

An Efficient Cluster Routing Protocol Based on 2-level Tree for Wireless Ad Hoc Networks

Lee Young Joon[†] · Kim Sung Chun^{**}

ABSTRACT

We proposed a 2 level tree based cluster based routing protocol for mobile ad hoc networks. it is our crucial goal to establish improved clustering's structure in order to extend average node life-time and elevate the average packet delivery ratio. Because of insufficient wireless resources and energy, the method to form and manage clusters is useful for increasing network stability. but cluster-head fulfills roles as a host and a router in clustering protocol of Ad hoc networks environment. Therefore energy exhaustion of cluster-head causes communication interruption phenomenon. Effective management of cluster-head is key-point which determines the entire network performance. The scheme focuses on improving the performance the life time of the network and throughput through the management of cluster-heads and its neighbor nodes. In simulation, we demonstrated that it would obtain averagely better 17% performance than LS2RP.

Keywords : Ad Hoc Networks, Routing Protocol, 2-Level Tree

무선 애드 혹 네트워크에서 에너지 효율적인 2-level 트리 기반의 클러스터 라우팅 프로토콜

이 영 준[†] · 김 성 천^{**}

요 약

본 논문에서는 MANET 환경에서 2-level 트리 기반 클러스터링 라우팅 알고리즘을 제안한다. 이를 통해 네트워크에 속해 있는 무선 노드들의 생존시간을 증가시키고 네트워크 데이터 전송률을 향상시키는 것이 주요 목표이다. 무선 네트워크의 기본 특성인, 무선 자원과 배터리 상의 제약으로 인해 클러스터 형성과 유지 및 관리기법은 네트워크의 안정성을 향상시키기 위해서는 필수적이다. 그러나 애드 혹 네트워크 환경의 클러스터링 기법에서 클러스터 헤드는 호스트의 기능을 수행함과 동시에 라우팅 기능을 담당하는 라우터의 역할까지 수행한다. 따라서 클러스터 헤드의 에너지 고갈이나 부재는 클러스터를 붕괴시키고, 전체 네트워크의 통신을 어렵게 하며, 나아가서는 네트워크의 수명을 단축시킨다. 결국 클러스터 헤드의 효율적인 관리는 전반적인 네트워크의 성능을 결정짓는 핵심이라고 볼 수 있다. 때문에 본 제안 기법은 클러스터 헤드와 이웃 노드들의 유지 및 효율적인 관리를 통해 전체적인 네트워크의 생존시간과 Throughput을 향상시키고자 하였다. 또한 시뮬레이션을 통해 MANET 환경에서 기존 LS2RP 기법에 비해 Qos 측면에서 평균 17%의 성능 향상이 이루어짐을 알 수 있었다.

키워드 : 애드 혹 네트워크, 라우팅 프로토콜, 2-Level Tree

1. 서 론

애드 혹 네트워크는 Access Point나 기지국과 같은 기반 구조 없이 이동 노드들로만 구성되는 자율적이고 독립적인 네트워크이다. 각각의 노드들은 이동성을 가지기 때문에 이

들로 이루어진 네트워크는 다양한 형태를 이루게 된다. 따라서 기존의 단일 홉을 이용하는 클러스터링 기법보다는 멀티 홉을 사용한 클러스터링 방법이 많은 노드들을 효율적으로 관리하는 데 유리하다. 또한 기본적으로 무선 자원과 배터리 상의 제약을 가지기 때문에 클러스터 형성과 유지 및 관리를 통해 네트워크의 안정성을 향상시키는 알고리즘이 필요하다.

특히, 클러스터 헤드가 호스트의 기능을 수행함과 동시에 라우팅 기능을 담당하는 라우터의 역할까지 수행하기에 클러스터 헤드의 에너지 고갈이나 부재는 클러스터를 붕괴시키고, 전체 네트워크의 통신을 어렵게 하며, 나아가서는 네

* 이 논문은 2013년도 정부(미래창조과학부)의 지원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2012R1A1A2009558).

[†] 준 회 원 : 서강대학교 컴퓨터공학과 공학석사

^{**} 정 회 원 : 서강대학교 컴퓨터학과 교수

Manuscript Received : February 6, 2014

First Revision : March 25; Second Revision : April 25, 2014

Accepted : April 28, 2014

* Corresponding Author : Kim Sung Chun(ksc@sogang.ac.kr)

트위크의 수명을 단축시킨다. 결국 클러스터 헤드의 효율적인 관리는 전반적인 네트워크의 성능을 결정짓는 핵심이라고 볼 수 있다.[1, 2]

클러스터링 기법은 클러스터 헤드를 중심으로 이웃노드들이 연결된 클러스터를 구성하고 유지/관리하는 기법으로 클러스터 헤드는 해당 클러스터에 존재하는 모든 노드들의 ID, 잔여 에너지와 같은 정보들을 저장하고 클러스터 관리에 활용한다. 기존에는 특정 수식을 통해 얻어진 값을 기준을 활용한 클러스터 헤드의 선정 및 클러스터 구성에 초점이 맞춰져있다.

하지만, 최근에는 RAID 시스템을 활용한 클러스터의 안정적인 유지/관리에 관심이 새롭게 집중되고 있다. RAID 시스템을 활용하여 클러스터의 안정성을 향상시키게 되면 기존에 생성된 클러스터의 변경이 적어질 뿐만 아니라 이전에 저장된 정보들의 재사용이 가능하기 때문에 클러스터 유지에 소모되는 에너지가 적어지게 되고, 이는 곧 전반적인 네트워크의 성능 향상을 가져온다.[3]

무선 애드 혹 네트워크에서 LS2RP(Link-State RAID Routing Protocol)와 같이 RAID기법을 활용하여 안정성을 강화한 클러스터링 기법이 갖는 가장 큰 문제점은 클러스터 헤드의 정보를 해당 클러스터에 속한 모든 노드들에게 복사하여 백업시킨다는 점이다. 이런 단순한 백업 작업은 불필요한 네트워크의 혼잡과 자원의 소모를 발생시킨다. 때문에 본 논문에서는 이러한 점을 개선한 2-level 트리 기반 클러스터링 기법을 통해 데이터 백업이 용이할 뿐만 아니라, 빈번한 네트워크 토폴로지의 변화에도 쉽게 적응할 수 있으며, 오버헤드를 최소화시키고자 하였다.

