Ant Colony Optimization Based Energy Efficient Routing Algorithms For Routing In Mobile Ad Hoc Networks

Deepika Dhawan, Rajeshwar Singh

Abstract: Routing protocols in Mobile Ad-hoc Networks (MANETs) has yielded optimistic results for a long time, but the conflicts begin when we start to focus on particular parameters of the algorithms, like packet delivery ratio, end-to-end delay, throughput, energy consumptions, etc. These factors are very crucial in an algorithm as these are the building blocks of the optimal solution. For example, if an algorithm has a satisfactory packet delivery ratio but the energy used/consumed by the nodes of MANETs is such high that is it not feasible to implement or beneficial to implement in a real-time issue, then the algorithm would not be a practical solution to the efficient routing problem. Ant colony optimization is a heuristic which has so far yielded results that are satisfactory compared to other nature-inspired heuristics. In this paper, we propose Ant Colony Optimization – Energy Efficient Routing Algorithm (ACO-EERA), an algorithm which has produced significantly good results in comparison with other algorithms. The algorithm implements a function which chooses less the nodes with low energy remaining and it reduces the loss of energy of packets being dropped. At the end of the research paper, we also compare our proposed algorithm with Ad-Hoc On-demand Distance Vector (AODV), for the factors such as Packet delivery ratio, End to End delay and total Energy consumption.

Index Terms: ACO, MANETs, Energy Efficient, AODV, ACO-EERA, Packet Delivery Ratio, End-to-End Delay.

---

1 INTRODUCTION

MOBILE Ad-Hoc Networks (MANETs) are a special kind of network as they are infrastructure-less networks. All the devices or nodes in the network possess mobility which makes the topology of the network dynamic. Therefore the devices in the network can move freely i.e. they can be connected to one network and suppose a node is moving away from the network and discover that another network is present near its new arrived position, so if it will be in the discovery area, it will be then connected to the new network. Mobility of the nodes in a network also raises a number of problems for routing algorithms. As the nodes are mobile we do not have a proper definite way of reaching a destination node as it is likely that it is also moving and it could be the case that by the time we need to send information to the destination node it has changed its position and is no longer available at its previous position and is out of the reach of the current node. And in best case scenario it could also be that may be the destination node has come next to the source node which, while looking for the path to send the data to the destination node finds that destination node is next to it, and simply send the data packet to the destination node without any intermediaries. This situation would yield the best result of all, the energy consumed in sending the data would be the least because no hops were made as the destination node was just next to the source node.

- *Deepika Dhawan is currently Research scholar, pursuing Ph.D. in Electronics and Communication Engineering at IKG Punjab Technical University, Jalandhar, Punjab, India. E-mail: deepika273@gmail.com*
- *Rajeshwar Singh is currently Director at Doaba Khalsa Trust Group of Institutions, SBS Nagar, Punjab, India. E-mail: rajeshwar.rajata@gmail.com*

