

# 이동 애드혹 네트워크를 위한 경로 지속성을 고려한 거리벡터 멀티캐스트 프로토콜

## (Durable Distance Vector Multicasting Protocol for Mobile Ad hoc Networks Based on Path-Durability)

이 세 영<sup>†</sup>      장 형 수<sup>\*\*</sup>  
(Se-young Lee)      (Hyeong-soo Chang)

**요약** 본 논문에서는 distance vector 라우팅 알고리즘에 경로 지속성에 대한 분석을 적용한 Durable Distance Vector Multicast(DDVM) 알고리즘을 제안한다. DDVM은 기존의 distance vector 알고리즘에 PATHS의 분석 내용을 기반으로 한 경로 지속성 정보를 포함하여 견고한 멀티캐스팅 경로를 구성한다. 또한 경로 정보에 목적지까지의 세부적인 경로의 지속성 정보 역시 포함하여 멀티캐스팅 경로 형성의 실패율을 줄이고 보다 지속성 있는 경로를 멀티캐스팅 경로에 포함시킨다. 이러한 경로들을 통해 멀티캐스팅을 수행함으로써 high mobility 환경에서 기존의 알고리즘보다 높은 전송률을 보이며, 실험 결과를 통해 이를 확인할 수 있다.

**키워드** : 이동 애드혹 네트워크, 멀티캐스팅, 디스턴스벡터 라우팅 알고리즘, ODMRP, 경로지속성

**Abstract** In this paper, we propose a novel on-demand multicasting algorithm, "Durable Distance Vector Multicasting (DDVM)," for Mobile Ad hoc Network(MANET). DDVM has a more efficient routing-control complexity than Link State Routing(LSR) and with the smaller complexity, DDVM provides a high delivery ratio for MANET with high mobility by adapting to dynamic topological changes. This is achieved by adding "combined path durability" statistic into the routing-control information of Distance Vector Routing(DVR). The routing-path duration statistic is computed in a fully distributed manner at each node in the network based on only local information from neighbour nodes. We show by simulation studies the effectiveness of DDVM compared with On-demand Multicast Routing Protocol(ODMRP).

**Key words** : MANET, Multicasting, Distance Vector Routing Algorithm, ODMRP, Path-durability

### 1. 개요

본 논문에서는 Mobile Ad hoc Network(MANET) [1]의 경로 지속성을 확률적으로 분석한 논문 "Analysis of PATH Duration Statistics and their Impact on Reactive MANET Routing Protocols(PATHS)"[2]를 바탕으로 멀티캐스팅 경로의 지속성을 증대하여 높은 이동속도(high mobility)를 가진 MANET에서도 높은 전송률(delivery ratio)을 보이는 Durable Distance

Vector Multicast(DDVM) 알고리즘을 제안하고자 한다.

MANET는 노드의 이동(mobility)에 의한 위상 변화가 잦은 반면, 전파 특성에 의해 한번 전송한 데이터를 주변의 여러 노드가 동시에 받을 수 있다는 broadcast 특성을 장점으로 가지고 있다. 이러한 MANET의 특성을 고려하여 다양한 접근방식을 사용하는 멀티캐스팅 알고리즘들이 있는데, 대표적인 형태의 멀티캐스팅 그룹 구성 방식으로는 tree 방식[3], mesh 방식[4] 등이 있고, overlay 방식[5], stateless 방식[6] 등이 새로이 연구되고 있다.

앞서 열거한 여러가지 접근 방식들은 각각 개별적인 특징들을 가지고 있고, 서로 다른 장점들을 가지고 있다 [7]. 하지만 기본적으로 라우팅, 멀티캐스팅 알고리즘이 추구하는 목표는 데이터 전송률 확보에 있으며, 이외에 제어 패킷의 부하 감소, 전송효율의 증대, 하위 layer와

· 이 논문은 한국학술진흥재단의 지원에 의하여 연구되었음  
(KRF-2004-003-D00294).

† 학생회원 : 서강대학교 컴퓨터학과  
sylee@smolab.sogang.ac.kr

\*\* 정회원 : 서강대학교 컴퓨터학과 교수  
hschang@sogang.ac.kr

논문접수 : 2005년 9월 3일

심사완료 : 2006년 10월 30일

의 연계 같은 문제들은 데이터 전송률과 함께 해결되어야 한다.

멀티캐스팅 알고리즘은 멀티캐스팅 그룹의 중심점 역할을 하는 노드들(주로 source 노드)의 경로 정보를 명시적, 혹은 암시적으로 필요로 한다는 점에서 기본적으로 라우팅 알고리즘의 중요성이 대두된다. MANET에서 대표적으로 사용되는 라우팅 알고리즘은 크게 Link State Routing(LSR) 방식과 Distance Vector Routing(DVR) 방식으로 나눌 수 있다[8]. 이 중 DVR은 LSR에 비해 적은 대역폭을 소모한다는 장점이 있다. 따라서 DVR의 라우팅 정보에 MANET 노드의 이동성에 의해 발생하는 문제를 해결할 수 있는 정보를 추가한다면 보다 효율적인 라우팅, 멀티캐스팅을 수행할 수 있을 것이다. 최근에 발표된 논문 PATHS[2]는 다양한 형태의 MANET 노드 이동 모델들[9]에 대한 분석을 통해 MANET 상의 단일 경로가 지속될 확률을 분석하여 수학적으로 기술하였다. 본 논문에서는 제안하고자 하는 DDVM은 이 단일 경로 지속 확률을 복합적인 경로 정보로 확장하여 DVR에 포함시킴으로서 적은 부하의 라우팅 정보로도 MANET의 위상 변화에 효율적으로 적용할 수 있는 라우팅 경로를 형성할 수 있다. 이는 전송률 면에서의 향상을 가져오고, 부가적으로 멀티캐스팅의 효율을 높이게 된다. 이는 4장의 시뮬레이션 결과를 통해 확인할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 MANET 환경에서의 멀티캐스팅 알고리즘들의 공통점을 소개하고, DVR 알고리즘에 PATHS에서의 분석 내용을 적용하는 방안 대해 간략히 설명한다. 3장에서는 DDVM 알고리즘의 구체적인 동작방식을 PATHS의 분석 내용과 연계시켜 설명한다. 4장에서는 실험 결과를, 마지막으로 5장에서는 본 논문의 결론을 내린다.

## 2. 연구 배경

현재 다양한 MANET 환경에서의 멀티캐스팅 알고리즘들이 존재하지만 기본적으로 멀티캐스팅에 직접적으로 참여하는 source 노드들과 destination 노드들을 연결하는 forwarding 노드들을 어떤 방식으로 포함시키느냐에 따라 성능이 좌우된다. Forwarding 노드를 효율적으로 결정하기 위해서는 직접적인 참여자들(source, destination 노드들)에 대한 경로 정보를 전달해야 할 필요가 있으며, 이를 위해 멀티캐스팅 알고리즘들은 명시적, 혹은 암시적인 방법으로 참여자들 전체 혹은 일부에 대한 라우팅 정보를 전달한다. 따라서 우수한 멀티캐스팅 알고리즘을 설계하기 위해서는 이러한 라우팅 정보를 전달하는 방법에 대한 기본적인 이해가 선행되어야 한다.

MANET 환경에서의 여러 라우팅 알고리즘들은 크게 LSR 방식[10,11]과 DVR 방식[12,13]으로 나눌 수 있다. LSR은 이웃노드와의 연결이 변화하였을 때 이를 다른 모든 노드에 알리고 라우팅 경로를 재계산하는 방식이다. DVR은 직접 통신 가능한 이웃노드에게만 자신의 라우팅 정보를 전달하고, 각 노드는 이웃노드의 정보를 통해 간접적으로 자신의 라우팅 정보를 계산하는 방식이다. LSR은 거의 즉각적인 라우팅 경로 형성이 가능하므로 경로 정보의 효율이 높지만 위상 변화가 많은 MANET에서는 대역폭을 많이 소모하게 된다. 반면 DVR은 여러 노드에 대한 라우팅 정보를 하나의 패킷으로 처리 가능하므로 대역폭 소모가 적으나, 라우팅 정보의 효율에 문제가 있다.

