

Distributed Mobile Agents for Reliable Cluster Management in MANETs

H. Hamad

The Islamic University of Gaza, Gaza, Palestine

Abstract—A mobile wireless ad hoc network is formed dynamically without a need for a pre-existing infrastructure. This kind of dynamic networks has been used in commercial, cultural, and environmental applications including emergency situations. Managing such networks is a challenging task. In this paper, I propose a management solution that achieves reliable cluster management in clustered ad hoc networks. The solution is based on mobile agent technology. The cluster head performs many jobs to manage the cluster. My proposed idea is to distribute the CH jobs among some nodes called job-carriers. I implement each job in a mobile agent which resides on one of the job-carriers. Election of the job-carriers is based on a periodically calculated number called Connectivity Number (CN). Each node calculates its CN by counting its neighbors that fall in the 1-hop range of the cluster. The highest-CN nodes have the chance to be job-carriers. Simulation results show accomplishing semi-static structure and enhancing the performance of the ad hoc network.

Index Terms—ad hoc networks, Connectivity Number, job-carrier, mobile agent

I. INTRODUCTION

Ad hoc network is a self-organizing multi-hop system of wireless nodes which can communicate with each other without pre-existing infrastructure. This type of networks has been used in several applications such as industrial, commercial, cultural and environmental including the emergency situations, sensor networks, communicating vehicles, etc. It is characterized by limited battery power, limited bandwidth, frequent network topology changes, and rapid mobility. Frequent topology changes result when nodes move or fail or when devices are turned on or off. These characteristics make the design of management solutions and routing protocols a great challenge [1].

Most researchers today focus on dividing the network into clusters. Conventionally, nodes are divided into three level of hierarchy: *normal* nodes, *cluster heads* (CHs), and *gateways*. A normal node, belonging to one cluster, sends data packets to the CH. The CH, acting as a cluster manager, transfers the packets to the next hop. Nodes that belong to more than one cluster are called *Gateways*. This dual membership allows the gateway to bridge the CHs and hence provides communication between clusters. A node that does not belong to any cluster is in *Undecided* state. The backbone of the network is formed by the CHs and gateway nodes as shown in Fig.1 [2].

Clustered ad hoc networks could be classified as one-hop or multi-hop. In one-hop networks a member mobile node uses single hopping to reach the CH while in multi-

hop networks a mobile node uses multi-hopping to reach the CH [3][4].

Many of the clustering algorithms and routing protocols are based on mobile agents [5] [6] [7] [8]. A mobile agent is a software agent that can migrate among nodes to perform specific tasks. Mobile agent technology can be employed in telecommunication systems, such as ad hoc networks, to provide solutions to the management problem. This technology provides several advantages over the traditional client/server one. In client/server model non-executable messages are exchanged between nodes in the network, whereas a mobile agent can carry code, data and status when it migrates to another node. This process migration allows executable code to access databases, file systems and other agents [5][9]. Mobile agents are reliable in failure cases; when the connection is down between two nodes, the mobile agent continues its execution on the host node, saves results and state, and travels to the other node when the connection becomes available again. Also, mobile agents improve bandwidth utilization, reduce network traffic, and reduce communication latency and connection time [5][10].

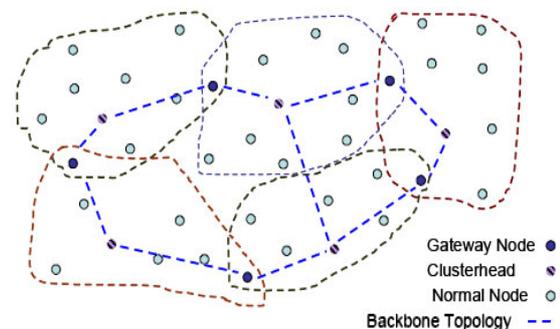


Figure 1. Clustered ad hoc Network

Management solutions for ad hoc networks have to support a distributed management mechanism that can efficiently adapt to the dynamicity of these networks. Since mobile ad hoc networks suffer from limited resources such as bandwidth and power, mobile agents can be employed in management solutions for these dynamic networks due to their robustness with limited network resources and ability to operate in a distributed manner. The CH performs many jobs to manage the cluster such as cluster formation and maintenance, route discovery, and routing. Performing these jobs guarantees the network performance while disrupting any of them may destroy the topology and causes re-clustering. This paper presents a solution for fault management in ad hoc networks based on mobile agent topology. The rest of the paper is organized as follows; section 2 provides State of

the art and section 3 defines the problem statement. Section 4 introduces the CH jobs in CBRP. Section 5 shows my proposed solution and section 6 discusses the simulation results. Finally, section 7 concludes the paper and proposes future work.

