Enhancing AODV Routing Protocol to Predict Optimal Path Using Ant Colony Algorithm in MANET

Mahmoud Abu Zant¹, Salah Jabareen², Dr. Amjad Rattrout³, Dr. Mohammad Hamarsheh⁴

Computer Science Department, Arab American University Jenin, Palestine.

Abstract
Mobile Ad-hoc Network (MANET) was a military idea. It's a network consists of mobile nodes that communicate and exchange information with each other using wireless links. MANET optimal operation faces many challenges, the most important one is selecting the optimal path between mobile nodes. It is not an easy process because of the characteristics of mobile nodes in MANET like energy restriction, node velocities, and coverage of mobile nodes, all of these parameters lead to a dynamic change in MANET topology. Ant Colony Optimization (ACO) algorithm is a form of a self-organized model that is based on Stigmergy concept. It is used in AODV to help in the selection of an optimal path between nodes based on different parameters. In this research, we proposed an enhancement of AODV routing protocol to predict optimal path between mobile nodes based on different parameters using ACO, we compared the proposed model (ORA_AODV) with AODV. The results showed that ORA_AODV has a higher Packet Delivery Ratio and a lower End to End Delay.

Keywords: Ant Colony Optimization, AODV, Mobility, Mobile Ad-hoc Network, Energy, Swarm Intelligence.

INTRODUCTION
Mobile Ad-hoc Network (MANET) is one of the applications of Wireless Ad-Hoc Network (WANET) that allows mobile nodes to communicate and exchange information with each other using wireless links while moving randomly without any central control. Mobile nodes communicate with each other directly while they are in the coverage area of each other in a single-hop connection, but in multi-hop connection, mobile nodes are not in the coverage area of each other, so they communicate by the help of intermediate nodes. Mobile nodes work as a bridge between other nodes or as an end node. There are three types of routing protocols used in MANET to adapt to the changes in the network topology, Proactive or as they called Table-Driven, Reactive or as they called On-Demand, and Hybrid routing protocols. Mobile nodes in Proactive routing protocols have a table that consists of information about other mobile nodes in the network. Tables are continuously updated by messages that are sent between nodes. Mobile nodes use the information stored in their tables to communicate with other nodes. Destination Sequenced Distance Vector (DSDV) is considered as one of the most popular Proactive routing protocols. Another type of protocols is Reactive routing protocols where mobile nodes obtain information about other nodes when a route is needed. Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distance Vector (AODV) protocols are the most popular Reactive routing protocols. Hybrid routing protocols like Zone Routing Protocol (ZRP) combines both advantages of Reactive and Proactive routing protocols [1]. MANET is an interesting research topic, especially its routing protocols, Security of routing protocols and the congestion control in MANET using TCP protocol variants [2]. MANET suffers from different problems that affect the performance of the network. Energy is considered as one of the main problems for MANET where nodes in the network have a limited Energy also nodes need Energy for transmitting packets through the antenna. Wireless interference is also a problem where nodes in MANET may interfere with each other while they are transmitting packets. Ensuring Security in MANET is a hard task and most of the native protocols are not provided with any security mechanism. MANET is vulnerable to different types of attacks that target the control of the path or the data itself. MANET suffers from Low Bandwidth, which could happen for several reasons such as collision, mobility, and obstacles [3]. Figure 1 shows an example of MANET.
BACKGROUND

AODV:

Ad Hoc On-Demand Distance Vector (AODV) Routing is a Reactive routing protocol that combines both features of DSR and DSDV. AODV has two phases, the first is route discovery and the second is route maintenance. In route discovery phase, AODV uses the same mechanism used in DSR by flooding the network with Route Request (RREQ) packets, each mobile node broadcasts RREQ to its neighbor until the RREQ reaches the destination or an intermediate node that has a path to the destination node. A Route Reply (RREP) packet is sent back to the source node. Source node starts to transmit data packets after it receives RREP through the discovered path. Each node in AODV stores information about routes in a routing table as in DSDV. Each record in the table contains destination node ID, hop count to the destination node, destination sequence number to indicate how fresh the route is to avoid loops. Each record also contains an expiry time entry, which helps in route maintenance to detect a link failure. If no packet received within the expiry time from a connected node, the link is considered to be disconnected and then a Route Error (RERR) packet is broadcasted. Each node receives RERR packet it removes any record in its table uses that link and start the route discovery phase in order to find an alternative path to the destination node [4]. In this research, we have selected AODV routing protocol because it outperformed all other Reactive routing protocols in different performance metrics according to [5].