LSR, DVR의 라우팅 정보 효율을 면밀히 분석해보면 다음과 같다. LSR에서 노드  $i$ 가 시간  $t$ 마다 라우팅 정보를 *flooding* 하고, DVR 역시 매  $t$ 시간마다 이웃에 라우팅 정보를 전달한다고 가정한다. 그리고 라우팅 패킷의 전송지연 시간을  $d$ 라고 하고, 라우팅 경로의 hop 수를  $h$ 라고 하자. 그러면 LSR에서 노드  $i$ 에 대한 라우팅 정보가 도착하는데 걸리는 시간은  $O(dh)$ 이고, 이를 이용하여 패킷 전송을 수행할 경우 최대  $O(t+dh)$  이전의 라우팅 정보를 이용하게 된다. 반면 DVR의 경우 라우팅 정보가 도착하는데 최악의 경우  $O((t+d)h)$ 의 시간이 걸리고, 이를 이용할 경우  $O(t+(t+d)h)$  시간 이전의 라우팅 정보를 이용하게 된다. 일반적으로  $d \ll t$ 이므로 LSR은 근사적으로 최대  $O(t)$  이전의 라우팅 정보를 이용하지만, DVR은  $O((h+1)t)$  이전의 라우팅 정보를 이용하게 된다. 따라서 MANET 환경과 같이 노드의 이동성이 고려되는 경우에 DVR의 정보는 매우 부정확한 정보가 될 수 있다.

반면 네트워크의 총 노드 수를  $N$ 이라 하고, 데이터를 보낼 노드의 수를  $S$ 라고 했을 때 DVR은  $S$ , 그리고 위상변화와는 관계 없이  $O(N)$ 개의 패킷만을 보내게 된다. 이러한 DVR의 장점을 살려 MANET 환경에서 사용하기 위해서는 우선 라우팅 정보의 부정확성을 줄여야만 한다. 이를 위해서 먼저 MANET 환경에서 경로 정보가 시간에 따라 얼마나 부정확하게 되는가를 수학적으로 분석해야만 한다.

PATHS[2]에서는 시뮬레이션을 통해 이러한 분석을 수행하고 수식화 하였다(식 (1)). PATHS에 따라 link를 두 통신 가능한 노드들의 논리적 연결이라 정의하고, 경로(path)를 loop를 포함하지 않은 link들의 연속이라고 정의할 때 길이  $h$ 의 경로가 형성된 지  $x$ 만큼의 시간 이후에 경로가 연결되어 있지 않을 확률  $F(x)$ 를 식으로 나타내면 다음과 같다.

$$F(x) = 1 - e^{-\frac{\lambda_0 h V}{R} x} \quad (1)$$

여기서  $R$ 은 MANET 노드의 propagation range,  $V$ 는 노드의 평균 이동 속도,  $\lambda_0$ 는 경로가 지속하지 않을 확률의 증가율을 나타내는 상수이다. 시간  $x$ 후에 길이  $h$ 의 경로가 지속하지 않을 확률은  $F(x)$ 의 확률분포를 따르게 된다. 위의 식을 보다 단순화하기 위해서 각 변수를 변경할 수 있다. 특정 기간동안의  $V$ 를 계산하면 상수이고,  $R$ 역시 상수이다. 따라서 위의 식에서  $\lambda_0 V/R = \mu$ 로 놓으면,

$$F(x) = 1 - e^{-\frac{\lambda_0 h V}{R} x} = 1 - e^{-\mu h x}$$

이고, 이를 경로가 지속할 확률  $G(x) = 1 - F(x)$ 로 바꾸면 어떤 상수  $\alpha \in (0, 1]$ 에 대하여,

$$G(x) = 1 - F(x) = e^{-\mu h x} = (1/e)^{\mu h x} = \alpha^{hx} \quad (2)$$

이고,  $\alpha$ 는  $(1/e)^\mu$ 에 의존적인 값이다. DVR의 라우팅 정보에 식 (2)의 정보를 추가하여 이용하면 보다 지속적인 라우팅 경로를 통해 데이터를 전달할 수 있게 된다.

이와 더불어 LSR, DVR의 경우 데이터를 받을 노드에 이르는 여러 세부 경로가 있지만 이들에 대한 정보는 무시된다. 그림 1은 이러한 세부 경로의 예를 보여준다. dst로부터 src까지 데이터를 전달하려고 할 때에 dst는 A, B, C를 통한 경로를 선택할 수 있다. 하지만 LSR이나 DVR을 이용할 경우 dst는 이들 경로가 어떠한 세부적인 경로들(B에서 E 또는 F로 가는 경로)을 가지고 있는지, 혹은 이러한 세부적인 경로들이 어떤 특징을 가지고 있는지 알 수 없다. A의 경우 D를 통해 src로 가는 하나의 경로만 존재하지만, B는 E나 F를 통해 가는 두 경로를 가지고 있다. 따라서 B는 E나 F로 가는 경로로 라우팅이 실패했을 경우 다른 경로를 다시 선택할 수 있다는 장점을 가지고 있다. 식 (2)를 적절히 변형하여 사용하면 src에 이르는 세부 경로에 대한 정보까지도 포함시켜 보다 나은 라우팅 경로를 선택하는데 이용할 수 있다. 이는 3.1에서 보다 자세하게 설명하도록 한다.

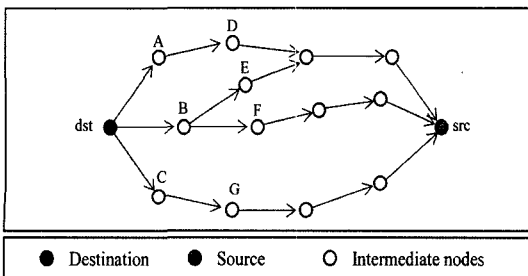


그림 1 라우팅 경로와 세부경로

### 3. Durable Distance Vector Multicast(DDVM)

DDVM은 다른 멀티캐스팅 알고리즘과 마찬가지로 직접적인 멀티캐스팅 참여자인 source 노드와 destination 노드, 그리고 이들을 연결하는 forwarding 노드로 전체 그룹을 구성한다. Source 노드가 새로운 그룹이 구성되었음을 알리고 나면, 네트워크 상의 각 노드는 distance vector 알고리즘에 따라 주기적으로 source 노드들에 대한 경로 정보를 이웃 노드(자신과 직접 통신 가능한 노드)들에게 전달한다. 다른 mesh 기반 멀티캐스팅 알고리즘과 마찬가지로 destination 노드는 각 source들에게 그룹 참여 메시지를 전달하고, source 노드 역시 자신을 제외한 다른 source들에게 그룹 참여 메시지를 전달한다. 이 과정에서 그룹 참여 메시지를 전달하는 경로상의 노드들은 forwarding 노드로 그룹에 포함된다. 그룹 참여 메시지의 경로는 이를 전달하는 노드가 가진 distance vector routing table(이하 경로 정보)을 통해 결정된다. MANET의 특성상 주기적인 그룹 참여 메시지가 필요하며, forwarding 노드는 일정한 시간 이상 다시 그룹 참여 메시지를 받지 못하면 그룹에서 탈퇴하게 된다.

DDVM의 보다 명확한 설명을 위해서 3.1에서는 DDVM의 멀티캐스팅 그룹 구성 및 유지에 대해 서술하고, 3.2에서는 멀티캐스팅 데이터 전송 방법에 대해 서술한다. 마지막으로 3.3에서는 DDVM의 성능을 개선하기 위해 사용될 수 있는 부가적인 방법들에 대해 서술한다.

#### 3.1 멀티캐스팅 그룹 구성 및 유지

DDVM의 멀티캐스팅 그룹 구성은 크게 Source Announce, Routing Update, Join Request, End Group의 4단계로 구분된다.

