Wireless Netw (2016) 22:2081–2100
DOI 10.1007/s11276-015-1082-1

A robust energy efficient ant colony optimization routing algorithm for multi-hop ad hoc networks in MANETs

P. Vijayalakshmi[1] · Sharmila Anand John Francis[1] · J. Abraham Dinakaran[1]

Published online: 12 October 2015

Abstract The Ant Colony Optimization (ACO) routing technique is an adaptive and reliable approach to find the paths in routing for Mobile Ad hoc Networks (MANETs). In ACO algorithms, the ant as agents traverses across the network to find the shortest path from the source to its destination. During the traversal, the ants deposits pheromones on its path. The path with high probability of pheromone is chosen as the optimized path to transfer data packets between the source and the destination. The quantity of the pheromone laid down on a path depends on its quality metrics viz. minimum number of hops, minimum energy path cost and packet transfer delay. Due to the high dynamic nature of MANETs, the path with high probability of the pheromone deposition may rapidly become unavailable due to link failures. Therefore, it is a challenging task in MANETs to construct a reliable path that is less likely to be disconnected for a long period of time. In this paper, a novel robust energy efficient ACO routing algorithm named AntHocMMP has been proposed. This algorithm is meant to achieve robustness of paths for reliable communication with adaptive re-transmission delays in MANETs. This proposed algorithm enhances the performance of Max–Min–Path (MMP) approach by using ant as agents to find the optimal path in the network. The selection of the optimal path is based on the residual energy of the nodes in that path. A robust route with minimum energy path cost with a short hop-count is chosen for pheromone deposition. Therefore, to keep the paths connected in a dynamic topology, the proposed AntHocMMP redistributes the pheromone based on the energy path cost and it constructs an alternative path for reliable communication between the source and the destination nodes. In this paper, a study has also been done on the effects of re-transmission delays spent in reliably delivering the packets to the destination nodes. The simulation result shows that AntHocMMP achieves a good packet delivery ratio in a wider communication range than the existing algorithms viz. R-ACO1, LAR, MMP and AntHocNet routing algorithms. The statistical analysis viz. *t* test and Anova-test have been performed that proves which in turn the proposed AntHocMMP dominates the other existing algorithms in a more significant way.

Keywords Energy efficiency · Energy path cost · Reliable communication · Residual energy · Re-transmission delay · Robust path · Routing protocols · Adaptive routing

1 Introduction

1.1 Background

Ant Colony Optimization (ACO) based routing algorithms for MANETs have been discussed in various research papers [1–7] that are more reliable and adaptive in nature. ACO is a random stochastic population based heuristic algorithm that stimulates the natural behaviour of ants [8–10]. In ACO algorithms, the ants as agents have been simulated by their foraging behaviour. Each ant communicates with the other ants by the deposition of the pheromones on the path during their traversal. These pheromone trials are used as local stigmergetic

✉ P. Vijayalakshmi
vijilak@karunya.edu

[1] Department of Computer Applications, School of Computer Science and Technology, Karunya University, Coimbatore, India

communication element to find the shortest path between the source and the destination. The amount of pheromone depends on the exploration of the positive feedback on the path that provides the best solution for the search mechanism in order to find the shortest path among the nodes in the network.

In ACO routing algorithms, the source node generates the ants to traverse across the network so as to find the shortest path between the source and its destination. The quality of the path is determined by the amount of pheromone laid down on that path based on the shortest number of hops. Therefore, the shortest path is selected to transfer the data packet between the nodes in the network. In this paper, a novel AntHocMMP ACO routing algorithm has been proposed. This algorithm selects the shortest path with minimized energy path cost. During the link disconnection, the adaptive re-transmission approach is utilized to detect the link disconnections and to find the alternate path. This is done to transfer the packets to the destination by achieving the robustness in the network.

1.2 Our contribution

In order to solve the problem of link disconnections between the nodes in a high dynamic MANETs, it is essential to construct a reliable and an adaptive path that is not likely to be disconnected for a long period of time. A novel ACO routing algorithm named AntHocMMP has been proposed based on the adaptability of paths between the source and its destination. This algorithm uses three combinatorial approaches that achieves a high packet delivery ratio and maximizes network lifetime of individual nodes with minimized re-transmission delays. The first approach computes the relative path using Max–Min–Path (MMP) routing algorithm [11] based on the residual energy and energy path cost functions of the nodes in the network.