Stigmergy:

The Stigmergy term is derived from a two Greeks words “stigma” which means trace, and “ergon” which mean process. Thus, “Stigmergy” means trace left after finishing a process, which guides others in performing a subsequent task. This term was first introduced by Pierre-Paul Grasse in 1959 after he studied the social insect societies. He discovered that insects don’t have a direct connection to each other, but each insect work independently and the trace left in the environment simulates the work of another insect, which forms a self-organizing behavior [6].

Swarm Intelligence (SI):

The SI concept formed from Stigmergy and the observation of different swarms and colonies in the natural environment. SI combines both Artificial Intelligence (AI) and nature to create algorithms that help nodes to adapt to the changes in the environment and to form a self-organizing structure. These algorithms are called optimization algorithms such as Artificial Bee Colony (ABC), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). Much research use and implemented SI in MANET especially ACO algorithm [7].

Multi-Agent System (MAS):

MAS is a system that consists of intelligent agents that are capable of solving a particular problem, each agent has a simple task to do. Sometimes agents are heterogeneous in the system, and they can work in both static and dynamic environment. These agents work together to solve any problem that is impossible to be solved by any single agent. MAS is capable of adapting to the changes in the topology of a system environment. This is why many researchers use MAS in MANET to help in finding the route in the dynamic environment of MANET topology [8]. Figure 2 describes Stigmergy Concept.

Ant Colony Optimization (ACO):

ACO helps in finding the optimal path between nodes that want to communicate with each other. It’s inspired by ant society in the natural environment. Worker ants leave their nests to find food in order to feed their queen. They use their antenna to smell food. While ants search for food, they deposit a chemical substance called pheromone that evaporates over time. When an ant finds the food source it turns back and deposits more pheromone to increase its intensity along the road to the nest. Worker ants come across a road with high-intensity pheromone. It leaves the nest and searches for the food by following the pheromone trail towards the food source. This type of communication is called indirect communication.

If the environment had changed and an obstacle had appeared in the path, worker ants choose to avoid the obstacle from a different direction with equal probability. The intensity of the pheromone will be more in the shortest direction because it has a shorter travel time. Over time all worker ants will use the shortest path [9] as shown in figure 3 and figure 4. ACO is very suitable for MANET because it’s ability to adapt to dynamic changes in the topology and ensures QoS along with the link failure handling [10]. We can ensure QoS of MANET by using Ant Colony Optimization algorithm [11].

Figure 2. Stigmergy Concept.
RELATED WORK

ACO is considered as one of the most used SI algorithms in MANET to improve the selection process of the optimal path, it has been implemented in different ways. In this section, some related research work about the implementation of ACO in MANET is discussed. It proves that the improvement of the performance of the routing protocols in MANET.

Eseosa Osagie et. al. [12] developed a new improved algorithm based on ant colony optimization in MANET called PACONET. PACONET focus on finding the optimal path between the source and the destination node using ant colony optimization. PACONET uses two types of ants, Forward ants (FANTs) and Backward ants (BANTs). FANT is used to find the route from the source node to the destination node. It visits all possible paths from source to the destination node and stores the path in its memory. While FANT travels between nodes it deposits pheromone value at nodes based on time and weight of each node. When the destination node receives FANT it sends BANT to the source node. While BANT traveling thought the path obtained from FANT memory, it updates the pheromone entry in each node table. When the source node receives BANT, it starts to transmit data to the destination node through the path obtained from BANT. PACONET is compared with AODV routing protocol under different performance metric, the results show that PACONET has a higher Packet Delivery Ratio (PDR) while increasing the Pause time, but it shows also a lower PDR while increasing the node mobility. Results also show that it has a higher control overhead while increasing the node mobility.