The packet delivery ratio would also be at significant ratio if the destination node were near the source node and the mobility of the destination node is low, so the number of packets sent and the number of packets received would be near equivalent. But unfortunately, that does not happen every time. So one way to approach this could be that we keep a routing table and a position table for every node in the MANET and update it every time any node in the network changes its position, but that implementation would be exhaustive and much more complex than it should be. There are many ways of finding a route to the destination but the main factor we desire is that it should not be consuming a lot more energy than it is supposed to, instead the protocol should be saving a significant amount of energy. The ways to conserve energy in MANETs have been rigorously tested in the past and researchers are developing and studying several routing techniques [1] that can achieve an energy-efficient algorithm which not only maximizes the residual energy but also the maintains an optimal level for the other factors that an algorithm is evaluated on. Therefore developing a routing protocol for MANETs is a challenging task. Ant colony optimization [2] meta-heuristic belongs to swarm intelligence field which is foraging behavior of the ants in nature. In ACO, artificial ants are used to find solutions to combinatorial optimization problems. As in real life the ant leaves the trails of pheromone on the path it travels so that other ants could follow it, as a result, if another ant follows the same path due to pheromone left by the earlier ant, the pheromone level on that path will be then stronger than before as the second ant, which followed the path will also lay down some level of pheromone. This process might attract many other ants. Pheromones, in this case, represent indirect information exchange within the ants. As real ants lay down pheromones on their path, in the simulation the simulated agents (artificial ants) record the quality of their solution and their position i.e. the path traveled by an ant in the simulation will have a record of quality of the solution. Which lets the simulation find the best path from the source node to the destination node, and when there is need to send packets, it will send it through the best path discovered out of all the paths discovered by the simulated agents. Ant colony optimization is one of the most promising nature inspired meta-heuristic, which is capable of finding the best, shortest and quality path. Ad hoc On-Demand Vector Routing (AODV) [3] is an algorithm which is also used in operations of MANETs. It allows nodes to quickly route to the destination and does need to maintain the routes to the destination. The AODV algorithm operates loop free. And when the nodes move in the network it quickly converges by avoiding the “counting to infinity” problem which is also known as Bellman-Ford “counting to infinity problem”. In AODV, nodes can dependently on link breakage and change in the network topology. When the link breaks, the affected set of nodes are notified so that they can discard the route using the lost link. AODV offers adaptation to the link conditions, memory overhead and calculates unicast routes to the destination node. In this paper, we propose an algorithm in which simulated agents (artificial ants) go through the whole network to find the shortest path between the source and the destination node while also caring for the metrics such as low energy consumption, less data loss, etc. In the rest of the paper, section 2 summarizes the literature review, section 3 explains the working of our approach to the problem in MANETs extensively, section 4 consists of simulation and results, and with section 5 we conclude the paper.

2 LITERATURE REVIEW

Ant-based algorithms for routing were observed and proposed by Dorigo, Maniezzo, and Colorni in the year 1991. It was an observation of the real-life behavior of ants in the search of their food. This approach has been widely accepted and used for some NP-Hard permutation and combinatorial problems such as Travelling Salesman Problem and the Quadratic Assignment Problem (QAP). Experimentation of the same led to the result that the behavior of the ants can also be used for finding the shortest path. Due to the dynamicity of the topology, the first introduced ant-based routing technique could be applied. The first ant-based routing schemes introduced were proactive protocols for wired and static networks as in AntNet [4]. The periodic unicast ant’s usages to discover the routes presenting longer time and restoration in the changes of topology were very slow. Ant-AODV [5] is presented as a hybrid algorithm which combines the feature of the basic AODV behavior with the ant-based exploration technique i.e. a specified number of ants traversing around the network in somewhat a random way, keeping a track of nodes visited and upon arriving at the node it proactively updates the routing table. PERA [6] is an on-demand routing algorithm technique which is based on the flooding mechanism and which uses the delay as a metric to reactively establish the route. In an AntHocNet [7,8] authors propose a Hybrid and stochastic approach which is aware of any type of congestion in its network and it only tends to find the path on demand, but once a route is established it is maintained proactively. In ANSI [9] authors propose that only the whole network is flooded when a required path is not available. EEABR [10] uses the artificial ants who are looking for paths, where each of the ants chooses its next hop with a function that is the quantity of pheromone available on the route between the nodes and the node energy. While choosing the next hop it evaluates the amount of pheromone present on the path and keeps updating it as a function of the energy level and the length of the path.

3 ACO-EE ROUTING ALGORITHM

(i) Ant Colony Optimization: ACO is a meta-heuristic which is based on the foraging behavior of real-life ants. The ants in real life work together to find the path to their food source. ACO is one of the research directions, applied in Swarm intelligence. In real life ants leave pheromone on their path while going in a direction when another ant finds the trail, it has a good possibility to follow the trail. If the subsequent ant follows the trail it also lays down some pheromone on its path. As a consequence the pheromone level on the followed path is increased and more of the ants are likely to follow the same path as it has now become the more traveled path and it has a high level of pheromone deposited on the route and the ants tend to follow the path where there is more pheromone, which leads other ants to follow the same path. The exchange of the information in between the ants is indirect and the exchange of information is done by pheromone. One of the positive feedbacks we get by implementing the ACO meta-heuristic is that it describes a framework which is common for approximating the NP-Hard problem and it able to adapt to the network changes in real time [11]. And simultaneously multiple solutions can be found for the considered problem. Every ant in a network starts at a node S and heads for the destination node D. The simulated ants follow a decision rule (probability decision rule) [12] to explore the network. The decision rule is a function of local pheromone trails. The applied probability rule can be represented as shown in (1).

