

Ad-Hoc 네트워크에서 클러스터 수를 최소화하기 위한 방안

A Method for Minimizing the Number of Clusters in Ad-Hoc Networks

방 상 원*
Sang-Won Bang

요 약

Ad-Hoc 네트워크에서 클러스터 구조는 여러 채널을 효율적으로 사용 가능하게 하고 제어메시지의 교환수를 감소시키며 네트워크의 확장성을 증가시킨다. 또한 클러스터 구조는 Ad-Hoc 네트워크에서 방송되는 메시지의 수를 감소시키기 위해 사용된다. 이를 위해 임의의 클러스터 구조는 작은 수의 클러스터를 가지는 것이 바람직하다. 일반적으로 연결성 기반의 클러스터 구성방법은 다른 방법들에 비해 작은 수의 클러스터를 생성한다. 그러나 연결성 기반의 방법은 네트워크의 위상에 따라 연결성을 고려하지 않는 방법보다 더 많은 클러스터를 생성할 경우도 있다. 본 논문에서는 클러스터 구성을 2단계로 나누어 수행하는 클러스터 구성방법을 제안한다. 제안방법은 첫 번째 단계에서 최소연결성을 가진 노드가 클러스터 구성을 시작한다. 이때 최소연결성의 노드들을 가능한 많이 포함하는 작은 수의 클러스터를 생성하기 위하여 일련의 조정절차를 이용한다. 두 번째 단계에서는 첫 번째 단계에서 생성된 클러스터에 포함되지 않은 노드들을 대상으로 최대 연결성을 가진 노드가 클러스터 구성을 시작한다. 제안된 방법은 첫 번째 및 두 번째 단계를 거쳐서 기존 방법보다 더 작은 수의 클러스터를 완전히 분산된 방법으로 생성한다. 본 논문에서 제안한 방법은 실험결과에 의해 LIDCP(3)와 HCCP(3)보다 우수한 것으로 평가되었다.

Abstract

In Ad-Hoc network, the cluster structure enables effective use of multiple channels, reduces the number of control messages, and increase the scalability of network. Also, it is employed for reducing the number of broadcast messages in an Ad-Hoc network. With the consideration of these advantages, it is desirable that a cluster structure keeps a few clusters in the network. Generally, the cluster formation scheme based on connectivity yields fewer clusters than the other schemes. However, the connectivity based scheme may yield even more clusters than the other schemes according to the network topology. In this paper, a cluster formation scheme dividing the cluster formation into two phases is proposed. In the first phase, the lowest connectivity host in neighborhood initiates the cluster formation. Then, an adjustment procedure for affiliating a lot of the lowest connectivity hosts is employed. In the second phase, the hosts which were not affiliated to the first phase clusters are grouped into one or more clusters through criterions of connectivity and host ID. As a result, the proposed scheme yields a fewer clusters compared with existing other schemes in fully distributed method. The simulation results proves that our scheme is better than LIDCP(3) and HCCP(3).

Keyword : Ad-Hoc network, Cluster Formation

1. 서 론

Ad-Hoc 네트워크에서 이동노드들을 그룹으로 묶어 관리하면 그룹단위로 채널을 관리할 수 있고, 제어메시지의 교환부하 감소 및 이동성 관리가 용이해진다[1]. 또한 임의의 Ad-Hoc 네트워크를 클러스터 헤드들간의 네트워크로 만들어 네트

워크의 확장성(scalability)을 증가시킨다.

이동 노드들을 일단의 기준에 따라 묶은 그룹을 클러스터(cluster)라 하고 이렇게 이동 노드들을 그룹화 하는 것을 클러스터 구성(cluster formation)이라 한다. 이때 클러스터내에서 노드들간의 최대 경로길이에 따라 1홉 클러스터(clique), 2홉 클러스터, 4홉 클러스터 등으로 구분된다. 1홉 클러스터와 4홉 클러스터는 특수한 구조로 클러스터 구조를 유지·보수하는데 많은 노력과 비용

* 정 회 원 : 송원대학 컴퓨터정보과 교수
swbang@songwon.ac.kr(제 1저자)

이 소모된다. 따라서 본 논문에서는 가장 일반적인 2홉 클러스터를 가정한다.

클러스터는 클러스터의 서버적인 클러스터 헤드와 클러스터 헤드의 서비스를 받는 멤버노드로 구성된다. 클러스터 헤드의 일반적인 역할은 네트워크의 위상관리, 멤버들에게 자원을 할당, 지역 방송자 역할[3], 라우팅 등이 있다. 2홉 클러스터의 구조는 클러스터 헤드가 모든 멤버들과 직접 통신이 가능하도록 연결되어 있다. 따라서 클러스터 헤드가 데이터를 방송하면 모든 노드들이 그 데이터를 동시에 수신할 수 있다. 그러므로 이러한 특성은 reactive 방식의 라우팅 프로토콜(DSR[12], AODV[13])이나 CBRP[9]와 같은 클러스터헤드 기반의 라우팅 프로토콜에서 경로요청 작업(route request)동안 방송되는 메시지의 재전송 횟수를 줄이는데 유용하게 이용될 수 있다. 예를 들어, 임의의 DSR경로요청 메시지가 클러스터 구성된 Ad-Hoc 네트워크 내에 전파된다고 가정해보자. 이때 그 경로요청 메시지는 소스노드에서 시작되어 (클러스터 헤드, 게이트웨이, 이웃 클러스터 헤드, 게이트웨이, 게이트웨이, 이웃 클러스터 헤드..)와 같은 경로를 거쳐 목적지에 도달하게 된다. 즉 클러스터 내에서 멤버들끼리는 메시지를 전파하지 않게 되므로 목적지까지 전파하는데 필요한 재전송 횟수가 줄어들 것이다.