II. STATE OF THE ART

Mobile agents have been widely used in clustering and routing and slightly in network fault management. *Mobile Agents for Routing, Topology Discovery, and Automatic Network Reconfiguration in Ad-Hoc Networks* [11] proposes that each node hosts a static agent and a mobile agent. The mobile agent is responsible for many operations such as collecting information generated from static agents; updating routing tables on mobile hosts; and discovering new routes. A problem here is that all these operations will halt if the mobile agent is destroyed due to some failure. Also, implementing all the jobs in one mobile agent requires more processing but with less performance. *Mobile Agent System for Network Topology Discovery* [9] presents a system for topology discovery and management in mobile ad hoc networks in which nodes are classified as cluster head or members. The system presumes that the manager agent will perform more processing than other agents. *ANMP: Ad hoc Network Management Protocol* [12] presents a solution to the management problem that is compatible with SNMP and not based on mobile agent technology. The proposed management solution is a distributed architecture where nodes are classified into manager, CHs, and member nodes.

III. PROBLEM STATEMENT

In clustered ad hoc networks the manager node, CH, is responsible on many jobs such as maintaining the cluster, updating routing tables, and discovering new routes [11]. Mobile agents have been employed in mobile ad hoc networks due to their characteristics that provide advantages to improve the network performance. A major problem in many approaches is that all the CH jobs are implemented in one mobile agent. This results in that the manager agent, residing on the CH, will be doing more processing than other agents residing on non-CH nodes [9]. Many disadvantages arise from such problem, (1) the manager mobile agent may be loaded with too much work resulting in bottleneck and weak responsiveness to requests. (2) More processing on the CH requires intensive power exhaustion from the limited battery power and may lead to CH failure. The failure of the CH causes re-clustering, in order to select a new CH, which may cause significant topology changes and furthermore completely new structure, which adversely affects the stability of the cluster. (3) When different nodes request different services from the CH, they have to wait in serial form until they are replied. A question may arise here, why do these nodes have to wait although they are requesting different services? Why does the CH have to do all the jobs? Why does the CH have to pay the cost of processing alone?

IV. CBRP

The Cluster Based Routing Protocol (CBRP) is a routing protocol designed for the use in mobile ad hoc networks. In CBRP, the network is divided into a number of overlapping clusters [13]. Each cluster has a CH which identifies the cluster and carries many jobs. The CH

maintains the cluster membership and performs the route discovery process and the routing process. These jobs are described as follows:

A. Cluster Formation

Cluster formation process aims to form a structure that groups the member nodes into clusters. This structure helps to reduce the number of messages propagate through the network. The CH initiates the cluster formation process. A node in the Undecided state is still in search of a CH. Periodically, an undecided node sends a Hello message. When a CH receives the Hello message, it sends a triggered Hello message to join the undecided node to the cluster. The CH maintains a Neighbor Table (NT) that contains the member nodes and their roles. During cluster maintenance phase the CH removes the entries of the nodes that have left the cluster from the NT.

B. Route Discovery

Route Discovery is the process whereby a source node S wishing to find a route to a destination D in order to send a packet to this destination. This process is done by flooding. Essentially only CHs are flooded with Route Request (RREQ) packets in search for a route. The source node S broadcasts a RREQ with the target node address set to D. This RREQ is received by the neighbors including the CH. When an ordinary member node M receives the RREQ it uni-casts the RREQ to D if D is its neighbor; else M will discard the RREQ. The RREQ is flooded to the neighboring CHs through the gateway nodes. The gateway nodes send the RREQ to the CHs and so on until reaching the destination D. The route request only records the CHs it has passed in the path. When the route request arrives at D, D sends a route reply packet (RREP) to S as a reply. The actual route is calculated during the returning of the RREP. Each CH records the optimal hop-by-hop route from the previous hop to the next CH in the path.

C. Route Caching

In CBRP, when the CH learns a route it will keep the route in its own Route Cache and when a node asks for some route, the CH checks its cache before performing the route discovery process.

D. Routing

The actual routing starts from the source node. The source node sends or relays data to the CH which transfers the collected packets to the next hop.

