

CBRP에서 보조헤더를 이용한 라우팅 고장 극복 개선 (An Improvement of Fault Tolerant for Routing using Secondary Header in CBRP)

허태성* 이균하**
(Tai-Sung Har) (Gyun-ha Lee)

요 약

MANET은 기존의 이동 통신망과는 달리 백본망이 없고, 동적으로 분산된 이동 호스트들로 구성되어 전이나 천재지변과 같은 긴급사태 및 기존의 이동 통신 기반 구조가 존재하지 않는 지역에서 사용되는 통신 구조이다. 따라서 MANET에 기존의 라우팅 알고리즘을 적용할 수 없으며, 새로운 라우팅 알고리즘의 연구가 활발히 이루어지고 있다. 본 논문은 기존에 제안된 라우팅 알고리즘 중 클러스터에 헤더(주헤더)를 두어 관리하는 클러스터 기반의 한 라우팅 프로토콜(CBRP)에 주헤더 뿐만 아니라 보조헤더를 추가한다. 보조헤더의 역할은 주헤더가 비정상적으로 되어 노드들 간의 송수신에 더 이상 참가할 수 없을 경우, 별도의 주헤더 재선출 과정 없이 보조헤더가 즉시 주헤더로 교체되어 기존의 주헤더 역할을 담당한다. 따라서 본 논문은 기존의 CBRP에 비해 효율적인 라우팅 알고리즘 CBRP-SH를 제안한다. 또한 본 논문의 우수성을 입증하기 위해 기존의 CBRP와 비교 평가한다.

ABSTRACT

Mobile Ad Hoc Network (MANET) is a network architecture which has no backbone network and is deployed temporarily and rapidly in emergency or war without fixed mobile infrastructures. All communications between network entities are carried in ad-hoc networks over the wireless medium. Due to the radio communications being extremely vulnerable to propagation impairments, connectivity between network nodes is not guaranteed. Therefore, previously developed routing algorithms in wired networks cannot be used. And many new algorithms have been undergone.

This study proposes the Secondary Header approach to the cluster based routing protocol (CBRP). In case, the main header becomes abnormal status so that the main header can not participate the communications between network entities, the secondary header immediately replaces the primary header without selecting process of the new primary header.

The performances of proposed algorithm CBRP-SH(Cluster Based Routing Protocol using Secondary Header) are compared with CBRP and results are presented in order to show the effectiveness of the algorithm.

* 종신회원 : 인하공업전문대학 부교수

** 정회원 : 인하대학교 교수

논문접수 : 2001. 7. 18.

심사완료 : 2001. 7. 24.

1. 서론

MANET(Mobile Ad Hoc Network)은 기존의 이동 통신 서비스와는 다르게 기지국과 같은 고정장치나 중앙 집중식 제어 없이 특수한 상황 즉 전시나 천재 지변으로 인한 긴급상태, 또는 기존의 이동 통신 기반 구조가 존재하지 않는 지역에서 노드간에 일시적으로 통신망이 형성되는 방식이다. 이 때 노드들 사이에는 무선으로 패킷을 송수신 한다. 그러나 노드들의 무선 전송은 제한된 영역을 가지므로 송신 노드의 전파 영역 밖에 있는 노드에 패킷을 전송하려면 송신 노드와 수신 노드사이에서 패킷을 전달하는 또 다른 노드가 존재하여야 한다[1,2].

MANET은 기존의 유선 망과 비교하여 대역폭이 좁고 신호간의 간섭이 상대적으로 심하여 망 형상(topology)이 수시로 변경될 수 있다. 또한 MANET에서 사용되는 노드는 유선 망에서는 고려되지 않는 소모 전력량이 제한적이므로 빈번한 제어 및 상태 정보 전달이 어렵다[3,4]. 따라서 기존 유선 망에서 사용되는 경로 설정 프로토콜은 MANET에서는 적용될 수 없으며 이에 대한 많은 라우팅 알고리즘이 연구되고 있다[5]. 대표적인 연구로는 거리 벡터(Distance Vector) 알고리즘의 변형, 동적 소스 라우팅(Dynamic Source Routing), 클러스터 기반 알고리즘(CBRP:Cluster Based Routing Protocol) 등이 있다[2,6].

이 중 특히 클러스터 기반의 알고리즘은 헤더를 이용하여 클러스터의 효과적인 관리를 제안하고 있다[7]. 클러스터 기반의 라우팅 알고리즘에서는 경로 형성 및 라우팅에 클러스터 헤더가 전적으로 제어한다. 따라서 효율적인 헤더 선출 및 관리가 중요한 문제로 대두된다. 헤더 선출 알고리즘은 헤더 선출이 시작되는 시간을 보장하고 클러스터 헤더 노드에 문제가 발생하여 새로운 헤더가 선출되었을 때 다른 노드, 즉 헤더가 아닌 멤버 노드들이 이에 동의하도록 설계되어야 한다. [8]. 또한 헤더에 고장이 발생하였을 경우 클러스터들의 모든 노드들은 새로운 헤더가 선출될 때까지 수행을 멈추게 되어 전체 네트워크의 성능을 저해하는 요소가 된다. 특히 MANET에서는 유선 망과는 달리 각 노드들의 이동성으로 인해 헤더 재선출이 빈번하게 발생하며, 이로 인한 부하가 헤더를 이용한 라우팅 알고리즘 성능의 중요한 부분이 된다[9].

따라서 본 연구에서는 헤더가 존재하는 클러스터 기반의 라우팅 알고리즘 중 헤더가 클러스터 밖으로 이동을 하거나 비정상적인 상태가 되어 더 이상 헤더의 역할을 수행할 수 없어 라우팅을 계속할 수 없는 경우, 이에 대한 빠른 대처를 위해 헤더를 선출할 때 주헤더와 보조헤더를 같이 선출한다. 보조헤더는 주헤더의 고장으로 더 이상 주헤더가 라우팅에 관여할 수 없을 경우, 즉시 주헤더의 역할을 수행함으로써 주헤더의 고장으로 발생하는 라우팅의 문제점을 최소화한다. 제안된 알고리즘은 CBRP의 단점을 최대한 보완한 알고리즘이며, 성능 평가를 통해 본 연구의 타당성을 확인하였다.

2. 관련 연구

MANET에서 동작하는 라우팅 프로토콜은 기존의 유선 망에서 사용하는 라우팅 프로토콜과는 달리 다음과 같은 요구사항이 추가된다[10,11].