일반적으로 이동 노드의 연결성(connectivity)이 높은 노드를 클러스터 헤드로 선출하는 방법은 상대적으로 보다 작은 수의 클러스터로 전체 네트워크를 클러스터 구성한다고 알려져 있다[3,10]. 전체 네트워크를 작은 수의 클러스터로 구성하면, 다음과 같은 이점을 얻을 수 있다. 하나의 클러스터에 하나의 코드(CDMA)를 할당한다면, 전체 네트워크에 할당되는 코드의 수를 줄일 수 있기 때문에 코드 이용률을 높일 수 있다. 또한 클러스터 헤드들 사이의 평균 경로길이가 줄어들기 때문에 방송 메시지(즉 route discovery와 같은)의 재전송 횟수 감소 및 경로발견 시간을 단축시킨다. 이러한 이유로, 클러스터 구성에 관한 기존 연구들은

전체 네트워크를 작은 수의 클러스터로 구성하기 위해 노드의 연결성(connectivity)을 클러스터 헤드 선출을 위한 첫 번째 기준으로 설정하였다[1,3,10]. 그러나 기존 방법들은 클러스터 구성과정에서 가장 연결성이 높은 노드들이 클러스터 구성과정을 트리거하고, 탐욕스런(greedy) 방법으로 클러스터를 구성한다. 이러한 방법은 네트워크의 가장자리 부근에 존재하는 연결성이 낮은 노드들이 클러스터 헤드로 선출되는 경우를 자주 발생시킨다. 따라서 클러스터의 수는 ID기반의 클러스터 구성에 비해 줄어들지만, 탐욕스런 구성 방법에 의해 클러스터수를 더 이상 줄이지는 못한다.

본 논문에서는 연결성이 낮은 노드들이 먼저 클러스터 구성과정을 시작하여 연결성이 낮은 노드들을 클러스터에 먼저 포함시킨 후에, 연결성이 높은 노드들이 클러스터 구성을 시작하는 2단계 클러스터 구성 방법을 제안한다.

먼저 첫 번째 단계에서는 가장 연결성이 낮은 노드들을 임의의 클러스터들에 편입시킨다. 연결성이 가장 낮은 노드들이 헤드 권유메시지를 자신의 이웃 중 연결성이 가장 높은 노드에게 전송한다. 헤드 권유메시지를 수신한 노드는 자신의 이웃 중에서 연결성이 자신과 같거나 높은 노드가 존재하면, 헤드 권유메시지를 헤드 시도메시지로 변경해서 전송하고 잠시 대기한다. 여기서 잠시 대기하는 것은 다른 노드의 헤드 시도메시지를 수신하기 위함이다. 즉 이로 인해 경쟁상황이 발생하는데, 이때 결정기준은 권유메시지 수신횟수, 최소연결성 노드의 연결성, 최소연결성 노드의 ID 순으로 한다. 위의 기준들에 따라 자신이 우성노드(dominant)로 결정되면, 자신이 클러스터 헤드가 되어 클러스터 헤드 선언 메시지를 전송한다. 만일 다른 우성노드가 존재할 경우, 메시지를 전송하지 않는다.

두 번째 단계에서는, 첫 번째 단계에서 클러스터로 그룹화 되지 않은 노드들 간의 경쟁을 통해 클러스터 헤드를 선정한다. 이때 가장 연결성이 높은 노드가 두 번째 단계를 시작하고 나머지 노

들은 각각의 클러스터에 편입되게 된다. 이렇게 클러스터 구성방법을 2단계로 나누어서 수행하면, 네트워크의 경계부분에 존재하는 최소연결성 노드들이 클러스터 헤드가 되지 않으므로, 보다 작은 수의 클러스터를 형성하게 된다.

본 논문은 다음과 같이 구성된다. 2장에서는 클러스터 구성 과 유지보수 방법에 관한 기존연구들에 대해 간략히 기술한다. 3장에서는 본 논문에서 제안하는 클러스터 구성방법을 기술한다. 4장에서는 시뮬레이션 결과를 보이고 5장에서는 결론을 내린다.