V. 2-HOP CLUSTERING

In 2-hop clustering a mobile node may use at most one other node to reach the CH. The CH transmission range indicates the 1-hop range of the cluster while the transmission ranges of the 1-hop member nodes indicate the 2-hop range as shown in Fig. 2. 2-hop clustering has advantages over 1-hop clustering in respect of stability as we showed in our previous paper titled *2-hop clustering to accomplish semi-static structure* [14]. Here we give a brief overview about that paper and start by defining a terminology called Connectivity Number (CN).

Each node periodically broadcasts a Hello message which includes its neighbor table (NT); a table contains information about the neighbor nodes. Therefore, by

examining the NTs of its neighbor, a node is able to gather sufficient information about the network topology. A node calculates its CN by counting its 1-hop neighbors that fall in the 1-hop range of the cluster. The point here is that the CN is a measure to the closeness the cluster center. The highest the CN of a node, the closest the node to the cluster center. Fig. 3 shows node M with CN equals 8 since it has 8 neighbors fall in the 1-hop range of the cluster (in the intersection area).

A. Connectivity Number (CN)

In [14], we proposed a solution to overcome the mobility of the CH since the CH mobility causes frequent re-clustering.

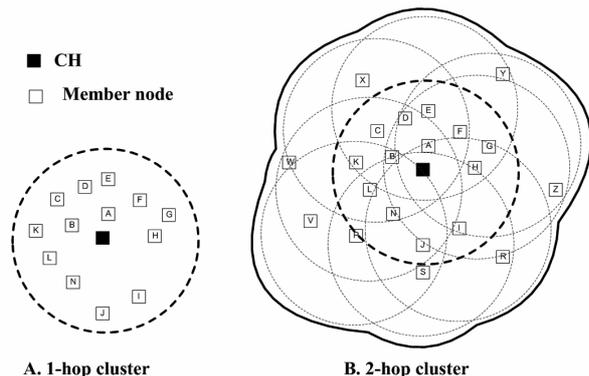


Figure 2. Clusters with respect to hopping

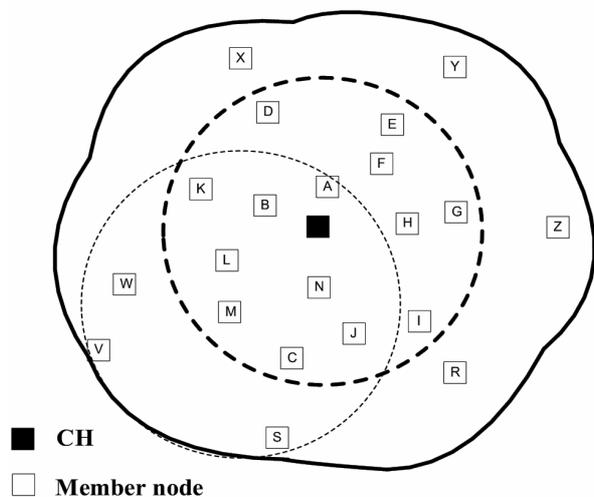


Figure 3. Node M has CN = 8

The solution introduced new identification of the cluster. We identify the cluster not only by the CH but also with the 1-hop members. That is, the 1-hop range of the cluster does not follow the CH when it moves. We try to keep the 1-hop range containing the same 1-hop members. This is accomplished by reassigning the CH according to the CN value. With every periodic Hello message each node calculates its CN and includes it in the Hello packet. Then nodes compare the CNs and the highest-CN node is assigned as the new CH. Note that the highest-CN node is the closest one to the center of the cluster. So the 1-hop range of the cluster remains nearly the same which results in semi-static structure.

VI. PROPOSED SOLUTION

We present a management solution that achieves reliable cluster management in 2-hop clustered ad hoc networks based on mobile agent technology. We believe that the reliability of a management solution in dynamic networks is related to the stability of the network structure. That is, when the network is more stable the network management is easier. Accordingly, our proposed solution is based on our previous publication *2-hop clustering to accomplish semi-static structure* [14].

In this paper the idea is quite different towards the definition of the CH. Instead of treating the CH as a physical node, we look at the CH as a number of jobs that provide services to the member nodes. The jobs are defined and implemented according to CBRP [13] protocol. In CBRP four jobs are defined: cluster formation, route discovery, route caching, and routing. Now, the CH is considered as 4 mobile agents that implement 4 jobs and hosted on 4 job-carriers.