The second approach is to find the optimal path using the *forward ants* to traverse across the network. Based on the energy path cost the *forward ants* update the pheromones on its path. This recurrent update of pheromone leads to a high probability of adaptiveness for dynamic network behaviour in MANETs.

The last approach predicts the link disconnections and finds the alternate path amongst the relative paths discovered by the MMP algorithm. In order to detect the link disconnections, each node broadcasts the *hello packets* to all its neighbourhood nodes. If a node predicts the link disconnection, it selects an alternative path based on probability of the pheromone deposition with a small-distance hop. In this paper, the performance of the proposed AntHocMMP algorithm is compared with the four existing algorithms viz. AntHocNet, R-ACO1, LAR and MMP. The AntHocNet and R-ACO1 are the well-known existing ACO routing algorithms and the LAR and MMP are not an ACO routing algorithms. The simulation results show that the proposed AntHocMMP algorithm outperforms the other existing algorithms in a more significant way.

1.3 Related works

The energy efficient routing in high dynamic network is a challenging task in MANETs [12] and many researchers are doing pioneering work with several routing algorithms [13–20] related to ant colony optimization techniques. Daisuke and Tomoko [21] proposed a position based routing algorithm that utilizes a robust path that is less likely to be disconnected among the GPRS enabled nodes in the network. During the link disconnections, the ant redistributes the pheromones and constructs an alternate path. This algorithm increases the control overhead of the processing nodes and thereby leads to high computational complexity. In addition to it, the author has not made attempts to consider the lifetime of the nodes for adaptive routing in the network.

Fernando and Teresa [22] have proposed a simple ant colony routing algorithm (SARA) that optimizes the routing process by adopting Controlled Neighbour Broadcast (CNB) mechanism. This algorithm incurs less overhead. But, the major disadvantage of this algorithm lacks in distributing the pheromone to the nodes in a dense network. A Position based Ant Colony routing algorithm (POSANT) [3] uses the multiple formulations of ants to construct paths during the link disconnection. During the link failure, this algorithm takes a shorter route establishment time and incurs less number of control packets. However, this approach takes a longer time to discover the route between the source and its destination nodes by varying the transmission range of the nodes in the network.

Mesut et al. [6] proposed the Ant colony based Routing (ARA) protocol that incurs less overhead as only a unique sequence number is transmitted for updating pheromones in the nodes and there is no interchange of routing information between the nodes in the network. However, the performance of this protocol tends to reduce a high network load that does not maintain pheromone concentration in the nodes.

Di Caro et al. [10] proposed the AntHocNet–ACO routing algorithm that reactively sets up multiple paths to speed up the routing process and then selects a path with a good heuristic probability based on the amount of pheromone deposition on the nodes. This approach fails to provide a solution during the link disconnections between nodes. The algorithm incurs high routing overhead as the ant traverses multiple paths to select the shortest path between the source and its destination nodes. In addition to it, the time complexity is also high as new path is constructed each time during the link disconnections.

Sardar et al. [23] proposed on demand ad hoc routing algorithm where the quality of service is enhanced by considering end-to-end parameters. The major drawback of this algorithm is that more number of control packets is used for enhancing the quality of service parameters that incurs high control overhead. Sisodia et al. [24], proposed a preferred link based routing protocol that incurs less control overhead during the routing process between the nodes in the network. The time complexity is higher as this protocol is involved in computing the shortest paths and the stabilizing information during the routing process.

Camilo et al. [1] proposed an Energy Efficient Ant Based Routing (EEABR) protocol that works efficiently by reducing the memory utilization of the nodes in the network. The drawback is as it tends to degrade its performance in a high dynamic network due to the assignment of random energy levels to the nodes in the network. This paper is organized as follows: Sect. 2 discusses the system models with its preliminaries. Section 3 gives a perspective on Ant colony optimization. Section 4 explains the design and construction of the proposed ACO routing algorithm. Section 5 compares the performance of the proposed ACO routing algorithm with other existing approaches in MANETs. Section 5.1 shows the simulation results and Sect. 5.2 shows the statistical analysis of the proposed AntHocMMP with the existing ACO routing algorithms. Section 6 discusses the conclusion and the summary of the proposed research work.