2. 관련연구

Ad-Hoc 네트워크에서 가장 잘 알려진 클러스터 구성방법은 LIDCP(Lowest ID Clustering Protocol)[3]와 HCCP(Highest Connectivity Clustering Protocol)[3]이다. 먼저 LIDCP는 각 이동 노드가 자신의 이웃 노드들과의 제어메시지 교환(각 노드의 ID)을 통해서 ID가 가장 낮은 노드를 클러스터 헤드로 선정하는 방법이다. 참고문헌 [5,7,9]는 LIDCP를 변형한 클러스터 구성방법을 제안하였다. 반면에 HCCP는 각 이동 노드가 자신의 이웃중에서 가장 연결성이 높은 노드를 클러스터 헤드로 선정하는 방법이다. 참고문헌 [10]에서는 2홉 구조의 HCCP를 4홉 구조로 확장 시키는 방안에 대해 기술하였다. HCCP는 연결성이 같은 경우에 낮은 ID를 가진 노드를 클러스터 헤드로 선정한다. LIDCP는 노드의 연결성을 고려하지 않으므로 HCCP에 비해 많은 수의 클러스터를 생성한다. 반면에 HCCP는 LIDCP에 비해 작은 수의 클러스터를 생성한다. 그러나 어떤 경우에는 LIDCP와 같거나 혹은 더 많은 수의 클러스터를 생성하게 된다. 이것은 클러스터 구성이 greedy한 방법으로 수행되기 때문이다. 두 방법은 모두 클러스터내의 멤버십에 변화가 있을 때 마다 클러스터를 다시 구성하므로 많은 통신 오버헤드와 잦은 클러스터 변화를 유발한다.

LCC(Least Clusterhead Change)[4]는 잦은 클러스터 헤드의 변화를 줄이기 위해, 클러스터 구성의 영역을 최소한으로 줄이는 방법을 사용하였다. 즉, 이 방법은 임의의 노드 자신 또는 이웃 노드들의 이동으로 인해, 클러스터 헤드가 유실된 노드들이나 클러스터 헤드끼리 경합을 벌이는 영역에 속한 노드들만 클러스터 구성을 다시 하도록 한다. 그러나 네트워크내의 전체 노드들을 대상으로 클러스터를 구성하는 것이 아니므로, 클러스터 구조는 계속해서 분할되게 된다. 따라서 LCC는 시간이 지남에 따라 클러스터의 수를 점증적으로 증가시킨다. 대부분의 클러스터링 기법[1,4,7,8,10,11]들은 LCC와 같거나 유사한 클러스터 유지보수 과정을 사용한다.

ABCP(Access Based Clustering Protocol)[11]은 클러스터를 구성하기 위한 메시지 교환 오버헤드를 줄이기 위해 조금 색다른 클러스터 구성 방법을 제시하였다. 즉, 이 방법은 임의의 공통채널을 하나 할당한 후에 이 채널을 먼저 접근하는 이동 노드가 클러스터 헤드가 되는 방법을 사용한다. 그리고 채널을 통해 클러스터 헤드 메시지를 수신하는 노드는 그 클러스터의 멤버가 된다. 이 방법은 노드들이 채널을 공정하게 접근하도록 하기 위해서, Negative ACK기반의 TPMA(Three Phase Multiple Access)를 사용하였다. 이 방법은 링크계층에서 채널접근에 의해 클러스터 구성을 수행하므로 이웃노드간의 메시지 교환을 수행하지 않을 뿐 아니라 빠른 클러스터 구성을 가능하게 한다. 그러나 이 방법은 비록 병합(merge)과정을 수행한다 하더라도, 상당히 많은 수의 클러스터를 생성한다.

DCA(Distributed Clustering Algorithm)[2]는 노드의 이동성을 고려한 클러스터를 구성하기 위해서 노드의 이동속도를 기준값으로 설정하였다. 즉 DCA는 이동속도가 가장 낮은 노드를 클러스터 헤드로 선정한다. 그러나 DCA는 노드 이동성에 대한 판단기준으로서 단지 이동속도만을 고려하였다.

WCA(Weighted Clustering Algorithm)[1]는 클러스터 헤드를 선출하기 위한 다양한 기준들을 제시하였다. 즉 WCA는 클러스터 구성을 위한 기준값들로 연결성, 전송전력, 이동속도, 노드의 가용전력 등을 고려하였다. 이 기준값들은 응용에서의 중요성에 따라 각각 다른 가중치를 부여받는다. 즉 차수가 중요한 네트워크에서는, 차수는 다른 기준값들보다 높은 가중치를 부여한다. 클러스터 구성을 위해 필요한 최종기준값은 부여된 가중치와 세부 기준값들을 서로 곱하고, 이들을 모두 더하여 생성된다. WCA는 클러스터 구성을 위한 하나의 프레임워크를 제공하였다.