Ahmed M. Abdel-Moniem et. al. [13] developed a new algorithm called MRAA based on AODV and ant colony optimization to find the best route in MANET. In route finding phase, if the source node does not have a route to destination node then it initializes forward ant and broadcasts to all neighbor nodes. Neighbor nodes check if they have an entry in their tables that have a high pheromone value to the destination node. If no entry is found, then nodes broadcast forward ant to their neighbor nodes, the broadcasting process continues until the destination node receives forward ant. While forward ant traveling between nodes, it stores nodes IDs in its memory. When the destination node receives forward ant, it sends a backward ant to the source. Backward ant updates also the pheromone values in nodes depending on the number of hops, travel time and link capacity. MRAA is compared with AODV routing protocol under different performance metrics and the results show that MRAA has a higher Throughput and lower average End to End delay than AODV.

Rajanigandha Metri et. al. [14] designed a new protocol called QAMR. It finds the optimal path between nodes in MANET based on ant colony algorithm. The protocol has three types of ants, forward ant (FANT), Route Request Ant (RREQA), and Route Reply Ant (RREPA). RREQA keeps updating nodes table when destination node receives RREQA it sends back RREPA using the path that is stored in the ant memory. RREPA also keeps updating the pheromone value based on delay and bandwidth. When the source node receives RREPA it starts to transmit data through the path if its pheromone value is higher than a specific threshold. QAMR is compared with AODV and ARMAN protocols. It showed a high Packet Delivery Ratio while increasing both mobility of the nodes and the number of nodes.

P. Kranthi Kumar et. al. [15] proposed a new algorithm based on Ant Colony Optimization in MANET. They enhanced the ACO protocol by using a new parameter called orientation which indicates the direction of the destination node. Using this new parameter along with reinforcement learning to select the shortest path between the source and the destination node, the proposed algorithm results showed a higher throughput than AODV while the simulation time increases.

Anjali Jagtap et. al. [16] developed an optimized algorithm based on Ant colony algorithm in MANET, the developed algorithm called Ant-Based Routing Algorithm (ARA). BANT deposits the pheromone value based on the number of hops from source to destination node collected from FANT. Performance of ARA while increasing both numbers of nodes and simulation time is studied and the results showed that the performance of ARA is stable even if the size of the network increased.

Adwan Yasin et. al. [17] enhanced the AODV routing protocol in MANET by taking new parameters such as Power and speed of the nodes. The speed parameter is computed using the time and distance between nodes. The speed helps
in predicting the time needed for a node to leave the range of another node which leads to a link failure after a specific time.

Somesh Maheshwari et. al. [18] proposed a technique to select a secure path between nodes based on Ant Colony Optimization. The proposed technique selects the path based on two parameters, packet forwarding ratio (PFR) and Maximum bandwidth (MBW). In order for an ant in the proposed model to select a route, it follows five rules.

Bibhash Roy et. al. [19] proposed a QoS routing algorithm based on Ant Colony optimization in MANET. An ant called Path Request is used in route discovery stage, the pheromone values depend on three parameters link bandwidth, link expiration time, and the delay between links.

Dr. Madhumita Dash et. al. [20] worked on modifying ARA algorithm by taking other metrics instead of the number of hops. The number of hops is not enough to determine the quality of pheromone in MANET especially in high mobility MANET. Other metrics are used to increase the quality of path selection between nodes. They compared the modified ARA with the old ARA along with AODV and the results show that the modified ARA has a higher Packet Delivery Ratio and Throughput than the other two protocols and lower End to End Delay.

PROPOSED MODEL

The proposed model uses Ant Colony Optimization to predict the optimal route from the source node to the destination node depending on different parameters such as energy and velocity of nodes, number of nodes in the path and travel time from the source to the destination node. This model uses four types of ant and contains five phases namely: Joining Network, Distance computing, Route Exploring, Evaporation rate, and Route Recovery.

Joining Network:

In this phase, when a mobile node joins the network, it broadcasts a Neighbors_Ant (NA) to all single hop adjacent nodes. When a node receives NA, it checks if the source ID exists in its table then it reset the record expiry time, else it creates a new record for the new node with initial pheromone value of 0.0 as shown in figure 5. Each mobile node broadcasts NA every specific time to inform other nodes about its existence near them. The broadcast time is set by the users based on the network environment. Algorithm 1 below describes the Joining Network phase using pseudo code.

