- OFN : 이웃 노드로부터 수신된 패킷 중 이웃 노드에서 시작된 패킷의 수
- RFU : 향후 사용용



(그림 1) CM에게 전송되는 report 메시지

(그림 2)(a)는 CM으로 전송되는 report 메시지의 예를 보인 것이다. 같은 순서 번호를 가진 report 메시지를 모아, CM은 세가지 측면에서 report들의 신뢰성을 확인한다. 첫째, 한 노드에서 전송된 모든 패킷의 수는 해당 노드의 이웃 노드들의 수신 패킷의 수와 같아야 한다. 예를 들어 노드 *n* 과 *m*이 이웃한 노드이고,  $A_{n,m}$ 은 RID와 NID가 *n*과 *m*인 메시지의 *A*필드라면,  $O_{n,m} = I_{m,n}$ 이다.

둘째, 한 노드의 전체 수신 패킷의 수( $\sum I$ )와 전체 종료 패킷의 수( $\sum T$ ) 사이의 차이는 포워딩 패킷의 수와 같아야 한다. 또한 전체 전송 패킷의 수( $\sum O$ )와 해당 노드에서 시

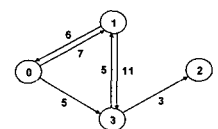
작된 패킷의 수( $\sum S$ )의 차이도 포워딩 패킷의 수와 같아야 한다. 그러므로  $F_n$ 이 노드 *n*에 의해 포워딩 된 패킷의 수이고  $A_n$ 이 노드 *n*의 이웃 노드들의 집합이라면,  $F_n$ 은 다음과 같다.

$$F_n = \sum_{m \in A_n} I_{n,m} - \sum_{m \in A_n} T_{n,m} = \sum_{m \in A_n} O_{n,m} - \sum_{m \in A_n} S_{n,m} \quad (1)$$

각 노드는  $F_n$ 에 비례하여 CM으로부터 *credit*을 받으며, 이 *credit*은 두 노드 사이의 평균 홉 수를  $H_{avg}$ 라고 했을 때  $\sum S \times H_{avg}$ 만큼 지불된다. 원칙적으로는 노드의 *credit*은 패킷이 목적지에 도착할 때까지 해당 패킷을 포워딩 한 노드의 수에 비례하여 감소된다. 그러나 AODV에서는 전체 경로 정보가 주어지지 않으므로 평균 홉 수를 이용한다.

마지막으로, 노드의 report에 포함된 S는 다음 홉 노드의 OFN과 동일해야 한다. OFN의 사용 용도는 악의 있는 노드가 앞서 설명한 식 (1)에서  $\sum T$ 와  $\sum S$ 의 값을 동시에 바꿈으로써 포워딩한 패킷 수를 조작하는 것을 막기 위함이다. OFN은 이웃 노드들로부터 수신된 패킷 중에서 이웃 노드에 의해 생성된 패킷의 수를 나타낸다. S값이 이웃 노드의 OFN 값과 동일하지 않다면, CM은 다음 홉 노드가 이전 노드의 포워딩 패킷 개수를 조작할 이유가 없다는 판단 하에 OFN 값을 취한다. 이와 같은 모든 측면에서의 신뢰성 확인 후, CM은 (그림 2)(b)에서와 같이 report 메시지로부터 토폴로지를 추론한다. 이러한 토폴로지는 노드의 이동성으로 인하여 실제 토폴로지와 다를 수도 있다. 그러나 이러한 사실은 포워딩 패킷 수를 계산하는 데는 아무런 영향도 미치지 못한다.

RID	NID	SEQ	I	O	S	T	OFN
0	1	128	6	7	7	2	2
0	3	128	0	5	1	0	0
1	0	128	7	6	2	3	7
1	3	128	5	11	7	1	3
2	3	128	3	0	0	3	2
3	0	128	5	0	0	4	1
3	1	128	11	5	3	9	7
3	2	128	0	3	2	0	0