MOBIC(Lowest Relative Mobility Clustering)[8]은 이동속도가 아닌 새로운 이동성 측정기준을 제시하였다. MOBIC에서 이동성은 연속 2회에 걸친 Hello메시지의 수신전력 측정에 의해서 측정된다. 즉 모든 이웃들과 연속적인 2회의 메시지 교환을 수행한 후, 이들 메시지들의 수신전력의 차이를 측정한다. 이 값들은 양의 값(positive value)을 가질 수도 있고 음의 값(negative value)을 가질 수도 있다. 따라서 0을 기준으로 수신전력 차이들의 분산을 구하는 것에 의해 각 노드는 자신과 이웃들의 통합적인 이동성을 가지게 된다. 이 통합적인 이동성은 다음 Hello메시지때 이웃들과 교환된다. 따라서 이 방법은 Hello메시지 사이에서 발생하는 연결성의 변화를 반영하지 못하고, 여타 방법에 비해 메시지 교환이 많다. 더구나 이 방법은 노드의 연결성을 고려하지 않으므로, 상대적으로 많은

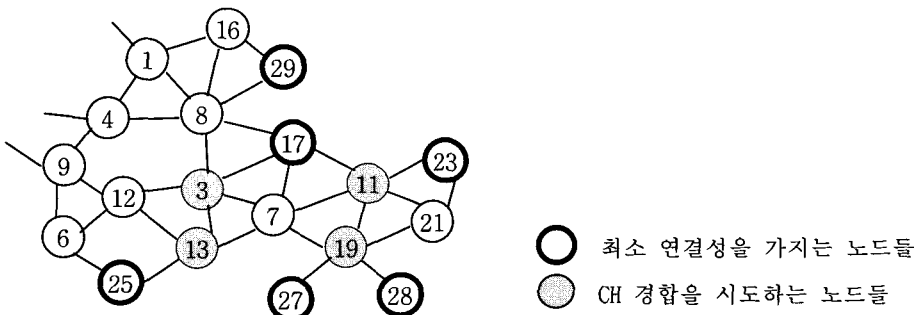
클러스터를 생성한다.

대부분의 기존 클러스터 구성 기법들은 작은 수의 클러스터를 생성하기 위해서 노드의 연결성을 고려하였다. 그러나 이 방법들은 클러스터 구성을 greedy한 방법으로 진행하기 때문에 반드시 더 작은 수의 클러스터를 생성한다고 보장할 수 없다. 따라서 보다 작은 수의 클러스터를 생성하기 위해서는 노드의 연결성을 고려하면서도 보다 융통성있는 방법이 요구된다. 본 논문에서는 네트워크 경계부분의 노드들을 먼저 클러스터에 포함시키고, 나머지 노드들에 대해서 연결성을 기준으로 클러스터를 구성하는 방법을 제안한다.

3. 클러스터 수를 최소화하기 위한 클러스터 구성방법

본 논문에서 제안하는 클러스터 구성방법은 2단계로 구성된다. 먼저 최소 연결성(경합상황에서는 더작은 id)의 노드들을 임의의 클러스터에 포함시키는 것이 첫 번째 단계의 목표다. 이를 위해 먼저 각 노드의 이웃간에 {id, 연결성}의 교환이 선행되어야 한다.

최소연결성의 노드들은 각각 자신의 이웃중에서 최고연결성을 가지는 노드(연결성이 동일할 경우는 더 작은 id)가 클러스터 헤드가 되기 위한 경합을 하도록 권유메시지를 전송한다. 이러한 권유메시지를 수신한 노드는 자신의 이웃에 자신과 같거나 보다 높은 연결성을 가진 노드가 있는 경우에만



〈그림 1〉 클러스터 구성 1단계에서의 경합시도 노드들의 선정

클러스터 헤드 경합을 시도하는 노드가 된다. 이것은 최고 연결성을 가진 노드가 첫 번째 단계의 클러스터 구성에서 먼저 포함되면 두 번째 클러스터 구성에서 클러스터의 분할효과로 클러스터의 수가 증가되기 때문이다. 클러스터 헤드 경합을 시도하는 노드들은 권유 메시지 수신횟수, 권유자 노드들의 연결성의 합, 시도자 노드들의 id를 비교하여 클러스터 헤드가 될 노드를 결정한다.

그림 1에서 최소연결성을 가지는 노드들은 노드 17, 23, 25, 27, 28, 29이다. 이들은 각각 3, 8, 11, 13, 19에게 권유 메시지를 전송하고 노드 3, 11, 13, 19는 CH(클러스터 헤드)경합을 시도하는 노드가 된다. 이때 노드 8은 자신이 이웃노드들 중에서 최대 연결성의 노드이므로 CH경합에 참여하지 않는다. 노드 3과 13의 경합에서는 권유자 노드의 연결성이 적은 13이 승자가 되고, 노드 11과 19의 경합에서는 권유메시지의 수신횟수가 많은 19가 승자가 된다.

두 번째 단계에서는 첫 번째 단계에서 클러스터로 묶이지 않은 노드들중에서 최고 연결성을 가진 노드가 클러스터 구성 과정을 시작 한다. 이때의 두 번째 단계를 시작 하는 노드가 대기해야 하는 최대시간 d 는 $3 \times MD$ 이다. 여기서 MD 는 최대 무선링크 지연시간+CTS/RTS에 의한 지연시간이다. 즉 이 지연시간 내에 자신의 역할을 결정하지 못한 최대연결성을 가진 노드는 자신을 클러스터 헤드로 설정하고 이것을 이웃노드들에게 방송한다. 이렇게 지연시간을 두는 이유는 이웃노드 사이에서 최대 연결성을 가진 노드들이 첫 번째 단계에서 클러스터에 포함이 될 수 있기 때문이다. 즉 이들이 첫 번째 단계에서 클러스터에 포함된다면, 두 번째 단계에서 이들은 제외되어야 한다.