- 가. 분산된 동작 체계를 갖추어야 한다.
- 나. 루프가 없어야 한다.
- 다. 망에 트래픽이 균일하게 분포된다는 가정보다는 요구 또는 필요에 따른 트래픽 형태에 따라 적용할 수 있는 라우팅 알고리즘이 필요하다.
- 라. 기존의 라우팅 알고리즘들은 양방향 링크로 설계되었다. 그러나 MANET서는 단방향 링크가 발생할 수 있으므로 이러한 단방향 링크를 수용할 수 있는 라우팅 알고리즘이 필요하다.
- 마. 노드는 배터리를 사용하기 때문에 전력 보호 또는 비정상적인 동작 등의 이유로 임의의 시간 동안 송수신 기능을 멈출 수 있다. MANET의 라우팅 알고리즘은 이러한 기간을 수용할 수 있어야 한다[12].

MANET에서 경로설정에 관한 많은 알고리즘이 제안되고 있다[1,6]. 이러한 알고리즘들은 각 노드가 라우팅 테이블을 생성하고 유지하는 방법상에서 차이를 보이고 있다. 가장 대표적인 라우팅 알고리즘으로 분산 벨만-포드 알고리즘을 사용해 순환 루프를 제거한 거리 벡터(Distance Vector)의 변형, 동적 소스 라우팅

알고리즘(DSR), 클러스터 기반 알고리즘(CBRP) 등이 있다.

2.1 AODV(Ad Hoc Demand Distance Vector) 라우팅

AODV[13,14]는 거리 벡터 알고리즘의 변형으로 MANET를 유지하는 노드사이의 Dynamic, Self-starting 및 다중 홉 라우팅이 가능한 알고리즘으로써 지속적으로 라우팅 정보를 유지하지 않고, 요구가 있을 경우에만 경로설정 절차를 수행하는 Reactive 라우팅 알고리즘으로 분류된다. 이러한 Reactive 프로토콜은 경로설정 시 지연 시간이 길고, 전체적인 검색을 수행할 때 심각한 제어 트래픽을 유발하는 단점을 가지고 있다. AODV는 이런 단점을 보완하기 위해 망의 위상 변화를 즉각 인지하여 이웃 노드에 알려주는 링크 파손 메커니즘을 사용하며, 각 노드는 캐쉬에 한번 수행된 경로설정에 관한 정보를 저장한다. 이 캐쉬에 저장된 경로 정보는 노드에 의해 생성된 목적지 순차 번호(Destination Sequence Number)로 어느 경로 정보가 최신 정보인지 구별될 수 있으며, 이를 이용하여 루프가 없는 경로를 보장할 수 있다.

2.2 DSR(Dynamic Source Routing)

DSR[15,16] 알고리즘에서는 링크 계층에서의 주기적인 정보 교환 없이 특정 노드가 데이터 전송을 요구할 때 목적지까지의 경로 질의를 통해 데이터 전송을 수행하게 된다. 이 방법의 단점은 경로 질의 패킷이 목적지까지 도착하기 위해 거쳐갈 모든 중간 노드들의 정보를 포함해야 하기 때문에 경로가 긴 경우에는 확장성의 문제가 있다. 그러나 각 노드에서 경로 질의 패킷(RREQ)에 기록된 중간 노드 리스트를 검사하여 순환 경로 유무를 판단할 수 있어 루프 문제를 쉽게 해결할 수 있다.

2.3 CBRP(Cluster Based Routing Protocol)

CBRP[6,7,9]는 망 전체를 노드들의 분포에 따라 중복되거나 분리된 여러 개의 클러스터로 나누어 관리한다. 그리고 하나의 클러스터에 클러스터 헤더가

존재하여 클러스터에는 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 CBRP는 클러스터를 형성하고, 유지하는 비용이 들며, 모든 경로가 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 단점이 있다. 특히 MANET에서는 유선 망과는 달리 각 노드들의 이동성으로 인한 헤더 재선출이 빈번히 발생하며, 이로 인한 부하가 헤더를 이용한 라우팅 알고리즘 성능의 중요한 부분이 된다.

따라서 본 연구에서는 이러한 CBRP의 단점을 보완하기 위해 기존의 클러스터 내에 하나만 존재하는 헤더를 두개의 헤더(주헤더, 보조헤더)로 나누고, 보조헤더를 이용하여 주헤더에 의한 클러스터 기반의 라우팅 알고리즘의 문제점을 보완한다.

3. 보조헤더를 이용한 클러스터 기반의 라우팅 알고리즘(CBRP-SH)

CBRP-SH는 MANET 환경에 맞도록 설계된 라우팅 프로토콜로 네트워크를 구성하는 노드들을 분포에 따라 여러 개의 중복되거나 분리된 클러스터로 나누어 관리한다. 그리고 하나의 클러스터에 클러스터 주헤더(Hp: Primary Head)와 보조헤더(Hs: Secondary Head)가 결정되어 클러스터에 속한 노드들의 정보를 관리한다. 클러스터 사이의 라우팅은 각 클러스터의 주헤더가 가지고 있는 정보들을 이용하여 동적으로 이루어진다. 또한 노드들을 그룹으로 관리함으로써 경로설정 절차시에 트래픽의 양을 효과적으로 줄이며 속도도 향상시킨다.

3.1 구성 및 동작

3.1.1 물리층/데이터링크층

MANET내의 모든 노드들은 송수신기 기능이 있어, 항상 양방향 통신이 가능한 것으로 가정한다.

3.1.2 링크/연결 상태 감지 절차

각 노드는 헬로우 메시지를 이용하여 이웃과 연결된 링크 상태를 알 수 있다. 각 노드는 <표1>과 같이 이웃 테이블을 관리한다.

<표 1> 이웃 테이블의 구조

<Table 1> Structure of neighbor table

이웃 노드의 ID	역할
이웃 1	Hp/Hs/MB/미결정
이웃 2	Hp/Hs/MB/미결정
.....
이웃 N	Hp/Hs/MB/미결정

각 노드는 자신의 이웃 테이블을 <표 2>의 헬로우 메시지에 포함하여 HELLO_INTERVAL 간격으로 방송한다.

<표 1>에서 첫 번째 항목은 헬로우 메시지를 방송하는 노드의 ID이며, 두 번째 항목은 전송 노드가 클러스터의 주헤더인지 보조헤더인지, 멤버인지를 나타낸다. 미결정이란 노드가 어느 클러스터에도 속하지 않고 있다는 것을 의미한다. 이웃 노드 B로부터 <표 2>와 같은 헬로우 메시지를 수신한 노드 A는 자신의 이웃 테이블을 다음과 같이 갱신한다.

- (1) 노드 B가 자신의 이웃 테이블에 속해 있는지 확인한다. 존재하지 않다면 항목을 추가한다.
- (2) 노드 B가 클러스터의 주(또는 보조)헤더이면 A의 테이블에 있는 노드 B를 클러스터의 주헤더(또는 보조헤더)로 표기한다.

이웃 테이블의 모든 항목은 타이머에 의해 관리된다. 이웃 테이블의 리스트에 있는 노드로부터 2번의 HELLO_INTERVAL의 시간동안 새로운 헬로우 메시

지를 수신하지 못하면 자신이 관리하는 테이블로부터 해당 항목을 삭제한다.

