MANET 환경에서의 이동 에이전트를 이용한 동적 클러스터링 기법

임원택° 김구수 선승상 엄영익
성균관대학교 컴퓨터공학과
{imcliff°, gusukim, threes, yieum}@ece.skku.ac.kr

Mobile Agent based Dynamic Clustering scheme in MANET

Won-tack Lim°, Gu Su Kim, Seung Sang Sun, Young Ik Eom
Department of Computer Engineering, Sungkyunkwan University

요 약

요 약 본 논문은 이동 애드혹 네트워크에서 이동 에이전트를 이용하여 동적으로 클러스터링을 구성하는 기법에 관한 것이다. 기존에 제안된 이동 애드혹 네트워크에서의 클러스터링 기법은 클러스터의 크기가 고정되어 있기 때문에 네트워크의 상태나 노드들의 이동성에 따라 클러스터 재구성의 오버헤드가 발생하였다. 본 제안 기법에서는 네트워크의 상태에 따라 클러스터 크기의 최대 임계치와 최소 임계치를 설정하고 이에 따라 이동 에이전트를 이용하여 클러스터를 병합 혹은 분할하면서 클러스터의 크기를 임계치 내에서 일정하게 유지시킴으로써, 클러스터 재구성의 오버헤드와 클러스터 내부의 경로 탐색의 오버헤드를 줄일 수 있다.

1. 서 론

최근 이동 단말기기의 보급과 유비쿼터스 컴퓨팅에 관한 관심의 증가로 인해 무선 통신에 관련된 연구가 활발하게 이루어지고 있다. 이 중 이동 노드로 이루어진 자율적이고 수평적인 네트워크 환경인 MANET(Mobile Ad-hoc Network)에 대한 연구가 IETF의 MANET WG을 중심으로 활발히 진행되고 있다. [1]

MANET는 기간망을 쓰지 않는 수평적 구조로 이루어져 있기 때문에 확장성이 떨어지는 문제점을 가지고 있다. table-driven 기반의 순향적(proactive) 라우팅 기법의 경우 n개의 노드가 통신을 할 때 발생하는 오버헤드는 $O(n^2)$가 된다.[2]

이러한 문제점을 해결하기 위해서 MANET에서의 클러스터링 기법이 연구되고 있다. 클러스터링 기법은 애드혹 네트워크상의 노드들을 계층적으로 관리해서 보다 효율적으로 네트워크를 운영할 수 있게 해준다. 하지만 네트워크 상태와 관계없이 고정적인 크기의 클러스터를 구성하게 되면 클러스터를 유지하는데 오버헤드가 발생하게 된다.

이를 해결하기 위해서 본 논문에서는 이동 에이전트를 이용해서 MANET에서 동적으로 클러스터를 구성하는 기법을 제안한다. 본 논문의 제안 기법에서는 네트워크 환경에 따라 클러스터 크기의 최대 임계치와 최소 임계치를 설정하고, 이동 에이전트를 이용해 클러스터를 병합, 분할해서 클러스터의 크기를 임계치 내의 수준으로 일정하게 유지함으로써, 기존 클러스터링 기법에서 나타나는 클러스터 유지 오버헤드를 줄일 수 있다.

본 논문의 2장에서는 기존의 MANET에서의 클러스터링 방법과 이동 에이전트를 이용한 MANET 라우팅 방법에 대해 살펴보고, 3장에서는 이동 에이전트를 이용해서 동적으로 클러스터를 구성하는 기법을 제안한다. 4장에서는 결론과 향후 연구 방향에 대해서 설명한다.

2. 관련연구

2.1 이동 애드 혹 네트워크에서의 클러스터링 기법

대표적인 이동 애드 혹 네트워크에서의 클러스터링 기법 기법으로는 가장 작은 ID값을 가지는 노드를 클러스터 헤드로 선출하는 LID(Lowest ID Clustering)기법[3]과 도메인 집합(Domain Set)을 기반으로 하는 CDS(Connected Domain Set)[4]가 있다.