##### 3.1.1 Source Announce 단계

DDVM의 멀티캐스팅 그룹 구성은 source 노드로부터 시작한다. Source 노드는 자신이 포함되어야 할 그룹이 이미 구성되어 있는지를 지역 정보의  $gID$ 를 통해 확인한다.  $gID$ 는 한 네트워크 내에 있는 여러 멀티캐스팅 그룹을 구분하는 고유한 ID이다. 이미 구성된 그룹이 없다면 Source Announce 패킷을 flooding 하여 새로운 그룹이 구성되었음을 알린다. 표 1(a)와 같이 Source Announce 패킷에는  $gID$ 와 source 노드의 ID인  $sID$  정보가 들어 있고, 이를 통해 다른 멀티캐스팅 그룹과 구분할 수 있도록 한다. 만일 또 다른 source 노드가 이미 형성된 그룹에 포함되고 싶을 때에는 포함되고자 하는 그룹의  $gID$ 와 자신의 ID를 Source Announce 패킷을 통해 flooding 한다.

##### 3.1.2 Route Update 단계

2장에서 설명한 것처럼 경로 정보 전달은 멀티캐스팅 알고리즘의 핵심 부분이고, 따라서 이 절은 DDVM을 이해하기 위해 가장 중요한 부분이다.

Source Announce가 완료되면 네트워크상의 노드는 source 노드로의 경로 정보를 Route Update 패킷을 통해 주기적으로 이웃노드에 전달한다. Route Update 패킷에 들어가는 정보는 표 1(b)와 같다. DDVM은 기본적으로 distance vector 알고리즘을 사용하므로 각 노드가 전달하는 정보에는 source 노드의 ID와 이 source까지의 거리(본 알고리즘에서는 hop 수) 정보가 포함된다. 이에 더하여 MANET의 잦은 위상변화에 대처하기 위해 경로의 지속 확률을 계산하여 전달한다.

먼저 source 노드  $s$ 와 노드  $i$ 와의 hop 수를  $D_i^s$  (Distance)라고 할 때 이를 계산하는 식은 다음과 같다.

$$D_i^s = \begin{cases} 0 & \text{if } i = s \\ \min_{j \in N_i} D_{j \rightarrow i}^s + 1 & \text{else} \end{cases} \quad (3)$$

여기서  $N_i$ 는 노드  $i$ 의 이웃노드 집합으로 이웃노드란 다른 노드를 거치지 않고 직접 통신 가능한 노드이다.  $D_{j \rightarrow i}^s$ 는  $i$ 의 이웃노드  $j$ 가  $i$ 에게 전달한 source 노드  $s$ 와의 거리(hop)이다. Source 노드는 자신과의 거리가 없으므로 항상 0을 전달한다.  $D_i^s$ 는  $i$ 의 이웃노드에게 전달되고, 이웃노드들은 이 정보를 통해 source까지의 최단거리를 찾고, 자신과 source와의 거리를 계산하는데 이용할 수 있다.

다음으로 source 노드  $s$ 와 노드  $i$ 와의 경로 지속 확률은  $C_i^s$  (Combined path-durability)로 정의된다. PATHS [2]에서 분석한 바에 의하면 MANET에서 형성된 경로의 지속 확률은 식 (2)에 의하여 시간(단위는 초)에 따라 지수적으로 감소하게 된다. DDVM에서 노드  $i$ 는 이웃노드 중에서 자신보다  $s$ 와 더 가까운 노드들을 선택하여 이들이 가진 경로 지속 확률을 적절히 조합함으로써 자신의 경로 지속 확률을 계산한다.  $C_i^s$ 를 계산하는 식은 다음과 같다.

$$C_i^s = \begin{cases} 1 & \text{if } i = s \\ 1 - \prod_{j \in N_i, D_{j \rightarrow i}^s < D_i^s} (1 - C_{j \rightarrow i}^s \cdot \alpha^{D_{j \rightarrow i}^s \cdot \Delta t}) & \text{else} \end{cases} \quad (4)$$

$t_c$ 는 현재 시각,  $t_{j \rightarrow i}$ 는 노드  $j$ 가  $i$ 에게 경로 정보를 전달한 시각이라고 할 때  $\Delta t$ 는  $t_c - t_{j \rightarrow i}$ 로 정의되고, 이는  $i$ 가  $j$ 의 경로 정보를 받은 이후 경과한 시간을 나타낸다.  $\alpha \in (0, 1]$ 는 경로 지속 확률의 감소율을 조절하는 상수이다. Source 노드  $s$ 는 자신과의 경로 지속 확률이 항상 1이므로 이를 이웃노드에 전달하고, 이외의 노드들은 자신보다  $s$ 와 가까운 이웃노드들을 통해 자신과  $s$ 의

“도달 가능 확률”을 식 (4)를 통해 계산한다. 도달 가능 확률  $\beta_i^s = \Pr(\text{시각 } t_c \text{에 } i \sim s \text{ 경로 중 적어도 하나가 지속하고 있을 확률})$ 이라고 정의하면, 이는  $(1 - \prod_{j \in N_i} (1 - \beta_j^s \cdot \alpha^{h_j^s \cdot \Delta t})) \cdot \Pr(\text{시각 } t_c \text{에 } j \in N_i \text{일 확률})$ 로 근사화 할 수 있다.  $h_j^s$ 는  $s$ 로부터  $j$ 에 이르는 경로의 길이이다. 이 때  $h_j^s$ 는 세부경로마다 다르므로 최단 경로만을 고려하도록 하고,  $\Pr(\text{시각 } t_c \text{에 } j \in N_i \text{일 확률})$ 를 근사적으로 1로 본다면  $C_i^s$ 와  $\beta_i^s$ 는 동일한 식이 된다. 식 (2)를 경로 형성 후  $x$ 만큼 지난 후에 길이  $h$ 의 경로가 지속할 확률과 동일하다고 가정하면 식 (4)는  $s \sim i$ 에 이르는 하나의 경로만이 있을 경우 식 (2)와 동일한 식이며, 여러 경로들에 대한 지속 확률을 계산할 수 있도록 변형한 식이 된다. 이는 단일 경로 정보만을 전달하는 기존의 LSR, DVR과는 차별되는 부분이다.

노드  $i$ 가 보낸 경로 정보는 이웃노드들에게 주기적으로 보내지고, 이를 받은 이웃노드들은 경로 정보들을 저장하는 경로 정보 테이블에  $i$ 의 정보를 저장하고 있다가 forwarding 노드를 결정할 필요가 있을 때 이를 사용한다. 이 방법은 3.1.3에 자세히 설명되어 있다.

### 3.1.3 Join Request 단계

DDVM에서 각 destination 노드는 모든 source 노드들에게 Join Request 패킷을 전달하고, 이 패킷을 받은 노드들을 그룹에 포함시킴으로서 source 노드들과 자신에 이르는 멀티캐스팅 경로를 형성한다. 또한 source 노드 역시 자신을 제외한 다른 source 노드들에게 Join Request 패킷을 보내어 보다 견고한 그룹을 구성할 수 있게 된다. Join Request 패킷에는 표 1(c)에 해당하는 정보가 포함된다.

하나의 Join Request 패킷으로 여러 노드에게 Join Request를 전달하는 방법은 논문 [4]에 자세히 설명되어 있으며, DDVM 역시 이를 따른다. 모든 Join Request 패킷은 최종적으로 원하는 그룹의 source 노드에 도달하는 것을 목표로 한다. Destination 노드 혹은 source 노드는 자신의 이웃노드들에 대한 경로 정보를 적절히 활용하여 효율적인 경로를 형성할 수 있는 하나의 노드 혹은 여러 노드에게 Join Request 패킷을 보낸다. 노드  $i$ 가 source 노드  $s$ 에게 Join Request 패킷을 보내기 위해 어떤 다음 노드(표 1(c)의  $nextID$ )  $k$ 를 선택할 것인가는 다음과 같은 식에 의해 결정된다.

$$k = \underset{j \in N_i, D_{j \rightarrow i}^s \leq D_i^s, j \neq pID}{\operatorname{argmax}} R(i, j, s) \quad (5)$$

$D_{j \rightarrow i}^s < D_i^s$  이거나  $D_{j \rightarrow i}^s = D_i^s \wedge C_{j \rightarrow i}^s \cdot \alpha^{D_{j \rightarrow i}^s \cdot \Delta t} > C_i^s$  이면  $R(i, j, s) = C_{j \rightarrow i}^s \cdot \alpha^{D_{j \rightarrow i}^s \cdot \Delta t}$ 이다.  $R(i, j, s)$  (Relative

*individual path-durability*)는 노드  $i$ 의 이웃 노드들( $j$ )에 대하여  $i$ 와의 상대적인 경로 지속 확률을 알려준다.  $i$ 의 이웃노드 중  $s$ 와 가까운 노드는 식 (5)에 의해 항상  $k$ 로 선택될 수 있지만, 같은 거리의 노드는 자신보다 경로 지속 확률이 높은 경우에만 선택될 수 있다.  $i$ 보다 먼 노드는 선택에서 제외되는데, 이는  $i$ 보다 먼 노드  $j$ 의 경우 식 (4)에 의해  $i$ 의 정보를 이용하여  $C_j^s$ 를 계산하였을 가능성이 매우 높고, 따라서 이를 선택할 경우 라우팅 loop를 형성하거나 긴 경로를 형성하여 낮은 효율의 그룹을 형성하게 될 가능성이 높기 때문이다. 다른 노드로부터 Join Request를 받아서 다시 보내야 할 경우에 이전 Join Request 패킷의  $pID$ 를  $k$ 의 후보에서 제외하는데 이 역시 loop를 방지하기 위함이다.