$$P_{ij}(t) = \frac{[\tau_{ij(t)}]^{\alpha}.[\eta_{ij}]^{\beta}}{\Sigma_{l\in \mathbf{N_i}}[\tau_{il(t)}]^{\alpha}.[\eta_{il}]^{\beta}} \text{ if } j\epsilon\ N_i, \text{else } 0 \qquad (1)$$

Where Pij(t) is the probability of ant to move from node i to j at the ith iteration, Ni is the current neighbor nodes of the node i, $\tau_{ij(t)}$is the pheromone intensity,$\eta_{ij}$is the heuristic information, it is a usually non-increasing function of moving cost. α and β are the parameters which control the impact of pheromone intensity. When α = 0 the ant select their next hop based on the heuristic information, and when β = 0 the ants are attracted by the pheromone intensity. And j not in Ni then it shows that the ants can only move to their neighboring nodes. The pheromone trails either gets increased or decreased. The updating of pheromone is calculated on the basis of the quality of the solution and the evaporation rate is shown in (2). [12]

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \textstyle\sum_{k=1}^{m}\Delta\tau^{k}{}_{ij} \qquad (2)$$

Where τ_{ij} is the pheromone value laid by the ants, $\rho\epsilon$ [0,1] is the pheromone evaporation rate, m is the number of ants, and $\Delta\tau^{k}{}_{ij}$ is the pheromone reinforcement deposited by the kth ant. The pheromone reinforcement of the kth can be evaluated as Eq. 3.

$$\Delta\tau^{k}{}_{ij} = \frac{Q}{C^{k}} \text{ if arc}(i,j) \in\ P^{k}, \text{else } 0 \qquad (3)$$

Where Q is the positive application specific constant, $P^{k}$is the set of edges chosen by the kth ant, Ck is the overall cost function of the path constructed by a kth ant. Many other variations of the ACO are available which may have minor changes from the above-stated equation of path search and pheromone update, but in general, the ACO implementation consists of three main functions [12], (a) ConstructAntSolutions, (b) UpdatePheromones and (c)

DaemonActions. ConstructAntSolutions is a procedure where the ants find the solution of the construction graph, UpdatePheromones is the procedure where the ants update the pheromone level and DaemonActions, which is implemented to add centralized actions, and this is also an optional procedure. One of the merits of the ACO meta-heuristics that it presents a framework which is common, to find solutions to the problem which are in the category of NP-Hard.

(ii) Ad-Hoc On-demand Distance Vector Routing: AODV is the most studied protocol for MANETs, AODV broadcast Route request when there is a need to send the packet to a destination node. The source node in AODV algorithm does not know the route to the destination node beforehand. The source node sends the Route Request (RREQ) to its neighbor nodes and the send the same to their neighbor nodes until there is a path found to the destination. The node which knows the information about the route for the destination node then sends the Route Reply (RREP) alongside the sent RREQ back to the source node. The path is established when the RREP is received by the source node. The neighbor nodes in AODV also sends HELLO messages to their neighboring nodes to make sure that the connection is still established, then the source node starts sending the data/packets to the destination node using the RREP received. The original AODV contains too many specifications which lead to a tendency to make mistakes, so in order to avoid that, AODVjr is used. AODVjr is the earliest simplified version of the AODV. We are using AODV by avoiding some parameters such as precursor list, RERR, etc. besides the loops can be avoided by the RREP sent by the destination node. AODVjr is much simpler than AODV, it can be extended by implementing some optimizations to improve performance, and some security features can also be added to the AODVjr.

