

애드혹 네트워크에서 가중치 클러스터링을 이용한 효율적인 코어-기반 멀티캐스트 트리

박 양 재[†]·한 승 진^{**}·이 정 현^{***}

요 약

본 논문에서는 이동 애드혹 네트워크의 코어-기반 멀티캐스트 트리 라우팅 알고리즘에 가중치 클러스터링을 이용하여 효율적으로 코어-기반 멀티캐스트 트리를 유지하는 기법을 제안한다. 코어-기반 멀티캐스트 트리 라우팅에서 가장 큰 문제점은 코어 노드의 위치를 결정하는 문제이다. 코어 노드의 위치에 따라서 데이터의 전송 거리가 달라진다. 코어노드의 이동으로 인하여 멀티캐스트 트리의 재구성으로 인한 오버헤드가 전체 네트워크에 미치는 영향이 크므로 가중치 클러스터링을 이용한다. 클러스터의 헤드와 게이트웨이 노드, 클러스터 헤드의 연결 경로를 멀티캐스트 트리 구성하여 데이터 전달과 제어 메시지의 전달 경로로 사용한다. 코어 노드의 선택은 코어 영역내의 클러스터 헤드 노드들 중에서 가중치가 가장 작은 값을 갖는 노드를 코어 노드로 선정한다. 성능평가에서 코어영역을 단계별로 증가시켰을 때 코어영역이 전송지연에 미치는 영향을 기존의 코어 기반 트리방식과 비교한 결과 전송지연과 오버헤드가 감소하였다. 이와 같이 가중치 클러스터링에 의하여 멀티캐스트 트리를 구성하고 유지함으로써 제안한 방식이 코어 노드의 위치와 이동성에 따라서 전송거리와 제어 오버헤드가 고정 CBT 방식보다 향상됨을 알 수 있었으며, 코어 노드를 선택할 때 이동성이 적고, 네트워크의 중심에 가까울수록 멀티캐스트 트리가 안정되며 전송거리도 짧아짐을 시뮬레이션을 통하여 검증하였다.

An Efficient Core-Based Multicast Tree using Weighted Clustering in Ad-hoc Networks

Yang-Jae Park[†] · SeungJin Han^{**} · Jung-Hyun Lee^{***}

ABSTRACT

This study suggested a technique to maintain an efficient core-based multicast tree using weighted clustering factors in mobile Ad-hoc networks. The biggest problem with the core-based multicast tree routing is to decide the position of core node. The distance of data transmission varies depending on the position of core node. The overhead's effect on the entire network is great according to the recomposition of the multicast tree due to the movement of core node, clustering is used. A core node from cluster head nodes on the multicast tree within core area whose weighted factor is the least is chosen as the head core node. Way that compose multicast tree by weighted clustering factors thus and propose keeping could know that transmission distance and control overhead according to position and mobility of core node improve than existent multicast way, and when select core node, mobility is less, and is near in center of network multicast tree could verification by simulation stabilizing that transmission distance is short.

키워드 : 애드혹 네트워크(Ad-hoc Networks), 무선 이동통신(Wireless Mobile Communication), 멀티캐스트(Multicast), 코어-기반 멀티캐스트 트리(Core-Based Multicast Tree)

1. 서 론

무선 네트워크는 기반구조(Infrastructure)를 이용한 기존의 이동통신 서비스와 특수한 환경에서 기지국과 같은 고정장치나 중앙 집중식 통제 없이 일시적인 통신망이 형성되는 MANET(Mobile Ad-hoc NETwork) 그리고 두 가지 방식을 조합한 Hybrid 무선 네트워크까지 다양한 형태로 발

전하고 있다[1-3]. 따라서, 노드의 라디오 전파범위의 제약으로 인해 무선 도메인에서의 경로는 다중 홉으로 구성된다. 또한 노드들이 임의로 예측할 수 없이 이동하므로 토폴로지가 동적으로 빈번하게 변화한다는 특성을 가지며, 고정 네트워크에 비하여 가용할 대역폭과 전력 양에 제약이 크다. 무선 애드혹 네트워크의 전형적인 응용분야는 재난복구, 전쟁, 응급구조 및 탐색작업 등과 같은 위급상황에서의 대원들간의 통신 혹은 콘서트나 컨퍼런스 와 같은 모임에서 구성원 간의 통신 등을 들 수 있는데, 이와 같은 경우의 통신은 대부분 구성원간의 공동작업을 수행하는 것이 목적이므로

† 정 회 원 : 가천김대학 전산정보처리과 교수
** 정 회 원 : 인하대학교 컴퓨터공학부 강의전담 교수
*** 종신회원 : 인하대학교 컴퓨터공학과 교수
논문접수 : 2002년 11월 23일, 심사완료 : 2003년 3월 10일

멀티캐스트 혹은 다중점-대-다중점 형태의 통신이 필요할 경우가 많다. 또한 애드혹 네트워크는 물리적으로 브로드캐스트에 의해 전송이 이루어지기 때문에, 멀티캐스트 형태의 통신을 효율적으로 지원하기에 매우 적합한 네트워크이다.

무선 애드혹 네트워크의 라우팅 방식은 경로를 결정하는 시기에 따라 순향적(Proactive) 라우팅 방식과 반응적(Reactive) 라우팅 방식, 혼합(Hybrid)방식으로 분류할 수 있다. 순향적 라우팅 방식은 주기적으로 네트워크 정보를 수집하여 모든 목적지에 대한 경로를 미리 결정해 두는 방식이다. 순향적 라우팅 방식은 미리 경로가 결정되어 있으므로 즉시 패킷을 전송할 수 있다는 장점을 가지고 있다. 기존의 유선망에서 사용하고 있는 거리벡터 알고리즘, 링크상태 알고리즘이 있다. 무선 애드혹 네트워크를 위한 순향적 라우팅 방식으로는 DBF[4], DSDV[5], WRP[6]이 있다.

반응적 라우팅 방식은 패킷을 보낼 필요가 있을 때 요구 기반(On-demand)방식으로 경로를 설정한다. TORA[7], DSR[8], AODV[9], ABR[10], RDMAR[11]이 있다. 순향적 라우팅 방식으로 설정된 경로는 라우팅 정보를 교환하기 위하여 많은 양의 대역폭과 제어 패킷을 발생시켜서 비효율적이다. 이전 연구에서 동적으로 변화하는 애드혹 네트워크의 특성상 순향적 라우팅 방식 보다는 반응적 라우팅 방식이 더욱 좋은 결과를 제공한다는 것을 발견하였다[12-14]. 혼합방식은 순향적 방식과 반응적 방식의 혼합형태로 ZRP(Zone Routing Protocol)와 같이 각 노드가 홉 카운트를 반경으로 하는 지역에 대해서 사전에 라우팅 정보를 유지하고 지역을 벗어나는 라우팅 정보에 대해서만 경로 설정 절차를 요구하는 방식이다.

본 논문에서는 애드혹 네트워크의 코어-기반 멀티캐스트 트리 라우팅 알고리즘의 가장 큰 문제점인 코어 노드의 위치를 결정하는 문제를 가중치 클러스터링(Weighted factors clustering)을 이용하여 효율적으로 코어-기반멀티캐스트 트리를 유지하는 기법을 제안한다.

서론에 이어 2장은 기존 연구로 무선 애드혹 네트워크에서의 라우팅 방식의 장단점을 설명하고 3장에서는 제안하는 가중치 클러스터링을 이용하여 효율적으로 코어-기반 멀티캐스트 트리를 유지하는 기법에 대하여 설명하였다. 4장에서는 시뮬레이션 환경 및 모델, 5장에서는 성능평가에 대해 설명하고 6장에서 결론을 맺는다.

