Energy and Path Aware Clustering Algorithm (EPAC) for Mobile Ad Hoc Networks

Waqar Asif and Saad Qaisar, *Member(IEEE)*

NUST School of Electrical Engineering and Computer Science,
H-12 Islamabad, Pakistan
{09mscsewasif,saad.qaisar}@seecs.edu.pk

Abstract. Node clustering is a technique that mitigates the change in topology in Ad hoc communication. It stabilizes the end to end communication path and maximizes the path life time. In SWARM communication, each cluster is assigned an objective and expected to complete it in the available resources. Most of the algorithms previously designed assume that the assignment of tasks can be done in any arbitrary manner and does not depend on the energy resources. In this work, we have emphasized that the number of nodes in a cluster is fundamentally related to the energy requirement of the objective. With the help of this new algorithm, we minimize energy consumption in a cluster by improving the mechanism for selecting objective, depending upon the amount of energy present at the nodes of that cluster.

Keywords: SWARM, Pareto Optimality, Cluster Head, Optimality

1 Introduction

Recent years have witnessed increased interest in the field of SWARM based cooperative communication for establishment and coordination in a Mobile Ad Hoc network. Swarm Communication plays an important role in creating a communication bridge between different independent devices working in a decentralized approach. Most of the current applications of SWARM based communication e.g. aerial swarm deployment [14] and disaster mitigation [13] require minimal energy wastage to maximize life time of the SWARM. Thus, while developing protocols for SWARM Communication, the prime objective is to reduce the energy consumption [6] and improve packet delivery ratio [2].

Clustering provides the flexibility in dividing tasks among multiple nodes to mitigate energy wastage and also to help in formulating stable routing mechanisms eventually resulting in minimal packet loss. A cluster in general constitutes of two types of nodes, a cluster head that forms the backbone of the network and is responsible for controlling a cluster and non-cluster head nodes that follow the instructions of a cluster head. Cluster heads negotiate among each other for a stable and reliable inter cluster routing path, hence whenever, there is a

change in cluster head the inter cluster routing path has to be reconfigured, resulting in wastage of energy resources.This change in cluster head is known as re-clustering. Usually, it is fashionable for re-clustering to be performed when the present cluster head lacks the energy requirement for controlling and monitoring the cluster. The frequency of re clustering can be reduced by efficiently selecting the cluster size as the consumption of energy and the size of cluster are directly proportional. For a large sized cluster, the distance between two neighboring clusters will reduce but distance between nodes inside the cluster will increase resulting in extra energy consumption for communicating within the cluster. On the other hand, if size of the cluster is kept smaller, amount of energy required for communicating within the cluster will reduce but the amount of energy required for communicating between any two clusters will increase. Hence, to have a stable cluster, energy consumption of the cluster for inter-cluster and intra-cluster routing should be considered [16].

Along with cluster stability, efficient resource consumption is also an important parameter for SWARM based networks. For a network with multiple objectives and each of them having a different priority, it is of high concern which objective should the cluster execute first and how many cluster should there be on a particular instance. Else, there is an increased high likelihood that a cluster will die out on its way of trying to complete a task that has a lower priority. In order to avoid such a scenario, various techniques and optimization algorithms have been proposed [9][8] . These algorithms consider priority of objectives and depending upon the geographic location and velocity of the nodes, select which objectives should be executed first. These algorithms are mostly build upon the basic Particle Swarm Optimization (PSO)[4] technique. They help in finding the optimal objective but do not provide a mechanism as to how these objectives should be achieved in an energy and time efficient manner. To solve this problem, we propose a new framework that focuses on improving energy efficiency and stability of the cluster by forming clusters based on the energy requirement per cluster.

The paper is further divided as: Section II explains some related work in this field, Section III defines the problem statement, Section IV gives the simulation and analysis of the algorithm and Section V gives the conclusion.

2   Related Work

For making a stable cluster, a clustering scheme should be devised that forms clusters based on some stability constrained mechanism. In [5], mobility of nodes is considered as cluster creation mechanism. When two cluster heads come close to each other, the node having the lower relative mobility and who is also close to all the cluster members becomes the new cluster head. Likewise in [7], nodes form clusters based on velocity constraint. Nodes having similar velocity form up a single cluster. Both these techniques lack in stability in a way that they do not select a stable cluster head during the initial cluster head selection procedure hence this results in frequent need for re-clustering and hence wastage of

resources.