(iii) Energy efficiency in MANETs: In MANETs the nodes work on a limited power stored in them if the algorithm executed is designed in a way that it that is not energy efficient then the Network might collapse and become dead. There is a very critical need of implementing an energy efficient algorithm which can save the power of the nodes and complete a process. Although the nodes work on a limited energy it does not mean that we cannot stretch the life of the node, in fact, the purpose of the energy efficient models in MANETs have the same aim, to save the energy of the nodes so that it can operate for longer period of time and do not get exhausted and dead. In the absence of the base station, each node has to participate in the protocols like building routes, etc. causing them to spend more energy. The nodes in the MANETs can alter their transmit power level to be connected to the network, while simultaneously increasing the network capacity, reducing energy consumption and reducing interference. The evaluation of the total energy consumed by the network is given in (4)

$$E_{TOTAL} = E_{ECHO} + E_{RD} + E_{DATA} \quad (4)$$

Where E_{RD} is the energy consumed in the route discovery process, E_{DATA} is the energy consumed during the data transmission, and E_{ECHO} is the energy consumed by the broadcast message. The energy consumed in the data transmission is evaluated as shown in (5).

$$E_{TX}=\begin{bmatrix} B \times E_{elec} + B \times \gamma_{fs} \times d^2, if\ d \leq d_0 \\ B \times E_{elec} + B \times \gamma_{mp} \times d^4, if\ d \geq d_0 \end{bmatrix} \quad (5)$$

Where E_{elec} is the energy disseminated per bit to run the recipient circuit, γ_{fs} is the energy expanded when $d \leq d_0$ and γ_{mp} is the energy devoured in the intensifier when $d \geq d_0$.

(iv) ACO-EE Routing Algorithm: Pheromone table is stored in each node's memory, the pheromone table can be visualized as the matrix where the neighboring nodes are listed in the row side of the matrix and the columns correspond to destinations. The probability equation to select the next hop is given in (6).

$$\frac{(\tau^k{}_{id})^{\eta}}{\sum_{j \in N_i} (\tau^k{}_{jd})^{\eta}} \quad (6)$$

The forward ant F_{SD} which was generated from the same source and destined to the same destination node have the same source of sequence number which is referred to as ant generation. Each forward antF_{SD} keeps track of the sequence of nodes it has visited. For every intermediate node, the forward ant F_{SD} sets up reverse paths, staring from the current nodes to the intermediate node by updating the pheromone table as explained in (2). When the forward ant F_{SD} reaches the destination node, the destination node generates a backward ant BDS destined to the source of the forward ant and the list of nodes which the forward ant came through is provided to the backward ant B_{DS} and the forward ant is then killed by the destination node. At each intermediate node the backward ant B_{DS} reaches, it set up a list of the path of the nodes it has visited just like the forward ant F_{SD}. Once the backward ant B_{DS} arrives at the sources node a path from the source to the destination is established. And the source and destination nodes can communicate from now on. ACO-EERA has implemented a different policy for sending the data packets and the ants. In routing when a node has multiple next hops for the destination d, it selects the next hop randomly with the probability as shown in (7).

$$\frac{(\tau^k{}_{id})^{\alpha} (E_i \overline{FTE_\iota{}^k})^{\beta}}{\sum_{j \in N_i} (\tau^k{}_{jd})^{\alpha} (E_i \overline{FTE_\iota{}^k})^{\beta}} \quad (7)$$

Where E_i denotes the energy level if the next hop node i, and $\overline{FTE_\iota{}^k}$ is the link quality and the congestion factor between node k (the current position of the node) and it's neighbor i (the node it is considering to hop), it is also known as frame transmission efficiency. E_i is computed as shown in (8).

$$E_i = E_{iremaining} / E_{iinitial} \quad (8)$$

Where $E_{iremaining}$ is the energy remaining in the next possible hop node and $E_{iinitial}$ is the energy of the next possible hop node's initial energy. Based on the probabilistic data, traffic will be distributed according to the probabilities for each neighbor in the routing table. The neighboring nodes which have less remaining energy and path with bad links are less selected.

4 SIMULATION AND RESULTS

In this section, we will be simulating the proposed protocol and it will be compared with the AODV. AODV is the most used protocol for the operations of the MANETs.

4.1 System environment: The proposed ACO-EERA for the analysis of factors such as Energy consumption, Packet Delivery Ratio, End to End delay is implemented in Network Simulator 2.35, with the following system configuration:

- Processor: Intel i3 processor
- Ram: 6 GB
- Operating system: Fedora 23

We have run the awk scripts on the output trace file, which was generated from the output of the TCL file. A trace file (.tr) contains the simulation details, network information, changes, etc.