2. 무선 애드혹 네트워크에서의 멀티캐스트 라우팅 방식

오늘날 컴퓨터 네트워크 상에서 다자간 통신을 효율적으로 지원하는 방법으로 멀티캐스트가 사용된다. 이 방법은 특정 그룹에 대하여 전송 트리를 만들고, 트리를 통하여 데이터를 전송하는 방법을 많이 사용하고 있다. 멀티캐스팅은 브로드캐스트라는 매체의 기본적인 특성 때문에 그룹 통신에

사용되는 경우에 무선 네트워크에서 큰 효율성을 지니고 있다. 그러나 무선 네트워크의 특성상 잦은 패킷의 분실과 전송 에러가 발생하기 때문에 데이터 전달과 데이터 전달 구조를 신뢰성 있게 유지하기 위해서는 많은 패킷의 재전송과 지연 오버헤드가 발생된다. 기존의 무선 네트워크에서 멀티캐스트 메시지를 신뢰적으로 전송할 수 있는 멀티캐스트 라우팅 방식에는 트리 기반 라우팅 방식과 매쉬 기반 라우팅 방식으로 분류된다.

2.1 트리 기반 라우팅 방식

멀티캐스트의 트리 구성 방식은 플러딩(Flooding)과 확장(Spanning)트리를 생성하며, 트리를 생성하는 방식에는 송신자 기반 트리와 공유 트리로 분류된다. 송신자 기반의 트리는 플러딩과 삭제(Prune)기법으로 트리를 생성한다. 이 방식은 송신자와 수신자 사이에 하나의 유일한 경로만을 제공하기 때문에 네트워크 토폴로지가 빈번히 변하는 애드혹 네트워크에서는 경로가 단절되는 점과 플러딩과 삭제기법을 사용함으로써 대역폭을 낭비하는 점과 멀티캐스트의 규모가 커질 경우에 멀티캐스트 그룹마다 또는 송신자마다 트리를 구축해야 하는 문제점이 있다. 더욱 큰 문제점은 애드혹 네트워크의 특징인 빈번한 노드의 이동으로 인하여 송신자가 동적으로 위치를 이동하는 경우 트리의 재구축 문제이다. 공유 트리 방식은 하나의 그룹에 대하여 트리를 공유하는 방식으로 최적의 트리를 생성하는 것이 아니라 확장성 측면과 자원 낭비 문제를 해결하기 위한 방식이다. 최적의 경로 보다는 확장성의 측면과 노드의 이동성이라는 점에서 멀티캐스트 그룹을 관리한다. 트리 기반 라우팅 방식에는 [15-17]이 있다.

2.1.1 AMRoute(Ad-hoc Multicast Routing Protocol)

AMRoute [17]는 기존의 IP 멀티캐스트 프로토콜들[18]이 갖은 트리 재구성으로 인한 과도한 시그널링 오버헤드와 데이터그램의 분실 때문에 애드혹 환경에 적합하지 않다는 점 때문에 동적인 환경에서도 안정적인 멀티캐스트를 수행할 수 있도록 고안되었다. 멀티캐스트 트리는 단지 멀티캐스트 멤버들 사이에서만 구성되는데, MBONE[19]에서 멀티캐스트 라우터들이 연결되는 방식과 유사하게 IP-in-IP 터널로 연결된다. 따라서, 멀티캐스트 그룹의 송·수신원이 아닌 노드들은 AMRoute를 지원할 필요가 없고, 멤버 노드 간에 유니캐스트 경로가 존재하는 한 이동성에 관계 없이 AMRoute의 멀티캐스트 공유 트리의 연결성은 유지된다. AMRoute는 동적인 환경에서도 멤버들 사이에만 트리를 구성함으로써 시그널링 오버헤드를 비약적으로 감소시키지만 구성의 복잡성과 비멤버 노드가 트리의 구성에 참여하지않기 때문에 비효율적인 경로가 만들어 질 수 있다. 이것은 대역폭의 낭비를 초래할 수 있으며, 패킷 지연 시간의 증가를 불

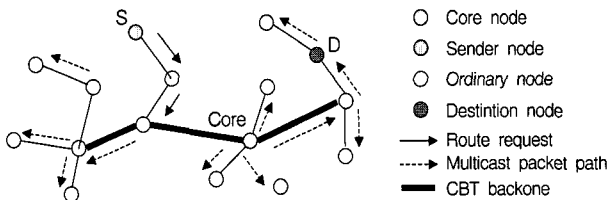
러 일으킬 수 있는 단점이 있다.

2.1.2 AMRIS(Ad-hoc Multicast Routing protocol utilizing Increasing id-numberS)

AMRIS[16]는 전체 멀티캐스트 트리 구성 오버헤드를 감소 시키기 위하여, 각 노드가 갖고 있는 MSM-ID(Multicast Session Member ID)를 이용하여 국부적으로 트리를 형성해 나가는 방식이다. 새로운 멀티캐스트 세션을 구성하고자 하는 노드는 NEW_SESSION 메시지를 브로드캐스트하게 되고, 이 메시지를 수신한 노드들은 자기 MSM-ID를 계산하여 할당하고 NEW-SESSION 메시지에 자신이 할당한 값을 넣어 하위 노드들이 참조하도록 한 후 다시 브로드캐스트하게 된다. 멀티캐스트 그룹에 참가하고자 하는 노드는 참가 요청 패킷을 주변 노드들에게 전송한다. 주변 노드가 이미 멀티캐스트의 멤버인 경우, MSM-ID를 주변 노드보다 크게 하여 자신을 주변 노드의 자식으로 멀티캐스트 트리에 접합(Graft)하고, 그렇지 않다면 확장 링 탐색을 통하여 멤버 노드를 찾게 된다. 주변 노드가 부모 노드가 되는데 적당하지 않는 MSM-ID라면 가지 재설정(Branch reconstruction)기법을 통하여 적당한 부모-자식 관계를 유지하게 된다.

2.1.3 CBT(Core Base Tree)

CBT[20]는 공유트리(Shared-tree) 개념을 사용하는 프로토콜로 루트가 되는 하나의 공유점이 코어노드가 된다. CBT는 송신자가 데이터를 보내면 데이터를 받은 수신자는 자신의 이웃에게 모두 데이터를 전달하는 양방향 공유트리를 갖으며, 하나의 전달 트리를 이용하여 그룹의 모든 구성원들에 의해서 공유된다. 장점은 확장성이 좋으며, 주기적인 포워딩이 필요하지않으므로 대역폭의 소비가 절약된다. 단점은 코어에 트래픽이 집중되어 병목현상(Bottleneck)가 발생하는 문제점이 있다. 코어의 위치를 수동적으로 네트워크의 중심에 선정하여 최단의 전송 경로를 유지할 때 가장 좋은 성능을 나타낸다. (그림 1)은 코어 기반 트리의 동작을 보여준다.