![Figure 5. A node broadcasts NA.](image)

![Figure 6. The source node broadcasts RFAs.](image)
Algorithm 1: Joining Network

Sender Node

1 if CurrentTime == NA_Time then
2 Initiate NA;
3 Broadcast NA to all neighbors;
4 Reset NA_Time;
5 end if

Receiver Node

1 search (NA Source ID in Table)
2 if ID exists then
3 Reset entry expire time;
4 else
5 Create a new entry in the table;
6 Add ID in NA to the new entry;
7 Set initial Pheromone value to 0.0;
8 Set entry expire time;
9 end if

Distance Computing:

In this phase, each mobile node has a Distance Calculation Timer that is set by the users based on the network environment. Each node computes the distance from other nodes using the power presented in the received radio signal using Received signal strength indication (RSSI) as shown in equation (1) [21]. It is based on Two-ray ground reflection model.

\[ P_r(i,j) = \frac{P_t G_t G_r H_t^2 H_r^2}{d^4} \]  

(1)

Where \( P_r(i,j) \): Receiving power between node i and j, \( P_t \): transmitting Power, \( G_t \): Gain of a transmitting antenna, \( G_r \): Gain of a receiving antenna, \( H_t \): height of the transmitter antenna, \( H_r \): height of the receiver antenna, \( L \): system loss factor which has nothing to do with the transmission, and \( D \): is the distance between the antennas. In this phase nodes store in its table the distance between them, nodes compute the difference between old distance and new distance (\( \Delta D \)), and the difference between times in computing distance (\( \Delta T \)). Algorithm 2 below describes the Distance Computing phase using pseudo code.

Algorithm 2: Distance Computing

1 if CurrentTime == DC_Time then
2 for each neighbor node do
3 Calculate D using RSSI (1);
4 \( \Delta D = new\ D - old\ D \);
5 \( \Delta T = new\ T - old\ T \);
6 Update D, \( \Delta D \), and \( \Delta T \) values in the table entry;
7 end for
8 Reset DC_Time;
9 end if

Route Exploring:

In this phase, source node broadcasts Route_Finder_Ant (RFA) to all single hop adjacent nodes. RFA contains a list of nodes that have been visited by RFA. Each node receives RFA, checks if it’s the destination node or not, if it’s not the destination node then it’s an intermediate node, RFA adds the intermediate node ID in the RFA list. The intermediate node unicasts RFA toward the next hop of the destination node using the high pheromone value indication. Otherwise, it rebroadcasts RFA to all single hop adjacent nodes, the rebroadcasting process continues until the destination node is reached. Figure 6 shows the broadcasting process of RFAs. When the destination receives RFA it sends back a Collector_Ant (CA) to the source agent. CA uses the path from the list inside RFA to find the route, while CA travels through the nodes within the path, CA deposits pheromone on the route based on equation (2):

\[ Ph_{i,j} = Ph_{i,j} + \Delta Ph_{i,j} \]  

(2)

Where \( Ph_{i,j} \): pheromone value from node i to node j and \( \Delta Ph_{i,j} \) is the function used to determine the pheromone value. Each pheromone has an evaporation rate, so each \( Ph_{i,j} \) is updated using equation (3):

\[ Ph_{i,j} = (1 - \rho) * Ph_{i,j} \]  

(3)

Where \( Ph_{i,j} \): pheromone value from i to j and \( \rho \) is the evaporation rate that is determined by the user. \( \Delta Ph_{i,j} \) is the function that is used to compute and determine the pheromone values this function depends on the following parameters, the energy of nodes, the distance between nodes, the velocity of nodes and active time between nodes. The relative velocity between two nodes can be computed using equation (4):

\[ V_{i,j} = \frac{\Delta D_{i,j}}{\Delta T} \]  

(4)

Where \( V_{i,j} \) the relative velocity between node i and j, \( \Delta D \) between i and j node, and \( \Delta T \) is the difference between the time in computing the \( \Delta D \). If \( V \) is positive then the nodes are moving away from each other, else if \( V \) is negative then the nodes are moving closer to each other; if \( \Delta D \) equal zero this means that nodes are moving together in same velocity and direction or that the nodes are not moving, in this case, \( V_{i,j} \) equals zero. In order to compute the active time from i to j (\( AT_{i,j} \)), which indicates the time expected for the nodes to leave the coverage of each other, \( AT_{i,j} \) can be computed using formula (5):

\[ AT_{i,j} = \begin{cases} \frac{CR - D_{i,j}}{|V_{i,j}|}, & V_{i,j} > 0 \\ \frac{CR - D_{i,j}}{|V_{i,j}|} + \frac{CR - D_{j,i}}{|V_{j,i}|}, & V_{i,j} < 0 \end{cases} \]  