The main idea in our solution is to distribute the CH jobs among some member nodes called job-carriers. Each job is implemented in a mobile agent which is hosted on a job-carrier node. The advantages of this idea are (1) the load on a specific node, previously the CH, is reduced since the load is distributed among 4 nodes. Reducing the load decreases the power exhaustion and so the possibility of failure decreases also. (2) When a job-carrier fails, only one job is temporarily unavailable while when the CH fails, in other algorithms, all the jobs are unavailable. (3) By employing mobile agents we overcome the mobility problem. If a job carrier moves away from the cluster, the mobile agent can migrate back to a node in the cluster and resume providing the service. (4) Distributing the jobs among 4 nodes allows member nodes to request services from these 4 nodes at the same time, i.e. up to 4 jobs can be executed in parallel.

At the start of the ad hoc network, the starting node is automatically assigned as a cluster formation job-carrier. Our assumption is that the starting node has the cluster formation mobile agent and each node is capable of creating the four mobile objects. The cluster formation mobile agent forms the cluster and creates the three mobile objects: route discovery, route caching, and routing. Each node computes its CN and includes it in the Hello message. After that, the cluster formation mobile agent sends the created mobile agents to the 3 highest-CN nodes. Now these nodes become job-carriers. Fig. 4.a shows the placement of nodes before starting the clustering process. Fig. 4.b shows node 1 forming the cluster. It also shows the CN of each node. Fig. 4.c shows the four mobile agents residing on four different mobile nodes.

The job-carriers advertise themselves through the periodic Hello messages. They specify the job in the ROLE fields of the Hello packets. So, the ordinary nodes know the job-carriers from the ROLE and ID fields. Fig 5 shows the structure of the Hello message. `my_NT` is a data structure contains information about the neighbor nodes. `my_CAT` is a table keeps information about adjacent clusters.

With each Hello message, the member nodes compute and exchange their CNs. Upon receiving the Hello messages, nodes compare the CNs and assign the

highest-CN node as the next cluster formation node. Then the cluster formation mobile agent migrates to this node. A point to notice here is that unless the cluster formation node moves, its CN will remain the highest.