2. 기존의 클러스터링 기반 라우팅 프로토콜

2.1 무선 애드 혹 네트워크에서 클러스터링 라우팅 기법

CBRP(Cluster Based Routing Protocol)는 망 전체를 노드들의 분포에 따라 중복되거나 분리된 여러 개의 클러스터로 나누어 관리한다. 그리고 하나의 클러스터에 클러스터 헤더가 존재하여 클러스터에 속한 노드들의 정보를 관리한다. 클러스터 사이의 라우팅은 각 클러스터 헤더가 가지고 있는 정보들을 이용하여 동적으로 이루어진다. 노드들을 그룹으로 관리함으로써 경로 설정 절차 시에 트래픽의 양을 효과적으로 줄이며, 속도를 향상시킨다. 그러나 CBRP의 단점으로는 클러스터를 형성하고, 유지하는 방법에 대한 비용이 들며, 특히 모든 경로를 클러스터 헤더에 의존하게 되어 헤더의 이동 또는 비정상적인 상태로 인해 헤더의 역할을 수행하지 못할 경우 새로운 헤더가 선출될 때까지 수행을 멈추게 되어 전체 네트워크의 성능을 저해하는 요소가 된다.[4, 5, 6]

이후, 이러한 문제점을 보완하기 위해 WCA와 WBACA가 제시되어졌다. WCA(Weighted Clustering Algorithm)는 GPS(Global Positional System)를 이용하여 얻어진 4개의 변수 값을 기반으로 가중치를 계산하는 알고리즘이며, 가

장 적은 가중치 값을 갖는 노드를 검색하기 위해 “global minima” 방식을 이용한다. 이것은 네트워크에 속해 있는 모든 노드가 서로의 가중치 값을 알고 있다는 전제가 필요하기 때문에 클러스터 헤드를 선정하기 위해서 많은 시간과 오버헤드가 발생하게 된다.[7, 8]

이를 보완하기 위해 제시된 기법인 WBACA(Weighted-Based Adaptive Clustering Algorithm)는 WCA와 같이 GPS를 이용하여 각 노드들의 위치 정보를 획득하는 알고리즘이지만 WCA와는 달리 transmission power, transmission rate, mobility, 그리고 battery power를 고려하여 클러스터 헤드로 가장 적합한 노드를 선정한다. WBACA는 클러스터 헤드 선정 시 발생하는 오버헤드를 줄이기 위해 “local minima” 방식을 사용하기 때문에 해당 클러스터 내에 존재하는 노드들은 자신이 포함된 클러스터를 제외한 다른 클러스터의 가중치 값을 고려하지 않아도 된다. 하지만 네트워크의 형태의 변화가 심하거나 이동성이 높은 경우, 재클러스터링이 빈번하게 발생하여 네트워크의 안정성이 떨어지는 단점이 있다.[9]

2.2 애드 혹 네트워크에서의 RAID 라우팅 기법

무선 애드 혹 네트워크에서 노드는 이동성을 가지고 있기 때문에 클러스터링 기법을 사용할 경우 클러스터 멤버 노드의 이동성에 대한 고려가 필요하다. 특히 클러스터 헤더가 해당 클러스터를 벗어날 경우 해당 클러스터의 동작에 큰 영향을 미치게 된다. 이런 취약점을 보완하기 위해 제시된 기법이 RAID 시스템을 이용한 LS2RP(Link-State RAID Routing Protocol)이다.[3]

LS2RP는 AODV(Ad hoc On-demand Distance Vector)에서 사용하는 query-reply 방식을 사용하며 네트워크는 Super Cluster와 Normal Cluster로 구성된다. Super Cluster는 Normal Cluster의 집합을 의미하며 Super Cluster Head는 Super Cluster 내에 존재하는 클러스터 헤더와 클러스터 멤버들의 정보를 저장하고 해당 Super Cluster를 유지/관리하는 역할을 수행한다. Super Cluster Head 역시 이동성을 가지고 있기 때문에 네트워크 전체의 통신이 불가능한 상황이 발생할 수 있다. 이를 방지하기 위해 미리 Super Cluster Head의 database copy를 이웃 노드들에게 분배 저장하여 Super Cluster Head가 해당 클러스터를 벗어날 경우 이웃 노드들이 가진 database를 결합하여 해당 클러스터 헤더의 정보를 복원한다.

Fig. 1은 LS2RP에서 Super-Cluster Head의 DB를 분산 저장하는 과정을 보여주고 있다. 이렇게 모든 database를 원형으로 분산 저장함으로써 클러스터 헤더가 database의 특정 부분을 이웃 노드들에게 요청했을 때, 모든 노드에게 병렬로 수신받음으로써 데이터 처리 속도를 향상시킬 수 있다.

하지만 LS2RP의 경우 Normal Cluster의 크기를 1-hop으로 한정 짓고 다수의 클러스터를 Super-Cluster로 관리하기 때문에 네트워크가 확장될 경우 관리 비용이 증가할 뿐만 아니라 해당 Super Cluster Head의 트래픽이 증가하는

문제가 발생한다. 또한 Super-Cluster Head의 DB를 이웃 노드들에게 균등하게 분배할 경우 모든 node가 재결합에 참여하는 불필요한 비용이 발생하게 된다. 이는 곧 해당 노드들의 freezing time이 증가함을 의미하며 전체적인 네트워크의 성능을 감소시킨다.