In [1], node mobility is considered as the constraint for the cluster head selection. In this algorithm every node monitors its relative speed with its neighbor and selects a node that has the lowest speed to be its head. Likewise, in [4] speed entropy is used to accumulate the nodes that are in close proximity to each other. Among these nodes, the node having the maximum energy is selected as the cluster head. These algorithms improve cluster stability but face convergence issues i.e., as relative speed measured on a specific node has local significance, hence different nodes have different measurements. This hinders the algorithm from reaching a unanimous decision.

The Size-bounded Multi-hop Clustering (SMC) [15] considers the local topology information. The node elects a cluster head depending on a relative weight value. This weight value decreases as the number of hops from the cluster head increase. This surely restricts the size of the cluster but does not guarantee stability and does not cater for unstable nodes.

The Affinity propagations for vehicular network (APROVE) [12] uses the Affinity Propagation Algorithm to form cluster of nodes having minimum distance and minimum difference of speed. Likewise the Mobility and Energy Aware Clustering Algorithm (MEACA) [16] consider node's mobility and energy into consideration for selecting the cluster head. In this algorithm, the energy and mobility factor of each node is advertised to its immediate neighbor. Nodes receive the advertised values and form a table on their basis. Node having most energy and least mobility is selected as the cluster head. Both these algorithms improve stability of the cluster but they ignore the importance of priority of the objectives and the energy required by each objective.

Multi objective Particle Swarm optimization (MOPSO) and Game Strategies Multi-Objective Particle Swarm Optimization (GMOPSO) both include the importance of the optimality of an objective. In these algorithms the objective that is close to the current position of the node becomes the optimal objective of that node [8]. This also helps giving high priority to a specific node and low priority to some other node, but these algorithms lack in determining the energy requirement for accomplishing those objectives.

Some of these algorithms improve stability but to the best of our knowledge algorithms that assure the completion of objectives with minimal energy consumption are almost non existing. The existing literature mainly concentrates on stability of the cluster for maximum time. A specific cluster may totally run out of power before performing a particular objective because the cluster formation algorithm did not consider this parameter.

3   Problem Statement

In SWARM based Ad Hoc Network groups of nodes put in a joint effort in completing a set of tasks. These nodes have limited energy and time resources and are required to complete these tasks in the available resources [14]. The geographic location of tasks can be either predefined or dynamically set. For the

scope of this work we consider that the location is fed in the nodes before they are deployed into the test area. Among these tasks, the nodes dynamically select the sequence in which the tasks have to be completed. To have a stable and organized movement towards these tasks, these nodes make up clusters. Each cluster then selects its own head. The cluster head changes over time depending upon the circumstances defined, in our case being the energy constraints. Different algorithms are used to introduce fairness in the selection of task per cluster, e.g. Pareto Optimality, Multi Object particle Swarm optimization etc. These algorithms help all the nodes decide which path to move on , depending upon the geographically closest objectives location, relative to all the nodes in that cluster. Various algorithms have been proposed that consider selection of path and which also design special metrics for forming cluster, but to the best of our knowledge none of them have considered the relation between the total energy of the cluster and the energy required for completing the task that cluster has selected as its prime objective. When a cluster inefficiently selects a tasks which it cannot complete with the available energy then the cluster will perish on the way towards that task, resulting in wastage of energy. To avoid this problem an algorithm should be devised that considers the relation between the energy of the cluster and the energy required to complete the task.

4   EPAC

In this section we present the details of the proposed clustering algorithm called EPAC(Energy and Path Aware Clustering). In our algorithm a node in a cluster can be in one of the five possible states: Initial (I), Cluster Head (CH) , Gateway Node (GN), Initial Cluster Head (ICH) and Ordinary Node (ON) .An I state is when the node has not heard from any of its neighbors and has yet to become a part of any cluster.A CH is at the center of the cluster and it communicates with all the nodes inside the cluster. The GN communicates with multiple cluster heads of different cluster and also relays messages coming from one source cluster to another destination cluster. A ICH is a node that becomes the cluster head at the start of the algorithm and calculates all the paths. An ON is a node inside a cluster and is not the GN or the CH and only communicates with the cluster head of the cluster which it is a part of.