(그림 1) 코어 기반 트리의 동작

2.1.4 Ad-hoc On Demand Distance Vector(AODV)[24]

AODV[23]는 유니캐스트와 멀티캐스트를 모두 지원하는 라우팅 프로토콜로서, 실제로 트래픽 플로우가 발생한 경우에만 경로를 계산하는 요구 기반 프로토콜이므로 필요하지

않은 목적지 혹은 그룹에 대한 라우팅 오버헤드를 피할 수 있다. AODV는 하나의 멀티캐스트 그룹에 대하여 하나의 공유 데이터 전달 트리를 구성하는데, 루프를 방지하기 위해 경로 요청에 대하여 여러 개의 경로 응답을 받는 경우 이들 가운데 하나를 선택해서 트리를 구성한다. 또한, 무한대까지 카운트해 나가는 수렴 문제를 방지하기 위하여 라우팅 정보에 순차 번호를 부여하고 더 큰 순차 번호를 더 최신 정보로 받아들인다. 그리고, IP 데이터그램을 이용해 이웃 발견을 구현하고 이를 통해 트리 분할을 발견하며 재 연결을 시도하는데, IP 계층에서 이와 같은 감시 작업을 하기 위한 오버헤드가 크다.

2.2. 메쉬 기반 라우팅 방식

송신자와 수신자 사이의 다수의 경로를 제공한다. 따라서 링크가 단절되더라도 다른 우회 경로를 사용하여 견고성이 높다는 장점을 가지고 있다. 메쉬 기반 라우팅방식이 트리 기반 방식보다 네트워크의 위상 변화에 더욱 견고한 것으로 밝혀졌다. 메쉬 기반 라우팅 방식에는[23, 25, 26]이 있다.

2.2.1 Core-Assisted Mesh Protocol(CAMP)

CAMP[25]는 목적지 노드에 대한 최단경로를 알고 있는 유니캐스트를 가정하여 멀티캐스트 메쉬 방법을 이용한다. 코어 노드의 역할은 멤버가 멀티캐스트 그룹에 참석하기 위하여 존재하며, 반드시 멀티캐스트 그룹에 속할 필요는 없다. 멀티캐스트 트리에 참석과 이탈은 CBT와 동일하며, 송신자 노드가 데이터 전송을 위하여 먼저 멀티캐스트 메쉬에 참여하여 단 방향 경로를 생성한 후 데이터를 전송한다. 만일 메쉬에 속한 노드가 데이터를 받았을때 그 데이터가 송신자로부터 최단거리를 통하여 전달된 것이 아니라면 최단 거리에 있는 이웃 노드에게 하트비트(Heartbeat) 메시지를 보내고, 이 메시지를 받은 이웃 노드는 하트비트 메시지를 보낸 노드에게 결합(Join)메시지를 보내 메쉬 링크가 연결되며 그 이후부터 전송되는 멀티캐스트 데이터는 최단경로로 전송된다.

CAMP는 하나 이상의 데이터 전달 경로를 제공함으로써 이동성에 대해 안정적이며, 코어 노드를 이용해 데이터 혹은 제어 패킷 플러딩 없이 멀티캐스트 그룹 가입이 가능하도록 하였다. 또한, CAMP는 송신원 혹은 코어 노드로의 유니캐스트 경로 발견을 위해 별도의 유니캐스트 라우팅 프로토콜에 의존하므로, 사용하는 유니캐스트 라우팅 프로토콜의 성능에 민감하게 영향을 받는다는 문제가 있다.

2.2.2 Forwarding Group Multicast Protocol(FGMP)과 On Demand Multicast Routing Protocol (ODMRP)

FGMP[26]와 ODMRP[23]는 매우 유사한 방식의 프로토콜로서 전자는 사전 결정 방식의 프로토콜인데 반해 후자

는 요구 기반 방식이라는 점이 차이점이다. 이 두 프로토콜은 모두 '제한적인 플러딩(Scoped flooding)'에 의해 데이터를 전달한다. 제한적인 플러딩이란 데이터 전달이 플러딩에 의해 이루어지되 네트워크를 구성하는 노드들 가운데 일부만이 플러딩에 참여하는 것을 의미한다. FGMP/ODMRP는 정기적인 수신원 혹은 송신원의 제어 메시지 플러딩에 의하여 이동으로 인한 토폴로지 변화에 맞추어 데이터 전달 구조를 변경한다. 고정 네트워크의 DVMP가 데이터를 플러딩 함으로써 데이터 전달 트리를 재구성한 것과는 달리, FGMP/ODMRP는 채널 오버헤드를 줄이기 위해 데이터 패킷보다 훨씬 크기가 적은 제어 패킷을 플러딩 한다. FGMP/ODMRP는 플러딩에 비하여 멀티캐스트 데이터 전달을 담당하는 노드의 수를 효과적으로 줄이고, 데이터가 최단 경로를 통해 전달되며, 멀티캐스트를 전달 구조를 구성하는 노드에 유지해야 하는 상태 정보가 극히 간단하다는 장점이 있다. 그러나, 이동성이 증가하면 이에 따라 정기적인 제어 패킷 플러딩 구간이 짧아져야 하고, 송신원 혹은 수신원의 수가 늘어남에 따라 제어 패킷 플러딩 오버헤드가 급격히 증가한다는 문제가 있다.

3. 제안 기법

3.1 제안 배경

무선 애드혹 네트워크의 코어-기반 멀티캐스트 트리 라우팅 알고리즘에 가중치 클러스터링을 이용하여 효율적으로 코어-기반 멀티캐스트 트리를 유지하는 기법을 제안한다. 코어-기반 멀티캐스트 트리 라우팅에서 가장 큰 문제점은 코어 노드의 위치를 결정하는 문제이다. 코어 노드의 위치에 따라서 데이터의 전송 거리가 달라진다. 유선망의 경우 가장 적절하다고 생각되는 장소에 위치한 노드를 수동적으로 코어노드로 결정할 수 있지만, 무선 애드혹 네트워크에서는 동적으로 코어 노드를 선택한다. 애드혹 네트워크 상에서 존재하는 모든 노드는 빈번하게 이동하며, 멀티캐스트 트리의 코어 노드 또한 마찬가지이다. 코어노드의 이동 정도와 네트워크 상에서의 위치에 따라 전송거리와 제어 오버헤드가 증가하므로, 멀티캐스트의 효율성을 증가시키기 위해서 코어노드를 동적으로 선택하여야만 한다. 따라서 코어

노드를 선택할 때 이동성이 작고 네트워크의 중심에 가까울수록 멀티캐스트 트리가 안정되며 전송거리가 짧아진다.

이와 같이 코어노드의 이동으로 인하여 멀티캐스트 트리의 재구성하는 경우에 전체 네트워크에 미치는 오버헤드가 크므로 가중치 클러스터링을 이용한다. 먼저 전체 네트워크에서 가중치를 부여한 클러스터링을 통하여 클러스터의 헤드들이 선택된다. 클러스터 헤드의 선택을 위하여 GPS의 정보인 위치, 평균이동속도, 이웃 클러스터 헤드들과의 거리 합, 클러스터 헤드로서 동작 시간에 조합 가중치를 적용한 값을 사용하였다. 가중치 중에서 위치에 대한 가중치를 가장 많이주어 선정하였으며, 그 이유는 코어노드의 위치가 네트워크의 중심에 가까울수록 멀티캐스트 트리의 안전성과 전송거리도 짧아지기 때문이다.