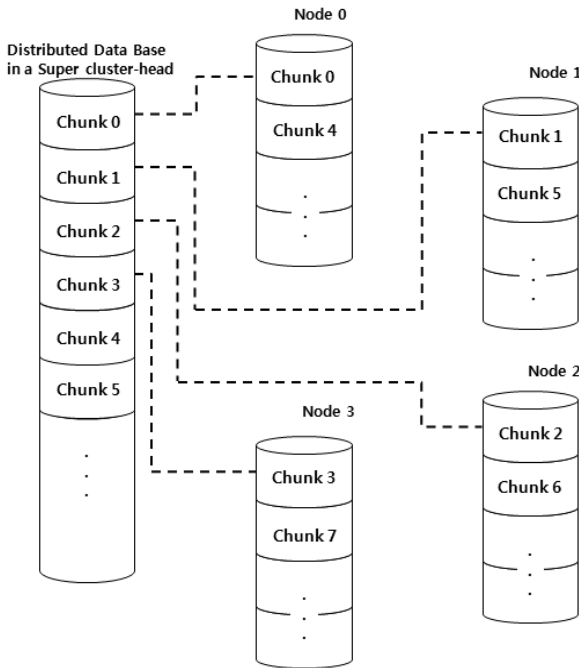


Fig. 1. DB process of Super-Cluster Head in LS2RP

3. 제안 기법

3.1 2-level Tree 기반의 계층적 Cluster

본 논문에서 제안하는 가장 중요한 특징은 2-level 기반의 클러스터를 이용하여 라우팅을 설정한다는 점이다. 또한, 이를 통해 노드 간의 통신을 수행하며, LS2RP에서 적용한 기존 클러스터링 기법에서 클러스터 헤드(CH)와 게이트웨이(GW) 설정 방식 외에는 동일하게 적용된다.[12]

위와 같은 과정을 통해 전체 노드들 간의 클러스터링 구성 작업이 완료되면, 각 노드들 간의 통신은 Query-reply AODV방식을 통해 소스 노드가 클러스터 헤드에게 경로 정보를 요청하여, 클러스터는 먼저 내부 노드를 살펴보고, 목적지 노드가 클러스터 내부에 존재하지 않으면 게이트웨이를 통해 외부 클러스터 헤드에게 목적지 노드의 유무를 질의하여, 최단 거리 경로를 통해 데이터 전송이 이루어진다. Fig. 2는 전체 네트워크 내의 모바일 노드들이 제안 프로토콜에 따라 클러스터링 구성이 완료한 상태의 그림이다. 클러스터 헤드나 게이트 웨이로서의 역할을 부여받지 못한 노드들은 일반 이웃 노드(NH)로 클러스터 내에 존재하게 된다.

또한, 1-level을 이용한 클러스터를 구성할 경우 네트워크의 변화에 능동적으로 대처 할 수 있다는 장점이 존재하지만 네트워크 내에 존재하는 클러스터의 개수가 증가하기 때문에

그만큼 클러스터 헤드의 숫자도 증가하게 된다. Super-Cluster 헤드는 클러스터 멤버들의 주소 및 위치 정보 저장, 라우팅 패스 설정 등의 추가적인 작업을 수행하기 때문에 일반 클러스터 멤버 노드에 비해 에너지 소모가 크다. 따라서 클러스터 헤드 수의 증가는 곧 전체적인 에너지 소비 증가를 의미하고 이는 전반적인 네트워크의 생존 시간이 줄어드는 것을 의미한다. 하지만 2-level 기반의 클러스터를 이용할 경우 클러스터 헤드의 숫자가 적어지기 때문에 네트워크의 생존 시간을 증가시킬 수 있다.

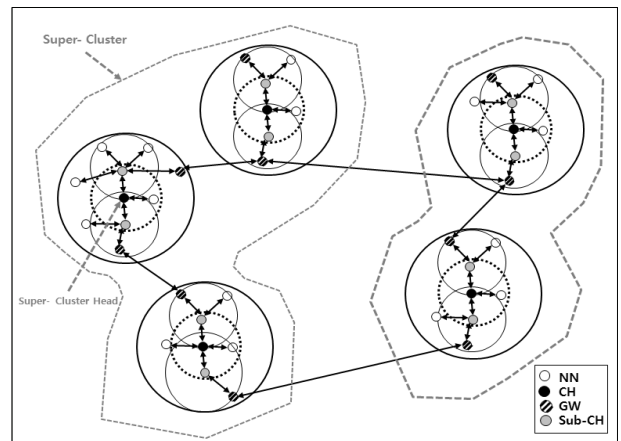


Fig. 2. 2-level tree-based Clustering Architecture

1) 2-level Tree 기반의 계층적 Cluster 구성

초기 네트워크 구성시 모든 노드들은 동일한 에너지를 갖고 시작한다는 가정하에 가장 작은 ID를 갖는 순으로 클러스터 헤드를 선정하는 Lowest ID 기법[12]을 이용하여 클러스터 헤드를 선정한다. 클러스터 헤드로 선정된 노드는 자신의 상태가 클러스터 헤드로 선정되었음을 알리는 CREQ(Cluster-Head REQuest) 패킷을 1-hop 거리로 브로드캐스트 한다.

Fig. 3는 클러스터 헤드가 전송하는 CREQ 패킷의 메시지 구성이다.

Cluster Head ID	Energy	Number of Neighbor Nodes	Cluster ID	Reserved
4 Bytes	4 Bytes	4 Bytes	4 Bytes	8 Bytes

Fig. 3. CREQ(Cluster-Head REQuest) Packet Structure

Cluster Head ID는 CREQ 패킷을 전송하는 노드의 ID를 저장하는 부분으로 노드의 ID는 초기 네트워크가 구성되면서 각 노드들이 가지는 고유값이다. 일반적으로 네트워크에 참여시 부여되며 NIC(Network Interface Card)의 값을 기반으로 한다. Energy는 노드의 현재 에너지 잔여량을 의미한다. #. of Neighbor Nodes는 CREQ 패킷을 전송하는 노드와 연결되어 있는 이웃 노드들의 숫자를 의미한다. 이 값이 클수록 해당 노드가 클러스터 헤드로 존재하는 클러스터의 규모

가 크름을 의미한다. 따라서 CREQ 패킷을 수신한 노드들은 이 값을 비교하여 좀 더 큰 값을 가지는 클러스터에 참여한다. Cluster ID는 클러스터 헤드가 속해 있는 클러스터의 ID 값으로 초기 클러스터 생성시 부여되는 고유 값이다. 각 노드들은 이 값을 이용하여 자신이 어느 클러스터에 속해 있는지 인식하고 클러스터들을 구별하는 데 사용된다. Reserved는 예약된 공간으로 긴급 메시지나 경고 메시지를 전송하는 데 사용되며, 평소에는 NULL 값으로 채워진다.

