

# MANET 환경에서 클러스터 기반 주소 할당 프로토콜

정희원 조영복\*, 종신회원 이상호\*\*

## A Cluster-based Address Allocation Protocol in MANET Environment

Young-bok Cho\* *Regular Member*, Sang-ho Lee\*\* *Lifelong Member*

### 요 약

MANETs(Mobile Ad-hoc Networks)에서는 네트워크에 참여하는 노드 간 통신을 위해 노드 식별 주소를 받아야 한다. 주소는 노드 자신 또는 다른 노드에 의해 생성되며, 이 주소가 사용 가능한 유일한 주소인지 검사하기 위해 주소 중복 검사(Duplicate Address Detection)를 수행한다. 하지만 이런 방법은 MANETs의 확장성이 떨어지는 문제가 발생한다. 본 논문은 MANETs에서 이동 노드들을 그룹으로 묶어 그룹 단위로 관리 할 수 있고 제어 메시지의 교환 부하 감소 및 이동성 관리가 용이해지는 클러스터의 특성을 주소할당 프로토콜에 적용하여 주소 중복 검사에 소요되는 시간을 최소화하고 확장성 문제를 해결하는 방법을 제안한다. 본 논문에서 제안한 방법은 시뮬레이션을 통한 결과 MANETConf[2]보다 노드 수 증가에 따라 우수한 성능을 보여준다.

**Key Words** : Mobile Ad hoc Networks, Cluster, Address Allocation Protocol

### ABSTRACT

I must receive node discernment address for communication between node that participate to network in MANETs(Mobile Ad-hoc Networks). Address is created by node confidence or different node. I achieve address redundancy check (Duplicate Address Detection) to examine whether this address is available unique address. However, this method happens problem that MANETs' extensity drops. This paper can manage by group unit binding transfer nodes to group in MANETs. I suggest method that apply special quality of cluster that exchange subordinate decrease and mobility government official of control message are easy in address assignment protocol minimize time required in redundancy check and solves extensity problem. Method that propose in this paper shows excellent performance according to node number increase than wave and MANETConf [2] through simulation.

### I. 서론

MANETs은 기반 통신 시설의 도움 없이 노드들 간에 자율적으로 구성되는 임시적인 네트워크이다. 각 노드는 이동성 및 라우팅 기능을 가지고 있는 이동 노드이다. 이러한 망 특성 때문에 천재지변, 전쟁, 재난 등으로 인해 기반 통신 시설의 이용이 불가능하거나, 사막이나 산간 지대 등 기반 통신 시설이 존재하지 않고, 설치가 용이하지 않은 지역에 적용이

가능하다<sup>1,2</sup>. 이러한 MANETs에서 지금까지의 연구는 주로 경로 설정 방법에 관한 문제 해결이 가장 주된 관심이었다. 하지만 경로 설정 문제에 앞서 MANETs이 패킷 형태의 IP 기반으로 동작할 경우 모든 노드는 MANET 내부에서 또는 MANET을 포함한 전체 인터넷에서 식별할 수 있는 자신의 주소를 필요로 한다. 특히 새로운 MANETs이 형성될 경우 내부에 위치한 모든 이동 노드는 해당 MANETs 내부에서 고유한 자신의 주소를 자동적으로 할당 받을

\* 충북대학교 전자계산학과 네트워크보안연구실(bogi0118@netsec.cbnu.ac.kr), \*\* 충북대학교 전자계산학과 (shlee@chungbuk.ac.kr)  
 논문번호 : KICS2007-01-022, 접수일자 : 2006년 1월 22일, 최종논문접수일자 : 2007년 9월 6일