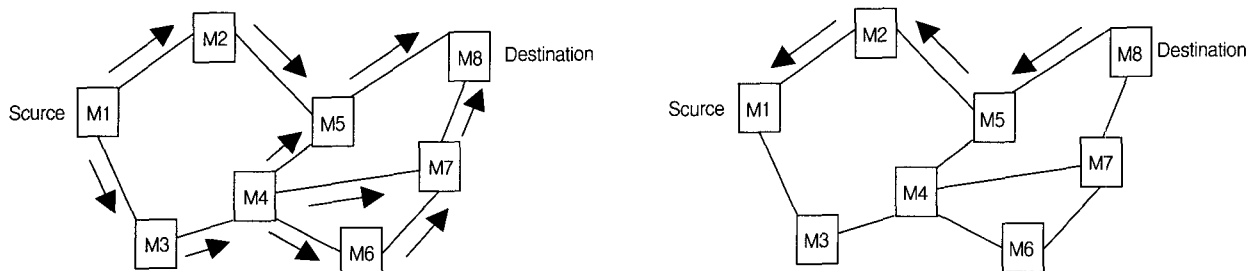
클러스터의 헤드와 게이트웨이 노드, 클러스터 헤드의 연결 경로를 멀티캐스트 트리로 구성하여 데이터 전달과 제어 메시지의 전달 경로로 사용한다. 멀티캐스트 트리 상의 클러스터 헤드 노드들 중에서 클러스터링 가중치가 가장 작은 값을 갖는 클러스터 헤드를 코어 노드로 선정한다. 이와 같이 클러스터링에 의하여 멀티캐스트 트리를 구성하고 유지함으로써 코어 노드의 이동시 새로운 주코어 노드의 선발에 따른 지연이 감소할 것이며, 기존의 멀티캐스트 트리를 재구성 없이 사용할 수 있으며 데이터 전달 구조 오버헤드를 감소시키는 기법을 제안한다.

3.2 가 정

일반적인 애드혹 네트워크의 논문들과 같이 네트워크의 크기와 위치는 고정된 환경으로 가정하며, 모든 노드들은 GPS(Global Positioning System)[27]에 의해서 자신의 위치 정보를 제공되고 있으며, 이 정보에는 위치, 평균이동속도, 이웃 클러스터 헤드들과의 거리 합, 클러스터 헤드로서 동작 시간에 조합 가중치를 적용한 값이 포함되어 있다고 가정한다.

3.3 경로설정 프로토콜

본 논문에서는 애드혹 유니캐스트와 멀티캐스트 알고리즘은 (그림 2)와 같이 AODV 방식을 일반화하여 코어-기반 트리에 적용하였다. AODV는 순방향 설정프로토콜인 DSDV

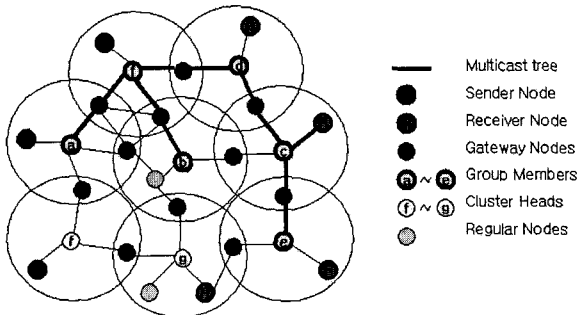


(그림 2) AODV의 경로요청/ 경로응답

의 반응식 설정방식 버전이라고 할 수 있다. DSDV 역시 기존 네트워크에서 많이 사용하는 RIP와 같은 Bellman-Ford 방식의 프로토콜로 차이점은 경로 설정시 제어 패킷에 순서번호를 사용하여 경로 상에 순환이 발생하는 것을 방지하고 있다는 점이다. AODV는 DSDV와 달리 경로 탐색 절차를 지속적으로 수행하지 않고, 송신자의 데이터 송신 요구가 있을때만 수행하는 요구기반 방식이다. AODV의 제어 패킷의 이름을 CBT에서 사용되는 일반적인 이름으로 사용하였다. AODV 방법을 사용하는 이유로는 경로설정 오버헤드를 줄이기 위한 On-Demand 방식이기 때문이다.

3.4 제안 시스템 모델

본 논문에서 사용되는 시스템 모델은 (그림 3)과 같이 다중 클러스터 기반 구조를 이용해서 코어-기반 멀티 트리의 데이터 전달 구조를 유지한다. 네트워크 내의 각 노드들은 멀티캐스트에 참여 하고 있는 멤버 노드와 비 멤버 노드로 구별된다. 송신자 노드가 포함된 노드들 중에서 가중치가 가장 작은 값을 갖는 노드를 클러스터 헤드로 선출하여 클러스터를 형성하게 된다. 형성된 클러스터와 클러스터들 사이를 연결하는 게이트웨이 노드를 경로로 하는 멀티캐스트 트리를 구성한다. 노드들 중에는 멤버 노드가 아닌 단지 수동적인 패킷 포워딩(Passive forwarding)만을 수행하는 노드가 발생할 수 있으며, 링크의 선택 기준으로는 최단거리 알고리즘에 의하여 결정된다. 데이터 송신을 원하는 노드는 브로드캐스트 ID와 IP주소로 이루어진 경로요청 패킷(Route Request Packet)을 브로드캐스트 하고, 목적지까지의 노드들 중에서 경로 요청 패킷을 받은 노드는 어느 노드로부터 경로 요청 패킷을 수신했는지 라우팅 테이블에 저장한다. 경로 요청 패킷을 수신한 목적지 노드는 가장 좋은 경로를 선택해서 경로 응답 패킷(Route Reply Packet)을 송신자에게 전달하는데 이 과정에서 라우팅 테이블에 기록된 송신자로의 경로 정보가 이용되어진다. 경로 응답 패킷은 경로 요청 패킷이 설정해 놓은 경로를 이용하기 때문에 기본적으로 양방향 링크를 지원해야 한다. 경로유지는 링크 실패 통지 메시지를 통해서 이루어지며, 링크 실패(Link Failure)를 감지한 노드는 링크 실패 통지 메시지를 상위 노드를 통하

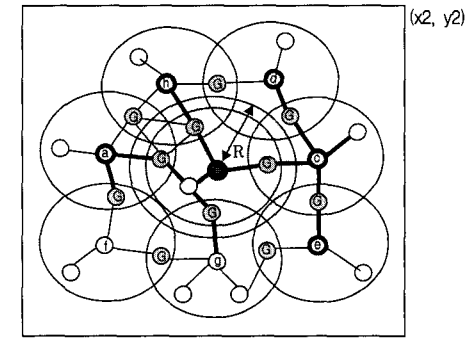


(그림 3) 시스템 모델

여 송신자에게까지 전달하게 되고, 송신자 노드는 다시 경로 설정 절차를 수행한다. 불필요한 패킷의 재전송을 방지하기 위해 이웃 노드들과 지속적으로 헬로우 메시지(Hello Message)를 교환하는 방법을 사용한다.

3.5 코어 영역

애드혹 네트워크의 코어-기반 멀티캐스트 트리 라우팅 알고리즘에서 가장 큰 문제점은 적절한 위치에 코어노드를 두는 것이다. 멀티캐스트 트리의 코어노드를 네트워크의 중앙에 위치시키기 위해서 네트워크의 중앙 영역인 코어 영역을 정의하며, 네트워크의 크기와 위치는 가정한 것과 같이 고정되어 있으며, 모든 노드들은 네트워크가 초기화 될 때와 각 노드들이 초기화 될 때에 자신의 위치 정보를 브로드캐스트 함으로써 알 수 있다. (그림 4)는 반지름이 R(Radius)인 코어 영역이다.