CREQ를 수신한 노드는 자신의 level을 1-level로 변경하고 해당 패킷에 포함된 정보를 저장한다. 그리고 자신의 주소(ID)가 추가된 NREQ(Neighbor REQuest) 패킷을 1-hop 거리로 브로드캐스트 한다. 만약 이전에 수신한 CREQ 패킷이 존재한다면 비교하여 이웃 노드의 개수가 더 많은 패킷을 저장하고 나머지를 버린다. NREQ 패킷을 수신한 1-level이 아닌 노드는 먼저 도착한 NREQ를 기준으로 해당 패킷에 저장된 정보를 저장하고 자신의 level을 2-level로 변경한다. 이후 NREQ의 송신자에게 RREP 패킷을 전송한다. RREP 패킷을 수신한 1-level 노드는 RREP를 송신한 2-level 노드까지의 경로를 저장하고 클러스터 헤드에게 RREP를 송신한다. 만약 1-level 노드가 일정시간 동안 RREP를 수신하지 못할 경우 자신이 RREP 패킷을 생성하여 클러스터 헤드에게 전송한다.

Fig. 4은 2-level Tree 기반의 클러스터 구성을 도식화한 것이다. 가운데 위치한 노드가 클러스터 헤드이다. 이를 중심으로 점선으로 표현된 원이 1-level을 의미하고 바깥에 표현된 원이 2-level을 의미한다. 1-level에 속한 노드들은 클러스터 헤드의 이웃 노드들로서 1-level 밖에 속해 있는 노드들과 클러스터 헤드 간의 연결을 담당한다. 2-level에 속한 노드들은 RREP 패킷을 1-level 노드를 거쳐 클러스터 헤드에게 전달함으로써 클러스터 내부의 라우팅 경로를 완성시킨다.

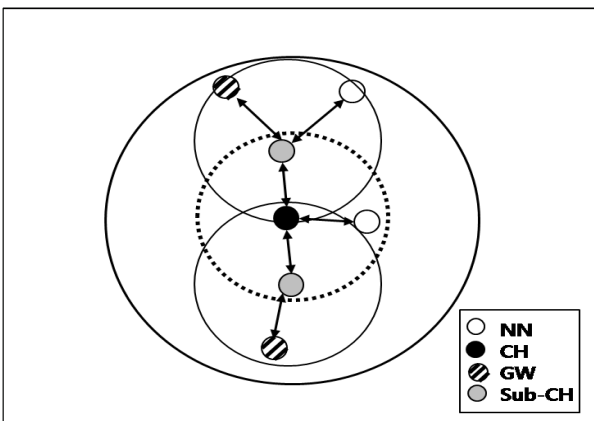


Fig. 4. 2-level Tree based Clustering

2) Gateway를 이용한 Cluster 간의 연결

클러스터 구성이 완료되면 2-level에 존재하는 노드들은 1-hop 거리 내에 자신의 클러스터에 속하지 않은 다른 클러

스터 멤버 노드들이 존재하는지 Hello Message를 전송하여 확인한다. 다른 클러스터 멤버 노드가 존재하는 경우 해당 노드와 연결을 완료하고 해당 노드의 주소와 자신이 외부 노드와 연결되었음을 알리는 기호(Flag)를 저장한 NACK(Node ACKnowledgment) 패킷을 자신이 속해 있는 클러스터의 클러스터 헤드에게 전송한다. 이에 대한 reply가 CH에 수신되면 해당 2-level 노드는 자신의 상태를 게이트웨이로 변경하고 클러스터 외부와 통신하는 통로로서의 역할을 수행한다.

Fig. 5는 제안 프로토콜의 클러스터 구성의 전체적인 흐름을 Level 0인 클러스터 헤드 선정부터 Level 2의 게이트웨이를 구성하는 것을 시간 순서대로 나타낸 순서도이다. 또한, 각각의 클러스터가 구성이 완료가 된 후, 초기 Super-Cluster 헤드 선정은 기존 클러스터 헤드의 선정과 마찬가지로 Lowest ID기법을 통해 임시로 선정하거나 클러스터링 내에 이웃노드 수가 많은 클러스터 헤드를 우선적으로 Super-Cluster 헤드로 선정한다.