(a) Report messages to CM      (b) Topology calculated with the reports

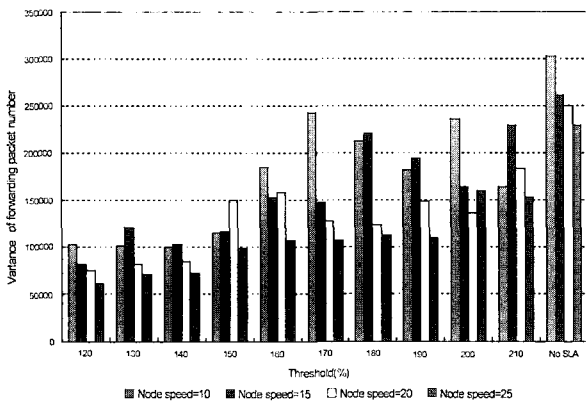
(그림 2) report 메시지로부터 현재 토폴로지를 계산하는 과정

MANET 환경에서 PIFA를 이용한 노드들은 자신의 패킷을 전송하기 위하여 *credit*을 획득해야 하므로 고의로 IGN\_REQ나 GIVE\_UP 상태에 들어가지 않을 것이다. 초기에 CM은 모든 노드들에게 정해진 개수의 *credit*을 할당하여 각 노드가 *credit*을 획득하기 이전이라 할지라도 얼마간의 패킷을 전송할 수 있도록 해준다.

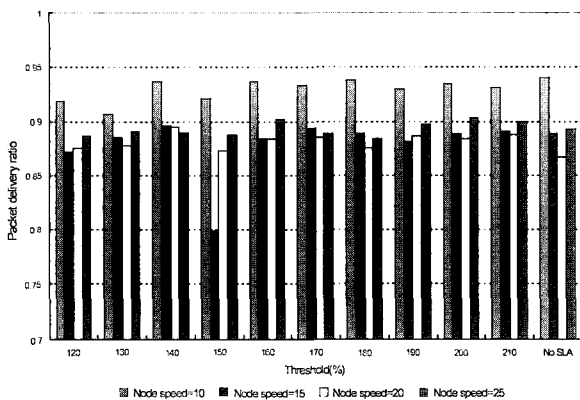
### 5. 성능 평가

SLA의 성능 평가를 위하여 ns-2 시뮬레이터를 사용하였다. 실험 환경은 2000m×2000m 범위 내 50 이동 노드로 구

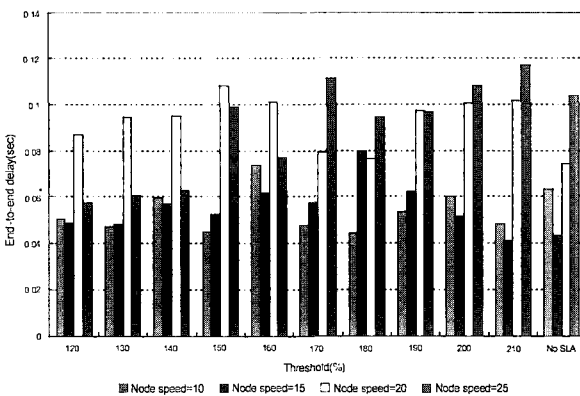
성되며 각 노드의 전송 범위와 채널 용량은 각각 250m와 2Mbps이다. 각 노드의 이동 방향은 무작위로 선택되었으며, 거리  $d$ 에 따라 신호 세기가  $1/d^2$ 로 감소되는 자유공간 전파모델을 사용하였다. 하부의 매체접근 제어 프로토콜로는 IEEE 802.11을 채택하였고 유니캐스트 라우팅 프로토콜로 AODV를 사용하였다. 소스 노드와 목적지 노드는 무작위로 선택하였고 각 소스는 초당 2개의 512바이트 패킷을 고정 비율(CBR)로 생성한다. 전체 실험 시간은 1000초이다.