노드  $i$ 는 이렇게 선택된  $k$ 를  $nxtID$ 로 하여 Join Request 패킷을 보낸다. 이 패킷을 받은 노드는 다시 식 (5)를 통해 다음 노드를 계산하여 보낸다.

만일 Join Request 패킷을 받은 노드가 최종 목적지 ( $s_1, s_2, \dots$ )일 경우 Explicit Passive Ack 패킷을 보낸다. 이 패킷의 포맷은 표 1(d)에 나와 있다. 이 패킷을 보내는 이유는 DDVM이 passive ack[4]를 사용하기 때문이다. 만약 노드  $i$ 가 노드  $j$ 에게 Join Request 패킷을 보냈고, 이를  $j$ 가 받았을 경우  $j$ 는 다른 노드에게 Join Request를 보내거나 Explicit Passive Ack 패킷을 보내게 된다. MANET의 broadcast 특성 때문에 노드  $i$ 는 이 둘 중 하나를 받을 수 있게 되는데, 만약  $i$ 가  $j$ 에게 Join Request를 보낸 이후 일정 기간( $join\_request\_timeout$ )이 지났는데도 위의 패킷들을 받지 못했다면  $i$ 는  $j$ 에게 보낸 Join Request가 실패한 것으로 간주한다. 그러면  $i$ 는  $j$ 를 제외한 이웃노드를  $nxtID$ 로 선택하여 다시 Join Request를 보낸다. 이를 통해 멀티캐스팅 경로 형성의 성공율을 높인다.

그림 1을 통해 DDVM의 경로 형성 방법을 설명하면 다음과 같다.  $dst$ 가  $src$ 로의 경로를 형성할 경우에 일반적인 LSR 혹은 DVR은  $dst$ 의 이웃인 A, B, C 중 하나를 선택하게 된다. 이들의 경로 정보에는 세부 경로에 대한 정보가 없으므로 어느 노드를 선택해야만 더 나은 경로를 형성할 수 있을지는 알 수 없다. 반면 DDVM에서는 각 경로에 대한 지속 확률을 가지고 더 나은 경로를 선택할 수 있다. 세부경로 정보는 식 (4)에 의해 종합되므로 많은 경우에 B의 경로 지속 확률은 A 또는 C의 지속 확률에 비해 높을 것이다. 세부경로가 많다는 것은 Join Request가 실패하게 되더라도 다른 경로를 선택할 여지가 많다는 것이므로 경로 지속 확률 정보가 유용함을 알 수 있다. 만약  $dst$ 가 B에 Join Request를 보냈다면 B는 E 또는 F의 경로 중 하나를 선택하여

표 1 DDVM의 각 패킷 포맷

(a) 노드  $s_1$ 이 보내는 Source Announce 패킷 포맷

$pID$	$sID$	$gID$	$nxtID$
$s_1$	$s_1$	$g_1$	BROADCAST

(b) 노드  $i$ 가 보내는 Route Update 패킷 포맷

$pID$	#srcs	$gID$	$sID$	$D_i^{sID}$	$S_i^{sID}$	$nxtID$
$i$	source 노드의 수	$g_1, [g_2], \dots$	$s_1$	$D_i^{s_1}$	$S_i^{s_1}$	BROAD CAST
		$g_2, \dots$	$s_2$	$D_i^{s_2}$	$S_i^{s_2}$	
		...	...	...	...	

(c) 노드  $i$ 가 보내는 Join Request 패킷 포맷

$pID$	#dest	$gID$	$sID$	$nxtID$
$i$	Join Request가 최종 도착할 source 노드의 수	$g_1$	$s_1$	$N_i$ 중 하나
		$[g_2]$	$[s_2]$	$[N_i$ 중 하나]
		...	...	...

(d) 노드  $i$ 가 보내는 Explicit Passive Ack 패킷 포맷

$pID$	#dest	$gID$	$sID$	$nxtID$
$i$	1-(c)와 동일	$g_1$	$s_1$	Join Request의 $pID$
		$g_2$	$s_2$	
		...	...	

(e) 노드  $i$ 가 보내는 End Group 패킷 포맷

$pID$	$sID$	$gID$	$nxtID$
$i$	$i$	$g_1$	BROADCAST

(f) 노드  $i$ 가 보내는 Multicast Data 패킷 포맷

$pID$	$sID$	$gID$	#sequence	$nxtID$
$i$	$i$	$g_1$	#	BROADCAST

Join Request를 보내야 한다. 식 (4)에 의해 경로 정보 전달 시간이 짧은 경로일수록 선호되므로 B는 둘 중에 경로 정보가 더 빨리 도착한 노드를 선택할 수 있게 된다. 식 (1)에 의해 경로가 오래될수록 붕괴될 확률은 더 높게 된다. 따라서 이는 효율적인 선택이 된다.

Join Request 패킷을 받은 노드가 직접적인 그룹 참여 노드가 아닐 경우 이는 forwarding 노드로 간주되어 그룹에 일정 기간( $forwarding\_node\_life\_time$ ) 포함된 다. Join Request는 source 노드들 혹은 destination 노드들에 의해 주기적으로 보내지며, 이를 통해 MANET 노드들의 이동에 의한 경로 붕괴에 대처한다.

### 3.1.4 End Group 단계

참여자들은 각각 다른 방식으로 그룹에서 나오게 된

다. 먼저 destination 노드들은 주기적인 Join Request를 중단함으로써 그룹에서 탈퇴하게 된다. Source 노드의 경우 명시적으로 End Group 패킷(표 1(e))을 flooding 하여 그룹을 탈퇴한다. End Group 패킷을 받은 노드는 해당 source 노드에 대한 경로 정보를 지우고, 더 이상 해당 source 노드의 정보를 이웃에 보내지 않는다. 만약 한 그룹의 모든 source 노드가 End Group 패킷을 flooding 했다면 해당 멀티캐스팅 그룹은 경로 정보에서 삭제되고 그룹이 종료된다.

### 3.2 멀티캐스팅 데이터 전송

멀티캐스팅 데이터 전송은 MANET의 broadcast 특성을 이용한다[4]. 각 그룹에 속한 source 노드는 Multicast Data 패킷(표 1(f))에  $gID$ ,  $sID$ , 그리고 멀티캐스팅 데이터의 순서를 담은  $\#sequence$  정보를 포함하여 보낸다. 만약 forwarding 노드를 포함한 모든 그룹 내 노드중 하나가 이 패킷을 받았을 경우 위의 세 가지 정보를 통해 자신이 한번도 받지 않은 Multicast Data 패킷임을 확인하면 이를 다시 이웃노드들에게 보내고, 이미 받은 패킷이면 이 패킷을 버린다. Source 노드는 자신이 생성한 Multicast Data 패킷을 보낼 때마다  $\#sequence$ 를 1씩 증가시켜 보냄으로서 모든 노드가 새로운 멀티캐스팅 데이터임을 인식할 수 있도록 한다.

이것으로 DDVM의 핵심적인 동작 방식에 대한 설명을 마치고, DDVM의 세부적인 개선 사항들과 이들을 적용하였을 때 얻을 수 있는 이점들을 살펴보도록 하겠다.