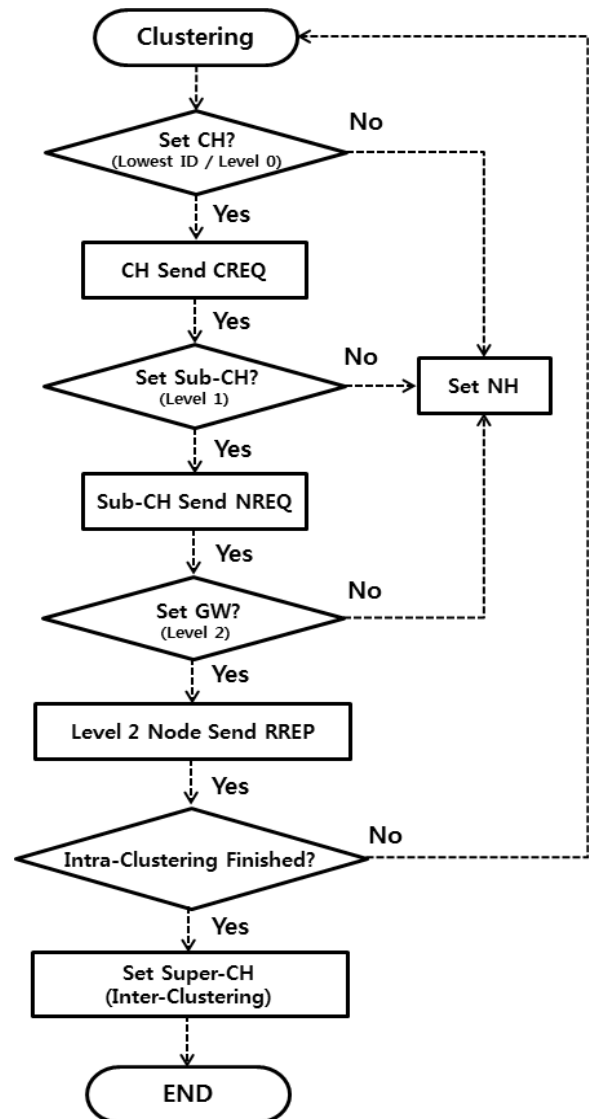


Fig. 5. 2-level Tree-based Clustering Flowchart

3.2 Super-Cluster Head의 Traffic Control

Super-cluster 내에 존재하는 cluster들이 자신이 가지고 있는 routing table 정보를 동시에 super-cluster head(SH)에게 전송하게 되면 패킷이 한꺼번에 몰려 트래픽이 증가할 수 있다. 이런 트래픽 증가는 멤버 클러스터들과의 정상적인 통신을 방해하고 대역폭 초과로 인해 해당 노드로 향하는 패킷이 유실되어 전반적인 네트워크의 성능을 감소시킬 수 있다. 따라서, 패킷 집중으로 발생하는 트래픽을 방지하기 위해 Super Cluster Head와 연결되어 있는 Sub-Cluster Head를 이용하여 트래픽 관리를 수행한다.

Sub-Cluster Head를 통해 트래픽을 관리하게 되면 Super-Cluster Head의 부담을 덜어줄 뿐만 아니라 균등한 에너지 소모를 이끌 수 있다. 또한 Super-Cluster Head로 향하는 데이터들을 중간에 저장함으로써 Super-Cluster Head의 DB를 백업하는 효과도 얻을 수 있다.

1) Sub-Cluster Head 설정

Super-Cluster Head는 자신이 저장하고 있는 라우팅 테이블을 검색하여 Gateway 역할을 수행하는 2-level의 노드들과 자신 사이에 존재하는 1-level 노드들을 선정한다. 선정된 노드들은 자신의 상태를 Sub-Cluster Head로 변경하고 Super-Cluster에 속해 있는 외부 클러스터 헤드와 Super-Cluster Head 사이의 통신에 참여한다.

2) Packet 분산을 통한 Traffic Control

Sub-Cluster Head로 선정된 노드들은 자신을 통해 Super-Cluster Head로 전달되는 모든 패킷을 확인한다. 해당 패킷에 저장된 정보가 라우팅 테이블이라면 해당 패킷을 자신의 DB에 저장한다. 그 이후 아래 3가지 경우로 나뉘어 처리된다.

- ① 해당 패킷의 크기가 패킷 헤더의 크기보다 작은 경우
- ② 해당 패킷의 일부가 이전에 전송된 정보인 경우
- ③ 비슷한 시간에 다수의 패킷이 수신된 경우

①의 경우, Super-Cluster Head에 집중되는 트래픽의 양을 감소시키기 위해 다음 패킷이 수신될 때까지 대기하였다가 합쳐서 전송한다. 하지만 만약 일정 시간동안 수신되지 않을 경우 해당 패킷을 바로 Super-Cluster Head에게 전송한다. (일반적인 네트워크 통신에서 대기하는 시간은 200ms이다.)

②의 경우, 수신된 패킷의 정보를 자신이 저장했던 DB의 정보와 비교하여 동일한 부분은 삭제하고 나머지 정보를 패킷에 담아 Super-Cluster Head에게 전송한다. ③의 경우, 다수의 패킷에서 라우팅 테이블 정보만을 추출 및 결합하여 패킷의 개수를 감소시킨 후 Super-Cluster Head에게 전송한다.

위의 3가지 경우를 처리한 후 Sub-Cluster Head는 해당 패킷의 송신자에게 reply 패킷을 전송하여 해당 송신자에게 성공적으로 처리되었음을 알려준다.

Fig. 6는 Sub-Cluster Head를 통해 Super-Cluster Head

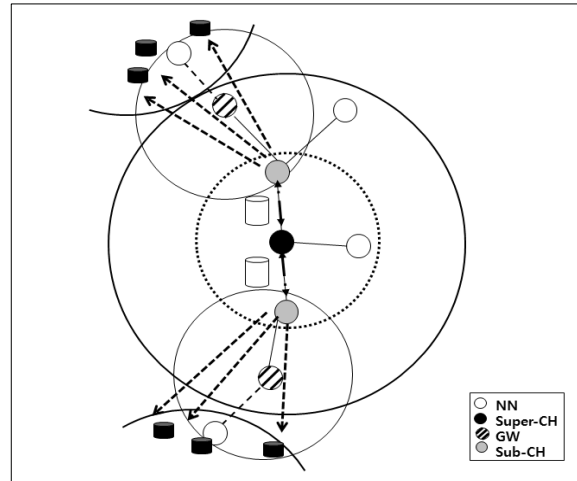


Fig. 6. Super-Cluster Head Traffic Management using Sub-Cluster Head

의 트래픽을 관리하는 과정을 표현하고 있다. Sub-Cluster Head는 클러스터 외부로부터 전달되어지는 패킷들을 수신한다. 해당 패킷의 정보가 외부 클러스터의 라우팅 테이블 정보를 담고 있고 새로운 정보라면 해당 패킷을 자신의 DB에 저장한다. 그렇지 않다면 패킷 헤더의 크기나, 정보, 도착 시간 등을 고려하여 패킷의 수정, 결합을 통해 통합 패킷을 생성한다. 이후 해당 패킷을 Super-Cluster Head에게 전달한다. Super-Cluster Head로부터 reply 메시지를 수신 받으면 외부의 클러스터 헤드들에게 해당 reply 메시지를 전달한다. 이렇게 외부 클러스터 헤드와 Super-Cluster Head 사이에서 중계자 역할을 함으로써 Super-Cluster Head에 발생하는 트래픽을 관리한다.

