MANET Address Assignment Protocol that use Cluster

Dong heui Lee*, Young bok Cho** Sang ho Lee**

*dept of Computer Science Network Security Lab, Chungbuk National Univ, 48 Gaeshindong Heungdukgu, Cheongju ChungcheongbukDo, KOREA,

**dept of Computer Science Network Security Lab, Chungbuk National Univ, 48 Gaeshindong Heungdukgu, Cheongju ChungcheongbukDo, KOREA,

Abstract:- It is suggested that applying the characteristics of the cluster to the address assigning protocol which can manage mobile nodes in a group, reduce exchange traffic and make it easier to manage mobile characteristics to minimize the Duplicate Address Detection time and solve the expansion problem. In the simulation results, it showed superior function when the number of nodes was increased over the existing MANET Configuration [2], node identification addresses are self-generated or generated by the other nodes and perform the Duplicate Address Detection to check whether it is an avaliable address. However, this can cause interior expansion of MANETs. In this thesis, it is suggested that applying the characteristic of the cluster to the address assign protocol which can manage mobile nodes in a group, reduce exchange traffic and make it easier to manage mobile characteristic to minimize the Duplicate Address Detection time, and solve the expansion problem. In the simulation results, it showed superior function when the number of nodes was increased over the existing MANET Configuration [2].

Key-Words: - Ad-Hoc, DAD, Cluster, Address Allocation Protocol, NS2

1. Introduction

MANETs is a temporary network that consists of nodes that don't require infra-structure communication, also each node has a mobile and routing function. Because of these characteristics, it can be used when it is impossible to use infra-communication facilities due to a natural disaster, war, or other disasters and it can be used in areas that have no infra structure or where an infra-structure is hard to set-up such as in a desert, or mountain area[3],[4]. So far, in the field of MANETs research, the main interest has been how to set the route.

However, before setting up a route, all nodes need their own identification address which can be identified on the internet including MANET or in a MANET if the MANETs work in the IP base is a packet type. Especially, if a new MANET is formed, then, all mobile nodes which are located in the MANETs need some way to get their own address automatically from the inside. The differences of how to assign addresses to those mobile nodes depends on how the nodes choose their own address, and Duplicate Address Detection can be applied to the addresses chosen. In this study, to minimize the address assignment time, it is suggested that applying the cluster characteristics which can control mobile nodes in the group, can reduce the exchange traffic of the control message and can easily manage mobile characteristics to minimize the address duplication delay time while assigning the address which is to provide mobility to nodes in the MANET, and to increase the expandibility of MANETs. The mobile nodes group which is gathered by a certain standard is called a cluster, and each mobile node is called a member, and the mobile node that controls the members is called a cluster head. First of all, the MANET consists of more than one cluster, and whenever a new mobile node requests an address, the cluster head generate an address, it doesn't flood an address to all nodes and only asks the cluster members to perform the Duplicate Address Detection and assigns an address to the new mobile node. The structure of this thesis is as follows. We will introduce the existing studies of assigning addresses in a MANET in the second chapter, and in the third chapter, we will explain the protocol and its operation which is suggested in this thesis. In the fourth chapter, we will describe the test environment and the comparison results with the existing studies, and in the fifth chapter, we will described our conclusion of this thesis.

2. Related Study

2.1 How to Present Operating Nodes Assign the Address

Temporary nodes which are already participating in a MANET should assign an address to the newly arrived nodes[3]. This means that newly arrived node *i* floods the *Neighbor_Query* message to the MANET and chooses the node as an "*Initiator*", which has already participated in the MANET, from among the nearby nodes. Also it sends a *Requester_Request*(requesting address assignment.) to chosen "*Initiator node j*" to request an address. Node *j* which has received a *Request_Request* message chooses a new address from the *Allocated_Pending* and floods the *Initiator_Request* message to all of the MANETs. Also they add the address to their own *Allocated_Pending* list. After node *j* receives an Affirmative Reply from all nodes, it assigns the address to the nodes and floods this information to the entire MANET.