### 3.3 개선 사항

#### 3.3.1 다양한 멀티캐스팅 그룹 형성 방법

기본적으로 DDVM은 높은 전송률을 얻기 위해 source 노드를 중심으로 한 mesh 기반 접근방식을 취한다. 하지만 경우에 따라서는 tree 기반 접근방식, 혹은 이 둘을 조합한 hybrid 접근방식[14] 등을 취할 수 있다. 일반적으로 broadcast 특성을 이용할 경우 참여자 노드의 수가 많으면 많을수록 높은 전송률을 보인다. 따라서 이러한 경우에 mesh 방식을 사용하면 오히려 그룹 유지에 필요한 부하를 늘리게 될 수 있다. 따라서 그룹의 한 source에게만 Join Request를 보냄으로서 shared tree를 구성하거나, source 노드들 중 일부에게만 Join Request를 보내어 그룹을 구성할 수도 있다. 또한 destination 노드들의 정보까지도 경로 정보에 포함시켜 이를 바탕으로 steiner tree[15]를 구성하는 overlay 방식을 취할 수도 있다. 이는 알고리즘의 사용 환경에 따라 달라질 수 있다.

#### 3.3.2 Join Request 실패 기록

DDVM에서는 어떤 노드  $i$ 가 노드  $j$ 로 보낸 Join Request에 대한 passive ack를 받지 못했다면 Join

Request가 실패한 것으로 간주하고, 다른 노드  $j'$ 를 선택하여 Join Request를 재전송한다. 이 때 Join Request가 실패한 노드  $j$ 의 경로 정보를 경로 정보 테이블에서 삭제하면 이후에 다시 Join Request를 보낼 때 실패를 줄일 수 있게 된다. 또한 이 정보를 삭제하게 되면 보다 정확한 경로 정보를 계산할 수 있으므로 경로 정보 효율성을 높일 수 있게 된다.

#### 3.3.3 중복된 Join Request 방지

만약 그룹 내의 어떤 노드가 Join Request 패킷을 받고 처리한지 얼마 되지 않았을 때 다른 노드로부터 Join Request를 받으면 또다시 이를 처리해야만 한다. 이것은 대역폭을 낭비할 수 있으므로 한번 Join Request를 받은 노드는 일정시간( $no\_join\_request\_interval$ ) 동안 받은 Join Request에 대해서는 source 노드처럼 Explicit Passive Ack만을 이전 노드를 향해 보냄으로서 중복된 Join Request로 인한 대역폭 소모를 방지한다.

#### 3.3.4 Quantization

DDVM에서 사용하는  $C_r^*$  정보, 즉 경로 지속 확률은 항상 0~1의 실수 값을 가진다. 일반적으로 컴퓨터에서 실수 계산을 위해 사용하는 데이터 타입은 float 형(4byte)과 double 형(8byte)이 있는데, 이를 통해 경로 지속 확률을 계산할 경우 저장 공간이나 대역폭을 낭비할 수 있다. 따라서 4바이트 미만의 저장 공간을 0보다 작지 않은 정수형을 표현하는 데이터형으로 사용하고, 이 공간이 표현 가능한 최대 정수로 나누어 0~1 사이의 실수로 mapping할 수 있다. 이 저장 공간을 이용하여 경로 지속 확률을 저장할 경우 몇 번의 타입 변환과 곱셈 및 나눗셈을 통해 데이터양을 줄이고 대역폭을 줄일 수 있게 된다. 대신 충분히 작은  $\alpha$  값을 사용하여 경로 지속 확률이 0으로 수렴하는 것을 방지할 필요가 있다(그림 2 참조).

이것으로 DDVM에 대한 모든 설명을 마쳤다. 다음 장에서는 DDVM과 다른 알고리즘의 성능을 시뮬레이션을 통해 알아보도록 한다.

## 4. 성능 평가

성능 평가를 위해서 사용한 시뮬레이션 툴은 NS-2 [16](version 2.28)이다. 하위 프로토콜로는 MAC 802.11만을 사용하고, 라우팅 프로토콜은 사용하지 않는다. 모든 시뮬레이션에 대해서 2500m × 2500m의 평면 위에 400개의 mobile 노드가 있다고 가정하고, 이 노드들은 반경 200m의 propagation range, 2.0Mbps의 대역폭을 가지며, 0~20m/sec의 속도로 이동한다. 노드의 이동은 random waypoint 모델[9]을 사용하고, 높은 이동성에서의 전송률을 평가하기 위해 정지시간(pause time)은 없다. Source 노드는 매 초마다 2개의 512byte

패킷을 정기적으로 보내고, 총 시뮬레이션 시간은 2000 초이다. 각 항목은 10회의 실험을 평균하였다.

4.1 성능 평가 항목

멀티캐스팅 효율은 데이터 전송률(Delivery ratio), 패킷 전송 효율(Packet transmission efficiency), 데이터 전송 효율(Data packet transmission efficiency), 제어 정보 효율(Control information efficiency)로 평가한다.

4.1.1 전송률(Delivery Ratio, %)

$G_1, \dots, G_p, p < \infty$  멀티캐스팅 그룹이 있다고 할 때, source 노드  $s \in G_i$ 의 destination 노드 수를  $l^{G_i(s)}$ 라 하고,  $s$ 가 보낸 데이터 패킷 수를  $w^{G_i(s)}$ , destination 노드들이 중복없이 받은 데이터의 총 수를  $m^{G_i(s)}$ 라 할 때 전송률은

$$\sum_{i=1}^p \sum_{s \in G_i} m^{G_i(s)} / \sum_{i=1}^p \sum_{s \in G_i} l^{G_i(s)} w^{G_i(s)}$$

이다. 데이터 전송률은 멀티캐스팅 알고리즘의 성능을 평가하는 가장 중요한 척도이다[7].

4.1.2 패킷 전송 효율(Packet Transmission Efficiency)

$$M = \sum_{i=1}^p \sum_{s \in G_i} m^{G_i(s)}$$

이라 할 때 패킷 전송 효율은 시뮬레이션 기간동안 보내진 모든 패킷의 수를  $M$ 으로 나눈 값이다. 데이터 패킷, 경로 정보 전달 및 그룹 구성에 사용되는 패킷 수를 합한 후,  $M$ 으로 나누면 도착한 데이터 패킷 당 필요한 패킷 전송 횟수가 나온다. MANET에서의 잦은 패킷 전송은 노드의 전원 및 대역폭을 소모시킬 뿐만 아니라 다른 패킷과 간섭을 일으켜 패킷 손실을 유발하기 때문에 이는 좋은 평가 기준이 된다.

4.1.3 데이터 전송 효율(Data packet transmission efficiency)

시뮬레이션 기간동안 모든 노드가 보낸 데이터 패킷의 총 합을  $g$ 라고 했을 때, 데이터 전송 효율은  $g/M$ 이다. 즉, 하나의 패킷을 목적지까지 전송하는데 몇 번의 데이터 패킷 전송이 필요한가를 나타낸다.

4.1.4 제어 정보 효율(Control information efficiency, bytes)

시뮬레이션 기간동안 보내진 제어 패킷(DDVM의 경우 Source Announce, Route Update, Join Request, Explicit Passive Ack, End Group)에 저장된 정보량(byte)과 Multicast Data 패킷에 저장된 제어 정보량(byte)을 합한 값을  $M$ 에 데이터 크기(512 byte)를 곱한 값으로 나눈 값이다. 즉, 데이터 1byte를 목적지까지 보내는데 몇 byte의 제어 정보가 사용되는가를 나타낸다. 이를 통해 제어 정보의 효율성을 판단할 수 있다.

4.2 비교 알고리즘 및 파라미터 설정

성능 비교를 위해 선택한 ODMRP 알고리즘은 여러 MANET 멀티캐스팅 알고리즘들[14,17,18] 중에서도 단순성, 효율성, 확장성을 갖춘 알고리즘으로 알려져 있고 [7], 최근에 제안된 멀티캐스팅 알고리즘들이 가장 많이 비교하는 알고리즘 중 하나이다[6,19]. 또한 source 경로 정보 전달 이외의 그룹구성 방식에 있어서 DDVM과 가장 유사하다. ODMRP는 기존의 DVR 방식의 멀티캐스팅 알고리즘에 비해 전송률이 뛰어나다고 알려져 있으며[20], 따라서 ODMRP와 DDVM를 비교하는 것이 경로 정보 전달 방식을 평가하는데 유용하다.