(그림 4) 코어 영역

3.6 코어 노드의 선정

멀티캐스트 트리의 코어노드를 네트워크의 중앙에 위치시키기 위해서 네트워크의 중앙 영역인 코어 영역을 정의하였다. 코어 영역의 중앙에 가장 가까이 위치한 클러스터의 헤드를 코어 노드로 선정한다. 코어 영역 안에 있던 코어 노드가 코어영역을 벗어나면 멀티캐스트 트리 상에 있는 클러스터 헤드들 중에서 클러스터 헤드의 위치가 네트워크의 중앙인 코어 영역에 위치한 클러스터 헤드로 교체되어 코어 노드의 역할을 대체한다.

3.7 코어 노드 선정 알고리즘

코어 노드의 선정은 다음과 같은 가중치 클러스터 헤드 선출 알고리즘이 완료된 후에 클러스터의 헤드들 중에서 현재의 위치가 네트워크의 중심에 가장 가까운 노드를 코어 노드로 선정한다. 클러스터 헤드 선출 알고리즘은 시스템이 활성화될 때와 현재의 도미넌트 셋(Dominant set)이 전체 노드들을 포함할 수 없을 때 진행된다. 이 과정의 출력은 도미넌트 셋이 된다.

단계 1 : 각 노드 v의 이웃노드를 찾는다. (즉, 모든 노드에서 전송 범위 내의 노드들을 찾아 그 개수인 노드

연결성(degree) dv 를 부여한다.

단계 2: 각 노드 v 에 대하여, 노드의 연결성 차 Δv 를 계산한다. 클러스터헤드가 이상적으로 처리할 수 있는 노드들의 수는 $\delta = 2$ 로 미리 설정한다. 이것은 클러스터헤드에 과부하가 걸리는 것을 막고 시스템의 효율성이 기대된 수준에서 유지되는 것을 보증한다.

$$\Delta v = |dv - \delta| \quad (1)$$

단계 3: 각 노드 v 에 대하여 그 이웃 노드들과의 거리 합 Dv 를 계산한다. Dv 에 대한 값은 임의로 선택되어지며, 주로 에너지 소모와 관계가 있다. 먼 거리 통신을 위하여 더욱 많은 전력이 요구되어지며, 거리의 제곱의 합으로 선형적인 것 보다 상당히 더 빠르게 증가하는 연결을 지원하도록 요구된다.

$$Dv = \sum_{v' \in N(v)} \{dist(v, v')\} \quad (2)$$

단계 4: 현재 시간 T 까지 모든 노드에 대하여 진행중인 평균 속도를 계산한다. 이것은 이동성의 척도를 제공하고 Mv 에 의하여 표시된다. Mv 에 대한 값은 임의로 선택되어지며, $Mv = 0$ 는 전혀 움직이지 않는 노드를 의미한다. 적은 이동성을 가진 노드가 클러스터헤드가 되면 안정된 클러스터를 유지할 수 있다. 시간 t 에서 노드 v 의 좌표는 (X_t, Y_t) 이고 시간 $(t-1)$ 에서의 좌표는 (X_{t-1}, Y_{t-1}) 이다.

$$Mv = \frac{1}{T} \sum_{t=1}^T \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2} \quad (3)$$

단계 5: 한 노드 v 가 클러스터 헤드로서 동작하는 동안의 누적되는 시간 Pv 를 계산한다. Pv 는 얼마나 많은 배터리 전력이 보통의 노드보다 클러스터헤드라고 가정했을 때 소모되었는지를 의미하며 임의의 값이 선택된다.

단계 6: 각 노드 v 에 대하여 조합된 가중치 Wv 를 계산한다.

$$Wv = \omega_1 \Delta v + \omega_2 Dv + \omega_3 Mv + \omega_4 Pv \quad (4)$$

각 노드 v 에서 $Wv = \omega_1 \Delta v + \omega_2 Dv + \omega_3 Mv + \omega_4 Pv$ 이며 가장 작은 조합 가중치 Wv 를 가진 노드가 클러스터 헤드가 된다. $\omega_1, \omega_2, \omega_3, \omega_4$ 는 시스템 파라미터에 대응하는 가중치로 이들 계수의 합은 1이다. 본 논문에서는 ω_1 은 노드의 위치계수 가중치 계수로 가장 높은 가중치를 부여하였다. 그 이유는 멀티캐스트 트리의 코어노드를 네트워크의 중앙에 위치시키기 위함이다. ω_2 는 이동성의 척도인 연속적인 노드의 속도 평균의 가중치 계수이며, ω_3 는 이웃 클러스터 헤드들과의 거리의 합(송신자로부터 홉 카운트)의 가중치 계수, ω_4 는 클러스터 헤드로서의 동작 시간에 대한 가중치 계수이다.

단계 7: 클러스터 헤드로서 가장 작은 값 Wv 를 가진 노드를 선택한다. 선택된 클러스터헤드의 모든 이웃 노드들은 더 이상 클러스터 헤드의 선출 과정에 참여하지 않는다.

단계 8: 모든 노드들 중에서 클러스터 헤드의 선출이 종료된 후에 선출된 클러스터 헤드 노드들 중에서 위치계수가 네트워크의 중심에 가장 가까운 노드를 코어 노드로 결정하고 종료한다.

단계 9: 아직 클러스터 헤드로서 선택되지 않은 노드로 남아 있거나, 클러스터로 할당되어져 있는 동안 단계 2에서 단계 7 과정을 반복한다.

3.8 멀티캐스트

멀티캐스트는 방송이라는 매체의 기본적인 특성 때문에 그룹 통신에 사용되는 경우에 무선 네트워크에서 큰 효율성을 지니고 있다. 그러나 무선 네트워크의 특성 상 잦은 패킷의 분실과 전송 에러가 발생하기 때문에 데이터 전달과 데이터 전달 구조를 신뢰성 있게 유지하기 위해서는 많은 패킷의 재전송과 지연 오버헤드가 발생된다. 멀티캐스트의 참석과 이탈, 유지 알고리즘을 정의한다.

3.8.1 멀티캐스트 트리 참석

송신자 노드는 패킷 전송을 시작하기 위해서 멤버 노드들에게 경로 설정에 참여해 주길 바라는 패킷을 브로드캐스트 한다. 멀티캐스트 그룹에 참석하고자 하는 노드 N 은 목적지 노드의 주소에 멀티캐스트 그룹 주소(Group address)의 값을 부가한 참가 요청(JOIN_REQUEST)패킷을 브로드캐스트 한다. 참가 요청 패킷을 수신한 멀티캐스트 트리의 멤버 노드들은 참가 인정(JOIN_ACKNOWLEDGE)패킷을 전송한다. 이 패킷에는 코어 노드까지의 홉(Hop) 수와 주소, 위치, 방향, 속도 등의 정보를 부가하여 노드 N 에게 전송한다. 노드 N 은 전송 받은 참가 인정(JOIN_ACKNOWLEDGE)패킷들 중에서 코어 노드까지의 최단거리 알고리즘에 의해 거리가 가장 짧은 멤버 노드를 찾고 그 노드에게 참가 확인(JOIN_CONFIRM) 패킷을 전송하여 멀티캐스트 트리를 형성하고, 그 이외의 경로에는 삭제패킷을 전송하여 다른 경로가 선정되었기 때문에 자신에게 더 이상의 패킷을 포워딩 할 필요가 없음을 상위 노드에게 알린다.

3.8.2 멀티캐스트 트리 이탈

노드 Q 가 멀티캐스트 그룹에서 벗어나기 원하면 자신의 부모 노드에게 이탈 요청(QUIT_REQUEST)패킷을 전송한다. 부모 노드는 노드 Q 에게 이탈 인정(QUIT_ACKNOWLEDGE)패킷을 전송함으로써 멀티캐스트 트리를 이탈하게 된다. 만일 부모 노드가 멀티캐스트 멤버 노드가 아니면서 멀티캐스트 트리에 속한 노드이고, 자식 노드가 없다면 부모 노드는 멀티캐스트 이탈 알고리즘을 반복하게 된다.