In this process, all nodes move the address from the *Allocated_Pending* list to the Allocated list and if they receive a Negative Reply, they set a new address and repeat the above process. In this way, the solution about the separation and combination of MANETs is suggested. If the partition is divided, then, the lowest address partition maintains the identifier and after some time, it uses all of the addresses in the other partitions. In the case of separated partitions, when a new node comes, it doesn't receive an Affirmative or a Negative Reply from the allocated address for some period of time, therefore the addresses which were used in the other partition are automatically deleted in the present partition. At this time, if the any of the nodes is using the same address in the partition, it follows the preference order according to the *MAC*(Media Access Control) address or follows the communication processing situation, also a lower preference node chooses one of their neighboring nodes as an initiator, gets a new address and floods it to the entire network to ensure that it is a unique address in the combined MANETs. The disadvantage of this method is that it can take a long time to assign the address because traffic occurs when the network expands.

2.2 How the Newly Participated Nodes Choose the Address Assigning

Perkins suggested using route set protocol to perform *DAD*(Duplicate Address Detection)[5]. In this method, when a new node tries to participate in the MANET, it temporarily chooses an IP address from among the addresses which are being used in the MANET and floods an AREQ(Address Request) message to the MANETs to check the DAD. When each node receive *AREQ* message, it checks whether it is the same address with theirs and if it is duplicated address then it sends *AREP*(Address Reply) message. If it doesn't receive any *AREP* from the all tries, then it regards that chosen address of receiving node is not assigned in the MANETs and assign that address to themselves. The problem of this way is that it doesn't suggest any solution for the case when MANETs separated or when the separated partitions are combined again.

3. Mobile Address Assigning Protocol in MANET

In the clustering method which were used for assign the address to the mobile node in MANET, cluster head will declare that itself is a cluster head if it didn't receive any message during consisting the cluster. It is possible that two nodes declare that themselves are the cluster head, however, in this case, lower MAC node will be chosen as a cluster header and also the "*Random Competition based Clustering*" [9],[10] which is used when cluster become separated or become combined, will be applied.

3.1 Initiating Algorism in the Address Assigning Protocol

Node n which is participated in MANETs in the first time, it floods *Neighbor_Query* message to the one hop seperated nods and set the *Neighbor_Query* timer as an initiation process. If n didn't receive any response messages from neighbor nods during the *Neighbor_Query* timer period, the above process will be repeated 3 times as *Neighbor_Query_Retries*.

```
int cnt_recev = 0 ; // The number to take the transmission
int retry = 0; // The number currently of the replay
//the initialization course
if ( m_vecNodes.size() > 0)   //If node exists
{                              // Neighbor_query message transmission to the node of the total
   pMsg Neighbor_query(m_vecNodes.size(),0,"");
   fprintf(OUT, "OUT, "New node waiting response \n");
   m_result.nNode++ ;    //increase the number of nodes
   pNode->m_nState = 0;
   pNode->m_nState_Act =0;
   // wait for the response
   While( retry < m_iRetry && pNode->m_nState_Act !=2 )
   {          //the message every node transmission
      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
              }
      } m_result.delayTime++ ;
if(pNode->m_nState_Act != 2) // can not take the receiving
{
   // can not communicate with different node by isolated node.
   // appoint to the cluster head.
   pNode->m_nState=1;
   pNode->m_ID=0;
   m_vecNodes.push_back(pNode);

   fprintf(OUT,"isolated node\n");
   fprintf(OUT,"cluster head generation Mac(%d)\n",pNode->m_ID);
}else
{  //take the receiving
   pNode->m_nState = 2;    //general insertion node
   pNode->m_nState_Act = 2;  // general insertion node

   fprintf(OUT,"the insertion to general node at the network \n");
```

Fig. 1. Initiating Algorism

If it doesn't receive any responses from neighbor nods in this time, then n will regard that there is nothing but itself in MANETs and it will generate a cluster which prefix it's MAC address and assign an address (MACaddr.0), then itself become a cluster head, and flood it to MANETs. The figure 1 is a algorism of the initiation process.