ODMRP의 동작방식을 간단히 살펴보면 다음과 같다. 먼저 멀티캐스팅 참여자 중에서 source 노드를 중심으로 한 그룹을 구성한다. Source 노드는 일정한 시간마다 Join Query 메시지를 flooding하고, 이 메시지를 받은 노드 중에서 destination 노드는 이 메시지에 대한 Acknowledge 패킷을 보냄으로서 그룹을 구성하게 된다. ODMRP는 이 패킷을 받은 노드를 그룹에 일정 기간(forwarding\_node\_life\_time) 포함시킨다.

두 알고리즘을 명확하게 비교할 수 있도록 시뮬레이션 한 알고리즘들의 파라미터들을 표 2에 정리하였다. 경로 정보 전달 주기는 기본적으로 [7]의 실험 환경을 따랐다.

표 2 시뮬레이션에 사용한 파라미터

항 목		DDVM	ODMRP
공통 항목	경로 정보 전달 주기	3초	3초
	그룹 구성 주기	3초	3초
	forwarding_node_life_time	6초	6초
	그룹 형성 방법	mesh	mesh
DDVM	$\alpha$	0.98	없음
	Quantization	2 byte	
	Join Request 재전송	2회	
	no_join_request_interval	1초	
	join_request_timeout	0.05초	
	Join Request 실패 기록	사용	
중복 Join Request 방지	1초		

4.3 시뮬레이션 상황 설정

멀티캐스팅 알고리즘을 효율적으로 평가하기 위해서는 여러 시나리오에 대한 적응성을 살펴보는 것이 중요하다. 먼저 DDVM의 파라미터인  $\alpha$ 가 성능에 어떤 영향을 미치는지 알아보기 위해 source 1, destination 1개의 노드일 경우에 대하여 전송률을 비교하였다. 다음으로 하나의 source 노드를 가진 하나의 그룹에 대해 destination 노드 수를 1에서 49까지 변화시켜가면서 그룹 크기 면에서의 성능 변화를 관찰하였다. 다음으로 참

여자가 적은 다수의 그룹(source 노드 하나에 destination 노드 수 9, 그룹의 수는 1~10), 참여자가 많은 다수의 그룹(source 노드 하나에 destination 노드 39, 그룹의 수는 1~8)에 대해 각각 실험하였다. 이는 각 알고리즘이 여러 그룹을 얼마나 효율적으로 제어할 수 있는지와 네트워크 부하에 얼마나 효율적으로 대처하는지를 알아보기 위해서이다.

4.4 시뮬레이션 결과 분석

4.4.1  $\alpha$  파라미터

그림 2는 DDVM에서 사용하는 파라미터  $\alpha$ 의 변화에 따라 source 노드와 destination의 1:1 데이터 전송에서 전송률이 어떻게 변화하는지를 보여준다. 먼저  $\alpha$ 가 1인 경우에는 식 (4)에 따라 모든 경우에 경로 지속 확률이 1이 된다. 이럴 경우 모든 경로 정보들에 대한 우선순위가 같아지게 된다. 이때의 전송률은 25%정도로,  $\alpha$ 가 0.98일 때 61%와 비교할 때 현저히 낮다. 이는 경로 지속 확률 정보가 효율적인 멀티캐스팅 경로를 형성하는데 큰 역할을 하고 있음을 보여준다.  $\alpha$ 를 지속적으로 감소시키면  $\alpha$ 가 0.98일 때와 비슷한 전송률을 보이다가 0.5 이하가 되면 역시 현저히 낮은 전송률을 보인다. 이는 source 노드와 거리가 먼 노드들의 경로 지속 확률이 큰 감소율( $\alpha$ )에 의해 0에 가까워지고, 다른 노드들과의 값을 비교할 수 없는 수준에 이르기 때문이다. 따라서  $\alpha$ 를 선택하는데 있어서 1이 아닌 큰 값을 선택하는 것이 성능에 도움이 된다.

4.4.2 단일 그룹의 참여자 수에 따른 성능 변화

그림 3~6은 단일 그룹의 참여자 수에 따른 성능 변화를 보여준다.

그림 3에서 두 알고리즘 모두 참여자 수가 증가할수록 전송률이 높아짐을 알 수 있다. 이는 broadcast 특성을 이용할 수 있는 MANET의 특징 때문이다. DDVM

은 어떠한 경우에도 항상 ODMRP에 비해 높은 전송률을 보임을 알 수 있고, 특히 그룹의 수가 작은 경우에도 더 큰 차이를 보임을 알 수 있다. 이는 DDVM이 사용하는 경로 정보가 ODMRP가 사용하는 LSR 방식의 경로

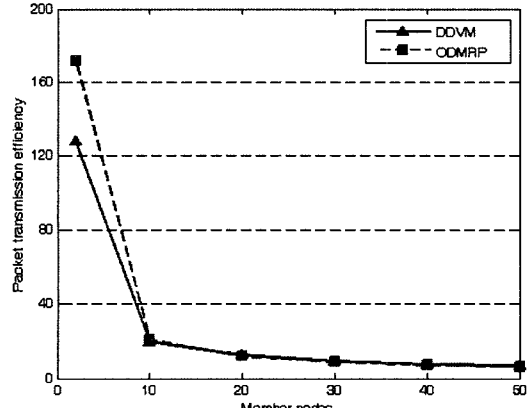


그림 3 단일 그룹에서의 전송률

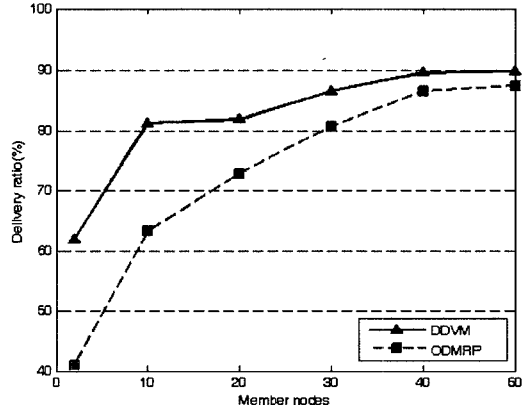


그림 4 단일 그룹에서의 패킷 전송 효율

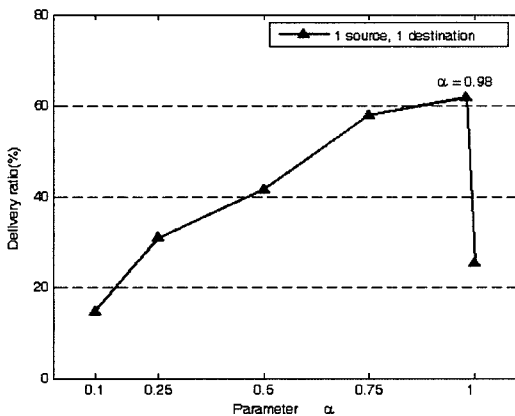


그림 2  $\alpha$ 에 따른 DDVM의 전송률 변화

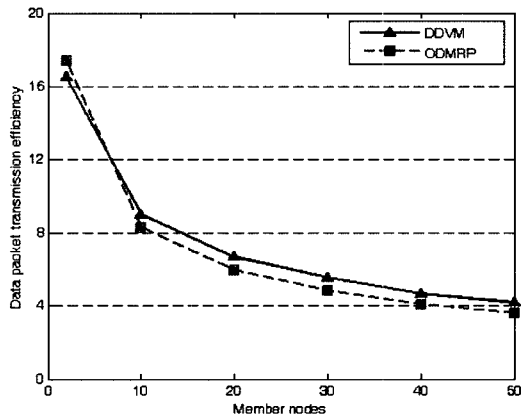


그림 5 단일 그룹에서의 데이터 전송 효율



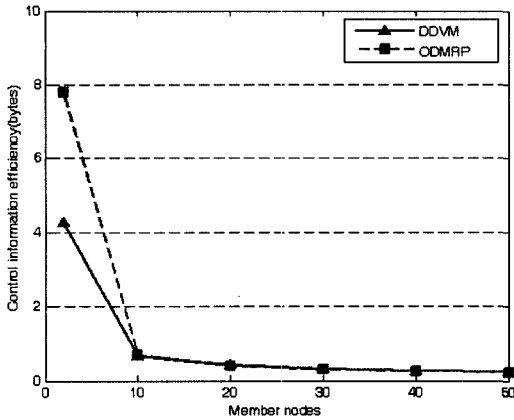


그림 6 단일 그룹에서의 제어 정보 효율

정보 보다 훨씬 우수함을 보여주는 것이다.