수 있는 방법이 요구된다<sup>34)</sup>. 이동 노드들에 주소를 할당 하는 방법의 근본적인 차이는 노드들이 자신의 주소를 임의로 선택하는 방법에 있어서 차이가 있으며 선택된 주소에 대한 주소 중복 검사를 수행하는 방법은 거의 동일한 방법을 적용할 수 있다. 일반적으로 주소를 선택하는 방법은 새롭게 추가되는 노드 자신이 주소를 선택하는 방법<sup>4)</sup>, 이미 MANET에서 동작하는 노드가 새롭게 MANET에 참가하는 노드에게 주소를 할당해 주는 방법<sup>1)</sup>, 상태합수를 이용해 동일한 IP 주소를 사용할 가능성을 줄이는 방법<sup>5)</sup>, 서버(MANET에 이미 접속해 있는 노드)는 사용 가능한 주소 풀을 관리하여 클라이언트(MANET에 새롭게 참가하는 노드)에게 주소 풀의 1/2를 전달하여 주소를 사용하는 접근 방법<sup>6,7)</sup> 등이 있다. 이 논문에서는 주소를 할당 할 때 발생하는 주소 중복 지연 시간을 최소화하고 MANETs의 확장성을 높이기 위해 MANETs에서 이동 노드들을 그룹으로 묶어 그룹 단위로 관리할 수 있고, 제어 메시지의 교환 부하 감소 및 이동성 관리가 용이해지는 클러스터의 특성을 주소 할당 프로토콜에 적용하여 주소 할당에 소요되는 시간을 최소화하는 방법을 제안한다. 이동 노드들을 일련의 기준에 따라 묶은 그룹을 클러스터라 하고 이 그룹에 속한 이동 노드들을 멤버라 하며, 이 멤버들을 관리하는 이동 노드를 클러스터 헤더라 한다. 먼저 MANETs은 하나 이상의 클러스터로 구성되며 새로운 이동 노드가 참가하여 주소 요청을 하면 클러스터헤더가 주소를 생성하여 모든 노드에게 플러딩 하지 않고 자신이 관리하는 클러스터의 멤버들에게만 주소 중복 검사를 요청 및 수행하여 고유한 주소를 새로 진입한 이동 노드에게 할당하도록 한다. 이 논문의 구성은 다음과 같다. 2장에서는 MANETs에서 주소를 할당하는 기존 연구에 대해 소개하고, 3장에서는 본 연구에서 제시하고 있는 프로토콜 및 동작을 설명한다. 4장에서는 시뮬레이션 환경과 이 논문에서 제안한 방법에 대한 성능 평가를 기존의 연구와 비교한 결과를 기술하고, 마지막으로 5장에서는 본 논문의 결론에 대해 기술한다.

## II. 관련 연구

### 2.1 새로운 노드 자신이 주소를 선택하는 방법

이 방식은 경로설정 프로토콜을 이용해 DAD (Duplicate Address Detection)를 수행하는 방식으로 PerKins가 제안했다<sup>4)</sup>. 이 방식에서는 새로운 노드가 MANETs에 참여 하려고 할 때, MANETs에 사용하는

주소 중에서 하나의 IP주소를 임의적으로 선택하고 중복 검사를 위해 AREQ(Address Request)메시지를 MANETs에 플러딩 한다. 이 메시지에 송신 노드가 앞으로 사용할 중복 검사를 하지 않은 주소와 주소 중복 검사용으로 선택된 일시적으로 사용하는 주소가 포함되어 있다. 각 노드가 AREQ메시지를 수신하게 되면 자신의 주소와 동일하지 검사하고, 만약 자신의 주소와 중복이 된다면 이에 대한 응답으로 AREP(Address Reply)메시지를 재전송 한다. 정해진 시간동안 기다려도 AREQ메시지가 수신되지 않으면 AREQ RETRIES만큼 반복 수행한다. 만약 모든 시도에서 AREP를 수신 하지 않으면 송신 노드는 선택한 주소가 아직 MANETs내에서 할당되지 않았다고 결론을 내리고 그 주소를 자신에게 할당한다. 이 방식 경우 MANETs이 분할되거나, 분할된 파티션이 다시 합병하는 경우의 해결책을 제시하지 않았다.