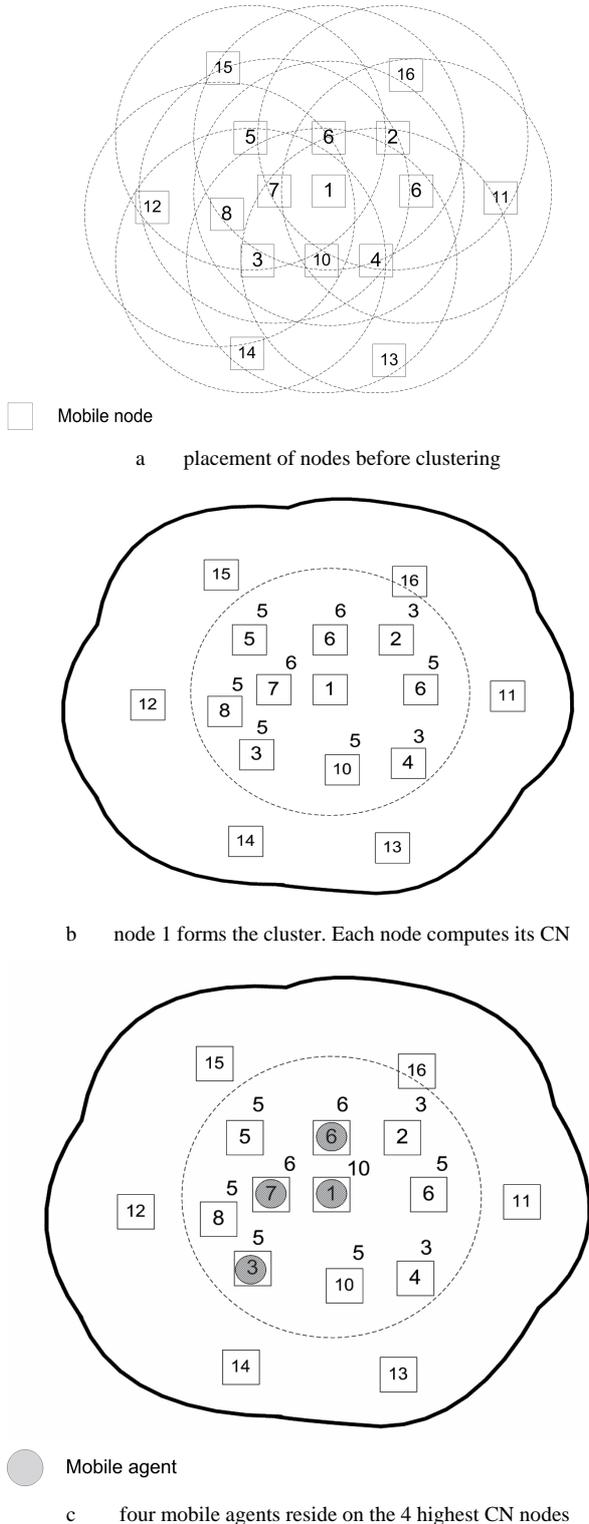


Figure 4. Illustration of forming the cluster and distrusting the mobile agents

ID	ROLE	CN
	my_NT	
	my_CAT	

Figure 5. Hello packet structure

On contrast, when this node moves away from the 1-hop range, its CN decreases which causes the cluster formation node to lose its role. Note also that reassignment of the cluster formation node occurs when it starts leaving the 1-hop range of the cluster. This does make sense because reassigning the cluster formation node achieves its goal in keeping the structure only when we are able to regain the mobile agent. If the cluster formation node left the 2-hop range, we can't regain the mobile agent.

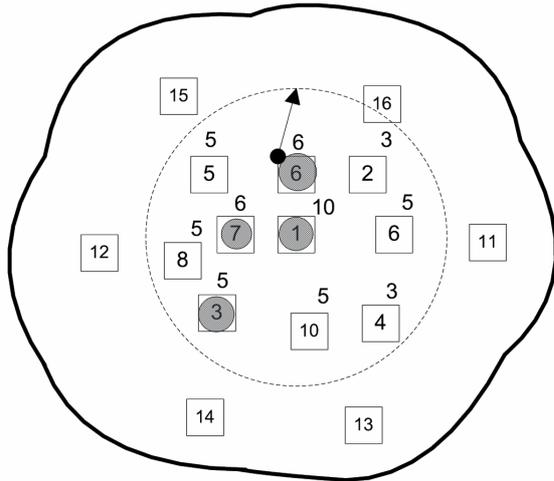
A. Fault Recovery

Mobile nodes in ad hoc networks are subjected to loss or failure. Mobility is the main factor of node loss. Failure occurs due to many reasons such as power exhaustion. Let us first take the situation in which a job-carrier j is lost. As mentioned above with each Hello period every node computes its CN and includes it in the Hello message; so, nodes can easily know the highest CN node. When j moves away from the 1-hop range j will discover, with the next Hello message, that it is leaving the cluster. This discovery is easily done when j compares its new CN with the old one and finds that the new CN is less than the old one. Upon this discovery, the mobile agent on j will migrate to the highest-CN ordinary node. Fig 6 illustrates this case. Figure 6.a shows the job-carrier 6 moving in the direction indicated by the arrow. In figure 6.b the job-carrier 6 has left the 1-hop range and it computes its CN to 1. Fig. 6.c shows the mobile agent migrating to the highest-CN ordinary node 5 assigning it as the new job-carrier. The advantage here is that the actual job carrier, which is the mobile agent, is not lost which guarantees providing the service. Also, we don't need to create a new object when a job-carrier is lost.

The other fault situation is when a job-carrier j_f fails suddenly. In this case the mobile agent hosted on j_f will be completely destroyed and the service provided by it will be down. Here, we need to create a new mobile object. Although all nodes have the capability of creating the four mobile objects, the process of creating any of these objects has to be managed. We assign this role to the job-carriers. Each of them is given a priority as in Table 1; the lowest the value the highest the priority. The responsibility on creating the new mobile objects ranges between these job-carriers according to their priorities. We assign the highest priority to the cluster formation mobile agent since it starts the cluster formation and maintains the cluster.

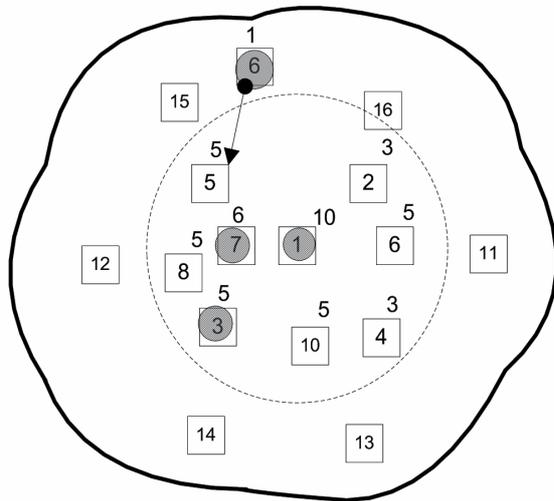
TABLE I.
MOBILE AGENT PRIORITIES

Mobile agent	priority
Cluster formation	1
Route discovery	2
Route Caching	3
Routing	4