CHs and GNs both forward messages hence this leads to more energy consumption of these nodes compared to ONs. We use path based clustering in order to reduce the energy consumption by forming intelligent clusters to retain the sameCHs and GNs for a longer time period. Our clustering technique works in two phases. In the first phase a single cluster is formed from all the nodes that are present in a close by location and the head of this cluster calculates all the possible paths and then forms sub-clusters based on the number of objectives and the amount of energy present in the cluster. This phase is called the Initialization phase. In the second phase clusters are formed based on the path that cluster has been assigned and the number of nodes that will be required. This
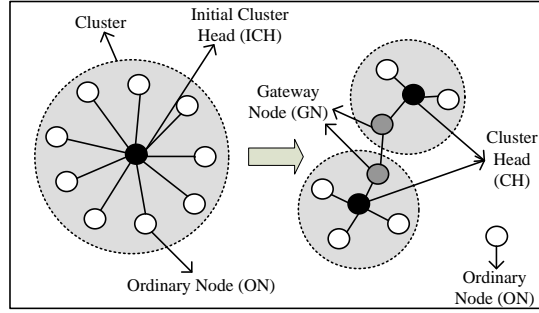
Fig. 1. Cluster Name Representation

phase is called the Clustering phase.

4.1 Initialization Phase

At the start of the algorithm all the nodes are in the initial phase. These nodes broadcast a Hello message based on the CSMA/CA scheme. These Hello messages contain node id and the remaining energy E_r[10]. Node that has a locally lowest ID, broadcasts a CHR. On receiving the CHR message, all the nodes with the highest ID become the ONs. Any node that finds itself as having a lower ID will also broadcast a CHR with its id. If the nodes sending the CHR message receives another CHR then it compares its own ID with the ID of the message received. If it finds its own id to be less, it becomes an ON. If the node sending the CHR message does not receive any other CHR message for two broadcasting time intervals, it sets itself as the ICH. Once the ICH has been decided then the ICH starts calculating the possible paths depending on the amount of energy available in the form of number of nodes.

4.2 Path Computation Phase

The objective of this phase is to have maximum paths covered within the amount of energy present with all the nodes in the network. Among all the set of paths that can be covered in the same amount of energy, the set carrying the most parallel paths is the most preferable set. This is because the more the parallel paths the less will be the average amount of time required per task.
We model this scenario using graph theory. Consider a graph represented by G=G(V,E), where, tasks are represented by the vertices V and the paths, represented by edges E.

The graph shown in Fig 2 is a completely connected graph of the six given tasks. The priority of the nodes is represented by weights of the tasks(higher
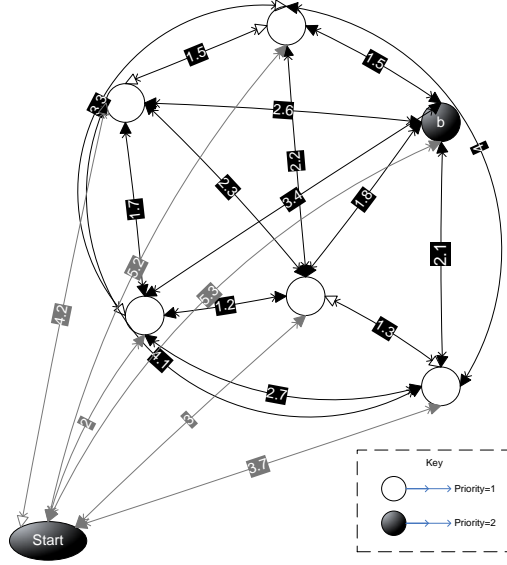
Fig. 2. A Graph Completely Connecting Six Target Locations

weights represent higher priority and lower weights represent lower priorities) and the edges represent the distances. Algorithm begins with ideal case of considering direct paths from starting point to all the tasks individually. Sum of these distances will provide energy requirement for completing these tasks in parallel. Energy required for executing task i located at distance d is given by:

$$E_{t,i} = (\frac{4kd(nT_x + (n-1)R_x) + Comp + nE_d}{3}) \qquad (1)$$

Here T_x represents the energy required for transmitting the signal from the non cluster head nodes to the cluster head and R_x represents the energy required for receiving the signal transmitted by the non cluster head nodes to the cluster head, Comp represents the energy required for the clustering process, E_d represents the energy required by each node for traveling a distance d and n represents the number of nodes. The sum of the energies required for completing all the paths is represented by $E_{t,t}$. If we assume that all the nodes contain an equal amount of energy then the total energy of all the nodes is represented by $E_{n,t}$. After the calculation of paths, we can have three different scenarios discussed below.