하지만 이러한 클러스터링 기법은 기본적으로 1홉 크기의 클러스터를 구성하므로 클러스터를 빈번하게 재구성해야하는 오버헤드가 존재한다. 3홉 크기의 클러스터를 구성하는 3hBAC(3-hop Between Adjacent Clusterheads) 기법[5]이 존재하지만, 3홉의 고정적인 크기의 클러스터를 구성해야 하는 한계성을 가진다.

2.2 이동 에이전트를 이용한 라우팅 기법

MANET 환경의 라우팅 프로토콜은 table-driven 기반의 순향적(proactive) 방식과 on-demand 방식의 반응적(reactive) 방식으로 분류될 수 있다. 하지만 두 방식 모두 문제점을 가지고 있다. 순향적 방식은 경로를 유지하기 위해 많은 네트워크 트래픽을 발생시키고, 반응적 방식은 경로 설정 과정에서 지연시간과 경로 실패로 인한 경로 재구성의 오버헤드가 발생한다.

이 같은 문제점을 해결하기 위해서 이동 에이전트를 이용한 라우팅 기법인 MARP(Multi-Agent Routing Protocol)가 제안되었다.[6] 이 기법은 각 노드가 이동 에이전트를 통해 MANET의 네트워크 토폴로지를 파악해서 목적지 노드까지의 경로를 찾아가는 방식으로, 순향적 라우팅 방식보다 적은 트래픽을 이용해서 지연시간 없이 경로를 설정할 수 있다. 하지만 네트워크의 크기가 늘어날수록 이동 에이전트가 유지해야 하는 정보가 기하급수적으로 늘어나는 문제점을 가지고 있다.

3. 제안기법

3.1 개요

본 제안기법은 한 클러스터 내의 노드들에게 연결 정보를 제공하는 Routing Mobile Agent(RMA)와 클러스터 정보를 가지고 다니면서 클러스터 헤드에게 인접 클러스터의 정보를 제공하는 Clustering Mobile Agent(CMA)을 이용한다.

각 노드는 RMA에게서 제공받은 노드 간 연결 정보를 이용해서 목적지 노드까지의 경로를 관리하고, 이웃 노드와의 연결성(connectivity)에 따라 클러스터와의 멤버십을 유지한다. 클러스터

헤드는 CMA에게서 제공받은 클러스터 정보와 네트워크 상태에 따라 설정된 최소 임계치와 최대 임계치를 이용해서 클러스터의 분할 혹은 병합 과정을 수행한다.

만일 클러스터의 크기가 최소 임계치보다 작다면 인접 클러스터와의 병합을 수행하고, 클러스터의 크기가 최대 임계치보다 크다면 클러스터 분할을 수행한다. 이와 같은 클러스터의 병합과 분할 과정을 통해 클러스터의 크기를 임계치 내의 수준에서 일정하게 유지시킴으로써 큰 규모(Large scale)의 MANET에서 클러스터 재구성의 오버헤드와 클러스터 멤버 수가 증가할수록 기하급수적으로 증가하는 라우팅 정보 유지 오버헤드를 줄일 수 있다.

3.2 패킷 및 자료구조

그림 1에서는 MANET의 모든 노드들이 유지하는 테이블인 Neighbor Table(NT)를 보인다.

NID	CID	Connectivity

<Neighbor Table>

그림 1 모든 노드가 유지하는 테이블

NT는 이웃 노드의 정보를 나타내는 테이블로써 NID는 이웃 노드의 ID, CID는 이웃 노드의 클러스터 ID, Connectivity는 이웃 노드와의 연결성을 나타낸다.

그림 2에서는 클러스터 헤드가 유지하는 테이블인 Neighbor Cluster Table (NCT)를 보인다.