그림 4와 6을 보면 단일 그룹일 경우 작은 그룹에서 패킷 전송 효율 및 제어 정보 효율면에서 DDVM이 ODMRP보다 나은 면을 보인다. 큰 그룹에서의 효율은 두 알고리즘의 거의 비슷하다.

그림 5의 데이터 전송 효율면에서는 두 알고리즘이 거의 비슷한 성능을 내거나 ODMRP가 조금 나은 면을 보인다. 하지만 이 차이는 매우 작으며, 특히 전송률을 고려하여 생각해보면 오히려 DDVM이 더 낫다는 점을 알 수 있다.

#### 4.4.3 작은 그룹에서 그룹 수에 따른 성능 변화

그림 7~10에서는 작은 그룹의 그룹 수에 따른 성능 변화를 보여준다. 두 알고리즘 모두 그룹의 수가 증가할수록 대역폭을 많이 소모하게 되고, 전파 간섭에 의한 패킷 손실이 늘어나면서 전송률이 저하된다. 하지만 DDVM에 비해 ODMRP의 전송률 저하가 더 크다는 것을 알 수 있다. 그림 8의 패킷 전송 효율을 보면, DDVM의 경우 그룹의 수가 늘어나면 늘어날수록 도착

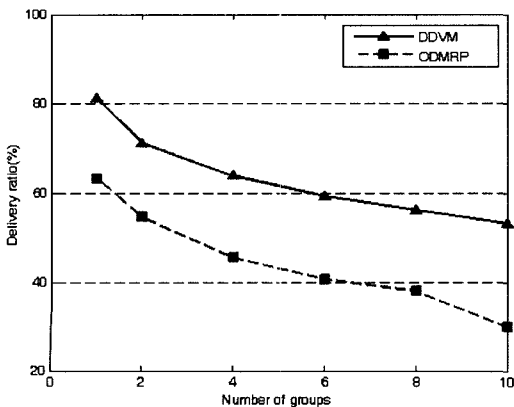


그림 7 다수의 작은 그룹에서의 전송률

한 데이터 패킷 당 전체 패킷 전송 횟수가 줄어드는 반면, ODMRP는 큰 폭으로 늘어났다는 것을 알 수 있다. 또한 그림 10에서 제어 정보 효율면에서도 DDVM은 데이터 1 byte당 필요한 제어 정보 byte가 ODMRP에

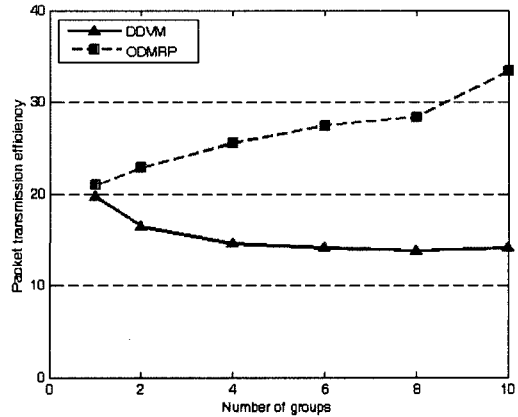


그림 8 다수의 작은 그룹에서의 패킷 전송 효율

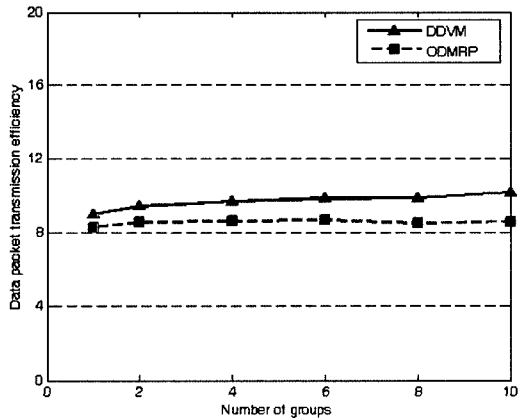


그림 9 다수의 작은 그룹에서의 데이터 전송 효율

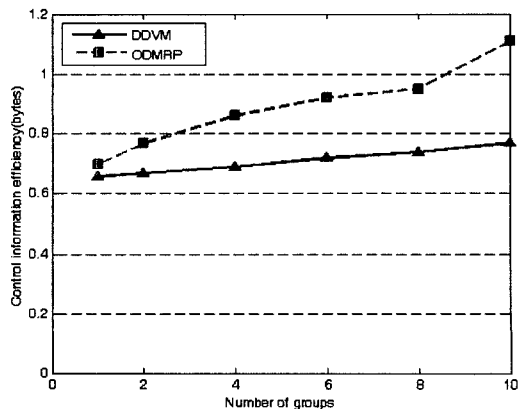


그림 10 다수의 작은 그룹에서의 제어 정보 효율

비해 적은 비율로 늘어남을 알 수 있다. 그림 9에서의 데이터 패킷 효율은 미세한 차이로 ODMRP가 더 좋지만, 전송률에 전송률을 고려할 때 전체 성능은 DDVM이 앞선다는 것을 알 수 있다.

이러한 실험 결과가 나타나는 것은 ODMRP가 사용하는 LSR 방식의 경로 정보 전달이 DVR 방식을 사용하는 DDVM에 비해 더 많은 대역폭을 소모하고, 이로 인한 패킷 손실이 더 많이 일어나기 때문이다. 즉, ODMRP는 데이터 패킷 전달을 위해 사용해야 할 대역폭을 경로 정보 전달을 위해 사용하고, 그룹의 수가 늘어남에 따라 더 많은 제어 정보 패킷을 보냄으로서 전파 간섭에 의한 패킷 손실이 훨씬 더 많이 일어나게 된다. 반면 DDVM에서는 그룹의 수가 늘어나더라도 경로 정보 패킷의 크기만 약간 늘어날 뿐, 패킷 전송 횟수는 일정하다.

4.4.4 큰 그룹에서 그룹 수에 따른 성능 변화

그림 11~14는 source 노드 수 1, destination 노드 수 39(총 40 노드)의 비교적 큰 멀티캐스팅 그룹이 여러 개 존재할 때에 성능 변화를 보여준다. 그림 11의 전송률에 있어서 4. 4. 3의 작은 그룹에 비해 전송률의 차이는 적다. 이는 두 알고리즘 모두 broadcast 특성을 이용하므로 그룹이 클수록 대체 경로가 더 많기 때문이다. 하지만 그룹 수에 따른 전체적인 성능 변화 양상은 4. 4. 3과 유사하게 진행된다. 그룹의 수가 늘어나면 늘어날수록 두 알고리즘의 전송률 차이가 커지고, 그림 12의 패킷 전송 효율면에서도 ODMRP의 경우 더 나빠지지만, DDVM은 오히려 더 좋아짐을 알 수 있다. 그림 13의 데이터 전송 효율은 ODMRP가 조금 낮지만, 전송률을 고려하여 볼 때 DDVM이 전체적으로 더 나은 성능을 보임을 알 수 있다.