### 2.2 MANET에서 동작중인 노드가 할당하는 방법

이 방식은 MANETs에 이미 합류되어 있는 임의의 노드가 MANET의 새로 도착하는 노드에게 주소를 할당하는 방식으로 Nesargi가 제안하는 방식이다<sup>1)</sup>. 즉, 새로 도착한 노드는 MANETs에 Neighbor\_Query 메시지를 플러딩하여 자신의 주위 노드 중 이미 MANETs에 참가 하고 있는 노드를 "Initiator"로 선택한다. 만약 MANETs에 참가한 노드가 없으면 자신이 Initiator가 되고 자신이 사용할 주소를 할당한다. 선택된 Initiator 노드j에게 Requester\_Request(주소할당요청) 메시지를 보내어 Initiator에게 주소 할당을 요청한다. 모든 노드는 현재 MANETs에서 사용되고 있는 주소 정보를 저장하고 있는 Allocated와 현재 주소 할당 진행 중이지만 완료되지 않은 상태의 Allocated\_Pending 리스트를 유지하고 있다. Request\_Request 메시지를 받은 노드 j는 Allocated와 Allocated\_Pending에 없는 주소를 선택해서 Initiator\_Request 메시지를 MANETs 전체에 플러딩한다. 이를 받은 노드는 자신의 Allocated와 Allocated\_Pending에 노드 j가 보낸 주소가 있는지 확인하고 없으면, 노드 j에게 Affirmative Reply를 전송하고, 그렇지 않으면 Negative Reply를 전송한다. 그리고 주소를 자신의 Allocated\_Pending에 추가한다. 노드 j는 접속한 모든 노드에게서 Affirmative Reply 메시지를 수신하면 그 노드에게 주소를 할당하고 MANETs 전체에 이 정보를 플러딩 한다. 이 과정에서 모든 노드들은 Allocated\_Pending에 위치한 주소를

Allocated로 이동하게 되며 만약 Negative Reply를 수신하게 되면 새로운 주소를 설정하고 위 과정을 반복한다.

이 방식에서는 MANETs의 분리와 병합에 대해 해결 방안을 제시하고 있다. 만약 두개의 파티션으로 나누어지면, 가장 낮은 주소를 가진 파티션은 식별자를 그대로 유지하게 되며 일정시간이 지나면 다른 파티션에 있던 주소를 모두 사용 가능한 주소로 사용하게 된다. 다른 파티션의 경우에는 결국 새로운 노드가 추가로 들어오게 되면 일정시간 동안 Allocated된 주소로부터 Affirmative 나 Negative reply를 수신하지 않기 때문에 다른 파티션에서 사용되는 주소들은 자동적으로 현재의 파티션에서 삭제된다. 이 때 가장 낮은 주소를 지닌 새로운 노드를 선출하여 새로운 식별자로 사용하게 되며 MANETs 전체에 플러딩 하게 된다. 두개의 파티션이 만났을 경우에는 처음에 만나는 두 노드가 자신의 Allocated 리스트를 주고받아 자신의 파티션에 플러딩 한다. 이 때 두 파티션에 동일한 주소를 사용하고 있는 노드들에 대해서는 MAC(Media Access Control) address에 의한 우선순위를 따르거나 통신 진행 상황에 따라 우선순위를 정할 수 있으며 우선순위가 낮은 노드들은 자신의 이웃노드 중 하나를 Initiator로 선택하여 새로운 주소를 할당받고 네트워크 전체로 플러딩하여 병합된 MANETs에서 유일한 주소임을 보장 받을 수 있다. 이 방식의 단점은 네트워크가 확장될수록 부하가 발생하여 주소 할당에 걸리는 시간이 길어진다.

### III. 클러스터를 이용한 주소할당 프로토콜

모든 디바이스는 MANETs의 네트워크에 참여하여 통신하길 원한다면 MANET에서 사용하는 통신 주소를 할당받아야 한다. 처음 MANETs환경에 참여한 모바일 노드는 통신에 참여할 통신 주소를 할당받기 위해 MANET에서 사용하는 주소 중 하나의 IP주소를 임의적으로 선택하고 중복검사를 위해 AREQ 메시지를 MANETs에 플러딩 한다. 기존 MANETs 환경의 통신에 참여하고 있던 노드들은 플러딩 메시지를 받고 자신의 주소 IP와 중복이 되는지 확인한 후 응답메시지(AREP)메시지를 재전송한다. 만약 모든 참여 IP에서 AREP메시지를 받지 못한다면 플러딩한 주소는 사용하지 않는 주소라 판단하고 플러딩한 디바이스는 그 주소를 할당하여 사용하게 되는 이 과정을 DAD(주소중복 검사)라 한다.