NCID	NCHID	GW	NoM	MaxH

<Neighbor Cluster Table (NCT)>

그림 2 클러스터 헤드가 유지하는 테이블

NCT는 이웃 클러스터의 정보를 나타내는 테이블로써, NCID는 이웃 클러스터의 ID, NCHID는 이웃 클러스터의 클러스터 헤드 ID, GW는 이웃 클러스터로 가기 위한 게이트웨이 역할을 하는 노드의 ID를 나타낸다. 또한 이웃 노드와의 병합 가능성을 알기 위한 정보로써 NoM(Number of Member)는 클러스터 멤버의 수, MaxH(Maximum Hop)은 클러스터 내부의 라우팅 경로 중 가장 긴 경로정보를 나타낸다.

CMA는 클러스터의 정보를 가지고 다니면서 병합 혹은 분할작업을 수행한다. 그림 3에서 CMA가 유지하는 테이블인 Cluster Information Table (CIT)를 보인다.

CID	CHID	NoM	MaxH	(NCID, GW)

<Cluster Information Table>

그림 3 CMA 가 유지하는 테이블

CIT은 클러스터의 정보들을 나타내는 테이블로써, CID는 클러스터 ID, 해당 클러스터의 클러스터 헤드 ID, NoM은 해당 클러스터를 구성하는 노드의 개수, MaxH은 해당 클러스터 내부에서 가장 긴 경로를 나타낸다. 마지막 필드는 이웃 클러스터의 ID와 게이트웨이 노드의 순서쌍들을 가지고 있다.

MANET의 모든 노드들과 RMA, CMA는 특정 클러스터의 모든 노드들의 연결성 정보를 가진다. 그림 4에서는 연결성 정보를 유지하는 Connectivity Table(CT)를 보인다.

	n1	n2	n3
n1	-	1to2	1to3
n2	2to1	-	2to3
n3	3to1	3to2	-

<Connectivity Table>

그림 4 연결성 테이블

CT의 각 항목은 노드들의 연결성을 나타낸다. 노드들의 연결성은 노드의 속도, 노드 간의 거리, 노드의 신호 도달 범위에 의해 결정된다.

3.3 동작 과정

본 제안 기법은 클러스터 구성, 클러스터의 통합, 클러스터의 분할의 세 가지 동작으로 구성된다.

네트워크가 초기화 되었을 때 클러스터를 구성하는 동작을 수행한다. 각 노드는 자신의 이동성에 기반한 우선순위를 이웃 노드에게 브로드캐스팅 한다. 이웃 노드로부터 수신한 우선순위보다 자신의 우선순위가 높으면 자신이 클러스터 헤드가 되었음을 인식하고 이웃 노드들에게 클러스터 정보와 클러스터 멤버 정보를 브로드캐스팅한다. 클러스터에 속하지 않은 노드가 클러스터 정보를 받으면 클러스터 헤드에게 ACK 신호을 전송한다.

초기 클러스터가 구성되고 나면 각 클러스터 헤드는 RMA와 CMA를 생성하여 이주시킨다. 그리고 CMA가 전달해주는 정보에 따라 병합과 분할 동작을 수행한다.

병합 동작은 자신의 클러스터 규모와 이웃 클러스터의 규모가 각각 최소 임계치보다 작을 때 수행된다. 클러스터 병합 작업은 클러스터링 에이전트를 이용해 두 클러스터의 CT를 교환함으로써 이루어진다. CT의 정보를 바탕으로 노드 간 연결성이 가장 좋은 노드가 병합된 클러스터의 클러스터 헤드로 선출된다. 클러스터 헤드로 선출된 노드는 클러스터 헤더 ID, 클러스터 ID, 클러스터 멤버들의 ID를 이웃 노드에게 브로드 캐스팅 한다. 그림 6에서는 두 클러스터의 규모가 일정 기준보다 작을 때 수행되는 클러스터 병합 동작의 예를 보인다.