3.2 New Nodes' Participating in MANETs

When a new nod participate, first it will flood *Neighbor_Query* message. The neighbor nods which received this message will send *Neighbor_Reply* message. After the new nod receive Reply message, it send address requesting message to the nods which sent Reply message.

```
while(pNode->m_nState_Act != 4 && retry < m_iRetry )  // address request
{
  pMsg2 = NULL;
  for(i1 =0; i1 < m_vecNodes.size() ; i1++)
  (
      if(pNode->m_nState_Act == 4 ) break ;
      fprintf(OUT, "%d request the MAC to the node  \n",i1);
      pMsg = m_vecNodes[i1] ->PushMsg(request_mac);
              int dbg =m_vecNodes.size();
      if(pMsg->m_iPacketType == 5)
      (
          fprintf(OUT,"get the response from the cluster \n",i1);
          pMsg2 = pNode->PushMsg(pMsg);  //transmit at the node
      }else if( pMsg->m_iPacketType ==7)
      {
          fprintf(OUT,"get the response form general node \n", i1);
          pMsg2=pNode->PushMsg(pMsg);  // transmit at the node
      }
      if( pMsg2 != NULL)
      (
        switch(pMsg2->m_iPacketType)
          {case 6: //a cluster leader delivered
               GetMacAddr(pNib+);//MAC address
                 fprintf(OUT,"assigned MAC address(%d) from cluster leader \n", pNode->m_macAddress);
                 pNode->m_nState = 2 ; // general node situation
                 pm_nState_Act = 4;  // add network
             m_vecNodes.push_back(pNode);
             fprintf(OUT,"order the append at the network to (%d) node \n",pNode->m_ID);
                break;
          }
      )
  }
}
```

Fig. 2. Address Request Algorism

The address requesting algorism is as figure 2. After the nod receive a Reply message about the address requesting, pMsg2 will check whether it is sent from the cluster, and if it is from the cluster, then, nod will add member nod state (m_nState=2) to network(m_nState_Act=4). If it didn't receive address, then it will repeat the Query message as the assigned number of times(retry++) to the neighbor nods. Figure 2 shows the address requesting algorism for new nod to participate in MANETs.

3.3 Node's Leaving the MANET

The nodes which leave MANET or cluster flood Leave message to its neighbor nodes. Cluster head can use the left node address through the above process. However, if a node left unexpectly, then, cluster head generate a new address to a new node and request DAD to members, and retries it when some nodes don't response in the time limit (*Addr_Reply_timer*), if they still don't answer in the time limit, then cluster head regards them as left nods and note this to the cluster members and make the left nodes address available.

3.4 Convergence and Separation of Cluster

MANET consists with more than one more cluster. However, cluster can be separated or be combined with other cluster because of the mobile characteristic of nodes. When each clusters be closed together and be combined, each cluster head recognize that there are other cluster existed, and send their *LEADER-BEACON* message to them. They check each other's MAC address through the above message, and the cluster which has lower MAC address become a head of the combined cluster and others became a member of the cluster. When cluster separated, the cluster information will be maintained where the cluster head exists, and the cluster members which have no head will not receive a message from the cluster head, and will be combined with the closest cluster, if there are no neighbor nodes. If there is no cluster head but there are neighbor nodes, then, it will generate a cluster and will declare itself as the cluster head. If there are no nodes in the cluster, the cluster will disappear.

4. Performance Test

4.1 Test Environment

The computer system capacity, operating system and software which will be used to perform this simulation are showed on Table 1.

Table1. Performance Test Tool

H/W	P-IV 2.4GHz system
O/S	RedHat Linux 9.0
S/W	Ns v2.26 C++, Tcl(Tool command Language)

Must be set for the mobile nodes which will be test through the above computer system because it consists of a hierarchical structure.

Table 1. Mobile Nodes Parameter

Parameter	Value
ChannelType	Channel/WirelessChannel
propType	Propagation/TwoRayGround
netif	NetIf/ShareMedia
macType	Mac/802_11
ifqType	Queue/DropTail/ProQueue
ifqLen	50
LLType	LL
antType	Antenna/OmniAntenna
adhocRouting	AODV

In the mobile node hierarchical structure, the LLtype sets the link which regulates band width, delay, and Q type. It assumes that the omni-antenna is set in the center of the mobile node and that it is 105m above the mobile node. Share Media in the network interface uses 944MHz wireless frequency and the range of the wave is 250m.