(그림 3) 노드 이동성에 따른 포워딩 패킷 수의 변화



(그림 4) 노드 이동성에 따른 패킷 전송률



(그림 5) 노드 이동성에 따른 종단간 지연시간

(그림 3)~(그림 5)는 노드 이동성(10, 15, 20, 25m/s)에 따

른 성능 변화를 보인 것이다. 노드는 60초간 제자리에 머무른 후 새로운 위치로 이동한다. 그림의 가로축은 전체 네트워크 평균 트래픽에 대한 노드의 트래픽 비율, 즉,  $M1$ 에서의  $L_i$ 를 나타낸다. SLA를 적용하지 않은 AODV의 성능은 그림에서 가장 오른쪽에 나타나 있다. (그림 3)은 일정한 시간 간격(10초)마다 각 노드가 포워딩한 패킷 수의 변화를 보인 것이다. 그림에서와 같이 SLA는 AODV의 성능을 두드러지게 향상시켰음을 확인할 수 있다. 그러나 노드의 이동성이 증가함에 따라 성능 향상의 정도는 차츰 둔화되며, 이는 AODV가 높은 이동성 환경에서 경로 설정을 좀더 자주 수행하기 때문에 그 결과 보다 나은 트래픽 분산 효과를 얻을 수 있기 때문이다. 또한 낮은 경계치를 사용할수록 더 좋은 트래픽 분산 효과를 나타낸다. (그림 4)는 노드 이동성에 따른 패킷 전송률을 보인다. SLA에서 GIVE\_UP 메시지를 수신한 소스는 대체 경로 설정을 시도하므로 혼잡 현상이 완화되어 패킷 전송률이 향상될 수 있으므로, SLA는 AODV보다 약간 더 나은 성능을 보인다. (그림 5)는 패킷의 종단간 지연시간을 보이고 있다. 매우 낮은 경계치는 경로 설정 배커니즘을 자주 동작시키므로 오히려 종단간 지연시간을 증가시킨다. 그러나 경계치가 아주 낮은 경우를 제외하고는 SLA는 AODV와 거의 같은 성능을 보인다. 그러므로 SLA는 다른 성능들을 악화시키지 않고서도 높은 트래픽 분산 효과를 얻음으로써 AODV보다 나은 성능을 보임을 알 수 있다.

### 6. 결 론

네트워크의 이동성이 낮은 환경에서 AODV나 DSR의 일부 노드들은 패킷 포워딩을 위하여 과부하 될 수 있다. 이러한 트래픽 집중화는 MANET 노드들의 제한된 에너지를 빠르게 소모시키는 문제를 야기시킨다. 이러한 문제를 해결하기 위하여, 본 논문에서는 부하 균등을 위한 새로운 방법인 SLA를 제안하였다. 이는 각 노드가 자신의 트래픽 상태를 독자적으로 확인하여 소스 노드에게 대체 우회 경로 설정을 요구하는 방식이다. SLA는 기존의 라우팅 프로토콜에 부가적으로 동작하는 독립적인 모듈이다.

시뮬레이션 결과는 SLA가 패킷 전송률을 저하시키지 않으면서도 트래픽 부하를 효율적으로 분산시킴으로써, 이동 노드들 사이에서 에너지 소모가 공정하게 이루어지도록 하였음을 보인다. 게다가 일부 이기적인 노드들을 패킷 포워딩에 참여시키기 위하여 어떠한 종류의 라우팅 프로토콜과도 동작할 수 있는 credit 기반 지불 방식인 PIFA를 추가적으로 제안하였다.

### 참 고 문 헌

[1] Perkins, C. E., Royer, E. M., Das, S. R., "Ad hoc On demand Distance Vector(AODV) Routing," IETF MANET Internet-draft, 2003.

[2] Johnson, D. B., Maltz, D. A., Hu, Y. - C., "The Dynamic Source Routing(DSR) Protocol for Mobile Ad hoc Networks," IETF MANET Internet-draft, 2003.

[3] Das, S. R., Perkins, C. E., Royer, E. M., "Performance Comparison of Two On-demand Routing Protocols for Ad hoc Networks," Proc. IEEE INFOCOM, pp.3-12, 2000.

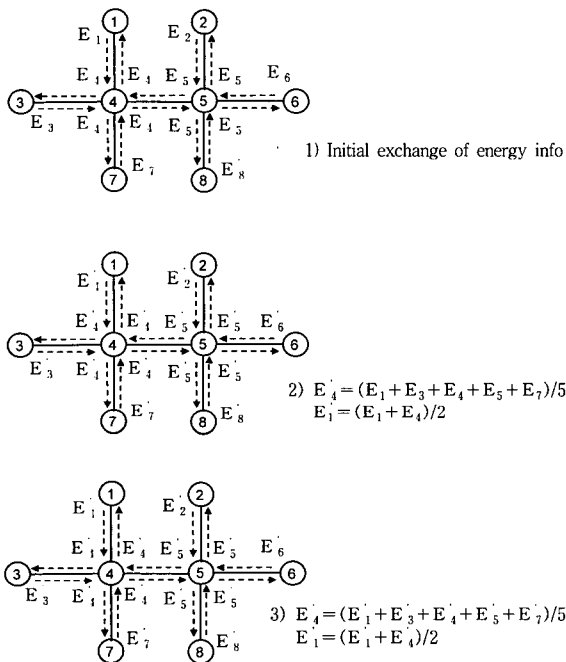
[4] Buttyan, L., Hubaux, J. - P., "Enforcing Service Availability in Mobile Ad-Hoc Networks," Proc. ACM MobiHoc, pp.87-96, 2000.