제안한 방법의 첫 번째 단계에서 클러스터 헤드가 되기 위해 시도하는 노드들은 임의의 기준에 따른 조정절차를 사용하므로 2홉 클러스터의 정의를 위배하는 클러스터를 생성하지 않는다. 또한 제안된 방법은 두 번째 단계에서 클러스터에 포함되지 않은 노드들만을 대상으로 클러스터를

구성하므로 역시 2홉 클러스터의 정의를 위배하지 않는다. 제안된 방법은 첫 번째 단계와 두 번째 단계에서 여러 개의 클러스터 구성 시작노드를 가질 수 있으므로 완전히 분산된 방법으로 수행된다.

제안하는 클러스터 구성방법의 절차를 기술하기 위해 다음과 같은 사항을 가정한다.

- 각 노드는 이웃 노드들과 주기적으로 Hello메시지를 교환한다. 초기의 Hello메시지는 노드의 ID와 연결성을 포함하고 이후의 Hello메시지는 노드의 상태에 따라 클러스터 가입메시지 혹은 클러스터 헤드 선언 메시지를 포함한다.

제안된 클러스터 구성방법의 첫 번째 단계는 다음과 같이 수행된다.

1. 최소연결성의 노드들은 자신의 최대연결성의 노드에게 CH권유메시지를 전송한다. 이때 CH권유메시지에는 {노드 id, 노드의 연결성}이 포함된다.
2. 권유메시지를 수신한 노드는 자신의 이웃중에 자신보다 연결성이 높거나 같은 노드가 존재하는지 검색하고 있으면 CH경합 시도자가 된다. CH경합 시도자가 된 노드는 권유메시지 수신 횟수와 메시지에 포함된 연결성을 누적한다. CH 시도자는 자신의 이웃노드들에게 CH시도메시지를 전송한다. CH시도 메시지에는 {노드 id, 권유메시지 수신횟수, 권유메시지 송신노드들의 연결성 합}이 포함된다.
3. 이웃 CH경합 시도자로부터 CH시도메시지를 수신하는 노드는 CH시도메시지의 내용을 비교하여 자신이 우성노드인지 체크한다 만일 그렇다면 자신을 클러스터 헤드로 선언하는 메시지를 방송한다.
4. 이웃 노드로부터 클러스터 헤드선언 메시지를 수신하는 노드들은 그 클러스터에 가입하고 이를 클러스터 가입 메시지를 이용해

서 이웃 노드들에 알린다.

제안된 클러스터 구성 방법의 두 번째 단계는 다음과 같이 수행된다.

1. 임의의 노드가 자신의 연결성이 클러스터에 포함되지 않은 이웃들중에서 가장 높다면, 자신을 클러스터 헤드로 선언하는 메시지를 방송한다.
2. 임의의 이웃노드로부터 클러스터 헤드 선언 메시지를 수신한 노드는 그 클러스터에 가입하고 이를 클러스터 가입메시지를 통해서 이웃노드들에게 알린다.
3. 임의의 노드가 자신보다 높은 연결성을 가진 모든 노드들로부터 클러스터 가입 메시지를 수신한 노드들은 비로소 자신을 클러스터 헤드로 선언하는 메시지를 방송한다.
4. 클러스터 재구성 영역에 포함된 모든 노드들이 임의의 클러스터에 포함될 때 까지 2. 부터 3.을 반복한다.

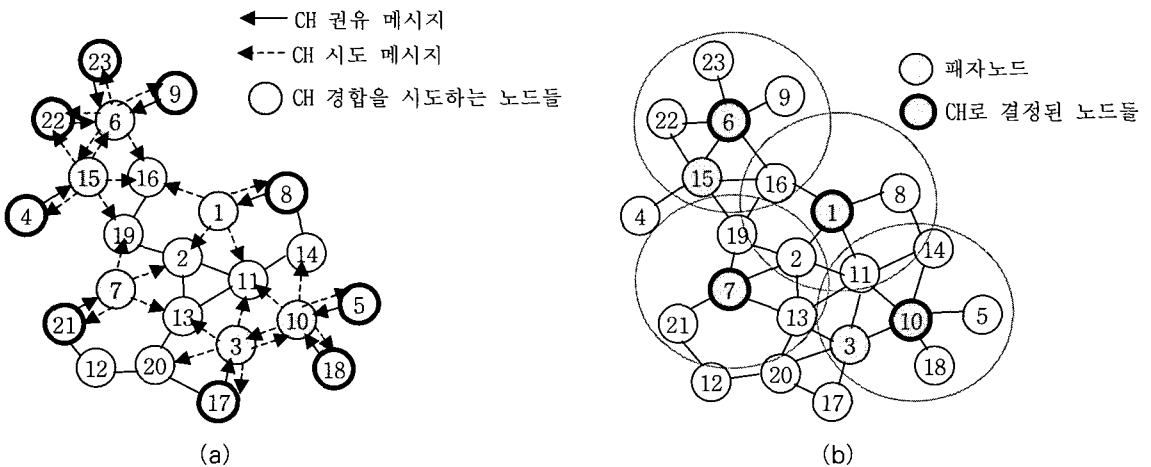
제안하는 클러스터 구성방법의 이해를 돕기 위해, 도식적인 예제를 도입해보자. 그림 2는 임의의 시점에서 클러스터 재구성이 필요한 노드들이 제안된 클러스터 구성방법의 1단계를 통해