<표 2> 헬로우 메시지 구조

<Table 2> Structure of Hello Message

노드의 ID	MY_MEMBERSHIP_STATUS (Hp/Hs/MB/미결정)
이웃 테이블	

이웃 노드들의 위상이 변하지 않을 때 각 노드는 링크를 가지는 모든 노드들의 정보를 가지게 된다. 이웃 테이블은 링크를 인식하는데 쓰여지는 것이 아니라, 클러스터 정보와 프로토콜의 다른 기능을 위해서 전달된다.

3.2 프로토콜 동작

CBRP-SH는 완전히 분산되어 동작되며, 기능면에서 세 개의 함수로 이루어져 있다.

3.2.1 클러스터 형성

클러스터 형성 알고리즘은 가장 작은 값의 ID를 갖는 노드를 클러스터 헤더로 선출하는 간단한 최저 ID 클러스터링 알고리즘이다. 노드는 헬로우 메시지로부터 클러스터 형성을 위한 정보를 얻는다. 자신과 한 홉으로 연결된 모든 노드들 중에서 가장 작은 값의 ID를 가지는 노드가 클러스터 주헤더가 되며, 그 다음 작은 값의 ID를 가지는 노드가 클러스터 보조헤더가 된다. 새로운 클러스터 주헤더는 자신의 헬로우 메시지 중 첫 번째 항목을 'Hp'로 기입하고, 보조헤더 노드가 누구인지를 기입한 후 방송한다. 클러스터 주헤더는 보조헤더를 제외하고, 자신과 링크를 가진 모든 노드들을 자신의 멤버 노드로 여긴다. 역으로 클러스터 주헤더와 링크를 가진 모든 멤버 노드들은 그 클러스터의 멤버로 속하게 된다. 멤버 노드는 여러 개의 클러스터 주헤더로부터 헬로우 메시지를 수신할 수 있으며, 여러 개의 호스트 클러스터를 가질 수 있다.

3.2.2 클러스터 관리

클러스터는 클러스터 주헤더에 의해 식별되므로 가능한 한 변화될 확률이 적은 노드를 주헤더로 가져야 한다. 클러스터 주헤더를 변경할 때 아래와 같은 규칙을 따른다.

(규칙1) 클러스터 주헤더(또는 보조헤더)가 아닌 노드는 현재의 클러스터 헤더들의 상태를 변경할 수 없다. 즉, 노드 X가 클러스터 주헤더(또는 보조헤더)인 노드 Y와 링크를 가지는 클러스터 주헤더(또는 보조헤더)가 아닌 노드라고 할 때 ID가 Y보다 낮은 값일 때도 클러스터 주헤더(또는 보조헤더)가 되지 못한다.

(규칙2) 두개 클러스터가 이동하여 주헤더들 간에 한 쌍의 링크가 설정될 때만 두 주헤더 중의 하나가 클러스터 주헤더의 역할을 잃게 된다.

클러스터 관리 절차는 다시 세 부분으로 나뉘어진다.

가. 노드 제거: 노드 X는 클러스터 주헤더의 링크를 잃게 되거나 또는 노드 자체에 문제가 있을 때 자신이 속한 클러스터로부터 제거된다. 두 경우 모두 클러스터 주헤더와 노드 X의 이웃 테이블은 타이머에 의해 상대방의 정보를 갱신하게 된다. 이때 노드 X가 클러스터의 보조헤더라면 클러스터 주헤더는 멤버 노드 중에서 가장 작은 값의 ID를 가지는 노드를 클러스터의 보조헤더로 선출하여 각 멤버 노드들에게 헬로우 메시지를 발송한다.

나. 노드 추가: 이미 존재하는 클러스터 주헤더가 새로운 노드보다 높은 ID를 가지고 있어도, 새로 들어온 노드가 헬로우 메시지를 통해 클러스터 주헤더의 링크가 설정되었을 때 해당 클러스터의 멤버 노드가 된다. 이 노드는 새로운 호스트 클러스터를 알게 되며 새로운 호스트 클러스터 주헤더는 새로운 멤버 노드에 대한 이웃 테이블을 갱신하게 된다. 노드가 초기화될 때 헬로우 메시지의 MY_MEMBERSHIP_STATUS는 일정 주기 동안 '미결정'으로 설정되며, 어떤 클러스터 주헤더와 링크를 가진다는 것을 발견할 때,

이 항목을 '멤버'로 바꾼다. 어떠한 클러스터 주헤더와도 링크를 가지지 않을 때 스스로 클러스터 주헤더가 되며 이를 헬로우 메시지로 발송한다.

다. 클러스터 헤더 충돌: 두 클러스터의 주헤더가 이동하여 서로 링크가 설정될 때 그 중 하나는 클러스터 주헤더의 역할을 포기해야 한다. 결과적으로 한 클러스터의 주헤더가 다른 클러스터의 주헤더로부터 링크가 설정되었다는 것을 가리키는 헬로우 메시지를 받으면 두 클러스터의 주헤더들은 ID를 비교한다. 작은 ID를 가지는 노드는 계속 클러스터 헤더로 남게 되고 높은 ID를 가지는 노드는 자신 및 자신의 보조헤더 노드의 MY_MEMBERSHIP_STATUS를 멤버로 바꾸어 헬로우 메시지를 발송하게 된다. 이로 인해 다른 새로운 클러스터가 형성하게 된다. 이때 클러스터 보조헤더는 자신이 속한 클러스터의 주헤더가 계속 클러스터 주헤더로 남을 경우 그대로 존재하며, 아니면 멤버 노드로 변경된다. 따라서 이 알고리즘은 엄격한 최저 ID 클러스터링 알고리즘은 아니다.

3.3 라우팅 고려 사항

CBRP-SH에서의 라우팅은 소스 라우팅을 기반으로 하며, 경로 발견, 라우팅, 잘못된 경로의 삭제 등 세 단계로 나눌 수 있다. 경로 발견 단계 동안에 패킷 트래픽의 증가를 최소화하기 위해 클러스터 구조를 도입하였다.

3.3.1 게이트웨이 발견

클러스터 X와 클러스터 Y에 각각 속한 한 개 또는 두 개의 노드 사이에 링크가 설정되었을 때 이 연결 노드를 게이트웨이라 하며, 게이트웨이 발견 절차에 의해, 클러스터 X의 주헤더는 주헤더와 링크로 연결된 이웃 클러스터에 대한 정보를 알게 된다. 이를 위해 <표 3>과 같은 인접 클러스터 테이블이 각 노드에 의해 관리된다.