2 Preliminaries

The proposed AntHocMax–Min–Path (AntHocMMP) is an energy-efficient routing protocol for a wireless ad-hoc communication network that minimizes the energy consumption to route a packet and thereby enhances the network lifetime of the whole network. In this algorithm, a wireless network is represented as a Directed Graph $G(V, E)$, where the vertices V represent the nodes and edges E which in turn represent the link between the nodes in the network. The degree of a vertex v in the graph G is the number of edges that are connected to that vertex v. The available energy of the ith vertex (node, v_i) is termed as the energy level or residual energy, $E_r(v_i)$ and it is designated as the weight w of that node. This energy level is classified into two types: saturation energy level and threshold energy level. When the energy of the node attains zero, its energy level is termed as saturation energy level. The edges connected to the saturated energy levels node are removed, to avoid the data transmission through that node. The minimum energy that is sustained to conserve the energy of the node is termed as threshold energy level. The threshold energy level is computed with the available energy, $E_r(v_i)$ and the degree, deg (v_i) of the ith node. The Linear Congruential Method (LCM) is a Random Number Generation (RNG) that is meant to initialize the residual energy of the nodes in the network. Since all the nodes do not have equal energy levels in the networks, the LCM is used to initialize random energy levels to the nodes.

The energy required for transmitting one data unit from node u to v is termed as the energy path cost e_c. The energy path cost, $e_c(P(v_0, v_k))$ is based on the Euclidean distance for the available paths P between the nodes v_0 and v_k in the Directed Graph G. The cost functions are divided into two types: transmission cost and reception cost. The transmission cost is the sum of the energy path cost along the paths between the source and the destination. The reception cost is defined as path cost function for AntHocMMP protocol and is discussed in Sect. 4.2. The proposed routing protocol (AntHocMMP) maximizes the network lifetime which is an NP-hard problem [8, 25, 26] by conserving the energy level of individual nodes thereby reducing the saturated nodes in the network. The AntHocMMP protocol consists of two phases: selection phase and energy conservation phase. The selection phase selects the optimal node with the maximum residual energy and minimum path cost. The energy conservation phase conserves the energy of the individual nodes by maintaining its energy above the threshold level thereby maximizing the lifetime of the network [11].

2.1 Selection phase

To select optimal node with maximum residual energy and minimum path cost, the source node initially computes the threshold value, $E_{threshold}$ by using the degree of the node and the residual energy as,

$$E_{threshold} = \left(\frac{(E_r(v_i))}{deg(v_i)}\right) \tag{1}$$

where $deg(v_i)$ and $(E_r(v_i))$ represents the degree and residual energy of ith node v.Whenever the source node wants to initiate the data transmission, it broadcasts the request packet (RE_Request) requesting the residual energy of its neighbourhood nodes. When receiving the request packet (RE_Request) from the source node, the neighbourhood node unicasts the reply packet (RE_Reply) with its information back to the source node. The source node receives the residual energy of its neighbourhood nodes and computes it's the energy path cost. The nodes whose residual energy are greater than or equal to its threshold value $E_{threshold}$ are selected for data transmission.

These selected nodes are termed as r*elative energy nodes*, N_{re}. Amongst the relative energy nodes N_{re}, the optimal node with the minimum energy path cost is selected by the ants. All the relative energy nodes in that path are called as relative paths. Let $\{rpi\}$ be the set of possible relative paths from the source s to its destination d. The source node initiates the data transmission through that selected optimal node. Consequently the optimal node repeats the same procedure of this phase to forward its packets to its next hop neighbor until it reaches to the destination node. The path from the original source node to its final destination node is the optimal path $p0$.