4.2 Parameters: In the simulation, we have performed the QoS parameters on the above system environment, and the parameter for the protocol is as below:

Table.1: *Simulation Parameters for ACO-EERA*

Parameters	Value
Number of Nodes	20
Initial energy	100J
Area (m)	1000 * 1000
Mac Type	Mac/802.11
Rx Power	0.3
Tx power	0.6
Protocol name	ACO-EERA

4.3 Execution and Results: The metrics chosen for the testing of our algorithm are very suitable as they define the crucial factor which an algorithm should be aware of. The metrics chosen are:

4.3.1 End to End Delay: End to end delay is defined as the time taken for a packet to reach from the source to the destination. Mathematically End to end delay is defined in (9).

End-to-End delay = $\frac{\sum e}{P}$, Where. e is Td – Ts (9)

After executing the end-to-end delay awk script on the output trace file generated by the Tcl of the protocols, we get the following results as shown in Table.2:

Table.2: *End to End Delay of AODV and ACO-EERA*

Routing Algorithm	End to End Delay (ms)
AODV	67
ACO-EERA	41

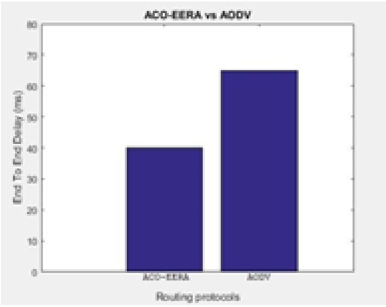

Figure 1 *showing the end to end delay AODV and ACO-EERA*

We can see in figure 1 that for our proposed algorithm, it takes significantly less time to send a packet from the source node to the destination node. The time taken to reach the destination node from the source node is a very satisfactory improvement.

4.3.2 Energy consumption: energy consumption in a network is that after completing the operation and completing the whole process when the simulation ends how much energy was consumed by the nodes in the network. This factor is crucial as the node work on a limited amount of energy and if the consumption is high then eventually they will lose all their energy and the nodes will be dead which will lead the network to be not much of a use. The total energy consumption in a network is defined as shown in (4).

$$E_{TOTAL} = E_{ECHO} + E_{RD} + E_{DATA}$$

This calculates the total energy spent in completing the whole process. After calculating and the following result is received.

Table.3: *Energy Consumption of AODV and ACO-EERA*

Routing Algorithm	Total Energy Consumption
AODV	2.6 KJ
ACO-EERA	1.2 KJ

As we can see in figure 2 that the energy consumed by our protocol is comparatively way less than the energy consumed by the node in AODV. This is due to the additional function, which before hoping to the next intermediate node, it first evaluates the link and path quality.

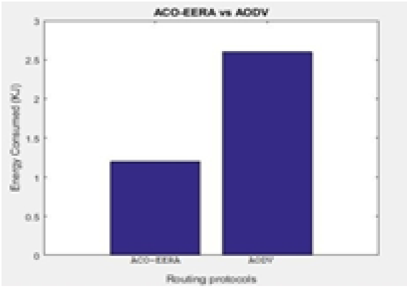

Figure 2 *shows the energy consumption comparison of ACO-EERA and AODV*

The energy consumed by the ACO-EERA is significantly lower than the energy consumed by AODV. In our approached algorithm the average energy consumption is near 1.2KJ whereas in AODV we can see that the energy consumed by the nodes is around 2.6KJ.

4.3.3 Packet Delivery Ratio: Packet Delivery Ratio in MANETs is the number of packets received over the number of packets originally sent. The packet delivery ratio is can be calculated using the (10).

$$PDR = \frac{\sum_i \text{No. Ofreceivedpacket}}{\sum_i \text{No.ofpacketsendbysource}} * 100 \qquad (10)$$

Upon calculating the packet delivery ratio of both the protocols we get the following result:

Table.4: *Packet Delivery Ratio of AODV and ACO-EERA*

Routing Algorithm	No. Of packets sent	No. Of packets received	PDF %
AODV	109	89	81.65
ACO-EERA	285	261	91.58