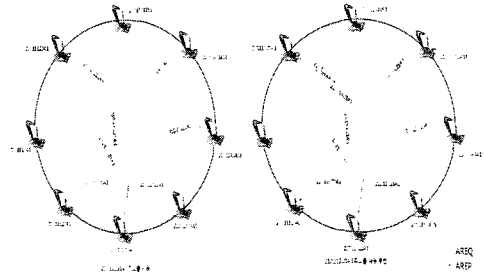


그림 1. 주소중복 검사 (DAD)

이런 DAD는 MANETs의 경우 MANETs을 구성하는 디바이스는 이동 디바이스이므로 MANETs이 분할되거나 분할된 파티션이 합병될 경우 문제가 된다. 이를 해결하기 위해 본 논문에서는 MANETs을 먼저 클러스터링을 구성하고 클러스터 헤더 선정 알고리즘을 이용하여 MANETs의 분할과 합병시 DAD되는 효율성을 높이는 알고리즘을 제안하였다. 기존 주소 중복 검사는 MANET의 네트워크 확장시 부하가 발생되어 주소가 할당되는 시간이 길어지는 문제점을 가지고 있었으나 제안 알고리즘을 사용하면 이를 해결하여 네트워크 확장시 노드의 수가 증가할수록 지연시간이 감소함을 알 수 있다.

#### 3.1 시스템 모델 고려사항

A Mobile Ad-hoc Network(MANET)은 독립적으로 이동하는 모바일 기기들의 집합으로 이루어진 네트워크이다. MANET에서 각 모바일 기기는 유선망에 비해 상대적으로 대역폭이 제한된 무선링크를 통하여 다른 기기와 통신을 하고 제한된 Radio Propagation 영역의 특징으로 인하여 대부분의 경로는 멀티 홉(multi-hop)특성을 갖는다. Ad-hoc 네트워크는 어떤 인프라 없이도 네트워크 구성이 가능하고 MANET을 구성하는 모바일 기기들은 자유로이 이동하면서 임의로 스스로를 구성할 수 있으며, 네트워크 토폴로지 자체는 자주 변화하게 되어 예측하기 힘들다<sup>[2]</sup>. 이와 같은 내용을 기반으로 본 논문에서 가정하고 있는 시스템 모델에서 고려되어야 할 사항은 다음과 같다

- 인터넷 같은 외부 네트워크에 연결이 없는 독립적인 MANETs 환경이다.
- 노드는 MANETs을 떠날 때 정상적으로 종료해야 하지만 배터리 부족, 노드의 이동 등으로 인해 예기치 않게 떠날 수도 있다.
- 하나의 이동 노드는 하나의 클러스터에만 속한다.

- MANETs은 하나 이상의 클러스터로 구성되고 각 클러스터는 분할되고 병합 될 수 있다.

이동 노드의 주소를 할당하기 위해 사용한 클러스터링 방법은 클러스터를 구성할 때 일정 시간동안 주위로부터 아무런 메시지를 받지 않았을 경우 자신이 스스로 클러스터 헤더임을 선언하도록 한다. 두 개의 노드가 동시에 클러스터 헤더 선언의 충돌이 발생 가능하다. 이러한 충돌을 방지하기 위해 충돌이 발생하면 두 노드 중 낮은 MAC 주소를 가진 노드를 우선적으로 클러스터 헤더로 선출한다. 또한, 클러스터가 분할되거나 합병될 때에도 같은 방법을 이용하는 "Random Competition based Clustering"<sup>[8,9]</sup> 방법을 적용한다.

### 3.2 제안 알고리즘

#### 3.2.1 MANET 초기화

MANETs에 처음으로 합류한 노드 n은 초기화 과정으로 원 흠 떨어진 노드들에게 Neighbor\_Query 메시지를 플러딩 한 후에 Neighbor\_Query 타이머를 설정한다. Neighbor\_Query 타이머 기간 내에 n이 이웃 노드들로부터 어떠한 응답 메시지도 받지 못한다면, 위의 과정을 Neighbor\_Query\_Retries 만큼 반복(3회 수행)한다. 이 기간 내에도 이웃 노드로부터 응답을 받지 못하면 n은 MANETs에 자신 외에 아무도 없다고 결론을 내리고 스스로 자신의 MAC 주소를 프리픽스로 하는 클러스터를 생성하여 주소를 할당하고 (MACaddr.0), 자신이 클러스터헤더가 되고 이를 MANETs에 플러딩 한다. 초기화 과정에 대한 알고리즘을 그림 2에서 보인다.

```

//초기과정
{
  cMsg Neighbor_query(m_vecNodes.size(),0,"");
  pNode->m_nState=0; //노드 합류되지 않음
  pNode->m_nState_Act=0
  while(retry<m_iRetry && pNode->m_nState_Act !=2)
  { for(i1=0;i1<m_vecNodes.size(); i1++)
    { pMsg = m_vecNodes[i1]->PushMsg
      (&Neighbor_query);

      if( pMsg != NULL)
      { cnt_recev++;
        pMsg = pNode -> PushMsg(pMsg);
        delete pMsg;
      }
      // 수신 받은 적이 없다면
    } m_result.delayTime++;
  } if( pNode->m_nState_Act !=2) // 수신 받지 못함.
  { // 클러스터 헤더로 지정,
    pNode->m_nState = 1;
    pNode -> m_ID = 0;
    M_vecNodes.Push_back( pNode);
  }
}
    
```