클러스터에 포함이 되는 과정을 보여준다.

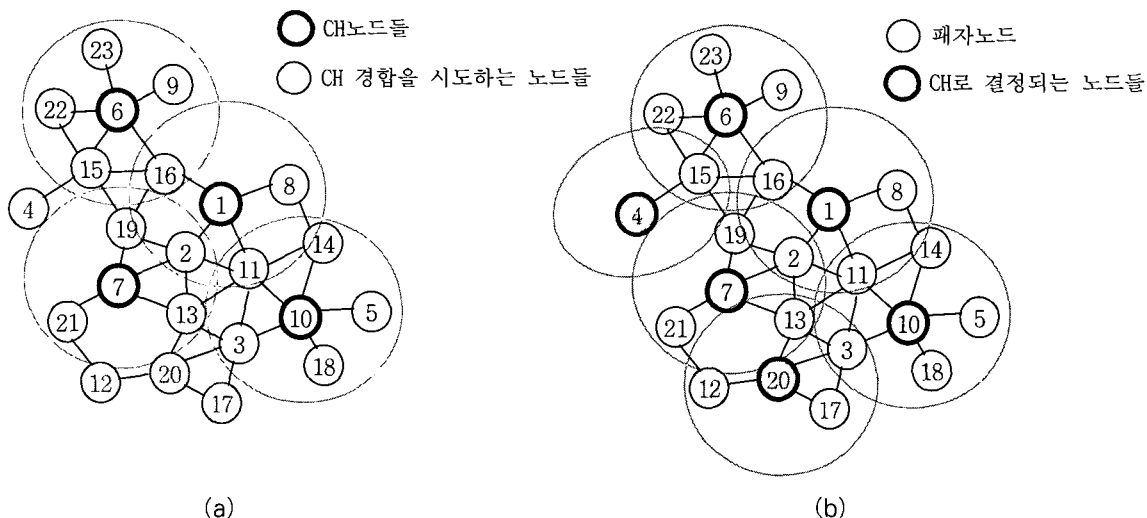
그림 2의 (a)에서 최소 연결성의 노드 4, 5, 8, 9, 17, 18, 21, 22 그리고 23은 자신의 최고 연결성 노드인 15, 10, 1, 6, 3, 7에게 각각 권유메시지를 전송한다. 이들은 모두 CH경합 시도자가 되고 각각 이웃에게 CH시도메시지를 전송한다. CH시도메시지들의 비교를 통해 승자 1, 6, 7, 10이 클러스터헤드가 되고, 나머지 CH시도자들은 이들의 멤버가 된다. 그림 2의 (b)는 이것을 보여준다. 그림 2의 (b)에서 클러스터에 포함된 노드들은 모두 앞에서 언급한 d시간 내에 자신의 역할이 결정된다. 이후 클러스터에 포함되지 않은 노드들중에서 가장 연결성이 높은(경합상황에서는 더 작은 id) 노드 20은 자신을 클러스터 헤드로 선언하게 된다. 마지막으로 노드 4는 자신의 모든 이웃노드가 다른 클러스터에 포함되었고 자신만 아직 클러스터에 포함되지 않았으므로 스스로 클러스터 헤드가 된다. 그림 3의 (a)와 (b)는 이것을 보여준다.

4. 시뮬레이션 결과

제안하는 방법의 성능을 평가하기 위해 Ad-Hoc 네트워크를 위한 시뮬레이터를 CSIM18엔



〈그림 2〉 제안된 방법의 1단계 클러스터 구성



〈그림 3〉 제안된 방법의 2단계 클러스터 구성

진과 C언어를 이용하여 개발하였다. 본 시뮬레이터는 실험영역 내를 이동하며 클러스터 구성을 위해 메시지를 교환하는 노드들 프로세스로 모델링 하였으며 클러스터 구성과정을 사건중심 기법에 의해 수행하였다.

본 실험에서는 초기에 45개의 노드가 500m × 500m의 평면상에서 랜덤하게 배치되며 임의의 시간에 임의의 방향으로 이동한다고 가정하였다. 시뮬레이션을 위한 입력 파라미터들은 표 1과 같이 설정되었다.