1. $E_{n,t} = E_{t,t}$ In the case of equality we prefer using the parallel paths for completing the tasks in the minimum time possible. Fig 3a shows the direct paths to all the tasks from the original start point where the initialization phase took place.
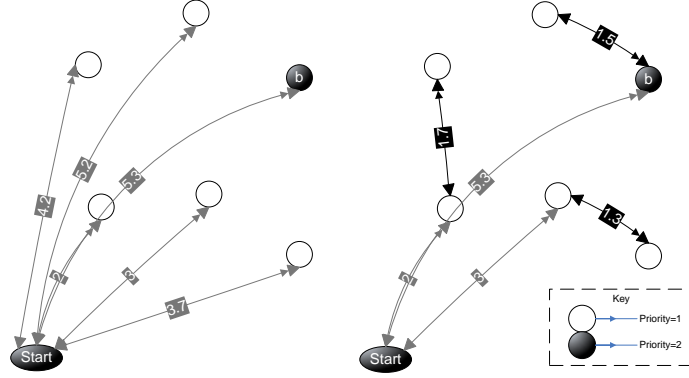
Fig. 3. a)Energy of Nodes = Energy of Paths, b)Energy of Nodes < Energy of Tasks

2. $E_{n,t} < E_{t,t}$ In this case, total energy of nodes is less than the energy required for completing tasks in parallel. In such a scenario, difference of the distances is calculated and the paths having the minimum priority and maximum weights are removed. These weights are added togather and compared with the difference between $E_{n,t}$ and $E_{t,t}$. When this weight increases the size of the difference then the tasks whose paths were removed are reconnected to the nearest task and the weights with the new connections are computed again. This algorithm repeats itself till the time.

$$E_{n,t} > E_{t,t}$$

Fig 3b shows that depending on energy of the nodes there are three parallel paths. Objectives that are geographically close to each other are laid on the same path and objectives that are away from each other are laid on a separate paths.

3. $E_{n,t} > E_{t,t}$ In this case the total energy of the nodes is larger than the energy required for completing the tasks in parallel. In such a scenario, since we have extra energy with the nodes hence we use it by placing extra nodes on a new self generated path. This path will be between two paths leading to two different tasks. The purpose of these specially designed paths will be to reduce energy required by one *GN* to transmit data to another cluster, by placing nodes that act as relay devices only. These nodes will help multi-hop communication between them and save energy of the gateway nodes present inside each cluster. As energy will increase, these relay paths will also increase till the time all the clusters are at one hop distance to each other.Fig 4 shows that there are 6 parallel paths on which the clusters will travel and the arrows in between the paths show the path where the relay nodes will travel. These paths as shown in the figure are in the center of the parallel paths.
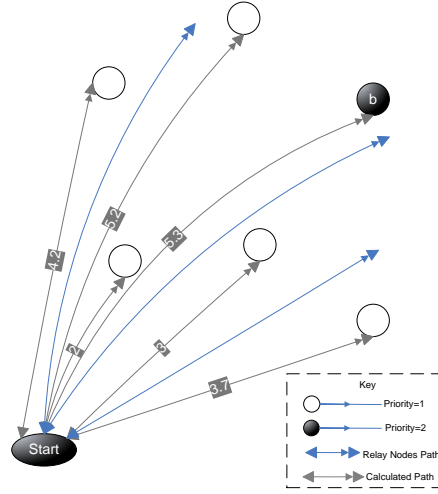
Fig. 4. Energy of Nodes > Energy of tasks

Cost function for the energy of a path The cost function of path energy is dependent on the energy consumed by initial clustering $Comp$ and the energy required for the transmission and reception of the data signals T_x and R_x. The $Comp$ function is defined by:

$$Comp = aE_cu + (N - a)E_{no-cu} \tag{2}$$

Here E_{cu} represents the energy consumed by the cluster head in forming a cluster and E_{no-cu} is the energy consumed by the non cluster head nodes in forming a cluster.