3.3 2-level Cluster Maintenance

제안 기법이 적용되는 환경은 무선 애드 혹 네트워크이기 때문에 노드의 이동과 에너지 고갈에 대한 고려가 필요하다. 특히, 제안 기법의 중요한 역할을 담당하는 Super-cluster Head에 대한 고려가 중요하다. 이러한 이동이나 에너지 고갈로 인해 발생하는 사례는 크게 2가지로 나뉠 수 있다. 또한, 기본 클러스터 헤드 역시 마찬가지로 재선정이 필요한 경우, 아래 수식을 통해 얻어진 가중치 값을 통해 새로운 클러스터 헤드를 선출한다.

- 1) Super-cluster Head가 본래의 역할 수행을 못하는 경우 이와 같은 경우를 대비하기 위해 Super-Cluster Head는 해당 클러스터에 속해 있는 1-level 노드들을 아래의 수식 (1), (2)를 통해 자체적으로 등급을 설정한다.

$$Node_n = \frac{1}{dist_n} * \alpha + r p_n * \beta + r t_n * \gamma \quad (\text{단, } \alpha + \beta + \gamma = 1) \quad (1)$$

$$dist_n = \sum (s P_i / r P_i) / i \quad (2)$$

수식 (1)은 거리, 에너지, 시간 등을 고려하여 각 노드들의 가중치 값을 계산하는 수식이다. $dist_n$ 은 Super-Cluster Head

와 노드의 거리를 의미한다. Super-Cluster Head와의 거리가 가까울수록 기존의 topology를 변경없이 유지할 수 있기 때문에 $dist_n$ 값이 낮을수록 높은 가중치를 얻는다.[10] rp_n 은 해당 노드의 현재 잔여 에너지량을 의미한다. 에너지 고갈에 따른 Super-Cluster Head의 재선정이 발생할 경우 많은 오버헤드가 발생한다. 따라서 노드의 생존시간을 나타내는 잔여 에너지량이 높을수록 높은 가중치를 얻는다. rt_n 은 해당 클러스터에 속해 있는 시간을 의미한다. 클러스터에 속해 있는 시간이 길어질수록 클러스터의 정보를 많이 저장하고 있기 때문에 클러스터들의 관리를 담당하는 Super-Cluster Head의 역할을 수행하는 데 용이하다. 따라서 이 값이 높을수록 높은 가중치 값을 얻는다.

수식 (2)는 수식 (1)의 $dist_n$ 을 측정하는 수식이다. 각 노드들은 이동성을 가지기 때문에 특정 시간 동안 이동한 거리의 평균을 통해 Super-Cluster Head와의 거리를 측정한다. 거리를 측정하기 위해 수신 전력과 전송 전력의 차이를 이용하여 측정하는 방법을 사용한다. i 는 특정 시간 간격(interval)을 의미하며 sP 는 전송전력, rP 는 수신전력을 의미한다.

수식 (1)을 통해 얻어진 가중치가 가장 높은 노드는 Backup-Node로 상태를 변경하고 Super-Cluster Head가 사라질 경우 Hello message를 통해 자신이 Super-Cluster Head임을 주변 노드와 클러스터들에게 알린다. 그리고 Hello message를 수신한 Sub-Cluster Head들은 자신들이 저장한 DB를 Backup Node에게 전송한다.

2) Super-cluster Head의 DB를 가지는 sub-cluster head가 해당 cluster를 벗어나는 경우

Super-Cluster Head는 해당 Sub-Cluster Head가 연결되어 있었던 Gateway 노드를 검색한다. 이후 해당 GW노드와 1-hop 거리에 존재하는 1-level 노드를 탐색한다. 만약 해당 노드로 연결되는 1-level 노드가 존재하지 않는다면 해당 Gateway 노드를 통해 연결되는 경로를 DB에서 삭제한다. 반대로 해당 GW노드가 존재할 경우 Super-Cluster Head는 나머지 Sub-Cluster Head들에게 DB_req message를 전송한다. Sub-Cluster Head들은 자신들이 저장하고 있는 Super-Cluster Head의 DB들을 DB_reply packet에 담아 Super-Cluster Head에게 전송한다. DB_reply 패킷을 수신한 Super-Cluster Head는 Exclusive-OR 연산을 통해 생성한 DB 조각들을 새로 선정된 Sub-Cluster Head에게 전송한다.

아래 Fig. 7은 Sub-Cluster Head로부터 수신한 DB의 조각들을 기존의 RAID기법처럼 Exclusive-OR연산을 통해 비교하여 선정된 DB조각을 Sub-Cluster Head에게 전송할 새로운 Backup을 생성하는 과정을 보여주고 있다.

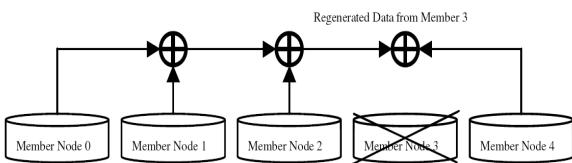


Fig. 7. DB Recovery by Exclusive-OR Operating

4. 성능 평가

4.1 시뮬레이션 환경

성능 평가를 수행하기 위해 NS-2 상에서 각 노드는 IEEE 802.11의 네트워크 인터페이스 카드의 실제 성능을 모델로 삼았으며 데이터 전송은 일정한 비트율로 데이터를 전송하는 CBR을 택하여 1초당 데이터 패킷을 4개씩 전송했다. 노드의 움직임은 이동할 목적지를 무작위로 선택하고 pause time을 주기로 일정속도로 그 목적지를 향해 움직이며, 그 목적지에 도달한 후 다시 pause time을 주기로 다른 목적지로 이동하도록 했다. 시뮬레이션은 총 10회 실시하였으며, 그 평균값을 기준으로 평가하였다. Table 1은 본 시뮬레이션에 사용된 환경과 파라미터 값들이다.