2.2 Energy conservation phase

In this phase, as the data transmission occurs along the path $p0$ between the source and the destination node, the residual energy of the nodes along the path varies from time to time. Hence, to conserve the energy of the individual nodes, their residual energies are checked periodically to improve the lifetime of the individual nodes. The following constraints are periodically checked to sustain the nodes energy threshold level and to prevent them attaining to its saturation level.

2.2.1 *Constraint 1: to sustain the energy threshold level of the optimal selected node during transmission*

The source node sends the RE_request packet, requesting the residual energy of its optimal selected node. On receiving the reply from that selected node, the source node checks the residual energy of optimal selected node along the path $p0$. If the optimal selected node attains the threshold energy level, the source node restricts the data transmission through that selected optimal node along the path $p0$, so as to conserve its energy level and thus the source node selects the next optimal node amongst its r*elative energy nodes*, N_{re} for further transmission.

2.2.2 *Constraint 2: the residual energy is equivalent to the energy path cost*

The source node sends the RE_request packet, requesting the residual energy of the optimal selected node. On receiving the residual energy from the optimal selected node, the source node checks whether received residual energy is equivalent to the energy path cost along the path $p0$. If it is equivalent to the energy path cost, the source node restricts the data transmission through that node along the path $p0$ and the source node selects an another optimal node amongst its r*elative energy nodes*, N_{re} for further transmission.

2.3 Computation of relative path using MMP algorithm: an overview

Let $G(V, E)$ represent a wireless network comprising of n nodes, where V represents the nodes $v_i, v_i \in V, where\ \ i = 0, 1, 2, \ldots, n-1$ and E represents the path (transmission link) between the nodes. Let $E_r(v)$ be the available energy at node v. Let $c(uv)$ be the energy path cost required for transmitting a packet from node u to node $v\ where\ \ c(u, v) \in V$ [11].

Let $P(v_o, v_k)$ be the path from the node v_0 to $v_k$and it is denoted by: $P(v_o, v_k) = v_0, v_1, \ldots v_k$. The cost of energy path is denoted by $e_c(P(v_0, v_k))$ and it is defined as [11]:

$$\begin{aligned} e_c(P(v_0, v_k)) &= c(v_0, v_1) + c(v_1, v_2) + \cdots + c(v_{k-2}, v_{k-1}) \\ &\quad + \cdots + c(v_{k-1}, v_k), \\ e_c(P(v_0, v_k)) &= \sum_{i=0}^{k-1} c(v_i, v_{i-1}) \end{aligned} \tag{2}$$

Let the residual energy path, $r(p(v_0,\ v_k))$ is defined as:

$$\mathrm{r(P(v_0, v_k)) = \min_{\mathit{i}}(E_r(v_i) - c(v_i, v_{i+1}))}, \quad 0 \le \mathrm{i} < k \tag{3}$$

When a data packet is routed between the source and its destination the following are the energy constraints that need to be considered:

1. To find the *minimum energy path cost:*

$$\min(e_c(P(v_0, v_k))) = \sum_{i=0}^{k-1} c(v_i, v_{i+1}) \tag{4}$$

2. To select the *maximum residual energy node with minimum energy path cost:*

$$\mathrm{max(r(P(v_0, v_k))) = \min_{\mathit{i}}((E_r(v_i) - c(v_i, v_{i+1})))}, \quad 0 \le \mathrm{i} < k \tag{5}$$

3. To find the *minimum residual energy of the selected node:*

$$\min(r(P(v_0, v_k))) = \min_i((E_{threshold}) - c(v_i, v_{i+1})), \quad 0 \le i < k \tag{6}$$

where $E_{threshold}$ is $\left(\frac{(E_a(v_i)}{deg(v_i)}\right)$ and $deg(v_i)$ represents the degree of ith node v.

If $\min(r(P(v_0, v_k)))$ is equivalent to the energy path cost and the residual energy attains the threshold value the data transmission through that node is restricted. i.e.,

$$\begin{aligned} &\text{if}(\min(r(P(v_0, v_k))) = \min\ (e_c(P(v_0, v_k))) \quad \text{or} \\ &\quad \min\ (r(P(v_0, v_k)) = E_{threshold})) \end{aligned} \tag{7}$$

With these three constraints, the MMP algorithm sustains the minimum residual energy at its threshold energy level by preventing the nodes attaining saturated level thereby reducing the number of saturated nodes in the network. Subsequently the MMP facilitates more number of data transmissions among the nodes in the network by incurring less overhead thereby enhancing the lifetime of the network.