$$E_cu = b_l(2E_{(}elec) + E_{(}mp)M^4 + E_{(}fs)d^2) \tag{3}$$

$$E_{no-cu} = b_l((a + 2)E_{(}elec) + E_{(}fs)d^2) \tag{4}$$

Here E_{elec} represents the energy required per bit and E_{fs} represents the energy per bit per unit distance. Number of bits in a packet are represented by b_l. E_{elec} and E_{fs} depends upon the environmental conditions in which the system is operating.

4.3   Clustering

Once paths have been calculated ICH starts forming sub-clusters. Number of these sub-clusters depends upon the number of parallel paths calculated. ICH uses available data to declare the authorities of a CH to a node having the highest energy. The new CH transmits a signal having the parameters $(id, E_r, P_i(d))$. This message initiates a three way handshake. The nodes receiving this message reply with a *join* message and then the cluster head replies with an accept

message along with $(id, E_r, P_i(d))$. Whenever a new node joins the value of $P_i(d)$ is reduced. This informs the surroundings that a new node has entered the cluster. When the cluster has fulfilled its needs then the value of $P_i(d)$ is set to zero showing that there is no more need of any new node. To ensure fair division of nodes among all the selected paths Ant Colony Optimization algorithm is used.

4.4   Ant Colony Optimization

In ant colony optimization (ACO), an artificial colony of ants is used to construct solutions guided by the heuristic information and the pheromone trails. ACO is inspired by the foraging behavior of the real ants. Initially, ants explore surrounding area of the nest in a random manner [3]. When they find a food source, they evaluate quality and quantity of the food and carry it back to the nest. On the way back they drop a pheromone trail on the ground whose quantity and quality guides the other ants towards the food source. This behavior of ants helps them find the shortest path to the food sources from their nests. If the $n_i(d)$ is the number of nodes on path i at distance d then the total number of nodes is

$$m = \sum_{i=1}^{j} n_i(d) \tag{5}$$

Now the intensity of trail represents number of nodes that are acting as cluster members. As number of non cluster node increases and energy level reaches close to energy required for the path, intensity of the trail also reduces. Intensity of the trail is represented by

$$\tau(d+1) = \tau(d) - \delta\tau(d, d+1) \tag{6}$$

Here $\tau(d)$ represents the maximum energy required for that particular path and $\delta\tau(d, d+1) = \sum_{k=1}^{m} \delta^k(d, d+1)$ represents sum of energies of nodes that have already joined the cluster. As number of nodes inside the cluster increase, the requirement of energy decreases hence the intensity of that cluster decreases. The *visibility* in ant colony optimization is the inverse of the distance of that particular path. The visibility $\eta_i = 1/d_i$. Now the number of nodes that a particular cluster will have after a distance d will be

$$p_i(d) = \frac{\tau_i(d)^\alpha \eta_i(d)^\beta}{\sum_{j=1}^{m} [\tau_j(d)]^\alpha [\eta_j(d)]^\beta} \tag{7}$$

4.5   Choosing the Gateway node

Gateway node selection also depends upon the selection of the path. A node will only become the gateway node if it has the sufficient energy and the suitable location. As all the cluster heads are aware of the paths that were originally
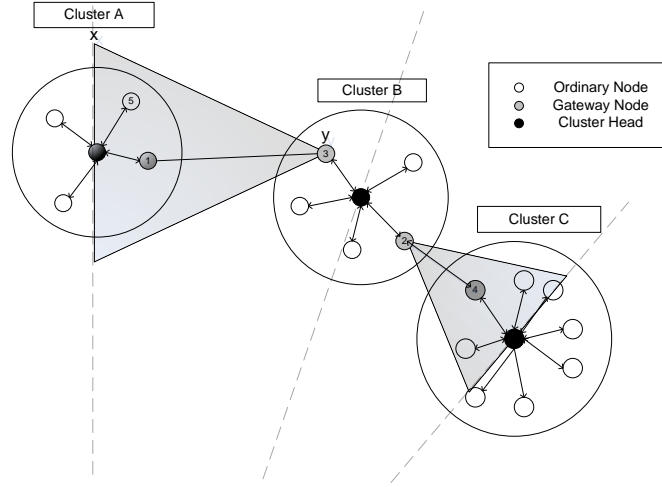
Fig. 5. Gateway Power Approximation

calculated by the ICH, hence considering the path and the average speed of the nodes the cluster head can calculate the expected location of the neighboring cluster. This calculation helps select the node that is best suitable for sending the data in that direction. Hence the gateway node selection will also vary over time. The amount of power required to send data to the neighboring node will also be calculated based on the location.