4.2 Performance Evaluation

4.2.1 Address Assigning Delay Time According to the Added Mobile Nodes

Figure 3 is the average delay time that it takes to assign an address to a mobile node which is combined to the network according to the number of added mobile nodes. More mobile nodes are added, which means that the network becomes larger, which requires more delay time, however the suggested method shows a lower average delay time than the existing way. As we can see in Figure 3, the existing MANET configuration took an average of 4.80 seconds to get an address when there were 25 mobile nodes in the network so 25 mobile nodes combined, therefore when it is 100, it took an average of 18.02 seconds. In the suggested test, when there are 25 mobile nodes combined, it takes 5.23 seconds, so when 100 mobile nodes are combined, it takes 13.93 seconds, so we checked that the address allocation average delay time is 0.43 seconds faster than the existing MANET configuration. We can see from these results, because there are newly added nodes in the network, it generates a new cluster and gets an address from the cluster head so it takes more time to choose a cluster. However, in the test results, compared to a MANET, if there are 50 mobile nodes, it is improved by 0.01 seconds (0.02%), if there are 75 mobile nodes, it improves by 2.09 seconds (18.82%), and if there are 100 mobile nodes, it improves by 4.09 seconds(22.70%).

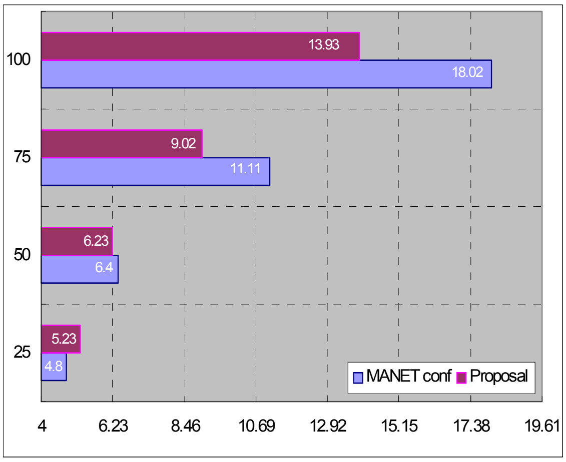

Fig.3. Address Assigning Delay Time According to the Added Mobile Nodes

4.2.2 Address Assigning Delay Time According to the Cluster Size

Figure 4 shows the most proper cluster in the address allocation algorism. We checked the address allocating time by changing the cluster size to 15, 30, and 45 mobile nodes and changing the number of mobile nodes of the beginning network to 25, 50, 75, and 100. As we can see from table 3, when the beginning network consisted of 25 mobile nodes, and when the cluster size was 45 nodes, it showed the shortest time of 5.14 seconds. When the network consisted of 50 mobile nodes, and the cluster size was 15 nodes, it showed 6.42 seconds, so compared to the clusters with 30 and 45 nodes, it took 2.5% longer. When the network consisted of 75 mobile nodes, and when the cluster size was 30 nodes, it took 6.99 seconds, so compared to the cluster of 15 nodes it took 0.58 seconds (7.6%) less and compared to 45 nodes, it took 0.12 second(1.6%) less. When the network consisted of 100 mobile nodes, and the cluster was 30 nodes, it took 7.28 seconds, so compared to the cluster with 15 nodes it took 1.22 seconds (14.5%) less and compared to 45 nodes, it was 0.42 seconds (10.1%) faster.

The total delay times according to the cluster size, when the clusters were 15 nodes, 30 nodes and 45 nodes, was 27.2, 25.73, and 26.58 seconds. Therefore, in the suggested method, when there were 30 cluster nodes, it showed the best performance.