By invoking these constraints, the energy conservation on individual nodes is maximized and hence the maximum number of packet is sent along the optimal path before the source and its destination gets disconnected.

2.4 The algorithm *Max–Min–Path (MMP)*

//$s \leftarrow$ Source Node, d $\leftarrow$ destination node, EG $\leftarrow$ *Energy Graph*, $adj(s) \leftarrow$ adjacent list that represents the neighbor of the *source node*, $weight(u, v) \leftarrow$ Capacity of edge (u, v), $width(u) \leftarrow$ weight function of the node $u, deg(s) \leftarrow$ degree of the source node s//

```
Algorithm Max-Min-Path - MMP($Energy, D, EG$)
Begin
Step 1: Initialize the required variables
        $route \leftarrow \phi$, $cost \leftarrow 0$
Step 2: Calculate the threshold energy
        $E\_threshold\_energy = ceil(width(u))/deg(s)$
Step 3: For every node in the network check the energy path cost
    for all adjacent vertices of $s$ ,say $n_A$ do
        if ( $width(s) \geq E\_threshold\_energy$) $i \leftarrow 1, \ldots n_A$ $\underset{nA}{e}(\cos t) = \underset{i}{e}(\cos t)$
        $e\_\cos t = \underset{i}{\min}(\underset{i}{e}(\cos t))$, $i \leftarrow 1 \cdots n_A$ $route \leftarrow route \| v$
        $new\_width \leftarrow width(s) - e\_\cos t$
        $start\_vertex \leftarrow v$
Step 4: Store the relative path as a new energy graph (EG') that satisfies the threshold level
        $EG' = EG - v$
        if ( $v \neq dest\_vertex$) do
        call Max-Min-Path ($new\_width, start\_vertex, EG$) $route, e\_\cos t$),
        $e\_less\_\cos t = \min(e\_less(\cos t)), \underset{j}{j} \leftarrow 1,2, \cdots n_A - 1$
Step 5: Repeat the function until all relative paths are identified between source and destination
        call Max-Min-Path ($e\_\cos t, s, EG - e\_less\_\cos t$)
        end if
        end if
        end for
end
```

By adhering to these three constraints, the proposed AntHocMMP algorithm uses ant as agent to traverse from one node to another node in the network. An ant can migrate from one to another node to transfer a data packet. An ant traverses from node u to node v at time t, only if node v is a neighbor of node u and both the nodes are in the same communication range. Each node in the network is provided with local buffer storage where both the nodes u and node v can update the local information of their movements in the network. When more ants traverse the nodes, they update the information in an arbitrary order that prevents the node attaining saturation energy level and maintaining the nodes energy to its threshold level. This enhances the individual lifetime of the nodes in the network. Since the lifetime of the individual nodes are enhanced as the saturated nodes in the network are reduced. Therefore, the proposed AntHocMMP algorithm increases the lifetime of the whole network to a greater extent.

3 Ant colony optimization routing: an perspective

The Ant Colony Optimization (ACO) routing technique uses a meta-heuristic approach to solve the problems focused by the communication networks. In particular, the ACO approach has been applied to routing problems to find the shortest path for communication between the nodes in the network. ACO routing is based on the principle of foraging behaviour of ants. Naturally ants will wander in search of food. When it finds the food, it reaches its colony by laying down the pheromone on the paths it traverses by allowing the other ants to follow. The other ants will eventually follow the pheromone trial to find its food and reaches its colony back by updating the pheromone trail on the same path used for traversal. Finally, the amount of pheromone on paths along which many ant traverses frequently will be the shortest path between the source and its destination. The amount of pheromones on the path remains high for a period of time and tends to evaporate if no ants traverse through that path. Subsequently, the ants may lose the path during its traversal.