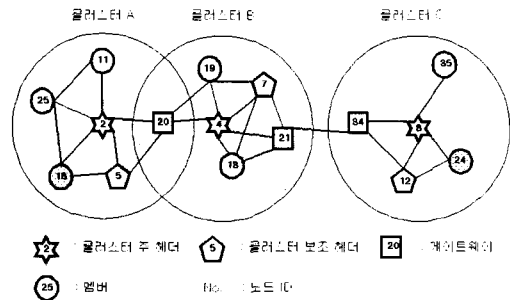
<표 3> 인접 클러스터의 테이블
 <Table 3> Cluster Adjacency Table

인접 클러스터1	인접 노드
인접 클러스터2	인접 노드
.....
인접 클러스터n	인접 노드

이 테이블은 노드로부터 주기적으로 수신하는 헬로우 메시지에 의해 갱신된다. 또한 보조헤더는 이 테이블의 정보 중 자신과 한 홉사이의 링크를 갖는 인접노드들의 정보만을 추출하여 저장 하게 된다. 이때 노드는 다른 하나의 클러스터 내의 여러 노드와 링크를 가질 수 있으며 그 중 하나만 인접 클러스터 테이블에 기록된다. 선택하는 방법은 같은 우선순위를 가진 링크 중에서 가장 낮은 ID를 가진 노드를 선택한다.

이 테이블은 멤버 노드의 호스트 클러스터 주헤더에게 주기적으로 보내진다. 클러스터 주헤더는 자신의 인접 클러스터 테이블을 구성하는데 이러한 멤버 노드들이 보낸 인접 클러스터 테이블을 이용한다. 규칙은 위와 동일하다.

클러스터 주헤더는 인접 클러스터 테이블에서의 연결관계(어느 노드로부터 어느 노드로 연결되었는지)를 알기 위해 홉수가 3인 메시지를 이웃 클러스터에 전송한다. 결과적으로 클러스터 주헤더는 이웃 클러스터의 링크 관계를 정확히 알게 된다. 예를 들어 [그림 1]에서, 클러스터 A는 클러스터 A가 노드 15를 통해 클러스터 B에 도달할 수 있다는 것을 알 수 있으며, 클러스터 B는 노드 21과 노드 14를 통해 클러스터 C에 도달할 수 있다는 것을 알 수 있다. CBRP-SH에서 클러스터 보조헤더는 클러스터 주헤더가 가지고 있는 정보를 저장하며, 만약 클러스터 주헤더에 문제가 발생할 경우에 저장된 정보를 이용하여 클러스터 주헤더의 역할을 대신하게 된다.



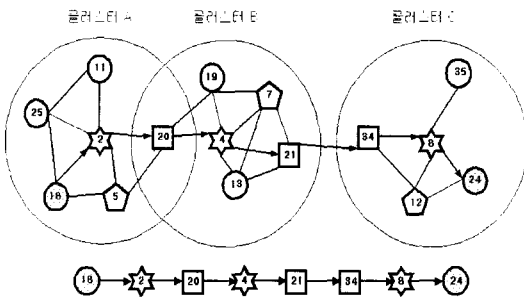
[그림 1] Ad-hoc 네트워크 모델
 [Fig. 1] Ad-hoc network model

3.3.2 경로 발견

경로 발견은 패킷을 보내고자 하는 노드 S에서 목적지 노드인 D로의 경로를 찾는 것이다. CBRP-SH는 다른 MANET 프로토콜들처럼 노드 S가 노드 D로의 경로를 찾는 방법에 플러딩 기법을 이용하지만, 다른 프로토콜에 비해 모든 노드를 대상으로 하지 않으므로 노드들의 부담이 적다. 노드 S는 노드 D로의 경로를 발견하기 위해 처음에 자신의 주소만 기록된 소스 경로 리스트가 담긴 경로 요청 패킷(RREQ)을 자신의 클러스터 주헤더에게 전송한다. 이 때 주헤더는 자신의 클러스터에 노드 D가 존재하는지를 먼저 찾아보고, 없을 경우 자신의 주소를 경로 요청 패킷에 추가하여 게이트웨이 노드를 통해 이웃 클러스터 주헤더에게 플러딩한다. 이러한 과정을 통해 노드 D를 찾으면 경로가 설정되어 경로 응답 패킷(RREP)이 소스 경로 리스트에 있는 역순으로 노드S에게 전달된다. 예를 들어 [그림 1]에서 노드 18이 노드 24와 연결하고자 할 때 게이트웨이 노드를 이용하여 자신과 연결되어 있는 클러스터로 전송하게 된다. 결과적으로 CBRP-SH에서 경로 요청 패킷은 목적지인 노드 24까지 포워딩되는 데에 [그림 2]와 같은 형태의 경로를 따를 것이다.

[그림 2]에서 보는 바와 같이 노드 18이 노드 24와 통신을 원할 경우, 소스 노드 18은 항상 클러스터 주헤더인 노드 2에게 경로 요청 패킷을 전송한다. 클러스터의 주헤더 노드 2는 자신의 클러스터에 노드 24가 있는지를 먼저 찾으며, 만약 없을 경우 자신의 주소를 경로 요청 패킷에 추가하여 게이트웨이 노드 20에 전달한다. 경로 요청 패킷을 받은 게이트웨이 노

드 20은 자신의 주소를 경로 요청 패킷에 추가한 후 클러스터 B의 주헤더인 노드 4에게 전달하다. 경로 요청 패킷을 수신한 각 클러스터 주헤더는 클러스터 A의 주헤더 노드 2와 같이, 자신의 클러스터에 목적 노드가 있는지 먼저 찾은 후 없으면 노드 2와 같이 소스 경로 리스트에 나타나지 않은 이웃 노드로 게이트웨이를 통해 라우트 요청 패킷을 포워딩한다. 이러한 절차는 목적지 노드까지 반복된다.



[그림 2] 경로 요청 패킷(RREQ) 흐름
[Fig. 2] A flow of RREP

경로 요청의 목적지인 노드 24가 경로 요청 패킷을 수신하면, 노드 24는 경로 요청 패킷에 기록된 소스 경로 리스트의 역 순서를 소스 노드 18로의 경로로 선택한다. 목적 노드 24는 발견된 경로를 경로 응답 패킷(RREP)에 복사하여 경로를 요청한 소스 노드 18에 전송한다. 이 소스 경로 리스트에는 소스 노드 18이 목적 노드 24로 패킷을 전달하기 위해 거쳐야 하는 노드들의 순서를 가지고 있다.