실험을 통해 관찰한 DDVM의 특성을 종합해보면 다음과 같다. 전송률 면에서는 어떠한 상황에서도 DDVM이

ODMRP에 비해 좋은 성능을 보인다. DDVM은 그룹의 수가 증가하면 증가할수록 패킷 전송효율이 좋아지고, 제어정보 효율면에서도 ODMRP보다 좋거나 비슷한 성

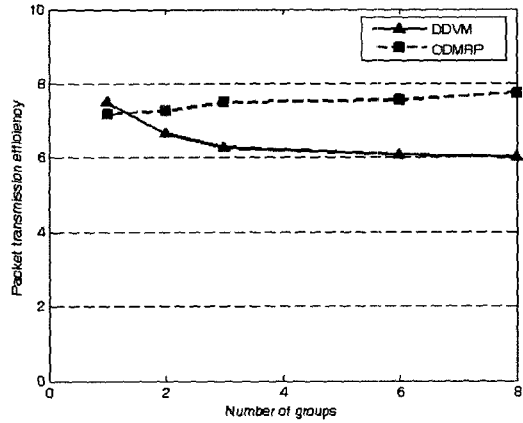


그림 12 다수의 큰 그룹에서의 패킷 전송 효율

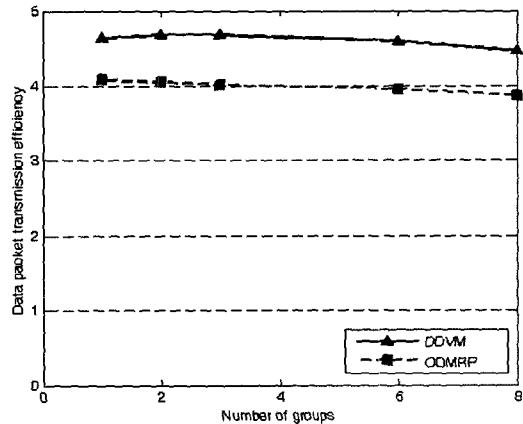


그림 13 다수의 큰 그룹에서의 데이터 전송 효율

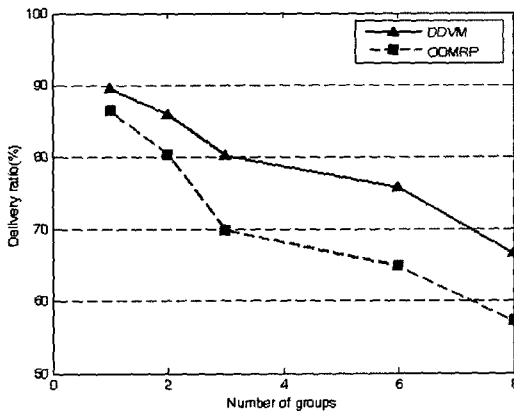


그림 11 다수의 큰 그룹에서의 전송률

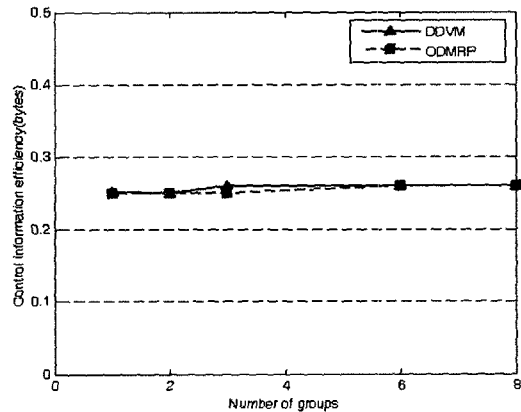


그림 14 다수의 큰 그룹에서의 제어 정보 효율

능을 발휘한다. 데이터 전송 효율면에서는 ODMRP가 조금 우수하지만, 전체적인 성능을 종합하여 볼 때 DDVM이 더 우수한 성능을 보임을 알 수 있다.

## 5. 결론 및 추후과제

DDVM는 mesh 방식의 MANET 멀티캐스팅 알고리즘으로서 DVR의 적은 패킷 전송이라는 장점을 살리고, 경로 지속 확률을 적용하여 단점을 보완함으로써 기존의 멀티캐스팅 알고리즘에 비해 높은 전송률을 발휘한다. 2500m×2500m, 400 노드의 큰 네트워크, 그리고 빠른 이동성을 가진 시뮬레이션 환경에서 좋은 성능을 발휘함으로써 경로 정보의 효율성을 보여준다. 이러한 경로 지속성을 고려한 경로 정보는 멀티캐스팅에서의 전송률 및 전송효율이 좋음은 물론 일반 유니캐스팅 알고리즘으로서도 충분히 활용가능한 가치가 있다. 따라서 향후 이러한 경로 정보를 바탕으로 다른 라우팅 알고리즘들의 장점을 수용하여 보다 나은 라우팅 알고리즘을 개발하고자 한다. 또한 현재 많이 연구되고 있는 계층적 멀티캐스팅 방법을 적용함으로써 보다 확장성 있는 라우팅, 멀티캐스팅 알고리즘을 개발하고자 한다.

## 참 고 문 헌

[1] M. Abolhasan, T. Wysocki and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks 2*, pp. 1-22, 2004.

[2] N. Sadagopan, F. Bai, B. Krishnamachari and A. Helmy, "PATHS: Analysis of PATH duration statistics and their impact on reactive MANET routing protocols," in *Proceeding of ACM MobiHoc*, pp. 245-256, 2003.

[3] E. Royer and C. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," *Mobile Computing and Networking*, pp. 207-218, 1999.

[4] S. Lee, W. Su and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *ACM Mobile Networks and Applications*, vol. 6, pp. 441-453, 2002.

[5] C. Gui and P. Mohapatra, "Efficient overlay multicast for mobile ad hoc networks," in *Proceeding of IEEE WCNC'03*, vol. 2, pp. 1118-1123, 2003.

[6] L. Ji and M. S. Corson, "Explicit multicasting for mobile ad hoc networks," *ACM Mobile Networks and Applications*, vol. 8, pp. 535-549, 2003.

[7] S. Lee, W. Su, J. Hsu, M. Gerla and R. Bagrodia, "A performance comparison study of ad hoc wireless multicast protocols," in *Proceeding of IEEE INFOCOM'00*, vol. 2, pp. 565-574, 2000.

[8] M. Steenstrup, "Routing in communications net-

works," *Prentice Hall*, Inc, Englewood Cliffs, NJ, 1995.

- [9] T. Camp, J. Boleng and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communication & Mobile Computing (WCMC)*, vol. 2, no. 5, pp. 483-502, 2002.
- [10] T. Clausen, P. Jacquet, A. Laouiti, et al, "Optimized link state routing protocol," *draft-ietf-manet-olsr-06.txt*, 2002.
- [11] M. Gerla, X. Hong and G. Pei, "Fisheye state routing protocol(FSR) for ad hoc networks," *draft-ietf-manet-fsr-03.txt*, 2002.
- [12] C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," in *Proceeding of IEEE 2nd Mobile Computing Systems and Applications*, pp. 90-100, 1999.
- [13] C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers," in *Proceeding of SIGCOMM'94*, pp. 234-244, 1994.
- [14] J. Xie, R. R. Talpade, A. Mccauley and M. Liu, "AMRoute : Ad hoc multicast routing protocol," *ACM Mobile Networks and Applications*, vol. 7, pp. 429-439, 2002.
- [15] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in graphs," *Mathmetica Japonica*, pp. 573-77, 1980.
- [16] K. Fall and K. Varadhan, editors, "NS notes and documentation," *The VINT Project, UC Berkely, LBL, USC/ISI and Xerox PARC*, 1997.
- [17] J. J. Garcia-Luna-Aceves and E.L. Madruga, "The core-assisted mesh protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1380-1394, 1999.
- [18] C.W. Wu, Y. C. Tay, "AMRIS: A multicast protocol for ad hoc wireless networks," in *Proceedings of IEEE MILCOM'99*, vol. 1, pp 25-29, 1999.
- [19] H. Moustafa, H. Labiod, "Multicast routing in mobile ad hoc networks," *Telecommunication Systems*, 2004.
- [20] R. S. Gray, D. Kotz, C. Newport, et al. "Outdoor experimental comparison of four ad hoc routing algorithms," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, Session. Routing in ad hoc networks, pp. 220 - 229, 2004.



이 세 영

2004년 서강대 컴퓨터학과 학사. 2006년 동대학원 석사. 현재 LS 산전 중앙연구소 연구원, 관심분야는 최적화 알고리즘, 무선 네트워크 등



장 형 수

1994년 미국 Purdue University, 전기및  
컴퓨터공학과 졸업, 동대학원 석사, 박사  
(1996, 2001). 2001년 9월~2002년 6월  
Institute for Systems Research, Uni-  
versity of Maryland, College Park,  
Research Associate, 2002년 6월~2003  
년 2월 고려대학교 정보통신 기술 공동연구소 연구교수,  
2003년 3월~현재 서강대학교 공과대학 컴퓨터학과 조교수