Table 1. Simulation Setting

Simulation Environment	
Simulation Area	250m * 250m
Simulation Time	100 sec
Number of Node	50
Node-Placement	Randomly
Mobility	Random Direction
Node-Speed	5,10 m/s
Node-Pause-Time	0,5,10,15,20 sec
MAC-Protocol	IEEE 802.11
Radio-Model	RADIO-ACCNOICSE
CBR packet size	512 bytes
Radio-Frequency	2.4 GHz
Initial Energy	5J
Transmit/Receive Power Drain	0.071 W + 10 * Transmission Power / 0.051 W
Idle Power Drain	0.027 W

4.2 시간의 흐름에 따른 노드의 생존 시간 비교

무선 애드 혹 네트워크에서 노드의 생존 시간을 측정하기 위해 일정 시간 동안 네트워크를 동작시킨 뒤 시간별로 작동하고 있는 노드의 수를 추출하였다. 기존의 LS2RP와 같은 RAID 시스템 기반의 클러스터링 알고리즘은 클러스터 헤드의 데이터 백업을 위해 이웃 노드들의 에너지 소모가 증가함으로써 네트워크의 생존시간을 단축시키는 단점을 가진다. 제안 기법은 이러한 기법의 단점을 개선하기 위해 1-level에 존재하는 Sub-Cluster Head데이터 백업 과정을 통해 에너지 분산을 우선적으로 고려하는 방법을 적용하였다. 또한 Super-Cluster Head에 집중되는 트래픽을 감소시키기 위해서 패킷의 크기와 정보에 따라 각기 다른 전송 방식을 적용함으로써 네트워크의 QoS를 증가시키고자 하였다. Fig. 8, 8-1에서 볼 수 있듯이 제안 기법은 LS2RP와 비교하였을 때, 이동 속도가 5m/s일 경우, 13%의 성능향상을 보였으며, 10m/s일 경우 8%의 성능향상을 보였다.

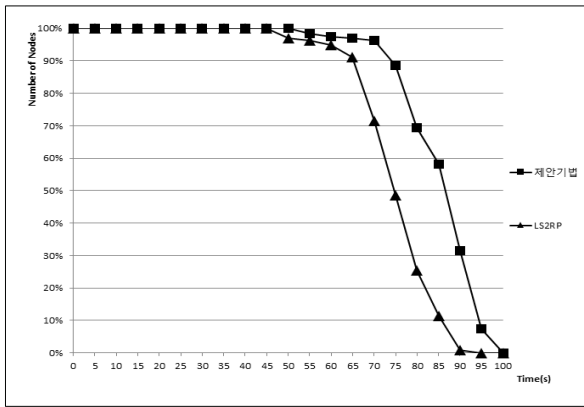


Fig. 8. Node Life Time according to Timeflow (v = 5 m/s)

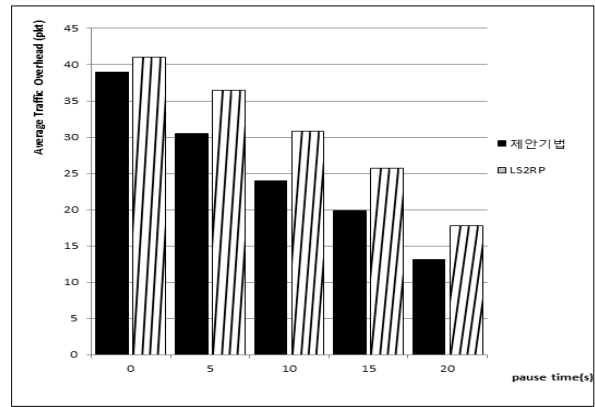


Fig. 10. Super-Cluster Head Traffic Overhead

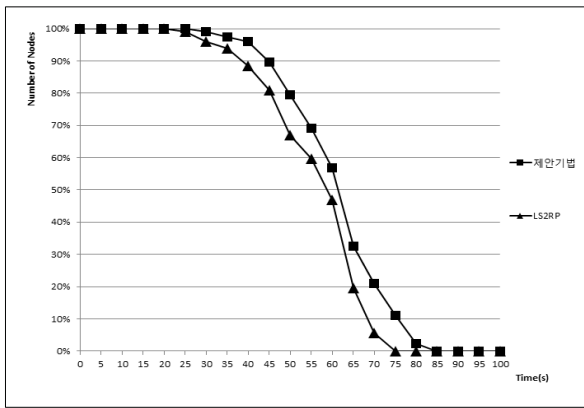


Fig. 8-1. Node Life Time according to Timeflow (v = 10 m/s)

4.3 전체 네트워크의 평균 패킷 전송률을 통한 성능 비교

전체 네트워크의 평균 패킷 전송률을 측정하기 위해 각 노드별로 송신 패킷과 해당 패킷에 대한 ACK 패킷의 비율을 측정하여 전송 성공률을 추출하였다. Fig. 9에서 나타나 있듯이 제안 기법은 Sub-Cluster Head를 이용한 클러스터 헤드의 DB 백업을 수행함으로써 클러스터 헤드가 해당 클러스터에서 벗어나거나 에너지가 고갈되어 작동을 못하는 상황에 대해 대처하였다. 그 결과 제안 기법이 LS2RP보다 평균 패킷 전송률에서 2.3% 성능 향상을 보였다.

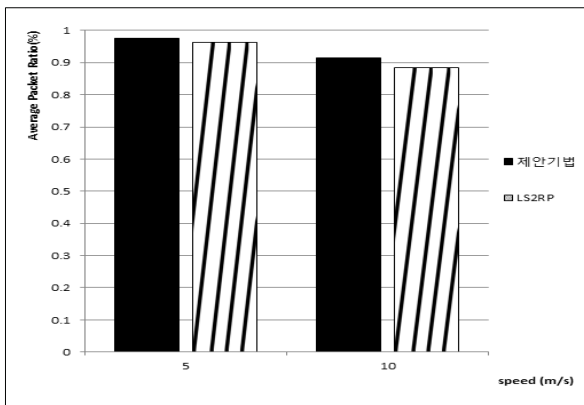


Fig. 9. Average packet transmission rate of Entire Network

4.4 이동주기에 따른 Super-Cluster Head의 Traffic 비교

Fig. 10은 이동주기에 따른 Super-Cluster Head의 트래픽에 관한 실험이다. 즉, 제안 기법의 효율적인 트래픽 관리 능력을 보여주는 실험이라고 볼 수 있다. 노드가 항상 움직이는 경우 LS2RP에 비해 5%의 성능 향상을 보였으며, 노드가 20초 주기로 이동하는 경우 27%의 성능 향상을 보였다. 또한, 노드가 항상 이동할 경우와 주기적으로 이동할 경우 평균 17% 정도 트래픽 감소를 시켜줌으로써 다양한 상황에 대한 QoS가 향상됨을 알 수 있었다.