3.3.3 라우트 삭제

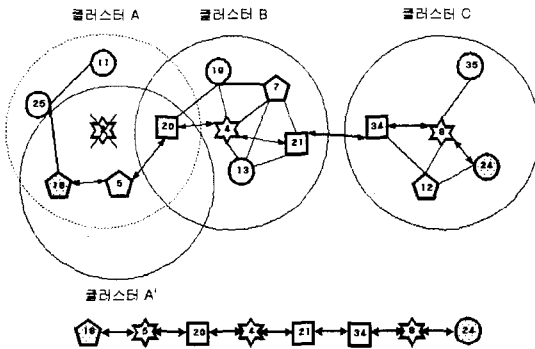
[그림 1]에서 소스 노드와 목적 노드 사이의 위상이 변하여 더 이상 이 경로를 사용할 수 없을 때 경로는 삭제된다. 그러나 소스 노드가 여전히 목적 노드와 통신을 하고 싶을 경우 새로운 경로를 찾기 위해 경로 발견 절차를 다시 시작한다. 이러한 경로 삭제의 경우는 다양한 이유가 있을 수 있다. 그러나 본 연구에서는 이러한 이유 중 클러스터의 주헤더에 문제가 발생한 경우만을 고려하기로 한다.

3.4 클러스터 주헤더의 고장 발생 및 경로 재설정

본 연구는 CBRP-SH에서 클러스터 주헤더가 클러스터 밖으로 이동을 하거나 비정상적인 상태가 되어 더 이상 주헤더로서의 역할을 못할 경우 이를 위한 보조헤더의 역할 및 재설정 문제를 연구하였다. [그림 2]와 같이 노드 18과 노드 24가 서로 통신중일 때, 클러스터 A, B, C의 주헤더인 노드 2, 4, 8에 문제가 발생할 경우 소스 노드 18과 목적 노드 24 사이의 경로는 문제가 발생한 클러스터의 다른 헤더가 선출될 때까지 동작을 멈추게 되며, 이에 따른 부하가 라우팅 알고리즘의 상당한 부담으로 작용할 것이다. 따라서 본 연구는 이러한 문제점을 다음과 같은 세 가지 경우로 나누어, 경로 재설정 방법을 제안한다.

3.4.1 소스 노드 18이 속한 클러스터 A에서 주헤더인 노드 2의 고장 및 경로 재설정

[그림 2]와 같이 소스 노드 18과 목적 노드 24사이의 경로가 형성되어 있을 때, 소스 노드가 속한 클러스터 A의 주헤더인 노드 2가 클러스터 밖으로 이동하거나 비정상적인 상태가 되어 더 이상 헤더의 역할을 수행할 수 없다면 [그림 3]과 같이 클러스터 A의 보조헤더인 노드 5가 즉시 주헤더의 역할을 대신하여 본래의 라우팅에 아무 영향을 주지 않으면서 라우팅을 복구할 것이다. 이 때 클러스터 A의 주헤더 노드 5는 클러스터 내의 다음 ID를 갖는 노드 18를 보조헤더로 선출하여 클러스터를 운영할 것이다.



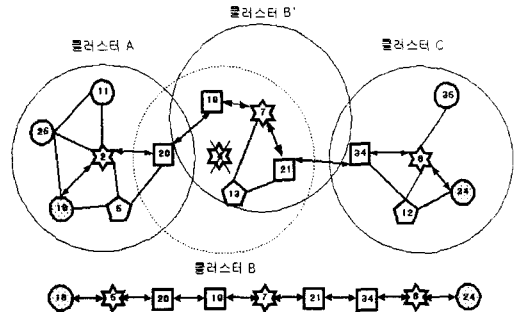
[그림 3] 클러스터 A에서 주헤더 노드 2에서 고장 및 경로 재설정

[Fig. 3] Fault tolerant and rerouting of primary-header node 2 in cluster A

또한 클러스터 A에서 클러스터 A'으로 변환 시 클러스터 외부에 존재하게 되는 노드 25, 11은 다른 클러스터의 멤버가 되거나, 자체의 클러스터를 형성한다.

3.4.2 경로의 중간을 담당하던 클러스터 B의 주헤더인 노드 4의 고장 및 회복

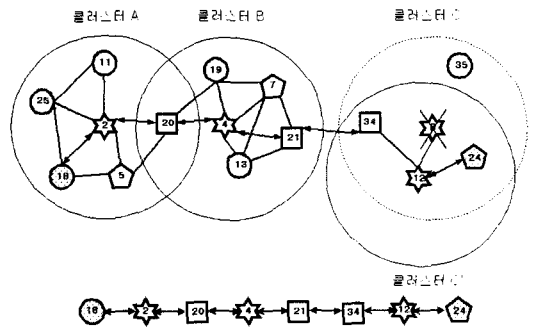
소스 노드 18과 목적 노드 24사이의 라우팅이 형성되기 위해서는 클러스터 B의 도움이 필요하다. 이때 클러스터 B의 주헤더인 노드 4가 클러스터 밖으로 이동을 하거나 비정상적인 상태가 되어 더 이상 헤더의 역할을 수행할 수 없어진다면 [그림 4]와 같이 클러스터 B의 보조헤더인 노드 7이 즉시 주헤더의 역할을 대신하여 클러스터 B'을 형성하게 된다. 그러면 [그림 4]에서와 같이 노드 19가 게이트웨이 노드가 되며, 본래의 라우팅을 복구하게 된다. 여기서 클러스터 B'의 주헤더 노드 7은 클러스터 내의 다음 ID를 갖는 노드 13을 보조헤더로 선출하여 클러스터를 운영할 것이다.



[그림 4] 클러스터 B에서 주헤더 노드 4의 고장 및 경로 재설정

[Fig. 4] Fault tolerant and rerouting of primary-header node 4 in cluster B

4.3 목적 노드 24가 속한 클러스터 C의 주헤더인 노드 8의 고장 및 경로 재설정



[그림 5] 클러스터 헤더 8의 고장 및 경로 재설정

[Fig. 5] Fault tolerant and rerouting of cluster header 8

소스 노드 18과 목적 노드 24사이의 라우팅이 형성되어 있을 때, 목적 노드가 속한 클러스터 C의 주헤더인 노드 8이 클러스터 밖으로 이동 하거나 비정상적인 상태가 되어 더 이상 헤더의 역할을 수행할 수 없어진다면 클러스터 C의 보조헤더인 노드 12가 즉시 주헤더의 역할을 대신한다. 노드 12는 본래의 라우팅에 아무 영향을 주지 않으면서 [그림 5]과 같이 라우팅을 복구할 것이다. 이 때 클러스터 C'의 주헤더 노드 12는 클러스터 내의 다음 ID를 갖는 노드

24를 보조헤더로 선출하여 클러스터를 운영할 것이다. 또한 클러스터 C에서 클러스터 C'으로 변환시 클러스터 외부에 존재하게 되는 노드 35는 다른 클러스터의 멤버가 되거나, 자체의 클러스터를 형성할 것이다.