3.8.3 멀티캐스트 트리 유지

애드혹 네트워크에서는 노드의 이동 또는 네트워크 상황 변화에 따라 네트워크가 2개 혹은 그 이상으로 분리될 수 있다. 초기 경로 설정 과정을 짧은 주기 안에 반복하게 되면 지속적으로 멀티캐스트 트리를 유지할 수 있게 되지만, 트리 구성 자체에도 많은 시간이 소요되며, 트리 구성을 위한 제어 트래픽도 많이 증가하게 된다. 이것은 무선 네트워크 환경에서 가장 중요한 요소 중의 하나인 대역폭의 낭비를 초래할 수 있다. 본 논문에서는 이와 같은 오버헤드를 줄이기 위해서 네트워크가 분리가 발생되었을 때 확장 링 탐색을 통하여 멀티캐스트 트리에 다시 접합하는 기법을 사용한다. 멀티캐스트 이탈 또는 네트워크 상황 변화로 멀티캐스트 트리로부터 분리된 노드는 다시 멀티캐스트 트리에 다시 접합하기 위해서 직접 교신이 가능한 한 홉 거리의 클러스터 헤드에게 멀티캐스트 멤버인지를 질의하는 접합 요청(GRAFT_REQUEST)패킷을 전송한다. 한 홉 노드들 중에 멀티캐스트 멤버가 존재하지 않는다면, 홉 카운트를 늘려가면서 부모 노드가 될 노드를 검색한다. 멀티캐스트 멤버인 노드들은 접합 요청에 응답하여 접합 패킷을 전송한다. 분리되었던 노드는 최초의 접합 패킷을 전송한 노드를 부모 노드로 선택하여 접합 인정(GRAFT_ACKNOWLEDGE)패킷을 해당 노드에 전송한다.

4. 시뮬레이션 환경

시뮬레이션 환경은 MS-Windows 2000 Server/MS-Visual C++6.0 환경에서 Global Mobile Simulation(Glomosim) 라이브러리를 사용하였다. Glomosim 라이브러리는 UCLA에서 무선 네트워크 시뮬레이션을 위하여 개발한 도구로서 라이브러리 기반으로 순차적이고 병렬처리가 가능하며 여러 개의 라이브러리 모듈로 구성되어 있어서 특정 통신 프로토콜의 시뮬레이션을 수행할 수 있다. 이 라이브러리는 C언어 기반 병렬 시뮬레이터 언어인 PARSEC으로 개발되었다[29]. 그러므로 새로운 프로토콜과 모듈들은 PARSEC을 사용하여 프로그램 되거나 수정하여 사용할 수 있다.

4.1 시뮬레이션 모델

이동 지역은 1000×1000m이며 이동 노드는 100개를 랜덤하게 분포하게 하였다. 코어 영역의 반경은 20m에서 20m씩 반경을 증가 시켜가면서 시뮬레이션 하였다. 각 노드의 무선 전송범위는 100m이며, 채널의 대역폭은 100KB/sec이다. 멀티캐스트 데이터 패킷의 크기는 데이터가 512Kbyte, 라우팅 테이블은 2Kbit, 제어 패킷은 500bit이며, 트래픽 패턴은 CBR(Constant Bit Rate)소스로 512byte이다. 송신자는 멀티캐스트 멤버들 중에서 임의로 선택된다. 멤버 노드들은 시뮬레이션 초기의 멀티캐스트 세션에서 조인되며 멤버로서 유지된다.

5. 성능 평가

본 논문에서 제안한 가중치 클러스터링을 이용한 코어 기반 멀티캐스트 트리와 기존의 코어 기반 멀티캐스트 트리의 성능을 비교하여 평가하였다.

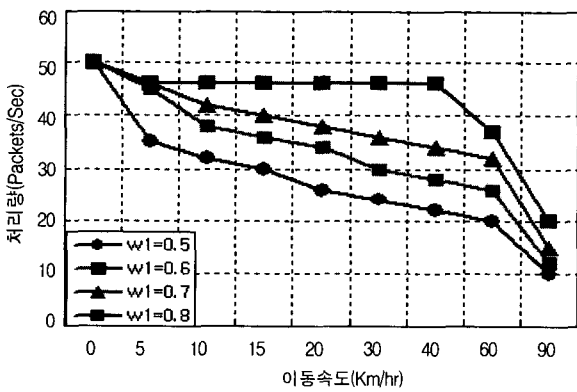
네트워크 안의 노드들은 지정된 이동지역인 1000×1000m 안에서 이동한다. 노드들은 초기에는 균등하게 분포되어 있으며, 정해진 시간동안 일정한 속도로 이동한다. 만일 이동 중에 네트워크의 밖으로 벗어나는 경우 다시 안으로 바운드되어 잔여 시간동안 이동을 계속한다. 두 이동 노드 상의 의 거리가 전송범위를 벗어나면 두 노드 사이의 링크는 없어지고, 모든 노드는 같은 전송 범위를 갖는다. 모든 무선 링크의 대역폭은 100KB/sec이다. 네트워크에는 오직 1개의 트리 만 존재하며, 멀티캐스트 송신자 노드가 지수분포로 데이터 패킷을 전송하며, 20KB의 데이터를 전송한 후 송신자 노드는 일반노드가 된다. 시뮬레이션에서 CBT와 제안된 WCBT(Weighted Clustering Core-Based Multicast Tree) 방식을 비교하였다.

비교 측정 단위로는 첫째 전송거리(Delay), 즉 멀티캐스트 데이터 패킷이 송신자 노드로부터 멀티캐스트 구성원 노드에게 도착하기까지의 평균 홉 수로 정의한다. 두 번째는 오버헤드로 시뮬레이션 시간동안 네트워크에 존재하는 모든 노드에서 발생한 플러딩 횟수로 정의한다. 이와 같이 정의한 이유는 애드혹 네트워크의 라우팅에서 사용되는 제어 패킷의 대부분은 플러딩에 의하여 발생되는 패킷이므로 다른 제어 패킷의 수는 작기 때문에 무시 할 수 있다.

멀티캐스트에서 플러딩이 발생하는 경우는 첫째 일반 노드가 멀티캐스트 트리에 참석하기 위하여 JOIN_REQUEST 패킷을 방송하는 경우 둘째는 링크 실패를 발견한 링크의 지식 노드가 링크 복구를 위해 JOIN_REQUEST 패킷을 방송하는 경우와 세 번째는 송신자 노드가 멀티캐스트 데이터를 보낼 때 트리의 경로를 알고 자고자 하는 경우에 ROUTE_REQUEST 패킷을 전송할 때, 네 번째는 코어노드의 이웃 노드가 존재하지 않을 때, 코어 영역에 존재하는 노드들을 찾기 위해서 NEW_CORE_REQUEST 패킷을 전송하는 경우이다.