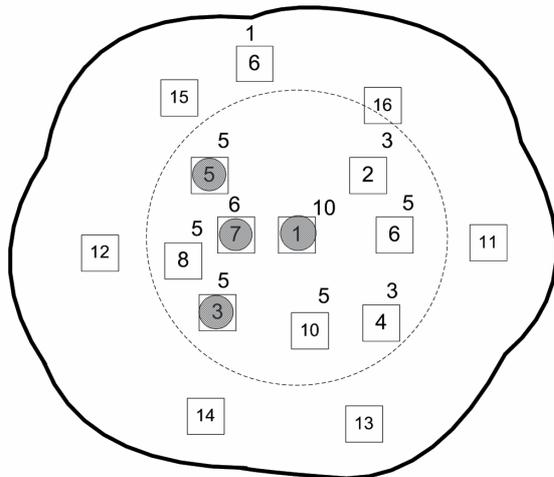


● → Motion direction

a the job-carrier 6 starts moving, its CN now equals 6



b the job-carrier 6 left the 1-hop range, its CN now equals 1



c node 5 has the highest CN so the mobile agent migrates to it

Figure 6. Reassigning the cluster formation

When a member node m requests a service from a failed job-carrier j_f , m will not receive the result and so will know that j_f has failed. Then, m will send an ERROR message to the cluster formation mobile agent to tell it about the failure of j_f . We have to be careful in that j_f may be too busy and does not respond to m 's request, so it is worth to ensure if j_f is available or not. Upon receiving the ERROR message from m , the cluster formation mobile agent calls a method *areYouAlive()* to ensure that j_f has failed or not. If j_f is really failed, the cluster formation sends a confirmation message (CONFIRM) to inform the other job-carriers that it will create the new object. The cluster formation mobile agent may receive ERROR messages about the same failure from many nodes. So, the agent sets a *prevent* flag to prevent processing the same failure. Then the cluster formation mobile agent creates a mobile object from the same class as the failed one. Fig. 7 illustrates the above steps. With the next Hello message, the new mobile object will migrate to the *free* highest CN node, which does not have a mobile agent, and will broadcast a message to advertise itself.

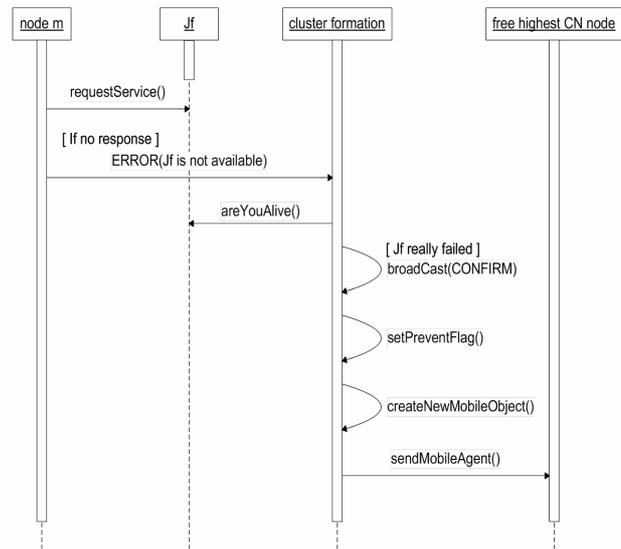


Figure 7. Failure discovery and creation of new mobile object

Let's extend the above case and assume that the cluster formation node has also failed in addition to j_f . In this case, when m sends the ERROR message to cluster formation node m will not receive the confirmation message. Then m will presume that the cluster formation node has failed and will send another ERROR message to next priority job-carrier j_n as in table 1. j_n will ensure that j_f and the cluster formation node are not available, again by calling *areYouAlive()* method. If they really failed, j_n will create the corresponding mobile agents. With the next Hello message the two mobile agents migrate to the two highest-CN nodes and advertise themselves. The

responsibility of creating the new agents ranges until reaching the lowest priority job-carrier. If all the job-carriers fail in broadcasts an ERROR message to tell the member nodes about the failure, the nodes exchange the CNs and choose the highest-CN node as a cluster formation job carrier which creates the four mobile objects and sends them to the three highest-CN nodes.