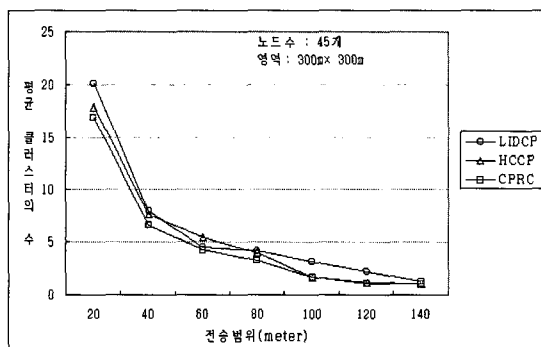
〈표 1〉 시뮬레이션 파라미터들

파라미터	값
노드의 수	15, 25, 35, 45
노드의 이동속도	A(0~5m/s), B(5~10m/s), C(10~15m/s)
전송범위	10~230meter
실험영역	300m × 300m, 500m × 500m
Hello메시지 간격	3초

시뮬레이션에서 노드의 이동속도는 세가지 클래스(A,B,C)중 하나에 속하게 된다. 본 논문에서 제안하는 방법의 성능을 측정하기 위해,

평균 클러스터수라는 척도를 이용하여 LIDCP[3] 및 HCCP[3]와 비교하였다. 이때 세 가지 방법들은 모두 매 Hello메시지 교환 후에 클러스터 구성을 수행하도록 한다. 또한 세 가지 방법들은 전송범위와 실험영역, 그리고 노드의 수와 같은 파라미터를 변화시켜 비교하였다.

그림 4는 전송범위의 증가에 따른 세 가지 방법의 클러스터 수의 변화를 보여준다. 여기서 제안하는 방법은 CPRC(Clustering Protocol for Reducing the number of Clusters)로 표기하였다. 이후부터는 모든 실험결과에서 제안하는 방법을 CPRC

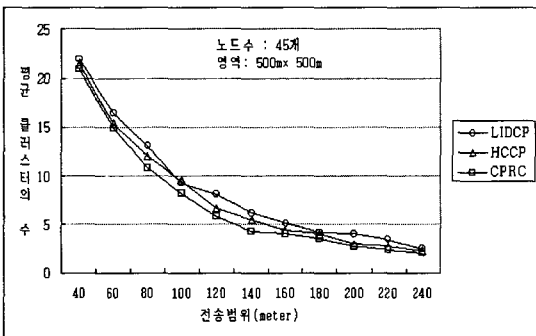


〈그림 4〉 전송범위의 증가에 따른 클러스터 수의 변화 (300m × 300m)

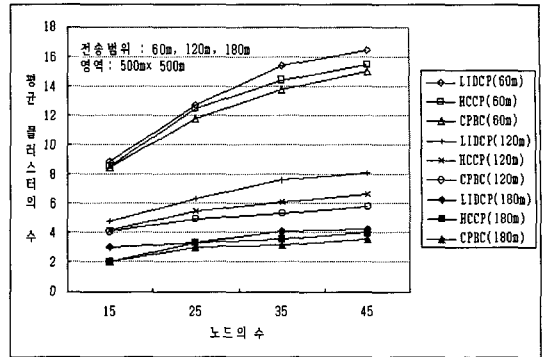
로 표기하기로 한다. 그림 4에서 보는 것처럼, 세 가지 방법 모두 전송범위가 증가함에 따라 클러스터의 수는 급격히 감소한다. 이는 전송범위가 증가할수록 하나의 클러스터에 포함되는 노드의 수가 많아져 전체적으로는 클러스터의 수가 감소되기 때문이다. 그러나 전송범위가 계속증가하면 (>60meter), 클러스터 수의 감소폭은 크게 줄어들을 알 수 있다. 이는 전송범위의 증가에 따라 클러스터간의 중첩된 부분이 증가하기 때문이다. 그림 4로부터 알 수 있는 또 하나의 사실은 제안하는 방법이 모든 전송범위에서 다른 방법들에 비해 작은 수의 클러스터를 생성한다는 점이다.

그림 5는 실험영역을 500m × 500m로 확장하여 실험한 결과를 나타낸다. 그림 5에서 보는 것처럼 실험영역을 확장시키면 같은 전송범위에서 보다 많은 수의 클러스터가 생성된다. 이는 노드들이 실험영역 내에 랜덤하게 배치되어서 좁은 영역에 비해 노드들간의 연결성이 급격히 감소하기 때문이다. 제안하는 방법은 확장된 영역에서도 모든 전송범위에서 다른 방법들에 비해 보다 작은 수의 클러스터를 생성한다.