그림 2. 초기화 알고리즘

#### 3.2.2 새로운 노드의 MANETs 합류

새로운 노드가 진입하면 그림 3처럼 우선 Neighbor\_Query 메시지를 플러딩 한다. 이 메시지를 받은 이웃 노드는 이에 대한 응답으로 Neighbor\_Reply 메시지를 전달한다. 새로운 노드가 Reply 메시지를 받으면 이 Reply 메시지를 전송한 노드에게 주소를 요청하는 메시지를 전송한다. 주소를 요청하는 알고리즘은 그림 4와 같다. 만약 일반 노드라면 이를 클러스터헤더에게 전달하고, 클러스터헤더가 주소 요청 메시지를 받았다면, 임의 주소를 할당하여 멤버들에게 주소 중복 검사를 요청한다. 주소가 멤버로부터 승인이 되면 새로운 노드에게 주소를 할당한다. 이때 새로 진입한 노드는 먼저 도착한 응답 메시지에 반응하고 다른 메시지는 폐기한다.

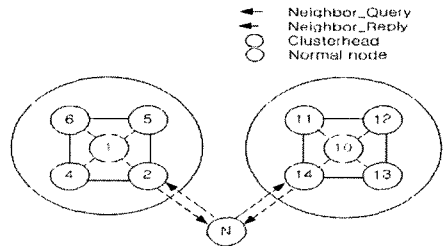


그림 3. 클러스터에서 새로운 노드 이웃 노드 질의 메시지

```

// 주소지정
while(pNode->m_nState_Act != 4 && retry <m_iRetry)
{
  pMsg2=NULL
  for (i1=0; i1 < m_vecNodes.size(); i1++)
  { if( pNode -> m_nState_Act ==4) break ;
    fprintf(OUT, "%d 노드에게 맥을 요청함 Wn"i1);
    pMsg = m_vecNodes[i1] ->PushMsg(&request_mac);
    int dbg =m_vecNodes.size();
    if(pMsg->m_iPacketType==5)
      pMsg2 = pNode->PushMsg(pMsg);
    else if (pMsg->m_iPacketType==7) {
      pMsg2=pNode->PushMsg(pMsg);
      delete pMsg;
    } if (pMsg2 !=NULL) {
      switch(pMsg2->m_iPacketType)
      {
        case 6:
          GetMacAddr(pNode);
          pNode->m_nState =2;
          pNode->m_nState_Act=4;
          m_vecNodes.push_back(pNode);
          break;
        case 7:
          break;
      }
      delete pMsg2;
      pMsg = NULL;
    }
    retry++;
  }
}
    
```

그림 4. 주소 요청 알고리즘

3.2.3 노드의 MANET 탈퇴

MANET 또는 클러스터를 탈퇴하는 노드는 탈퇴 메시지를 자신의 Neighbor에게 플러딩 한다. 이를 통해 클러스터헤더는 탈퇴하는 노드의 주소를 재사용할 수 있다. 하지만 노드가 예기치 않게 탈퇴하는 경우 클러스터 헤더가 새로운 노드에게 주소를 생성하고 이를 멤버들에게 주소 중복 검사를 요청하여 주어진 시간(Addr\_Reply\_Timer) 동안 응답을 하지 않는 노드에 대해 주어진 횟수만큼 재시도하여 그때 까지도 응답을 하지 않는다면 탈퇴 노드로 간주하고 이를 클러스터 멤버들에게 알리고 주소를 재사용할 수 있게 한다.