A slightly different scenario occurs when only the cluster formation node fails. The member nodes don't request services directly from the cluster formation node and so they can't discover its failure expect after the Hello period. With the next Hello period, the job-carriers don't receive a Hello message from the failed cluster formation node and so discover the failure. The Route Discovery mobile agent, in this case it is the highest priority job-carrier, creates a new cluster formation mobile object which migrates to the highest-CN node. Again, if the Route Discovery node has failed the next priority carries the responsibility on creating the mobile agent and so on.

VII. SIMULATION RESULTS

The performance of the jobs-distribution solution is evaluated via simulations using JIST-SWANs simulator [15][16]. The simulation attempts to compare between the performance of our solution, which is applied on CBRP, with CBRP [13]. Our evaluation is based on the simulation of 50 mobile nodes in 1000*1000 square meters. Random way point mobility model is used in our experiments with pause time of 30s [17]. In this model, a node travels towards a randomly selected destination in the network. After the node arrives, it pauses for the predetermined pause time and travels towards another selected destination.

The undecided time is the total time in which a node is in the Undecided state. A node that frequently leaves a cluster has large undecided time. So, the undecided time is a measure of stability of the structure. Fig. 8 shows the average undecided time of the nodes over the simulation time. I observe that the average undecided time decreased significantly in my solution compared with the original CBRP. This means that the solution achieved more stable structure.

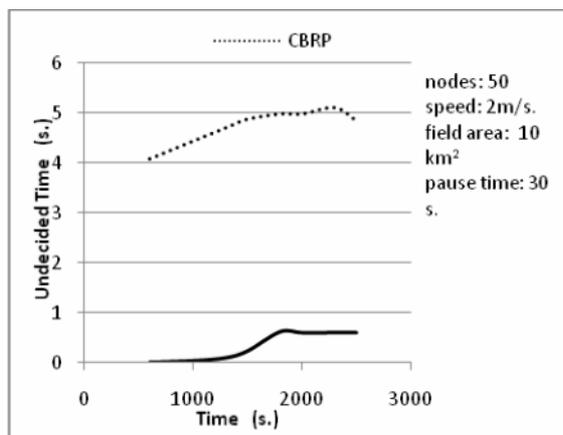


Figure 8. Undecided time over simulation time

Fig. 9 shows the Packet delivery ratio over simulation time. The figure shows that a significant improvement is achieved via our solution. The high packet delivery ratio implies that the network is more stable since fewer

packets are lost. We also observe that the packet delivery ratio saturates, to some extent, to a high value which indicates to a semi static structure.

Fig. 10 shows the packet delivery ratio vs. nodes speeds. In original CBRP the packet delivery ratio decreases significantly with high speeds while in our solution the ratio is nearly constant close to 1.

VIII. CONCLUSION

In this paper, I have proposed a management solution to accomplish stable structure and overcome faults. This solution can be applied on several MANET clustering protocols. The main objective consists respectively in ensuring the stability of the clusters and in reducing the re-clustering. We achieved this objective by implementing each CH job in a mobile agent and distributing the agents among some member nodes. The advantages of this idea are: (1) distributing the load among nodes reduces the failure possibility and hence avoids re-clustering. (2) Overcoming the nodes mobility by employing mobile agents and benefiting by their migration.(3) When a job-carrier fails, only one job is temporarily unavailable while when the CH fails, in other algorithms, all the jobs are unavailable. (4) Distributing the jobs among 4 nodes allows member nodes to request services from these 4 nodes at the same time, i.e. up to 4 jobs can be executed in parallel. Simulations showed that the proposed solution achieved stable structure by reducing the undecided time, reducing the packets loss, and overcome the faults that result from mobility.