Fig 5 shows that the CH of Cluster B expects the Cluster A to be inside the left sided shaded region.If the distance between point y in cluster B and point x in cluster A is d, power required by the GN of cluster B to convey the information to cluster A in free space will be [11].

$$P_t = \frac{S_r(4\pi)^2 d^2}{G_r G_t \lambda^2} \tag{8}$$

Here S_r is the receiver sensitivity, G_r and G_t are the receiver and transmitter gain respectively and λ is the wavelength the system is working on. Suppose the transmission time is T then the sending energy will be $W_t = P_t T$. Equation 8 can be re-written as.

$$W_t = S_r T d^2 k, k = \frac{(4\pi)^2}{G_r G_t \lambda^2} \tag{9}$$

Now in this case cluster B will require W_t amount of energy to transmit data to cluster A. Likewise, as shown in Fig 5, the distance between cluster B and cluster C is less, hence, it will require less energy to transmit data from cluster B to cluster A. This adaptive mechanism will reduce the energy consumption

of GN as GN will not be required to keep a constant connection with all the neighboring clusters.

5 Simulation Results

We conducted simulation experiments to provide a performance comparison of our framework with APROVE [12] and MEACA [16]. We take a test area of $1000x1000m^2$ and consider a random uniform distribution of 50 tasks over the whole test area. Initially, it is assumed that each mobile node contains 10 joules of energy. For averaging results we have taken 10^7 iterations for each task. The number of mobile nodes varies from 5-200.
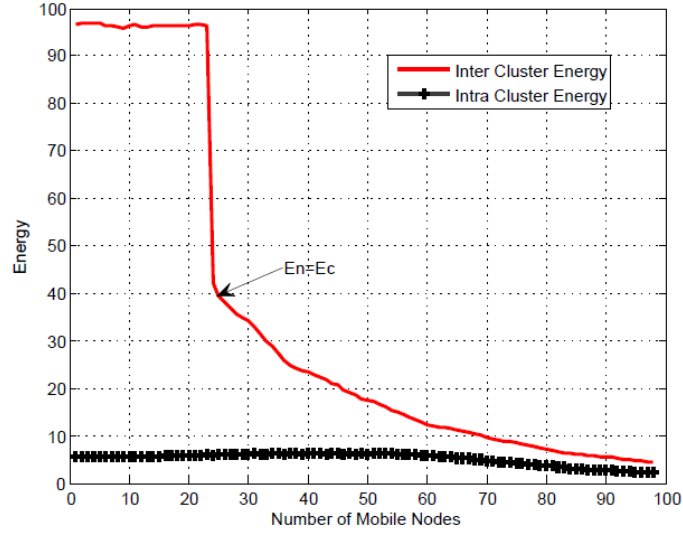

Fig. 6. Energy requirement for Inter-Cluster and Intra-cluster Communication

Fig 6 shows variation in energy required for inter-cluster and intra-cluster communication, based on the amount of resources present. The intra-cluster communication remains the same for different number of mobile nodes. This is due to the fact that all nodes inside a cluster are at a one hop distance. Thus, nodes can communicate while using a low amount of energy. On the other hand, the energy required for the inter-cluster communication varies with the amount of mobile nodes present. Initially, as the total energy of the nodes is less than amount of energy required for covering the paths in parallel, hence different paths are formed each having multiple tasks appended together. Hence, more amount of energy is required for inter-cluster communication. At a point when