3.2.4 클러스터의 합병 및 분할

MANETs은 하나 이상의 클러스터로 구성된다. 하지만 노드들의 이동성으로 인해 클러스터가 분할될 수 있으며, 다른 클러스터와도 합병될 수 있다. 각각의 클러스터가 서로 근접하여 합병될 경우 각각의 클러스터헤더는 다른 클러스터가 존재함을 인식하고 자신의 LEADER-BEACON 메시지를 전송한다. 이 메시지를 통해 서로의 MAC 주소를 확인 한 후 보다 작은 MAC 주소를 가진 클러스터헤더가 합병된 클러스터의 헤더가 되고 나머지는 클러스터의 멤버로 합류하게 된다. 그림 5는 이렇게 클러스터가 합병된 후 클러스터헤더가 새로 합류한 노드들에게 주소를 할당하는 과정이다.

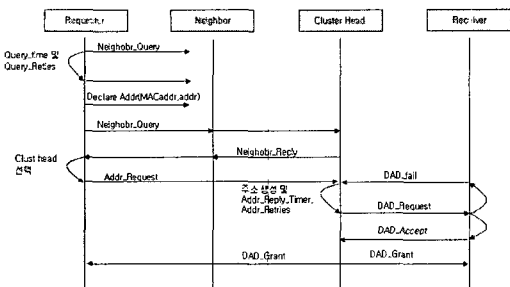


그림 5. 주소 할당 프로토콜 절차

클러스터가 분할되는 경우는 클러스터헤더가 있는 곳은 그대로 클러스터 정보를 유지하고 헤더가 없는 클러스터의 멤버들은 클러스터헤더로부터 메시지를 받지 못하게 되면, 이웃 노드가 없는 경우 가장 가까운 거리에 있는 클러스터의 멤버로 합류한다. 만약 다른 클러스터헤더가 없고 이웃 노드가 있는 경우라면 클러스터를 생성하여 자신이 클러스터헤더임을

선언하여 클러스터를 생성한다. 만약 클러스터에 모든 노드가 남아있지 않으면 클러스터는 소멸된다.

IV. 성능평가

4.1 실험 환경

이 논문에서 NS2 시뮬레이터를 사용하여 성능평가를 했다. 성능평가를 위해 몇 가지 환경을 설정하였다. 망의 크기는 1km x 1km으로, 네트워크 합류시간은 0초에서 1000초 사이, 클러스터의 구성은 하나의 클러스터에서 30개의 멤버 노드를 서비스하며, 클러스터에 합류한 노드는 시뮬레이션이 끝날 때 까지 클러스터를 벗어나지 않는다. 또한 네트워크에 25개의 노드가 존재하고 있으며 네트워크에 한번 추가시 노드를 25개씩 추가하여 합류시킴으로 노드 합류시 걸리는 평균시간을 측정하였다. 이때 정확한 측정을 위하여 노드의 수를 50개, 75개, 100개의 노드로 구성된 환경에서 시뮬레이션을 수행하였으며 추가되는 노드가 합류시 노드들의 속도는 5m/s로 하였다. 두 노드들 사이의 최대 전송 범위는 250m로 AODV 라우팅 프로토콜을 사용하고 패킷은 CBR 트래픽을 발생하였다. 이때 발생된 패킷을 UDP를 통해 목적지 노드에 전송하도록 설계하였다.

4.2 성능 평가

제안 모델의 성능을 평가하기 위해 노드 개수를 25, 50, 75, 100개로 25개씩 증가 시키면서 새로 네트워크에 합류한 노드가 주소 할당을 받는 데 걸리는 평균 시간을 표 1에 나타낸 것이다.

표 1. 주소할당 평균 지연시간 (단위:초)

Node 수	25개	50개	75개	100개
주소할당시간				
MANETconf	4.80	6.40	11.11	18.02
제안모델	5.23	6.39	9.02	13.93

표 1에서 기존의 MANETconf 방식은 네트워크에 이동 노드가 25개로 구성되어 있을 때 새로운 이동 노드가 25개가 합류해서 주소 할당을 받는 시간이 평균 4.8초에서 이동 노드가 100개 합류할 경우 평균 18.02초가 걸렸다. 그러나 기존 MANETconf 방식에 비해 제안한 방식은 25개 새로운 이동노드가 합류해서 주소를 할당받을 경우 평균 5.23초에서 이동노드가 100개 할당받을 경우, 13.98초가 걸렸다. 그림 6는 주소할당 평균지연시간을 그래프로 나타낸 것이다.