In ACO routing algorithms, ant as agents, traverse the network to search a path from a source node to a destination node. When an ant finds a path, it lays down the pheromone on the traversed path and the amount of pheromone is based on the communication range, and the number of hops. The amount of pheromones on the link is maintained by the pheromone table in each node. The pheromone table contains the destination node, the source node, the neighbouring node, the relative paths, and the energy path cost of the link between the nodes. The value p_{uv} of a node u's pheromone table is the amount of pheromone on a link (u, v) of the paths to the destination node d. Thus, a data packet is allowed to transfer to the destination node based on the amount of p_{uv} of the pheromone on the link (u, v) of the path in the network which determines how good the link is (u, v). The pheromone table also updates the energy cost function ep_c (v) of the node v along the path in the network. The stack routing table contains an array of the visited nodes energy path cost, the source id, the next-hop node id and the hop count.

4 Design of AntHocMax–Min–Path (AntHocMMP) Algorithm

In this section, the proposed ACO routing algorithm uses the energy model to find the optimal shortest path that improves the adaptability and enhances the network lifetime in MANETs. In order to establish a path from a source s to a destination d, the source s generates a *forward ant* and broadcasts it to all of its neighbours [13]. Initially, a small amount of pheromone τ_0 is assigned to all nodes in the network. The aim of the *forward ant* is to search a path from source s to destination d in the network. τ_{uv} is the pheromone trial for a path or a link (u, v) in the connected Graph G(V, E). These pheromone trials are the depositions of pheromones by the ants that traverse the path from source s to destination d. Each node in the network maintains a pheromone table. The *forward ant* chooses the next node based on the pheromone in the pheromone table of its neighbor node. The transition probability [27] for the Kth forward ant from one state i to the next state j for Ant System (AS) is given by [28],

$$P^k_{uv}(t) = \begin{cases} \dfrac{[\tau_{uv}(t)]^{\alpha}[\eta_{uv}]^{\beta}}{\sum\limits_{k \in allowed_k} [\tau_{uv}(t)]^{\alpha}[\eta_{uv}]^{\beta}} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where τ_{uv} is the trail intensity on edge (i, j), $\eta_{uv} = (1/d_{uv})$ is the heuristic function, d_{uv} is the Euclidean distance calculated for that particular edge, α is the relative importance of the trail ($\alpha \geq 0$), β is the relative importance of the visibility ($\beta \geq 0$) $allowed_k$ is the set of available states (uth to v th state) from which the kth ant can choose [28].

In AntHocMMP, a dynamic network is maintained by calculating the evaporation factor ρ where $0 \leq \rho < 1$, at regular intervals in the pheromone table as follows, [28]

$$\Delta\tau_{uv} = \sum_{k=1}^{m} \Delta\tau^k_{uv} \tag{9}$$

where $\Delta\tau^k_{uv}$ is the change on pheromone concentration. A data packet is transferred based on the amount of pheromone in pheromone table. When a node u sends a data packet to a destination d through the neighbour node v and, the selection of the neighbour node is based on the probability as follows [28],

$$P^d_{uv}(t) = \begin{cases} \dfrac{[\tau_{uv}(t)]^{\beta}}{\sum\limits_{d \in uv} [\tau_{uv}(t)]^{\beta}} \end{cases} \tag{10}$$

where β corresponds to the exploration of randomness and lies between $0 \geq \beta \geq 1$.

When the *forward ant* reaches its destination, it gets expired and the new ant called *backward ant* is created by the destination node. The *backward ant* will traverse the same path as *forward ants* and update the pheromone at each node and reaches back to its source node. The pheromone table at each node is also been updated simultaneously. The traversed path by the *backward ant* is the shortest path that is determined by the quantity of the pheromones laid down by the ants that frequently uses that path. The energy path cost is based on the number of hops and the transmission delay between the two nodes. The proposed AntHocMMP is based on the following three stages and it is discussed in Sects. 4.1–4.3

Stage 1: Selection of relative paths through MMP algorithm
Stage 2: Robustness of energy path cost by forward ants
Stage 3: Reconstruction of relative paths on link disconnections.