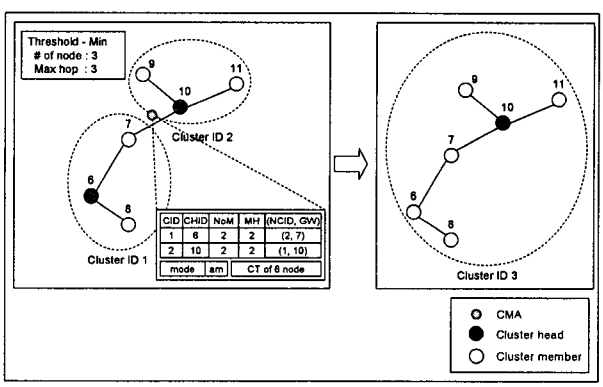

그림 5 클러스터의 병합

분할 작업은 자신의 클러스터 규모가 최대 임계치보다 클 때 수행된다. 클러스터 헤드는 CT를 이용해서 후보 클러스터 헤드를 선출하고, 선출 된 후보 클러스터 헤드에게 CMA를 통해 분할 이후의 CT와 CIT를 전송한다. 해당 정보를 전송받은 후보 클러스터 헤드는 이웃 노드에게 클러스터 정보와 클러스터 멤버 정보를 브로드캐스팅 함으로써 클러스터 헤드가 된다. 그림 6에서는 클러스터의 규모가 일정 기준보다 클 경우 수행되는 클러스터 분할 동작의 예를 보인다.

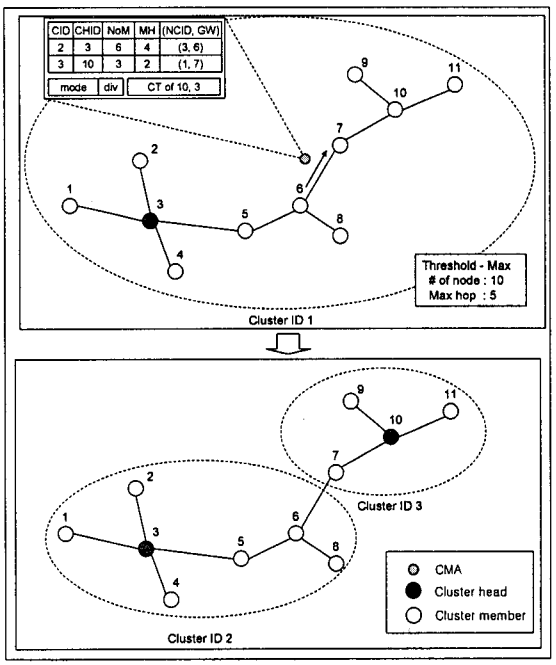

그림 6 클러스터의 분할

4. 성능평가

본 논문에서는 기존의 클러스터링 기법과 제안된 클러스터링 기법의 효율성을 알아보기 위해서 한정된 공간에서 클러스터의 개수와 클러스터 내에서 이동 에이전트가 전달해야하는 정보의 크기를 시뮬레이션 하였다. 시뮬레이션에서는 가로 3km, 세로 1.5km의 공간 안에서 각각 100개, 300개의 노드가 MANET을 구성한다고 가정하였다.

동일한 네트워크에서 클러스터의 개수가 많아질 경우 클러스터를 재구성하는 오버헤드가 증가한다. 그림 7에서 병합 기준에 따른 클러스터 개수의 변화를 보인다.