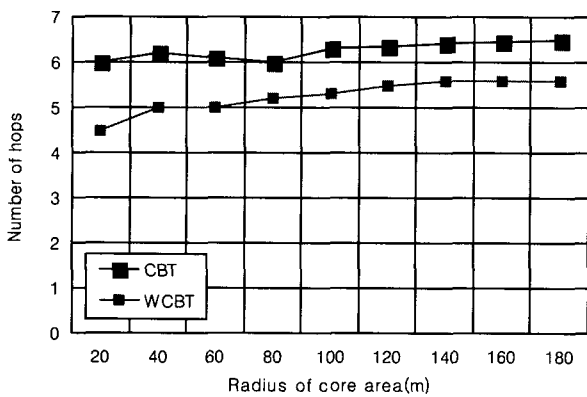
코어 노드 선정 기준으로는 GPS의 정보인 위치, 평균이동속도, 이웃 클러스터 헤드들과의 거리 합, 클러스터 헤드로서 동작 시간에 조합 가중치를 적용한 값을 사용하였다. 성능평가에서 위치계수의 가중치 ω_1 을 0.5, 0.6, 0.7, 0.8인 경우에 $\omega_1 + \omega_2 = 0.9$ 로 가정하고 ω_1 의 위치계수 가중치를 이동속도 변화에 대한 패킷 처리율(패킷 수)과 각 가중치에서 이동속도가 증가할 때 성능을 평가하였다. 그 결과 위치계수의 가중치가 0.8일 때 가장 좋은 성능을 (그림 5)와 같이 나타내어 가중치를 0.8로 설정하였다. 코어 영역에서 ω_1 은 위치 계수(클러스터헤드 위치)로 가중치를 0.8, ω_2 는

이동성의 척도인 연속적인 노드의 속도 평균의 가중치는 0.1, ω_3 는 이웃 클러스터 헤드들과의 거리 합(송신자로부터의 홉 카운트)으로 가중치를 0.05, ω_4 는 클러스터 헤드로 동작 시간은 가중치를 0.05에 두고 조합된 가중치 Wv 를 계산한다. 각 노드 v 에서 $Wv = \omega_1 \Delta v + \omega_2 Dv + \omega_3 Mv + \omega_4 Pv$ 이며 가장 작은 조합 가중치 Wv 를 가진 노드가 클러스터 헤드라 되며, 이 클러스터 헤드 노드들 중에서 위치계수가 네트워크의 중심에 가장 근접한 노드를 코어 노드로 선정한다. $\omega_1 \sim \omega_4$ 계수는 시스템 파라미터에 대응하는 가중치로 이들 계수의 합은 $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$ 이 되도록 임의로 선택된다.

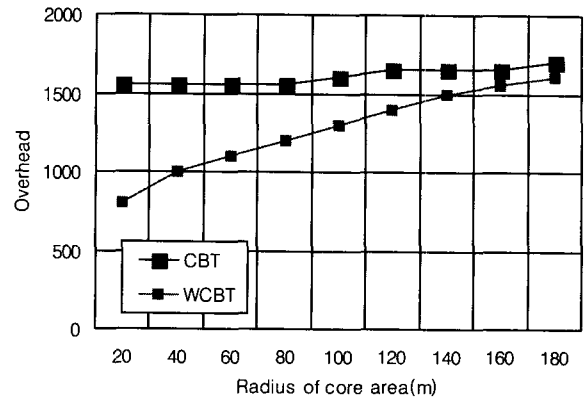


(그림 5) 위치 계수 가중치별 이동속도에 따른 처리율

따라서 시뮬레이션에서 반경과 이동속도의 값을 변화시켜가면서 제안된 알고리즘을 평가하였다. 반경의 변화에 따른 성능평가는 (그림 6)과 같이 제안한 방식이 고정 CBT보다 전송거리에서는 5~15% 감소되었으며, 오버헤드도 20~30% 감소하였으며 반경이 증가할수록 전송거리와 오버헤드가 증가함을 알 수 있었다. 위치 계수가 코어 영역 밖이면 패킷 전송 실패 및 지연 오버헤드가 크게 발생되며, 코어 영역을 20m에서 20m씩 증가하면서 코어 영역이 전송 지연에 미치는 영향을 (그림 6)과 (그림 7)에 나타내었다.



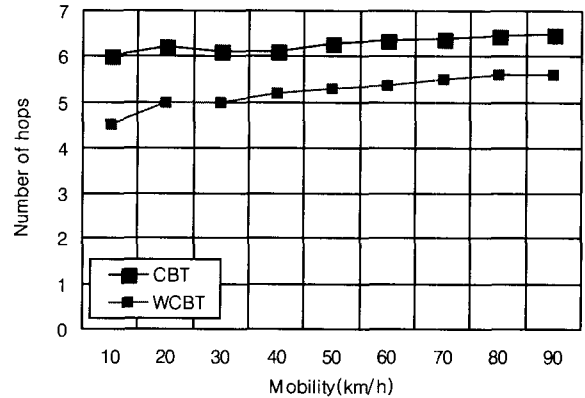
(그림 6) 코어 영역의 반경과 전송 지연 비교



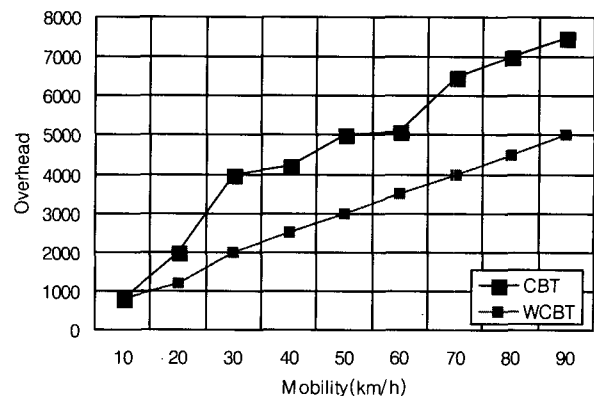
(그림 7) 코어 영역의 반경과 오버헤드 비교

(그림 6)과 (그림 7)은 코어 영역의 반경이 증가할수록 전송 지연(홉 수)가 증가하며 가중치 클러스터링 방식이 기존의 방식에 비하여 효율적이며, 오버헤드도 감소함을 알 수 있었다.

(그림 8)과 (그림 9)는 이동 속도의 평균(이동성)에 따른 전송 지연 및 오버헤드를 비교하였다. 이동성의 척도인 노드의 속도 평균이 커질수록 경로 재설정과 재전송 횟수가 증가하게 되며, 코어 노드로서 동작 시간이 클수록 안정적인 멀티캐스트 트리가 유지됨을 알 수 있었다.



(그림 8) 이동성과 전송지연 비교



(그림 9) 이동성과 오버헤드 비교

가중치 클러스터링을 이용한 코어-기반 멀티캐스트 트리 방식이 코어 노드의 위치와 노드의 이동성에 따라서 전송 거리와 제어 오버헤드가 기존의 코어 기반 멀티캐스트 방식보다 향상됨을 알 수 있었으며, 코어 노드를 선택할 때 이동성이 적고 네트워크의 중심에 가까울수록 멀티캐스트의 트리가 안정되며 전송거리도 짧아짐을 알 수 있었다.