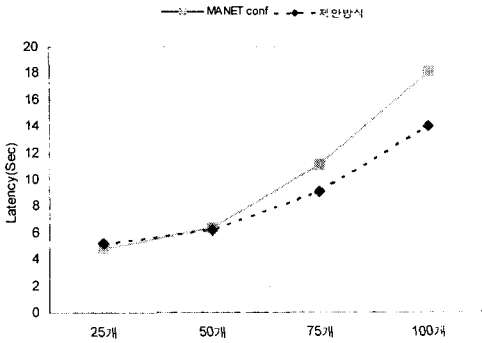


그림 6. 노드 개수에 대한 주소 할당 시간

위 실험 결과와 같이 처음 25개의 이동노드에서는 기존 MANETconf 방식에서 주소할당시간이 제안방식에서보다 0.43초 감소 되었다. 이 결과는 새로운 네트워크에 합류한 노드가 5개까지는 같은 클러스터 헤더에 의해 주소를 할당받기 때문에 각 노드는 이 클러스터 선택하는 시간이 소요되기 때문에 제안 방식에서는 기존 MANETconf 방식보다 더 많은 지연시간이 발생된 것이다. 그러나 MANET에 50개의 경우 0.01초(0.02%) 개선되고, 75개의 경우 2.09초(18.82%) 개선되며, 100개의 경우 4.09초(22.70%) 성능이 개선됨을 알 수 있다. 또한 클러스터 크기에 따른 평균 지연시간을 측정하기 위해 클러스터의 크기를 15개, 30개, 45개의 이동 노드로 변경하면서 초기 네트워크를 구성하는 이동 노드의 수를 25, 50, 75, 100개로 구분하여 네트워크 크기에 대해 주소할당에 소요되는 시간을 측정하여 표 2로 나타내었다. 이때 각각의 경우 신규로 추가되는 이동노드의 수는 25개로 동일하게 제한하였다.

표 2. 클러스터크기에 따른 평균 지연시간 (단위:초)

클러스터 수 \ 노드 수	25개	50개	75개	100개
15개	5.19	6.42	7.57	8.52
30개	5.23	6.23	6.99	7.28
45개	5.14	6.23	7.11	8.10

초기 네트워크가 25개의 이동노드로 구성된 경우 클러스터의 크기가 45노드의 경우 5.14초로 가장 적게 소요되었다. 표 2를 보면 각 클러스터 크기에 따른 총 지연 시간은 클러스터 크기가 15노드인 경우 27.7초, 30노드인 경우 25.73초, 45노드인 경우 26.58초 소요되었다. 따라서 제안 방법은 클러스터를 구성

하는 노드의 개수는 30노드일 때 가장 좋은 성능을 보임을 알 수 있다. 그림 7은 클러스터 크기에 따른 평균지연시간을 그래프로 나타낸 것이다.