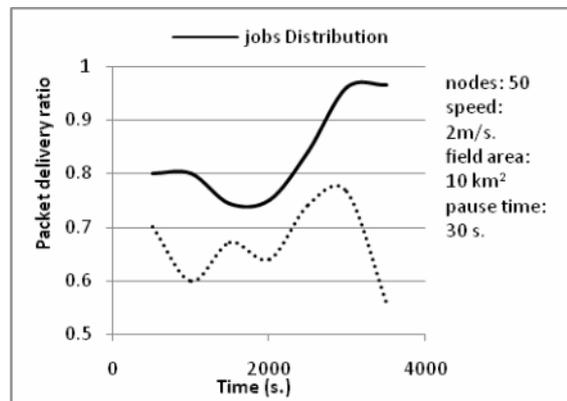


Figure 9. Packet delivery ratio over simulation time

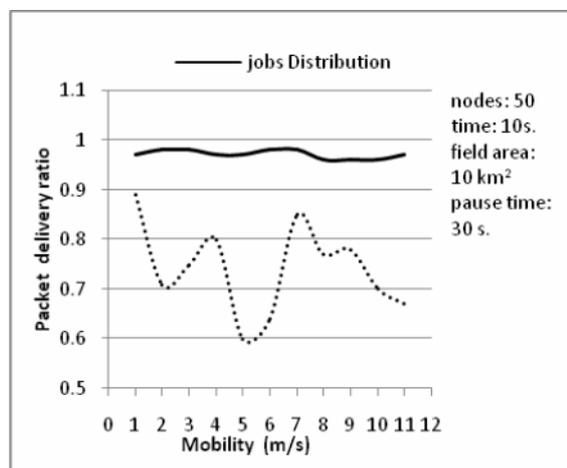


Figure 10. Packet delivery ratio vs. speed

ACKNOWLEDGMENT

The author would like to thank the research assistant Eng. Abdessalam H. Elhabbash for his assist in doing this work.

REFERENCES

- [1] Shang Y. and Cheng S. "A Stable Clustering Formation in Mobile Ad hoc Network", *IEEE*, 0-7803-9335-X/05, 2005.
- [2] D. Wei and H. Chan. "Clustering Ad Hoc Networks: Schemes and Classifications", *IEEE*, 1-4244-0626-9/06, 2006.
- [3] Fernandess Y. and Malkhi D., "K-clustering in Wireless Ad Hoc Networks", *ACM*, 1581135114/02/0010, 2002.
- [4] Mhatre V. and Rosenberg C., "Homogeneous vs Heterogeneous Clustered Sensor Networks: A Comparative Study", *IEEE*, 0-7803-8533-0/04, 2004.
- [5] DENKO M. K. "The use of Mobile Agents for Clustering in Mobile Ad Hoc Networks", *Proceedings of SAICSIT*, P: 241 – 247, 2003.
- [6] A. Ahmed and B. Far, "Topology discovery for network fault management using mobile agents in ad-hoc networks", *IEEE*, 0-7803-8886-0, 2003.
- [7] R. SUGAR AND S. IMRE S. "Adaptive Clustering Using Mobile Agents in Wireless Ad Hoc Networks". *Lecture Notes in Computer Science. Spring-Verlag*, 2001.
- [8] S. McGrath, E. Entin, R. Gray, and L. Shay., "The actcomm project: mobile agents and ad hoc routing, meeting military requirements for information superiority", *IEEE*, 0-7803-7225-5, 2001.
- [9] Ahmed A. and Far B., "Mobile agent system for network topology discovery", *IEEE CECE/CCGEI*, 1-4244-0038-4, 2006.
- [10] Gray R., et al, "Mobile-Agent versus Client/Server Performance: Scalability in an Information-Retrieval Task", *Proceedings of International Conference Mobile Agents (MA'2001), LNCS vol.2240*, pp.198-212, Springer, December 2001.
- [11] N. Migas, W. Buchanan, K. McCartney, "Mobile Agents for Routing, Topology Discovery, and Automatic Network Reconfiguration in Ad-Hoc Networks", *10th IEEE International Conference on Engineering of Computer-Based Systems (ECBS 2003)*, pp. 200-206, 2003.
- [12] Wenli C., Nitin J., and Suresh S. "ANMP: Ad Hoc Network Management Protocol," *IEEE Journal on Selected Areas in Communications*, Vol. 17, NO. 8, pp 1506 -1531, August 1999.
- [13] Azzedine Boukerche., "A Simulation Based Study of On-Demand Routing Protocols for Ad hoc Wireless Networks", *IEEE*, 0-7695-1092-2/0, 2001.
- [14] Hamad H., "2- hop clustering to accomplish semi-static structure", unpublished, available online: www.iugaza.edu/users/hhamad/recent.html
- [15] Barr R., March, "JiST- Java in Simulation Time User Guide", Department of Computer Science, Cornell University, Ithaca, NY 14853, 2004.
- [16] Barr R., "JiST SWANS- Scalable Wireless Ad hoc Network Simulator User Guide", Department of Computer Science, Cornell University, Ithaca, NY 14853, 2004.
- [17] W. Navidi and T. Camp. "Stationary distributions for the random waypoint mobility model". *IEEE Transactions on Mobile Computing*, 3(1), 2004.

AUTHOR

H. Hamad is with the Electrical and Computer Engineering Department, The Islamic University of Gaza, Palestine, (e-mail: hhamad@iugaza.edu.ps).

Manuscript received 1st Mars 2008. Published as submitted by the author.