(5)

Where CR: coverage Range, \( D_{i,j} \): the last computed Distance between i and j nodes, \( V_{i,j} \): the relative velocity between i and j nodes, in case, that \( V_{i,j} \) equal zero then the value of \( AT_{i,j} \) is...
determined by user, the value of this variable depends on the nature of the system environment, and in case of the $V$ is negative we added $2 \cdot D_{i,j}$, which indicates the time needed for both nodes to swap their positions.

$$\Delta P_{i,j} = AT_{i,j} \cdot E$$ \hspace{1cm} (6)

Where $E$ is the energy of the node. CA stores the minimum pheromone value between all visited nodes in the path because this value represents the lifetime of that path. CA also stores the number of nodes within the path and the travel time from the source node to the destination node, when source node receives CA it computes the Intensity of pheromone in the path using equation (7):

$$I = \frac{M}{N \cdot T}$$ \hspace{1cm} (7)

Where $M$ is the minimum pheromone value between all visited nodes in the path, $N$ is the number of nodes within the path and $T$ is the travel time from the source node to a destination node. When the source node receives a multiple CA it checks the pheromone value collected within each CA and determines the highest Intensity of pheromone path and starts to transmit through it if $I$ is higher than the threshold that is determined by the user. Figure 7 shows that the source node may receive multiple CA during Route Exploring phase. Algorithm 3 below describes the Route Exploring phase using pseudo code.

**Algorithm 3: Route Exploring**

**Sender Node**

1. Initialize RFA;
2. Set Destination node ID in RFA;
3. Add source node ID to RFA memory;
4. Broadcast RFA;

**Receiver Node**

1. for each RFA received do
   2. if Receiver Node ID in RFA memory then
      3. Kill RFA; //Avoid route looping
   4. else
      5. Add Receiver Node ID to RFA memory;
   6. end if
   7. if Receiver Node ID is not the Destination node then
      8. if Receiver Node has a path to the Destination node with a high Pheromone value then
         9. Unicast RFA to the Destination node;
      10. else
      11. Broadcast RFA;
   12. end if
   13. else // Receiver Node ID is the Destination node
      14. Initialize CA;
      15. Copy RFA memory to CA memory in a reverse way;
      16. Set the number of hops to 0;
      17. Set the time of travel to CurrentTime;
      18. Unicast CA to the Source node;
   19. end if
20. end for

21. for each CA received do
   22. if Receiver Node is not the Source node then
      23. Compute relative velocity using equation (4);
      24. Compute active time using equation (5);
      25. Compute $\Delta Ph$ using equation (6);
      26. Compute node pheromone value using pheromone equation (2);
      27. Increase the number of hops in CA by 1;
      28. if Receiver Node pheromone value is lower than $M$ in CA memory then
         29. Update M value in CA;
      30. end if
   31. else // Receiver Node is the Source node
      32. Update travel time in CA;
      33. Compute pheromone Intensity of path using the equation (7);
      34. if path pheromone Intensity is bigger than the threshold && bigger than the used path then
         35. Stop using the old path ;
      36. Start using the new path stored in CA memory;
   37. end if
38. end if
39. end if
40. end for
Evaporation rate:
In this phase, each mobile node updates the pheromone value for each node in its table using the evaporation equation (3) every specific period of time that is set by the users based on the network environment. Algorithm 4 below describes the Evaporation rate phase using pseudo code.

Algorithm 4: Evaporation rate

1 if CurrentTime \(\equiv\) ER_Time then
2 for each neighbor node do
3 Update Pheromone value in table entry using Evaporation equation (3);
4 end for
5 Reset ER_Time;
6 end if

Route Recovery:
In this phase, if a source node detects a link failure, it broadcasts an Error_Ant (EA) to all its neighbor nodes to stop using this node in transmitting and to reduce its link pheromone value to zero. If an intermediate node detects a link failure, it unicasts EA to the source node to stop transmitting and starts the Route Exploring phase again. Algorithm 5 below describes the Route Recovery phase using pseudo code.