4. 성능 분석 및 실험 고찰

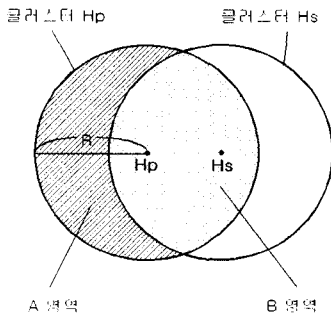
4.1 성능 평가

본 논문에서 제안한 CBRP-SH의 성능을 분석하기 위해서 CBRP에서 각 클러스터를 집합으로 표현하여 확률을 계산하였다[17]. 본 성능 분석의 최종 목표는 제안한 방법을 적용했을 때 각 클러스터의 주헤더에 고장이 발생할 경우 보조헤더가 즉시 주헤더의 역할을 대신함으로써 얻어지는 클러스터 관리상의 주헤더 재선출을 위한 비용 및 경로 재설정 비용 절감을 도출하는 것이다.

이를 위해서 클러스터 주헤더를 중심으로 반경 R인 클러스터를 정의 하였으며, 클러스터 내에 n개의 노드들이 균등 분포 되었다고 가정한다.

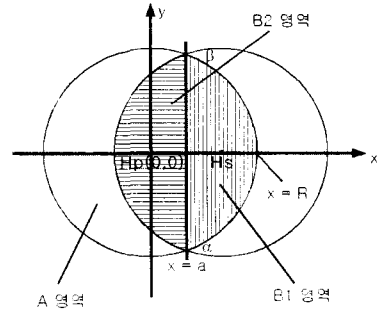
[그림 6]은 한 클러스터내에서 주헤더가 고장이 발생하여 보조헤더가 주헤더의 역할을 대신 할 때 클러스터의 변화를 표현하였다.

[그림 6]에서 클러스터의 영역은 주헤더로부터 R을 반지름으로 하는 원의 영역으로 하였으며, 클러스터의 주헤더(Hp)에 고장이 발생하여 보조헤더(Hs)에게 주헤더의 역할을 이전하게 되면 클러스터의 영역은 클러스터 Hp에서 클러스터 Hs로 변화된다.



[그림 6] 클러스터 주헤더와 보조헤더의 클러스터 변화
[Fig. 6] A change of the cluster Hp and cluster Hs

이때 B 영역에 있는 노드들은 변화가 없으나, A 영역에 있는 노드들은 다시 헤더를 선출하여야 한다. 따라서 A 영역과 B 영역의 노드 수를 계산하기 위해 A와 B 영역의 넓이를 계산하여야 한다. 이를 위해 Hp를 원점으로 좌표 축을 도입하여 [그림 7]으로 표현하였다.



[그림 7] 좌표 축을 이용한 주헤더 클러스터와 보조헤더 클러스터의 변화

[Fig. 7] A change of the cluster Hp and cluster Hs using coordinate axis

[그림 7]에서와 같이 클러스터 Hp와 클러스터 Hs가 겹치는 B영역의 면적을 계산하기 위해 B영역을 B1과 B2로 구분한다. 이때 B1영역과 B2영역의 넓이는 같다. 또한 클러스터 Hp와 클러스터 Hs가 겹치는 점, 를 구하고, 와 를 연결하는 선(x=a)을 설정하여 B1과 B2으로 이등분한다.

Hp를 중심으로 반경 R인 원에서 영역 B의 면적을 제외한 것이 영역 A의 면적이다. [그림 7]에서 B1 영역의 면적을 계산할, 를 구하기 위해, 원의 공식

$$x^2 + y^2 = R^2 \text{에 } x = a \text{를 대입하여}$$

$$y^2 = R^2 - a^2 \text{를 유도하고 해를 구하면}$$

$$y = \pm\sqrt{R^2 - a^2} \text{가 된다.}$$

따라서, Hp를 중심으로 하는 원과 $x = a$ 를 만족하는, 는 다음과 같다.

$$\alpha = -\sqrt{R^2 - a^2}, \beta = \sqrt{R^2 - a^2}$$

여기서 B영역의 넓이를 S(B)라 하면
 $S(B) = S(B1) + S(B2) = 2S(B1)$ 이며,

$$S(B1) = \int_a^R (x-a)dy$$

$$= \int_{\sqrt{R^2-a^2}}^R \sqrt{R^2-y^2} dy - \int_{\sqrt{R^2-a^2}}^R a dy$$

여기서 $t=R\sin y$ 로 치환하면

$$= \int_{\sqrt{R^2-a^2}}^R R\cos y \cdot R\cos y dy$$

$$- \int_{\sqrt{R^2-a^2}}^R a dy$$

$$= 2R^2 \int_0^{\sqrt{R^2-a^2}} \cos^2 y dy - 2a\sqrt{R^2-a^2}$$

$$= R^3 \sin \sqrt{R^2-a^2}$$

$$+ \frac{R^2}{2} \sin(2R \sin \sqrt{R^2-a^2}) - 2a\sqrt{R^2-a^2}$$

이 때 $0 < a \leq \frac{R}{2}$ 이다.
 따라서

$$S(B) = 2S(B1)$$

$$= 2R^3 \sin \sqrt{R^2-a^2}$$

$$+ R^2 \sin(2R \sin \sqrt{R^2-a^2})$$

$$- 4a\sqrt{R^2-a^2} \tag{식 1}$$

이다.

여기서 a는 0에서 $\frac{R}{2}$ 의 범위에 있으며, 평균 값을

중앙값($\frac{R}{4}$)으로 한다면

$$S(B)_{(a=R/4)} = 2R^3 \sin\left(\frac{R}{4}\sqrt{15}\right)$$

$$+ R^2 \sin\left(2R \sin\left(\frac{R}{4}\sqrt{15}\right)\right)$$

$$- \frac{R^2}{4}\sqrt{15} \tag{식 2}$$

따라서 영역 A의 면적은 Hp를 중심으로 하는 원의 넓이에서 영역 B의 면적을 제외한

$$S(A) = \pi R^2 - S(B) \tag{식 3}$$

으로 할 수 있다.

결과적으로 영역 A에 존재하는 노드의 수(N(A))는

$$N(A) = \frac{S(A)}{\pi R^2} * n \tag{식 4}$$

이다.

이를 이용하여 CBRP와 본 논문에서 제안한 CBRP-SH에서의 주헤더 고장시 이를 회복하기 위해 클러스터 내의 발생 패킷 수를 계산한다.

CBRP에서 헤더의 고장이 발생할 경우 각 노드는 헤더의 재선출을 위해 헬로우 패킷을 발생하게 되며, 이를 이용하여 각 노드는 헤더 선출 알고리즘을 이용하여 자신이 헤더인지를 인식한다. 만약 자신이 헤더일 경우 이를 주변의 노드들에게 방송한다.

따라서 CBRP에서 헤더 재 선출을 위한 패킷 발생은

$$A(\text{CBRP}) = n + 1 \tag{식 5}$$

이다.