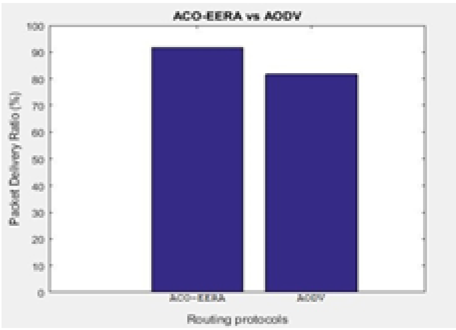

Figure. 3 *shows the packet delivery ratio of ACO-EERA and AODV*

In figure 3 we can see that the ACO-EERA has successfully delivered a large percentage of the packet and the difference between the packet delivery ratio of ACO-EERA and AODV is significant. ACO-EERA has a more than 91.58% of the packet delivery ratio, whereas the AODV protocol has near 81.65% of the packet delivery ratio. Thus we have backed our protocol with solid evidence that it can perform better than the AODV algorithm in all three metrics.

5 Conclusion

In this paper, we have proposed an algorithm that has proven to perform better in MANETs routing and yield better and satisfactory results. We have used AODV for comparison with our proposed ACO-EERA. AODV still has some drawbacks such as the energy consumption is high less packet delivery ratio than other protocols, the higher end to end delay. Our proposed algorithm has proven to be better than AODV and the results to back the statements are presented above. We have implemented the function to choose the next hop wisely using the information about the bad linkage or the path quality etc. The calculated steps taken to choose the next hop in a network has made the algorithm a great tool to solve problems in MANETs. We can see in the Simulation and Results section that our algorithm performs better than the AODV. ACO-EERA has yielded great and significantly improved result in case of energy consumption, end-to-end delay and packet delivery ratio.

References

[1] A. H. Mohsin, K. A. Bakar, A. Adekiigbe, K. Z. Ghafoor, "A survey of energy-aware routing and mac layer protocols in MANETs: trends and challenges", Network Protocols and Algorithms, vol. 4, pp. 82-107, June 2012.

[2] Dorigo, M., &Stützle, T. (2003). The ant colony optimization metaheuristic: Algorithms, applications, and advances. In Handbook of metaheuristics (pp. 250-285). Springer US

[3] Perkins, C.; Belding-Royer, E.; Das, S. (July 2003). Ad hoc On-Demand Distance Vector (AODV) Routing. IETF. doi:10.17487/RFC3561

[4] Dorigo M. and D. Caro G., "AntNet: a mobile agents approach to mobile adaptive routing", Tech. Report, IRIDIA- Free Brussels University, Belgium, 1997.

[5] S. Marwaha, C. K. Tham and D. Srinivasan, "Mobile agents based routing protocol for mobile ad hoc network", Proceeding of IEEE GLOBECOM 2002, Nov 2002, Taipei, Taiwan

[6] J. S. Baras, H Mehta, "A probabilistic emergent routing algorithm for mobile ad hoc network", Proceeding in Modelling and Optimization in Mobile, Ad hoc and Wireless Networks, 2003

[7] F. Ducatelle, G. Di Caro and L.M Gambardella, "AntHocNet: an ant based hybrid routing protocol for mobile ad hoc networks", IDSIA/USI-SUPSI, DalleMolle Institute for Artificial intelligence Galleria 2, Technical Report IDSIA 28-04, 2005.

[8] F. Ducatelle, G. Di Caro and L.M Gambardella, "AntHocNet: an ant based hybrid routing protocol for mobile ad hoc networks", Proceeding of PPSN VIII Eighth International Conference on Problem Solving from Nature Lecture Notes on Computer Science, 2004

[9] S. Rajagopalan, C.C Shen, "ANSI: an unicast routing protocol for mobile ad hoc network using swarm intelligence", Journal of systems Architecture, Nature Inspired Application and Systems, vol. 52, issue 8-9, 2006

[10] T. Camilo et al, "An energy efficient ant based routing algorithm for wireless sensor networks", Ant Colony Optimization and Swarm Intelligence, vol 4150, 2006

[11] H. Ahmed and J Glasgow, "Swarm intelligence: concepts, model and applications", School comput., Queens., Kingston, ON, Canada, Tech. Rep. 2012-585, 2012

[12] Dorigo, M., & Stützle, T. (2003). Ant Colony Optimization, Cambridge,, MA, USA, MIT Press, 2004.