number of mobile nodes are 25, energy required for completing paths in parallel
is equal to total energy of the nodes . After this point, the amount of available
energy increases the required amount of energy, hence relay paths similar to
those shown in Fig 3b are deployed. This causes the graph to decrease after the
point where the amount of mobile number of nodes equal 25. The amount of
relay paths increase as the number of extra nodes (that are not required by the
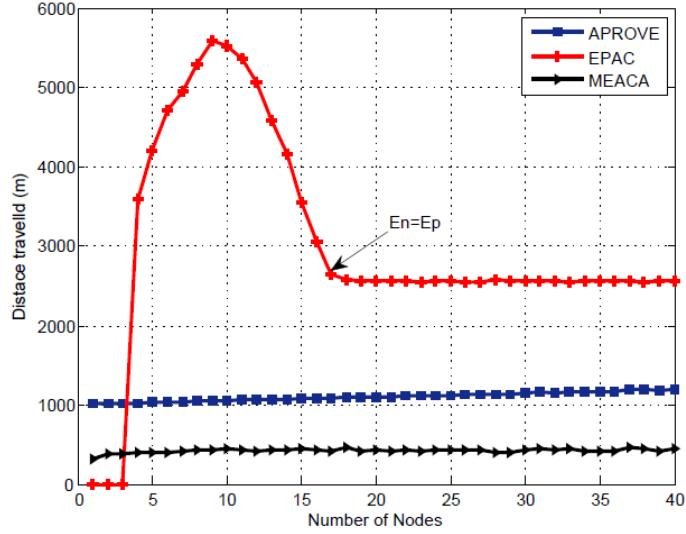cluster to complete the task) increases.

Fig. 7. Average maximum distance covered by different protocols

Fig 7 shows the difference in the average distance covered by each node under
different algorithms. In EPAC, average distance covered per node increases for
an increase in number of nodes from 0 to 12 because at this time the amount of
energy required at the nodes is not sufficient to handle all the tasks in parallel.
The tasks are executed in serial manner as shown in Fig 3b. As the number of
nodes increases from 12 to 17, the average distance covered per node decreases
as the amount of energy available at the nodes becomes sufficient enough to
handle more and more tasks in parallel. After the number of nodes crosses 17,
the average distance covered per node becomes constant, since, from this point
onwards the E_n exceeds E_p and all the extra nodes start following the self gen-
erated paths as shown in Fig 4. On the other hand, when APROVE is executed,
covered distance increases linearly as the number of nodes increase, but this in-
crease is of a very low gradient. This is because APROVE does not consider the
relation between amount of energy required for completing a task and amount

of available energy. Likewise MEACA shows a linear behavior, the amount of distance covered remains same for any amount of nodes present.
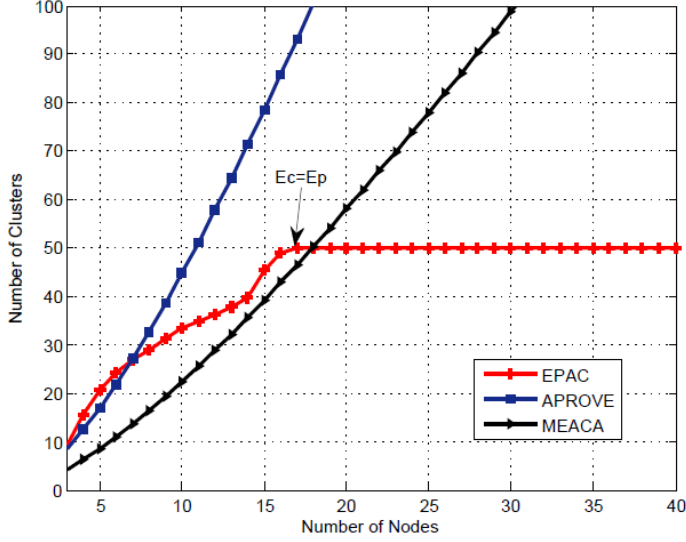

Fig. 8. Average maximum amount of clusters required

Fig 9 shows the difference in the average maximum amount of clusters required to complete 50 tasks. In EPAC, number of clusters formed for less than 12 nodes increases as E_c is less than E_p. Thus, after performing each task, re-clustering is required. This results in increase in number of clusters. After the point when E_c is equal to E_p, number of clusters become stable as all tasks here onwards are completed with similar patterns. On the other hand, number of clusters in APROVE and MEACA increase exponentially. The reason behind such a behavior is that when a new node is added into a network operating under APROVE, all the nodes with the same velocity and in a close proximity to each other join the same cluster resulting in excessive energy utilization of the cluster head for handling the cluster hence provoking extra re-clustering. Likewise the amount of clusters required to complete maximum tasks in MEACA increases as new nodes are entered into the network. Due to the addition of the new nodes size of a cluster increases as more nodes come into close proximity of each other, resulting in spending extra energy in controlling the cluster.