5. 결론 및 향후 과제

애드 혹 네트워크에서는 특히 노드의 에너지에 대한 제약이 크기 때문에 클러스터의 효율적인 관리를 통해 네트워크의 안정성을 향상시키는 알고리즘이 필요하다.

기존의 LS2RP와 같은 RAID 시스템 기반의 클러스터링 기법은 클러스터 헤드의 부재에 대한 대처 효율을 증가시킬 수 있지만 클러스터 헤드에 많은 양의 패킷이 집중됨으로써 트래픽이 증가하여 Bottle-Neck 현상과 같은 문제가 발생하는 단점을 가진다.

이를 해결하기 위해 본 논문에서는 무선 애드 혹 네트워크 환경에서 에너지 효율적인 2-Level 트리 기반의 클러스터 라우팅 프로토콜을 제시하였다. 클러스터 헤드는 Sub-Cluster Head를 통해 데이터 백업을 수행하고, Sub-Cluster Head는 패킷의 수정, 결합과 Super-Cluster Head의 업무를 분담하여 전체적으로 발생하는 트래픽의 양을 감소시킨다.

NS-2를 이용하여 제안 기법과 LS2RP의 성능 비교를 수행한 결과 네트워크 유지시간과 데이터 전송 성공률, 트래픽에 대해 모두 좋은 성능을 기록한다는 것을 확인하였다. 네트워크 유지시간에 대해서 제안 기법이 LS2RP보다 평균적으로 10%의 성능 향상이 나타났으며, Super-Cluster Head의 트래픽에 대해서는 제안 기법이 LS2RP보다 평균 17%의 성능 향상을 보임을 알 수 있었다.

이처럼 본 논문에서 제안한 기법은 Sub-Cluster Head를

활용하여 균등한 에너지 소비와 QoS를 동시에 만족시킬 수 있었다. 하지만 패킷의 긴급도에 따른 전송 방식의 고려를 하지 않아 긴급한 상황에 대처하기 힘든 단점을 보였다. 향후에는 다양한 상황에 대한 전송 방식을 제공하면서도 각 전송 방법에서 발생하는 손실을 최소화 할 수 있는 연구를 진행하고자 한다.

Reference

[1] JANE Y. YU and PETER H. J. CHONG, "SURVEY OF CLUSTERING SCHEMES FOR MOBILE AD HOC NETWORKS," in IEEE COMMUNICATIONS SURVEYS, Vol.7, No.1, pp.32-48, 2005.

[2] Abolfazle A, Ali K. and Naser L, "Clustering Algorithms in Mobile Ad Hoc Networks," in Proc. of Fourth International Conference on Computer Sciences and Convergence Information Technology, pp.1509-1513, 2009.

[3] Nabil TABBANE, "LS2RP: Link-State RAID Routing Protocol," in Proc. of Eight International Conference on Networks, pp.99-105, 2009

[4] Charles E. Perkins, AD HOC NETWORKING, Addison-Wesley, 2001.

[5] M. Jiang, J. Li and Y. C. Tay, "Cluster Based Routing Protocol(CBRP) Function Specification," in Internet Draft draft-ietf-manet-cbrp-spec, August, 1998.

[6] E. Gafni and D. Bertsekas, "Distributed Algorithms for Generating Loop-Free Routes in Networks with frequently changing topology," in IEEE Trans. Communications, January, 1981.

[7] Mainak C. and Sajal K. DAS, "WCA: A Weighted Clustering Algorithm for mobile Ad Hoc Networks," Cluster Computing, Vol.5, No.2, pp.193-204, Apr., 2002.

[8] M. Chatterjee and et al., "An On-Demand Weighted Clustering Algorithm for Ad hoc networks," in proc. of IEEE Globecom, pp.1697-1701, Dec., 2000.

[9] Wonchang C. and Miae W., "A Distributed weighted Clustering Algorithm for mobile Ad Hoc Networks," AICT, In proc. of IEEE Computer Society, Feb., 2006.

[10] Z. Junna, Y. Peiyan and L. Ping, "Research and Simulations of Cluster Routing Protocols in Ad Hoc Networks," in Proc. of World Congress on Software Engineering, pp.509-512, 2009.

[11] Huda Al Amri, Mehran A. and Tadeusz W., "Scalability of MANET routing protocols for heterogeneous and homogenous networks," in Trans. on Computers and Electrical Engineering, pp.1-14, 2009.

[12] D. J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm", IEEE Transactions on Communications COM-29(11), pp.1694-1701, 1981.

[13] PK Jaggi, and AK SinghSingh, "Staggered Checkpointing and Recovery in Cluster Based Mobile Ad Hoc Networks-Advances in Parallel Distributed Computing, Vol.203, pp.122-134, 2011.

[14] Changlin Ma, Nian Liu, and Yuan Ruan, "An Energy-efficient Clustering Algorithm in Mobile Sensor Networks", in Proc. of Signal Processing, Communication and Computing (ICSPCC), pp.1-4, Aug., 2013.



이 영 준

e-mail : keii0602@gmail.com
 2008년 서강대학교 컴퓨터학과(학사)
 2010년 서강대학교 컴퓨터공학과
 공학석사
 관심분야: 무선통신망, 센서 네트워크 등



김 성 천

e-mail : ksc@sogang.ac.kr
 1975년 서울대학교 공업교육학(전기전공 학사)
 1979년 Wayne State Univ. 컴퓨터공학 (공학석사)
 1982년 Wayne State Univ. 컴퓨터공학 (공학박사)

1982년~1984년 캘리포니아주립대학교 조교수
 1984년~1985년 금성반도체(주) 책임연구원
 1985년~현 재 서강대학교 컴퓨터학과 교수
 관심분야: 병렬처리시스템(Parallel Computer Architecture, Interconnection Network), WDM technology를 이용한 cluster system, 유비쿼터스 컴퓨팅, Pervasive Computing