[5] Zhong, S., Chen, J., Yang, Y. R., "Sprite : A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," Proc. IEEE INFOCOM, pp.1987-1997, 2003.

[6] Hassanein, H., Zhou, A., "Routing with Load Balancing in Wireless Ad hoc Networks," Proc. ACM MSWiM, pp.89-96, 2001.

[7] Lee, S. - J., Gerla, M., "Dynamic Load-Aware Routing in Ad hoc Networks," Proc. IEEE ICC, pp.3206-3210, 2001.

<부 록>



(그림 6) 노드 에너지 평균 값 계산 과정

부록에서는 MANET에서 버퍼링된 패킷 수의 평균 또는 에너지의 평균을 계산하는 방법에 대하여 제안한다. 본 논문에서는 TTL=1인 RREP 메시지를 Hello 메시지[1]로 사용한다. Hello 메시지는 RREP나 다른 2계층 메시지가 HELLO\_INTERVAL(일반적으로 1000ms)동안 브로드캐스트 되지 않을 때 링크 연결을 확인하기 위하여 브로드캐스트 된다. 이러한 방식은 거리 벡터(distance vector) 프로토콜의 동작과 유사하다. Hello 메시지로 사용되는 RREP는 헤더의 9비트가 현재 사용되지 않고 있으므로, 해당 필드에 노드의 에너지 정보와 버퍼링된 패킷 수를 담아 이웃 노드와 교환한다. 9비트 중 5비트에 버퍼링된 패킷 수를, 나머지 4비트에

16단계로 나뉘어진 잔여 에너지 정보를 넣는다. 자세한 동작은 다음과 같다. 편의상 에너지 정보만을 가지고 (그림 6)에서 설명하였다.

- ① 초기에 노드  $i$ 는 이웃 노드들과 자신의 에너지 정보  $E_i$ 를 교환한다.
- ②  $E_i$ 의  $E_i'$ 와  $E_j$ (노드  $j$ 는 노드  $i$ 의 이웃 노드)를 계산한 후 노드  $i$ 는 이웃 노드들에게  $E_i'$ 를 알린다.
- ③ 노드  $i$ 는  $E_i'$ 의 평균과 모든  $E_j'$ (노드  $j$ 는 노드  $i$ 의 이웃 노드) 값을 기반으로 정기적으로  $E_i'$ 을 갱신한다.

시뮬레이션 결과에 의하면  $E_i'$  값이 안정적이고, MANET이 30 노드로 구성되어 있을 때, 3단계 이후의 평균 값이 거의 같음을 알 수 있었다.



안 상 현

e-mail : ahn@uos.ac.kr

1986년 서울대학교 컴퓨터공학과(학사)  
 1988년 서울대학교 대학원 컴퓨터공학과(석사)  
 1989년 University of Minnesota 컴퓨터학과(박사)

1988년 (주) 데이콤 연구원

1994년 세종대학교 컴퓨터학과 전임강사/조교수

1998년~현재 서울시립대학교 컴퓨터과학부 조교수/부교수

관심분야 : 애드혹 네트워크, 센서 네트워크, 홈 네트워크, 이동통신, 라우팅 프로토콜 등



유 영 환

e-mail : ymomo73@gmail.com

1996년 서울대학교 컴퓨터공학과(학사)

1998년 서울대학교 대학원 컴퓨터공학과(석사)

2004년 서울대학교 대학원 전기컴퓨터공학부(박사)

2004년~현재 University of Cincinnati, ECECS Department, Post-Doc

관심분야 : 애드혹 네트워크, 센서 네트워크, 이동통신 등



임 유 진

e-mail : yujin@suwon.ac.kr

1995년 숙명여자대학교 전자계산학과(학사)

1997년 숙명여자대학교 대학원 전자계산학과(석사)

2000년 숙명여자대학교 대학원 전자계산학과(박사)

2000년 서울대학교 Post-Doc

2000년 서울시립대학교 연구교수

2003년 University of California Los Angeles, Post-Doc

2003년 삼성종합기술원 전문연구원

2004년~현재 수원대학교 정보미디어학과 전임강사

관심분야 : 센서 네트워크, 애드혹 네트워크, 홈 네트워크 등