Table 3. Mean/Total delay time by cluster sizes

Network / Cluster	25	50	75	100	Total Delay Time
15	5.19	6.42	7.57	8.52	27.7
30	5.23	6.23	6.99	7.28	25.73
45	5.14	6.23	7.11	8.10	26.58

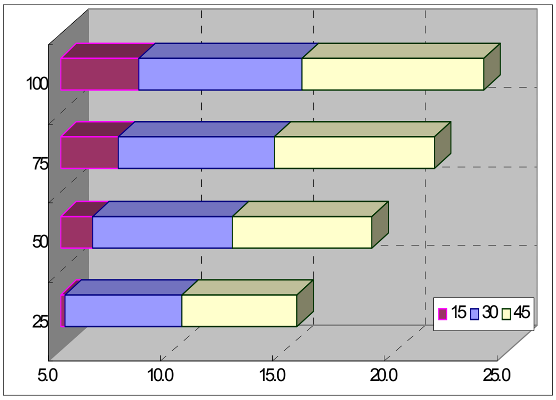

Fig. 4. Address Assigning Delay Time According to the Cluster Size

Before setting the route in a MANET, each node needs an identifiable address. Because the network topology can be changed dynamically, the MANET environment requires another access method instead of using a center focused address allocation method such as DHCP. In this thesis, cluster based address allocation protocol was suggested to solve the problem when it occurs while providing an identifiable address. Also we compared the existing methods by dividing the nodes into cluster units. In the suggested methods, compared to the MANET configuration, it took more time to assign the address if there were 25 nodes.

However, when there were 50 nodes, the suggested model showed a shorter average time of 13.93 seconds. Also in the suggested model, the cluster had the most appropriate average delay time according to the number of clusters and in the 100 nodes, 30 clusters case, 7.28 seconds was the most appropriate. When the number of nodes is more than 25 in the existing method, the address assignment time was 6.40 seconds in 50 nodes, and 18.02 seconds in 100 nodes because when the number of nodes is increased, more time is needed. However, when the number of nodes was 50 in the existing method, the address assignment time was 6.39 seconds, and 13.93 seconds in 100 nodes. It increased by 11.62 seconds in the existing method, and it increased by 7.54 seconds in 100 nodes. Therefore, we can see that the suggested model is more superior when the network size is bigger than 50, and also 30 clusters was superior to 15 or 45. Furthermore, further study is required in the evaluation of considering the failure of the cluster head to assign an address and for a safety mechanism which isn't greatly influenced by the network size.

References

[1] *Ministry of Information & Communication* “Promotion Plan of Next Generation Convergence Network”,203,7.

[2] Pradeep Parvathipuram, Gicheol Yang, Sooyeol Oh, “Evaluation and Choosing Control Algorism for Mobile Ad Hoc Network”, *Korean Information Process Society Paper collection,* Volume11-C No.6, December 2004,PP.829-834.

[3] Hye-Yeon Kwon, Jae-Wook Shin, Bok Byeong, Ji-Hyeok Choe, Sang-Woo Nam, and Seon-Bae Lim, "Mobile Ad Hoc Network Technique Trend", *Analysis of Electric Communication Trend* Volume18, No.2, April, 2003

[4] C.E Perkins, "Ad Hoc Networking" *Addison Wesley*

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

[6] 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.

[7] 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. pp. 1304 - 1311 vol.2

[8] 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.

[9] 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

[10] 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.

[11] 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

[12] The Network Simulator-ns-2, *http: //www.isi.edu/nsnam/ns/*

[13] Jae Chung, Mark Claypool, NS by Example, *http://nile.wpi.edu/NS/*

[14] Seonho Kim, Miyeon Yoon, Yongtae Shin,“Choosing Server Algorism of the Mobile Ad Hoc Network for Ubiquitous Computing”, *Korean Information Process Society Paper collection,* Volume 11-Arnjs, No. 6, Oct, 2004,pp.413-418.

[15] Hwajeong Lim, Younghuwan Cha, “ Multipath Routing Dynamic Source Protocol for the Mobile Ad Hoc Network ”, *Korean Information Process Society Paper collection,* Volume 12-Crnjs, No.1, Feb, 2005, pp.111-120.