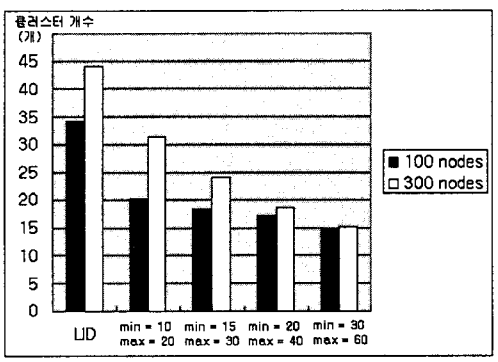

그림 7 병합 기준에 따른 클러스터의 개수

그림의 x축에 나타난 값은 클러스터 멤버 수의 최소 임계치와 최대 임계치이다. 임계치가 커질수록 클러스터의 크기도 함께 커지기 때문에 클러스터의 개수가 줄어드는 것을 볼 수 있다. 본 제안 기법으로 클러스터를 구성하였을 경우 LID(Lowest ID) 기반의 클러스터를 구성하였을 경우보다 클러스터의 개수가 25%~65%까지 줄어듦을 확인할 수 있었다. 이와 같은 결과는 클러스터 재구성 오버헤드의 감소로 이어질 수 있다.

표 1에서 병합 기준에 따라서 이동 에이전트가 유지해야하는 경로 정보와 클러스터 정보의 크기를 나타낸다.

Node 수	min=10 max=20	min=15 max=30	min=20 max=40	min=30 max=60	MARP
100개	388	478	547	702	160000
300개	1461	2479	4118	6233	1440000

표 1 이동 에이전트가 유지해야 할 정보의 크기 (단위 : byte)

연결성 정보를 유지하기 위해서는 노드 수의 제곱에 비례하는 정보를 가지고 있어야 한다. MARP는 클러스터 구성없이 노드의 정보를 관리하므로 이동 에이전트는 노드의 수가 100개일 경우 160KB, 300개일 경우 1.44MB 크기의 정보를 전송해야 한다. 하지만 본 논문에서 제안한 클러스터링 기법을 사용할 경우 노드의 수가 300개일 경우에도 라우팅 정보와 클러스터링 정보를 모두 더하여도 클러스터 당 최대 6.2KB 크기의 정보만을 전송함으로써 클러스터의 경로를 유지할 수 있었다.

5. 결론 및 향후 과제

본 논문에서는 MANET에서 이동 에이전트를 이용해서 동적으로 클러스터를 구성하는 방법을 제안하였다. 기존의 클러스터링 기법은 클러스터의 크기가 작고 고정적이기 때문에 클러스터 재구성의 오버헤드가 발생한다. 본 논문에서는 네트워크의 상태에 따라 최대 임계치와 최소 임계치를 생성하고, 그에 따라 일정한 크기로 클러스터의 크기를 유지하므로 클러스터 재구성의 오버헤드와 클러스터 내부의 라우팅 오버헤드를 줄일 수 있다.

향후 연구 과제로는 MANET의 크기와 노드의 개수, 그리고 노드의 이동성에 따른 최적의 클러스터 크기를 구해내는 연구가 필요하다.

참고문헌

[1] J. Z. Sun, "Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing," In Proc. ICII 2001 International Conference on Info-tech and info-net, Vol. 3, pp. 316-321, 2001.

[2] E. M. Bedling-Royer, "Hierarchical Routing in Ad Hoc Mobile Networks," Wireless Communications and Mobile Computing, Vol. 2, no. 5, pp. 515-32, 2002.

[3] A. Ephermides, J. E. Wieselthier, and D. J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," in Proc. IEEE, Vol. 75, pp. 56-73, 1987.

[4] J. Wu and H. L. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," Proc. of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7-14, 1999.

[5] J. Y. Yu and P. H. J. Chong, "3hBAC (3-hop between Adjacent Clusterheads): A Novel Non-overlapping Clustering Algorithm for Mobile Ad Hoc Networks," in Proc. IEEE Pacrim'03, Vol. 1, pp. 318-21, 2003.

[6] R. R. Choudhury, K. Paul, and S. Bandyopadhyay, "MARP: A Multi-Agent Routing Protocol for Mobile Wireless Ad Hoc Networks," Autonomous Agents and Multi-Agent System, Vol. 8, No. 1, pp. 47-68, 2004.