6. 결 론

본 논문에서는 애드혹 네트워크의 코어-기반 멀티캐스트 트리 라우팅 알고리즘에 가중치 클러스터링을 이용하여 효율적으로 코어-기반 멀티캐스트 트리를 유지하는 기법을 제안하였다. 코어노드의 이동 정도와 네트워크 상에서의 위치에 따라 전송거리와 제어 오버헤드가 증가하므로, 멀티캐스트의 효율성을 증가시키기 위해서 코어노드를 동적으로 선택할 필요가 있다. 따라서 코어노드를 선택할 때 이동성이 작고 네트워크의 중심에 가까울수록 멀티캐스트 트리가 안정되며 전송거리가 짧아진다. 이와 같이 코어노드의 이동으로 인하여 멀티캐스트 트리의 재구성으로 인한 오버헤드가 전체 네트워크에 미치는 영향이 크므로 가중치 클러스터링을 이용한다. 제안된 코어 노드 선정 알고리즘으로 전체 네트워크에서 가중치를 부여한 후 클러스터링을 통하여 클러스터의 헤드들이 선택된다. 클러스터 헤드의 선택을 위하여 GPS의 정보인 위치, 평균이동속도, 이웃 클러스터 헤드들과의 거리 합, 클러스터 헤드로서 동작 시간에 조합 가중치를 적용한 값을 사용하였으며, 가중치에서 위치에 대한 가중치를 가장 많이 주어 선정하였다. 그 이유는 코어노드의 위치가 네트워크의 중심에 가까울수록 멀티캐스트 트리의 안전성과 전송거리가 짧아지기 때문이다. 이와 같이 클러스터링에 의하여 멀티캐스트 트리를 구성하고 유지함으로써 코어 노드의 이동 시 새로운 코어 노드의 선발에 따른 지연의 감소와 기존의 멀티캐스트 트리를 재구성 없이 사용할 수 있어 데이터 전달 구조 오버헤드가 감소하였다.

참 고 문 헌

[1] David B. Johnson and David A. Maltz, "Mobile Computing-Dynamic Source Routing in Ad-hoc Wireless Networks," Kluwer Academic Publishers, pp.153-181, 1996.
 [2] Rohit Dube and et al, "Signal Stability-Based Adaptive Routing(SSA) for Ad-hoc Mobile Networks," IEEE Personal Communications, Feb. 1997.
 [3] C-K. Toh, "Wireless ATM and AD-HOC networks," Kluwer Academic Publishers.
 [4] D. Bertsekas and R. Gallager, *Data Network, Second Ed.*, Prentice Hall Inc., pp.404-410, 1992.
 [5] C. Perkins, P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (DSDV) for Mobile Computers," ACM SIGCOMM, Oct., 1994.
 [6] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," ACM Mobile Net-

works and Applications Journal, Special issue on Routing in Mobile Communication Networks, 1996.
 [7] V. Park and S. Corson, "Temporally-ordered routing algorithm (TORA) Version 1," Internet draft, IETF, Aug., 1998.
 [8] Josh Broch, David B. Johnson and David A. Maltz, "The dynamic source routing in Ad-hoc wireless networks," In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, pp.153-181, 1996.
 [9] Charles Perkins, Elizabeth M. Royer and Samir R. Das, "Ad-hoc on demand vector (AODV) routing," Internet draft, IETF, June, 1999.
 [10] C. K. Toh, "Long-lived Ad-hoc routing based on the concept of associativity," Internet draft, IETF, Mar., 1999.
 [11] G. Aggelou and R. Tafazolli, RDMAR : "A bandwidth-efficient routing protocol for mobile ad-hoc networks," proceedings of The Second ACM International Workshop on Wireless Mobile Multimedia (WoWMoM), Seattle, Washington, Aug., 1999.
 [12] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad-hoc network routing protocols," proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, ACM, Dallas, Oct., 1998.
 [13] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc network," MobiCom '99, Aug., 1999.
 [14] S. Corson and J. Macker, "Mobile ad-hoc networking (MANET) : Routing protocol performance issues and evaluation considerations RFC 2501," IETF, June, 1999.
 [15] Elizabeth Royer and Charles E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," Mobicom '99, Aug., 1999.
 [16] C. W. Wu, Y. C. Tay and C-K, Toh, "Ad-hoc multicast routing protocol utilizing increasing id-numbers (AMRIS) Functional Specification," Internet draft, IETF, Nov. 1998.
 [17] E. Bommaiah, M. Lui, A. McAuley and R. Talpade, "AM-Route : ad-hoc multicast routing protocol," Internet draft, IETF, Aug., 1998.
 [18] Chuck Semeria and Tom Maufer, "Introduction to IP Multicast Routing," Technical Report, 3Com, 1996.
 [19] <http://pado.etri.re.kr>.
 [20] A. Ballardie, P. France and J. Crowcroft, "Core Based Trees(CBT) : An Architecture for Scalable Inter-Donain Multicast Routing," ACM SIGCOMM'93, pp.85-95, Oct., 1993.
 [21] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," IEEE Journal on Selected Areas in communication, Sep., 1997.
 [22] J. T. Tasi and M. Gerla, "Multicluster, mobile, multimedia radio network," ACM/Baltzer Journal of Wireless Networks, 1995.
 [23] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," Proc. Of MOBICOM '99, 1999.
 [24] 이미정, "Ad-hoc 네트워크를 위한 멀티캐스트 라우팅 프로토콜", Telecommunications Review, 제12권 제3호, June, 2002.
 [25] J. J. Garcia-Luna-Aceves and Ewerton L. Madruga, "The

Core-Assisted Mesh Protocol," ACM/Balzer Networks and Applications Journal, 1999.

[26] C. C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," Baltzer Cluster Computing, Vol.1, No.2. 1998.

[27] G. Dommety and R. Jain, "Potential networking applications of global positing system (GPS)," Tech. Rep. TR-24, CS Dept., The Ohio State University, April, 1996.

[28] Mainak Chatterjee, Sajal K. Das and Damla Turgut, "A Weighted Clustering Algorithm for Mobile Ad-hoc Networks," Cluster Computing, May, 2002.

[29] UCLA Computer Science Department Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, Glomosim : A Scalable Simulation Environment for Wireless and Wired Network Systems.



박 양 재

e-mail : yjpark@gcgc.ac.kr
 1999년 인하대학교 공과대학 전자공학과 (공학사)
 1990년 인하대학교 산업대학원 정보공학과 (공학석사)
 2002년 인하대학교 대학원 전자계산공학과 (박사과정 수료)

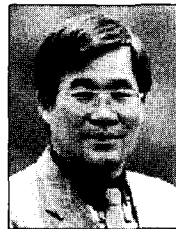
1984년~1985년 주식회사 서통 전자사업부 기술개발과 근무
 1985년~1993년 인하공업전문대학 전자과 조교
 2001년~2002년 주식회사 이메디피아 원격의료연구소 연구소장
 1993년~현재 가천길대학 전산정보처리과 부교수
 관심분야 : MANET, 피어투피어 네트워킹, HCI, 음성신호처리



한 승 진

e-mail : softman@inha.ac.kr
 1990년 인하대학교 전자계산학과(이학사)
 1992년 인하대학교 대학원 전자계산공학과 (공학석사)
 2002년 인하대학교 대학원 전자계산공학과 (공학박사)

1992년~1996년 대우통신 종합연구소 교환연구단
 1996년~1998년 SK텔레콤 디지털 사업본부
 2002년~현재 인하대학교 컴퓨터공학부 강의전담 조교수
 관심분야 : Mobile IP, MANET, 멀티미디어통신, IMT-2000, 음성신호처리, 홈네트워킹



이 정 현

e-mail : jhlee@inha.ac.kr
 1977년 인하대학교 전자공학과(공학사)
 1980년 인하대학교 대학원 전자공학과 (공학석사)
 1988년 인하대학교 대학원 전자공학과 (공학석사)

1979년~1981년 한국전자기술연구소 시스템 연구원
 1984년~1989년 경기대학교 교수
 1989년~현재 인하대학교 컴퓨터공학과 교수
 관심분야 : 자연어처리, HCI, 정보검색, MANET, 음성신호처리