그러나 본 논문에서 제안한 CBRP-SH에서는 [그림 6]에서 처럼 보조헤더가 즉시 주헤더의 역할을 담당하므로 B영역에 존재하는 노드들은 변화가 없으며, 단지 영역 A에 있는 노드들만이 CBRP에서와 같이 동작한다. 따라서

$$A(\text{CBRP-SH}) = (\text{A영역에서의 주헤더 재 선출을 위한 패킷 수}) + (\text{B영역에서 주헤더의 교체를 알리기 위한 패킷})$$

이며,

(식 4)를 이용하여 다시 표현하면

$$A(\text{CBRP-SH}) = (N(A) + 1) + 1 \tag{식 6}$$

가 된다.

여기서 클러스터 내의 주헤더에 문제가 발생하며 주헤더 재선출로 인한 비용 계산을 패킷 발생 측면만 고려한다. (식 5)와 (식 6)에서 단위 패킷 발생 비용을 k로 한다면 주헤더 부재로 인해 클러스터 내의 주헤더가 재 선출을 위한 비용(CHp; Cost for reelection

Hp)을 계산할 수 있다.

따라서 주헤더 부재로 인한 문제로 인한 주헤더 재선출의 비용은 CBRP의 경우

$$CHp(CBRP) = A(CBRP) * k \quad (\text{식 7})$$

이때 k는 패킷 발생 비용이며, 본 논문에서 제안한 CBRP-SH의 경우는

$$CHp(CBRP-SH) = A(CBRP-SH) * k \quad (\text{식 8})$$

이다.

또한 주헤더에 문제가 생겨 소스 노드와 목적 노드사이의 송수신에 문제가 발생하면 결국 소스 노드와 목적노드 사이의 송수신은 문제가 발생한 클러스터가 복구되어질 때 까지 송수신이 중단된다. 이에따른 경로 재설정 비용(CRR; Cost of ReRouting)을 계산하기 위해 경로 설정을 다음과 같이 구분하여 고려하였다.

첫째, 소수 노드와 목적 노드가 인접한 클러스터에 있을 경우 CBRP는

$$CRR(CBRP_2) = (P(C1) + P(C2) - P(C1 \cap C2)) * CHp(CBRP) \text{이며,}$$

이때, 첨자 2는 통신에 참여하는 클러스터의 수이다. CBRP-SH는

$$CRR(CBRP-SH_2) = (P(C1) + P(C2) - P(C1 \cap C2)) * CHp(CBRP-SH) \text{ 이다}$$

이때 P(Ci)는 클러스터 Ci에서 주헤더가 문제가 생길 확률이다.

그러나 P(C1 ∩ C2)는 클러스터 C1과 C2가 공통 부분이 있을 경우에만 해당되며, 만약 겹친 부분이 없다면, 0이 될 것이다. 따라서 평균적으로는 (식 2)를 이용하여 다음과 같이 표현할 수 있다.

$$P(C1 \cap C2) = N(B) * k$$

둘째, 소수 노드와 목적 노드가 중간 클러스터 1개를 두고 있을 경우,

CBRP는

$$CRR(CBRP_3) = (P(C1) + P(C2) + P(C3) - P(C1 \cap C2) - P(C1 \cap C3) - P(C2 \cap C3)) * CHp(CBRP) \text{이며,}$$

CBRP-SH는

$$CRR(CBRP-SH_3) = (P(C1) + P(C2) + P(C3) - P(C1 \cap C2) - P(C1 \cap C3) - P(C2 \cap C3)) * CHp(CBRP)$$

이다

이를 일반화 하면 CBRP는

$$CRR(CBRP_n) = \left(\sum_{i=1}^n P(C_i) - \sum_{i=1}^{n-1} (C_i \cap C_{i+1}) \right) * CHp(CBRP)$$

CBRP에서는

$$CRR(CBRP-SH_n) = \left(\sum_{i=1}^n P(C_i) - \sum_{i=1}^{n-1} (C_i \cap C_{i+1}) \right) * CHp(CBRP - SH)$$

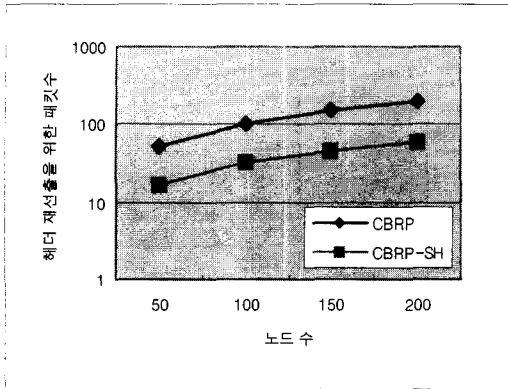
이다.

4.2 고찰

본 논문에서 제안한 CBRP-SH의 타당성을 확인하기 위해 노드에 따른 헤더 재선출을 위한 패킷수, 비용 및 헤더 고장으로 인한 경로 재선출 비용을 산출하였다.

[그림 8]은 주헤더의 고장으로 인해 주헤더의 재선출을 위한 패킷수를 CBRP와 제안한 CBRP-SH에 클러스터내의 노드 수 변화에 따른 변화를 표현하였다.

[그림 8]에서 보는 바와 같이 CBRP에 비해 CBRP-SH는 주헤더의 고장으로 인한 주헤더 재선출을 위한 패킷 수가 상대적으로 적은 것을 확인할 수 있었다.

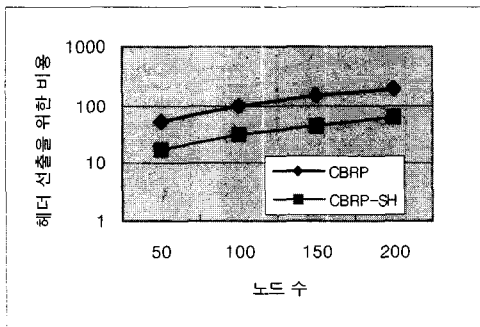


[그림 8] 노드 수에 따른 헤더 재선출을 위한 패킷 수의 변화

[Fig. 8] A change of the number of packet for the number of node

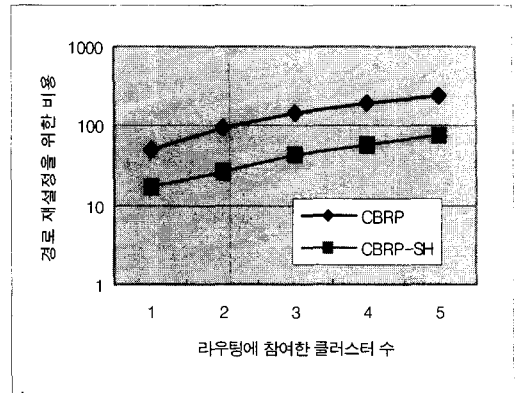
[그림 9]는 (식 7)과 (식 8)에서 클러스터내의 노드 수 변화에 따른 주헤더 재선출 비용($k=1$)을 클러스터내의 노드 수 변화로 나타낸 것이다.