그림 6은 세 가지 클러스터 구성 기법의 확장성을 평가하기 위해 실험영역을 고정(500m × 500m)시키고, 노드의 수를 점차로 증가시켜서 클러스터 수의 증가를 비교하였다. 이때 전송범위를 각각 다르게 하여 전송범위의 증가에 따른 영향을 평



〈그림 5〉 전송범위의 증가에 따른 클러스터 수의 변화 (500m × 500m)



〈그림 6〉 노드수의 증가에 따른 클러스터수의 변화

가하였다. 그림 6에서 보는 것처럼 세 가지 방법 모두 노드의 수가 증가함에 따라 클러스터의 수도 같이 증가된다. 이것은 노드들이 랜덤하게 실험영역 내에 배치되고 전송범위가 제한되어 있기 때문이다. 즉, 일정한 전송범위에서 노드의 수가 증가하면 노드들 간의 연결성이 저하되어 클러스터의 수의 증가로 이어지기 때문이다. 만일 전송범위가 증가하면 노드수의 증가에 따른 클러스터수의 증가율이 크게 감소됨을 알 수 있다. 그림 6을 통해 알 수 있는 또 다른 사실은 전송범위 60meter와 120meter에서 노드수의 증가에 따른 제안방법의 클러스터수 증가율이 다른 방법들에 비해 작다는 것이다. 전송범위가 일정한 상태에서 노드수의 증가는 네트워크 가장자리에 존재하는 최소연결성 노드 또한 증가시키고 이들은 곧 클러스터의 증가로 이어진다. 그러나 제안하는 방법은 최소연결성 노드의 수가 증가하더라도 첫 번째 단계의 클러스터 구성에서 이들을 클러스터에 포함시킴으로써 최소연결성 노드의 증가에 따른 영향을 최소화 시킨다.

5. 결론

본 논문에서는 최소의 연결성을 가지는 노드들을 먼저 클러스터에 포함시킨 후에, 남은 노드들을 연결성과 id를 기준으로 클러스터를 구성하는

2단계 클러스터 구성방법을 제안하였다. 첫 번째 단계에서는 최소연결성의 노드들을 가능한 많이 포함하는 작은 수의 클러스터를 생성하기 위하여 일련의 조정절차를 이용한다. 두 번째 단계에서는 첫 번째 단계에서 생성된 클러스터에 포함되지 않은 노드들을 대상으로 최대 연결성을 가진 노드가 클러스터 구성을 시작한다. 그 결과로서 제안된 방법은 여타 클러스터 구성기법들에 비해 더 작은 수의 클러스터를 생성한다. 실험결과를 통해 제안하는 방법은 LIDCP와 HCCP에 비해 약 20%와 11%정도의 클러스터 감소효과를 증명하였다. 또한 제안하는 방법은 실험결과에 의해 중간이하의 전송범위에서 LIDCP나 HCCP에 비해 확장성이 높은 것으로 평가된다.

참고문헌

- [1] M. Chatterjee, S. Das and D. Turgut, "An On-Demand Weighted Clustering Algorithm(WCA) for Ad hoc Networks," Proc. on GLOBECOM, vol. 3, pp. 1697-1701, 2000
- [2] S. Basagni, "Distributed Clustering for Ad Hoc Networks," International Symposium on Parallel Architectures, Algorithms and Networks, pp. 310-315, 1999
- [3] M. Gerla and C. Chiang, "Multicluster, mobile, multimedia radio network," ACM-Baltzer Journal of Wireless Networks, Vol. 1, No. 3, pp. 255-265, 1995
- [4] C. Chiang, H. Wu, W. Liu, M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel (CGSR)," Proc. of IEEE SICON'97, pp. 192-211, 1997
- [5] K. Liu, Jiandong Li, K. Mukumoto, A. Fukuda, "Adaptive Control Protocol in Mobile Wireless Ad Hoc Networks," APCCAS 2000, pp. 13-17, 2000
- [6] P. Krishna, N. H. Vadiya, M. Chatterjee, D. K. Pradhan, "A Cluster-based Approach for Routing in Dynamic Networks," ACM SIGCOMM Computer Communication Review, 49, pp. 49-64, 1997
- [7] C. R. Lin, M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," IEEE JSAC, 15(3), pp.1265-1275, 1997
- [8] P. Basu, N. Khan, T. D. C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," Proc. of IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile Computing, pp. 413-418, 2001
- [9] M. Jiang, J. Li, Y. C. Tay, "Cluster Based Routing Protocol," IETF draft, Aug. 1999, Work in Progress
- [10] G. Chen, F. Garcia, J. Solano and I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks, CD Proc. of IEEE Hawaii Int. Conf. System Science, 2002
- [11] T. C. Hou and T. J. Tsai, "An Access-Based Clustering Protocol for Multihop Wireless Ad Hoc Networks," IEEE JSAC, 19(7), pp. 1201-1210, 2001
- [12] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing for Multihop Wireless Ad Hoc Networks," Ad Hoc Networking, ed. C. E. Perkins, pp.139-168, 2000
- [13] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," Proc. of IEEE WMCSA' 99, pp.90-100, 1999

◎ 저 자 소개 ◎



방 상 원

1985년 전남대학교 계산통계학과 졸업(학사)

1988년 전남대학교 대학원 계산통계학과 졸업(석사)

1995년 전남대학교 대학원 계산통계학과 졸업(박사)

1987년~현재 송원대학 컴퓨터정보과 교수

관심분야 : Mobile Computing, 컴퓨터그래픽, 소프트웨어공학

E-mail : swbang@songwon.ac.kr