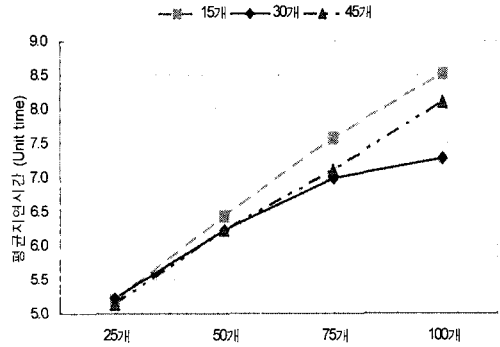


그림 7. 클러스터 크기에 따른 평균 지연시간

## V. 결론

MANET에서 경로 설정이 수행되기 전에 각 노드들은 식별할 수 있는 주소가 할당되어야 된다. 네트워크 토폴로지가 동적으로 변화하는 특성으로 인해 MANET 환경은 DHCP같은 중앙 집중형 방식으로 주소를 할당하는 방법 대신 다른 접근 방법이 필요하다. 이 논문에서는 MANET 내의 노드들에게 동적으로 식별 가능한 주소를 제공할 때 발생하는 문제를 해결하기 위해 클러스터 기반 주소 할당 프로토콜을 제안하였다. 노드들은 클러스터 단위로 나누어 관리하는 특성을 이용하여 기존 방법과 비교평가를 수행하였다. 제안 방법은 MANETconf 방법에 비해 노드의 개수가 50개미만의 경우 주소 할당 소요시간은 더 소요되었다. 즉 기존 모델에서 100개의 노드의 소요시간이 18.02였다. 그러나 제안모델에서는 클러스터 크기에 따른 평균 지연시간을 계산해 본 결과 최적의 클러스터를 계산하였고, 실험 결과 100개의 노드에서는 30개의 클러스터가 가장 적합하다는 것을 실험을 통해 도출하였다. 따라서 제안 모델에서 100개의 노드를 30개의 클러스터로 구성된 네트워크에서 주소할당 지연시간을 실험한 결과 7.28초가 소요되었다. 즉 이 논문에서는 클러스터의 특성을 이용해서 MANET의 크기가 대용량이라도 주소할당을 효율적으로 적용 가능한 방법을 제안 하였다. 시뮬레이션을 통해 제안하는 방법은 기존의 MANETconf. 방법에 비해 노드의 개수가 많아질수록 보다 효과적으로 주소를 할당함을 보였다. 향후 연구로는 주소할당

을 위해 클러스터에서 전송 되는 메시지의 감소를 위한 클러스터 유지 방법의 연구가 선행 되어야 할 것이다.

참 고 문 헌

[1] 권혜연, 신재욱, 이병복, 최지혁, 남상우, 임선배, "이동 Ad Hoc 네트워크 기술 동향", 전자통신동향분석 제18권 제2호 2003년 4월

[2] C.E Perkins, "AD HOC NETWORKING" Addison Wesley

[3] S. Nesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network", in Proceedings of the 21th IEEE Conference on Computer Communications, pp. 1059-1068, June 2002.

[4] C.E. Perkins, J.T. Malinen, R. Wakikawa, E.M. Belding-Royer, and Y. Sun, "IP Address Autoconfiguration for Ad Hoc Networks, draft-ietf-manet-autoconf-01.txt", IETF Working-Group, November 2001.

[5] Hongbo Zhou, Lionel M. Ni, Matt W. Mutke, "Prophet Address Allocation for Large Scale MANETs", INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE ,2 ,30 March-3 April 2003. Pages:1304 - 1311 vol.2

[6] Mansoor Mohsin and Ravi Prakash, "IP Address Assignment in a Mobile Ad Hoc Network", IEEE Military Communications Conference (MILCOM 2002), volume 2, pp. 856-861, October 2002.

[7] Abhishek Prakash Tayal, L.M. Patnaik, "An address assignment for the automatic configuration of mobile ad hoc networks, February 2004, Personal and Ubiquitous Computing, Volume 8 Issue 1

[8] K. Xu, X. Hong, and M. Gerla, An ad hoc network with mobile backbones, in Proceedings of the IEEE International Conference on Communications (ICC), New York, NY, Apr. 2002.

[9] C. Cramer, O. Stanze, K. Weniger, M. Zitterbart, "Demand-Driven Clustering in MANETs". In Proceedings of International Workshop on Mobile Ad Hoc Networks and Interoperability Issues (MANETII04), Las Vegas, USA, April 2004

[10] The Network Simulator-ns-2, <http://www.isi.edu/nsnam/ns/>

[11] Jae Chung, Mark Claypool, NS by Example, <http://nile.wpi.edu/NS/>

[12] 김관웅, 배성환, 김대익, "MANETs,에서 차등서비스 제공을 위한 트래픽 관리 기법" 한국정보처리학회논문지 ,제13-C권 제4호 ,Vol 13-CN04, pp455-460, Aug 2006

조 영 복 (Young-bok Cho)

정회원



2003년 2월 한밭대학교 컴퓨터공학과 졸업  
 2005년 8월 충북대학교 전자계산학과 석사  
 2006년 3월~현재 충북대학교 전자계산학과 박사과정  
 <관심분야> Ad-hoc Wireless Networks, Routing Protocol,

이 상 호 (Sang-ho Lee)

종신회원



1972년 3월~1976년 2월 숭실대학교 전자계산 공학사  
 1979년 3월~1981년 2월 숭실대학교 시뮬레이션 공학석사  
 1985년 3월~1989년 2월 숭실대학교 컴퓨터네트워크 공학박사  
 <관심분야> Protocol Engineering, Network Security, Network Management, Network Architecture