[그림 9]에서 클러스터 내의 주헤더 고장으로 인한 주헤더 재선출 비용면에서 CBRP-SH는 기존의 CBRP에 비해 상대적으로 적응을 확인하였다. 또한 클러스터 내의 노드수가 많을수록 비용 절감의 효과가 증대됨을 확인하였다.



[그림 9] 노드 수에 따른 헤더 재선출을 위한 비용 변화($k=1$)

[Fig. 9] A cost change for header re-election by the number of node



[그림 10] 라우팅에 참여한 클러스터 수에 따른 경로 재설정 비용 변화

[Fig. 10] A rerouting cost change by the number of cluster participating in routing

라우팅 형성 후 라우팅에 참여한 클러스터 중 한 개 이상의 클러스터에서 주헤더 고장으로 인해 주헤더가 재선출된다면 기존의 라우팅은 재설정된다. [그림 10]은 라우팅에 참여하는 클러스터 수에 따른 라우팅 재설정 비용을 표현하였다.

[그림 10]에서 보는 바와 같이 라우팅에 참여한 클러스터의 수가 많을 수록 라우팅 재설정 비용이 CBRP보다 본 논문에서 제안한 CBRP-SH에서 상당한 비용 절감의 효과있음을 확인하였고, 이는 먼 거리의 라우팅에서 CBRP-SH가 기존의 CBRP에 비해 훨씬 효과적임을 나타낸다.

5. 결론

MANET은 백본망이 없고, 동적으로 분산된 이동 호스트들로 구성되어 전이나 천재지변과 같은 긴급사태 및 기존의 이동 통신 기반 구조가 존재하지 않는 지역에서 사용되는 통신망이다. 또 기존의 유선 망과 비교하여 대역폭이 좁고 신호간의 간섭이 상대적으로 심하여 망 형상이 수시로 변경될 수 있다. 따라서 기존의 라우팅 알고리즘을 적용할 수 없으며, 새로운 라우팅 알고리즘 연구가 활발히 이루어지고 있다. 본 논문은 새로운 라우팅 알고리즘중 클러스터 기반의

라우팅 프로토콜(CBRP)에서 클러스터 주헤더의 효율적인 관리를 위해 보조헤더를 두어 클러스터 주헤더의 부재시 주헤더를 재선출할 때까지의 부담을 줄이고자 CBRP-SH를 제안하였다.

본 연구에서 제안한 CBRP-SH는 [그림 8]에서 확인한 바와 같이 클러스터 내의 주헤더 재선출시 비용 면에서 우수한 성능을 나타내었으며, [그림 10]에서 보는 바와 같이 라우팅에서 경로 재설정 비용이 훨씬 효과적인 것을 확인하였다. 특히 클러스터 내의 노드 수가 많을 수록 그 성능은 기존의 CBRP에 비해 비용 절감 효과가 더 크게 나타났으며, MANET의 특성상 클러스터 내의 노드는 밀집되어 분포되며, 이러한 상황에서 본 연구의 CBRP-SH는 CBRP에 비해 보다 더 효과적임을 확인하였다.

※ 참고문헌

- [1] Macker, J., "Mobile Ad Hoc Internetworking.", MILCOM'97 panel on Ad-Hoc Networks, Monterey, CA, November 3, 1997.
- [2] J. Broch, D. B. Johnson, D. A. Maltz, Yih-Chun Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols." <http://www.monarch.cs.cmu.edu/papers.html>
- [3] C. Perkins, "Mobile Ad Hoc Networking Terminology", draft-ietf-manet-term-00.txt, October, 1997.
- [4] M.S. Corson and A. Ephremides, "A Distributed Routing Algorithm for Mobile Wireless Networks.", Wireless Networks 1, 1995.
- [5] V. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks.", Proc. IEEE INFOCOM '97, Kobe, Japan, 1997.
- [6] Charles E. Perkins, AD HOC NETWORKING, Addison-Wesley, 2001
- [7] M. Jiang, J. Li, and Y. C. Tay, "Cluster Based Routing Protocol(CBRP) Function Specification." Internet Draft draft-ietf-manet-cbrp-spec-00.txt, August, 1998.
- [8] <http://www.ietf.org/html.charters/manet-charter.html>
- [9] T. S. Hur, Y. T. W. Tscha, H. K. Min and K. H. Lee, "A Management of Cluster using Secondary Leader in MANET.", PDPTA'2001, pp.1968-1973, June, 2001
- [10] E. Gafni and D. Bertsekas, "Distributed Algorithms for Generating Loop-Free Routes in Networks with frequently changing topology.", IEEE Trans. Commun., January, 1981.
- [11] M.S. Corson, and A. Ephremides, "A Distributed Routing algorithm for Mobile Wireless Networks.", ACM/Baltzer journal on Wireless Networks, January, 1995.
- [12] Z.J. Haas, "A New Routing Protocol for the Reconfigurable Wireless Networks", ICUPC'97, San Diego, CA, Oct, 1997.
- [13] C.E. Perkins. "Ad-Hoc On-Demand Distance Vector Routing.", MILCOM'97 panel on Ad-Hoc Networks, Monterey, CA, November, 1997.
- [14] C.E. Perkins, "Ad Hoc On Demand Distance Vector(AODV) Routing.", draft-ietf-manet-aodv-00.txt, 2001
- [15] J. Broch D.B. Johnson, D.A Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks.", draft-ietf-manet-dsr-00.txt, 2001.
- [16] J. Broch D.B. Johnson, D.A Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks.", in Mobile Computing, T. Imielinski and H. Korth, editors, Kluwer, 1996.
- [17] F. Balazs, G. Jeney and L. Pap, "Capacity Expansion Capabilities in ODMA systems.", Proc. Of WPMC'99, pp.207-213, September, 1999

허 태 성



- 1984 인하대학교 전자계산학과 졸업(이
학사)
- 1987 숭실대학교 대학원 전자계산학과
졸업(공학석사)
- 1984.3~1991.8 인하대학교 전자계산소
근무
- 1999.2 ~ 2000.2 Southern Illinois
University of Computer Science 객
원교수
- 1991.9~현재 인하공업전문대학 부교수

이 균 하



- 1977. 3 ~ 1981. 8 광운대학교 조교수
- 1986. 3 ~ 1986. 8 인하대학교 전자계산
소 개발부장
- 1986. 9 ~ 1987. 8 Purdue Uni- versity
School of Electrical Engineering
객원교수
- 1988.2 ~ 1990.1 인하대학교 전자계산소
개발부장
- 1990.2 ~ 1992.1 인하대학교 전자계산소
소장
- 1981.9 ~ 현재 인하대학교 교수