Algorithm 5: Route Recovery

Sender Node

1 if a link failure is detected then
2 Initialize EA;
3 Set failure node ID in EA;
4 Update Pheromone value of failure node ID to 0.0;
end if

Receiver Node

1 if Receiver Node is not the Source node then
2 Stop transmitting to failure node ID;
3 Update Pheromone value of failure node ID to 0.0;
4 Unicast EA to the source node;
5 else // Receiver Node is the Source node
6 Start Route Exploring;
7 end if

Figure 8 shows that a source node received two CAs. According to the proposed model, CA is going to store the minimum of all pheromones in nodes within the path, which in this case CA in path 1-7-4 will store value 10 and CA in path 3-5-8 will store value 20, by looking at the two paths we can see clearly that path 1-7-4 has a higher accumulative pheromone values but we can see that node 4 has only 10, which predict that the link will break sooner than path 3-5-8, so source node will choose path 3-5-8 to transmit data through it.

METHODOLOGY

The simulation of the proposed model ORA_AODV (Optimal Route using ACO in AODV) was performed using NS-2.35 simulator. The environment parameters were defined in TCL scripts like the number of nodes, antenna model, routing protocol, and other parameters. Table 1 describes the environment parameters used in the simulation. CBR (Constant Bit Rate) is used as the type of traffic between nodes. The placement and the movement of the nodes were created randomly using an NS-2.35 tool called CMU tool by using “setdest” command, the coverage of nodes was set to 150m and the size of the terrain is 500x500m.
Table 1. Network environment parameters.

<table>
<thead>
<tr>
<th>Simulation Environment Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
</tr>
<tr>
<td>Maximum 25 mps</td>
</tr>
<tr>
<td>Pause Time</td>
</tr>
<tr>
<td>Pause: 5s</td>
</tr>
<tr>
<td>Time</td>
</tr>
<tr>
<td>Simulation Time 200s</td>
</tr>
<tr>
<td>Terrain</td>
</tr>
<tr>
<td>Coordination 500*500 m</td>
</tr>
<tr>
<td>Connection</td>
</tr>
<tr>
<td>CBR (Constant Bit Rate)</td>
</tr>
<tr>
<td>Item size 512(byte)</td>
</tr>
<tr>
<td>Radio/physical layer parameters:</td>
</tr>
<tr>
<td>Radio type: 802.11b Radio</td>
</tr>
<tr>
<td>Data rate: 0.5 Mbps</td>
</tr>
<tr>
<td>MAC Protocol: 802.11</td>
</tr>
<tr>
<td>Routing Protocol: AODV &amp; ORA_AODV</td>
</tr>
<tr>
<td>Transport Protocol: UDP</td>
</tr>
<tr>
<td>Node: 15, 25, 50, &amp; 75</td>
</tr>
<tr>
<td>Node Placement: Random</td>
</tr>
<tr>
<td>Transmission range: 150 m</td>
</tr>
<tr>
<td>ANT_TTL: 4 hops</td>
</tr>
</tbody>
</table>

We compared AODV with ORA_AODV in three different performance metrics, End to End Delay, Packet Delivery Ratio (PDR) and Average Energy Consumption (AEC). An AWK scripts where used to analyze the information in the Trace file and to compute the performance metrics values.

A. Average End to End Delay: represents the time needed for a packet to travel between two nodes in the network, it can be computed using equation (8):

\[
A_{end} = \frac{\sum_{i=1}^{n} R_{ti} - S_{ti}}{n} \quad (8)
\]

Where \( A_{end} \): is the Average End to End Delay, \( n \): number of nodes, \( R_{ti} \): Receives time of packet \( i \) and \( S_{ti} \): Send time of packet \( i \).

B. Packet Delivery Ratio (PDR): represents the total number of received packets to the desired node divide on the number of the sent packets, PDR show the ratio where packets are delivered successfully from the source node to the destination node, it can be computed using equation (9):

\[
PDR = \frac{\sum R_{p}}{\sum S_{p}} \quad (9)
\]

Where \( PDR \) is: the Packet Delivery Ratio, \( R_{p} \) is the number of the received packet at the destination node and \( S_{p} \) is the number of the sent packet from the source node.