These results show that when the number of clusters is made dependent on number of objectives provided, we may have more efficient use of resources. The proposed EPAC algorithm gives us a more stable cluster by judicious use of energy for inter-cluster communication and undertaking clustering based on available energy. This algorithm will reduce the chance of wastage of energy by

not selecting a task that has less priority or a task for which it does not have sufficient energy. This algorithm will also remove the chance for unnecessary network partitioning due to re-clustering since once a cluster has started moving towards a specific task then for re-clustering, a new cluster head will be selected from the nodes that are moving towards the same destination point. The new cluster head will not have to decide a new destination, hence network partitioning is intelligently avoided due to re-clustering. This algorithm also support parallel execution of tasks, hence it reduces the time consumed for the completion of the tasks.

6   Conclusion

In this paper a cluster head selection algorithm based on the amount of energy required per objective is proposed for mobile ad hoc networks. It produces energy efficient and stable clusters that cover more tasks in parallel to reduce the time required to complete all the tasks. It also reduces the chance of network partitioning due to re-clustering or node failure by forming clusters strong enough to complete a task on its own. Clusters are only given tasks that they can handle, this reduces the chance of re-clustering. Hence by reducing the energy wastage, the clusters formed are more stable compared to the previously developed protocols.

References

1. B. An and S. Papavassiliou. A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks. *International Journal of Network Management*, 11(6):387–395, 2001.
2. C.C. Chiang, H.K. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. *Proc./E SlCON*, 97:197–21, 1997.
3. M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.
4. B. Gu and X. Hong. Mobility identification and clustering in sparse mobile networks. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1–7. IEEE, 2010.
5. Y. Gunter, B. Wiegel, and H.P. Grossmann. Cluster-based medium access scheme for vanets. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 343–348. IEEE, 2007.
6. V. Kawadia and PR Kumar. Power control and clustering in ad hoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 459–469. IEEE, 2003.
7. O. Kayis and T. Acarman. Clustering formation for inter-vehicle communication. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 636–641. IEEE, 2007.
8. Z. Li, S. Liu, D. Xiao, J. Chen, and K. Li. Multi-objective particle swarm optimization algorithm based on game strategies. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pages 287–294. ACM, 2009.

9. S.K. Mishra, G. Panda, and S. Meher. Multi-objective particle swarm optimization approach to portfolio optimization. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 1612–1615. IEEE, 2010.
10. M. Nasim and S. Qaisar. Hierarchical MIMO: A clustering approach for Ad hoc wireless sensor networks. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1–6. IEEE, 2010.
11. T.S. Rappaport. *Wireless communications.* Prentice Hall PTR New Jersey, 2002.
12. C. Shea, B. Hassanabadi, and S. Valaee. Mobility-based clustering in VANETs using affinity propagation. In *Global Telecommunications Conference, 2009. GLOBE-COM 2009. IEEE*, pages 1–6. IEEE, 2010.
13. Timothy Stirling and Dario Floreano. Energy Efficient Swarm Deployment for Search in Unknown Environments. In Marco Dorigo, editor, *Proceedings of the 7th International Conference on Swarm Intelligence (ANTS 2010)*, volume 6234 of *Lecture Notes in COmputer Science (LNCS)*, pages 562–563, Berlin, 2010. Springer-Verlag.
14. Timothy Stirling and Dario Floreano. Energy-Time Efficiency in Aerial Swarm Deployment. In *Proceedings of the 10th International Symposium on Distributed Autonomous Robotics Systems.*, 2010.
15. Y. Wang and M. Medidi. A Distributed Size-bounded Multi-hop Clustering for Mobile Ad Hoc Networks. 2006.
16. Y. Wu and W. Wang. MEACA: Mobility and Energy Aware Clustering Algorithm for Constructing Stable MANETs. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7. IEEE, 2007.