C. Average Energy Consumption (AEC): represents the average amount of energy consumed by every node in the network, it can be computed using equation (10):

\[
AEC = \frac{\sum E_{c}}{n} \times 100 \quad (10)
\]

Where \( AEC \) is: the Average Energy Consumption, \( E_{c} \) is the total energy consumed by every node in the network, and \( n \) is the total number of nodes in the network.

RESULTS

As shown in figure 9 there is a slight difference between the average of End to End Delay in both ORA_AODV and AODV because both of them depends on the lowest number of hops while selecting the path. While increasing the number of nodes the average End to End Delay decreases because there is more than one possible path to the destination node. When the number of nodes was 15 there are a few paths lead to the source node so both of the protocols showed the same average but when the number of nodes was 75, ORA_AODV showed a lower average because there is more than one path to the destination node and the proposed protocol does not only depends on the number of hops while selecting the path like AODV. ORA_AODV always tries to select the path that may live for the longest possible time among other paths.

Figure 9. Results of the average of End to End Delay.
As shown in figure 10 ORA_AODV Packet Delivery Ratio is higher than AODV because of the same reason discussed earlier and working principle of ORA_AODV. ORA_AODV takes the advantage of multiple nodes and explore all the possible paths to select the optimal one.

As shown in figure 11 there is a difference between the average Energy Consumption of both ORA_AODV and AODV. This is because ORA_AODV has a higher overhead in routing packets which indeed consumes more energy than AODV. ORA_AODV holds also more nodes than AODV before selecting the optimal path. It is interesting to show that in the case of 15 and 50 there is only a slight difference. The proposed protocol offers a higher PDR and a lower average End to End Delay at that low cost of AEC. We should mention that the maximum number that each RFA could hop from node to node may affect the performance were increasing the maximum number of hops leads to increase the AEC and increases the congestion of the network but it helps in selecting a better path. Table 2 shows the numeric results of comparison between ORA_AODV and AODV in terms of End to End Delay, PDF, and AEC.

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>ORA_AODV</th>
<th>AODV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg End to End Delay (ms)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.833</td>
<td>0.836</td>
</tr>
<tr>
<td>25</td>
<td>0.644</td>
<td>0.673</td>
</tr>
<tr>
<td>50</td>
<td>0.534</td>
<td>0.643</td>
</tr>
<tr>
<td>75</td>
<td>0.475</td>
<td>0.505</td>
</tr>
<tr>
<td>Packet Delivery Ratio (PDR) (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.252</td>
<td>0.168</td>
</tr>
<tr>
<td>25</td>
<td>0.368</td>
<td>0.341</td>
</tr>
<tr>
<td>50</td>
<td>0.390</td>
<td>0.343</td>
</tr>
<tr>
<td>75</td>
<td>0.470</td>
<td>0.364</td>
</tr>
<tr>
<td>Average Energy Consumption (AEC) (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>6.866</td>
<td>6.577</td>
</tr>
<tr>
<td>25</td>
<td>6.627</td>
<td>5.144</td>
</tr>
<tr>
<td>50</td>
<td>7.316</td>
<td>7.080</td>
</tr>
<tr>
<td>75</td>
<td>6.448</td>
<td>5.311</td>
</tr>
</tbody>
</table>

CONCLUSIONS AND FUTURE WORK
There are many factors that affect the topology of MANET like Energy, velocity of nodes, and the number of nodes in the network. An algorithm is needed to provide a self-organized structure to adapt to the dynamic changing in the topology. Ant Colony Optimization can provide a self-organized structure for nodes and can adapt changes in topology. Using ACO, an optimal path can be selected by depending on different parameters such as Energy and velocity. This will increase the stability of the network and will reduce the probability of link failure during the transmutation process between nodes. In this research, we have tried to consider all the parameters that affect the link between nodes and we provided a model that can predict the optimal path between nodes. The results show a higher Packet Delivery Ratio and a lower average End to End Delay comparing to AODV at a low cost of Average of Energy Consumption. In the future, we aim to compare our proposed model with other proposed models under different performance metric in order to prove its efficiency.

REFERENCES