4.1 Stage 1: Selection of relative paths through MMP algorithm

Let $G(V, E)$ represents a wireless network comprising of n nodes, where V represents the nodes $v_i, v_i \in V, where\ i = 0, 1, 2, \ldots, n-1$ and E represents the path (transmission link) between nodes. Let $E_r(v)$ be the residual energy at node v. Let $c(u, v)$ be the energy path cost required for transmitting a packet from node u to node $v\ where\ c(u, v) \in V$. Let $P(v_o, v_k)$ be the path from the node v_0 to v_k and it is denoted by: $P(v_o, v_k) = v_0, v_1, \ldots v_k$ [11]. The energy path cost is denoted by $e_c(P(v_0, v_k))$ and it is defined as:

$$\begin{aligned} e_c(P(v_0, v_k)) &= c(v_0, v_1) + c(v_1, v_2) + \cdots + c(v_{k-2}, v_{k-1}) \\ &\quad + \cdots + c(v_{k-1}, v_k), \\ e_c(P(v_0, v_k)) &= \sum_{i=0}^{k-1} c(v_i, v_{i-1}) \end{aligned} \tag{11}$$

To select the Relative paths (Rpath) with *maximum residual energy nodes with minimum energy path cost is given as:*

$$\begin{aligned} \text{Rpath} &= \max(r(P(v_0, v_k))) \\ &= \sum_{i=0}^{k-1} \left(\min_i (\mathrm{E_r(v_i) - c(v_i, v_{i+1})}) \right), \\ & 0 \le \mathrm{i} < k \end{aligned} \tag{12}$$

The relative paths discovered by the MMP algorithm are utilized for finding the optimal path amongst the other nodes in the network. The lists of relative paths are stored in the pheromone table. In AntHocMMP, the source node generates *forward ants* to traverse along the relative paths to reach the destination nodes. Once these ants reach the destination node without any link disconnections, the destination node generates the *backward ant* through that path and reaches the source node indicating it's the optimal path among the other relative paths discovered by MMP algorithm. This optimal path is stored in pheromone table for further data transfer operations.

4.2 Stage 2: robustness of energy path cost by forward ants

The task of these *forward ants* is to search a path from the source s to the destination d and to update the pheromone for the paths that the ants have traversed. The forward ants generated by source node is allowed to traverse to all its relative paths generated by MMP algorithm [11] as in Eq. (12). As it traverses, the ant updates the pheromone table of each node visited along its relative paths. Firstly, the *forward ant* updates the information of the previous node P_n in the pheromone table such as the hop count HP_c, and the energy path costs ep_c. This information in the pheromone table is used by the nodes for link disconnections during traversal along its relative paths. Secondly, the *forward ant* updates the pheromones on the each node to evaluate the robustness of the path. The proposed AntHocMMP, uses the energy path cost functions to compute the robustness of the path based on the transmission cost and the Euclidean Distance in the following Sects. 4.2.1 and 4.2.2.

4.2.1 *Robustness of energy path cost based on transmission cost*

The energy path cost is computed based on the total energy cost spent by traversing along the path $P(v_o, v_k)$ between the source and its destination. When a data packet is routed along the path(v_0, v_k), an energy decrement operation is performed on each node by *forward ants* by reducing its energy level by the amount of energy that is required along the path to transmit the data packet and it is denoted by the Eq. (13) as:

$$v_k : E_k(v_i) = E_r(v_i) - c(v_i, v_{i+1}), \quad 0 \le i < k \tag{13}$$

Therefore, the energy path cost function is defined in as:

$$E_k(v) = E_r(v_i) - e_c(P(v_0, v_k)), \quad 0 \le i < k \tag{14}$$

4.2.2 *Robustness of energy path based on Euclidean distance*

The energy decrement operation computed by the *forward ants* is highly correlated with the Euclidean distance between the source and destination node. Let there be n nodes in the energy path E_p (v_0, v_k) where v_0 is the source node and v_k is the destination node and the packets that are transmitted through the intermediary nodes $v_1, v_2, \ldots v_{k-1}$. Let the distance between v_0 and v_1 is r_0, v_1 and v_2 is r_1 and v_{k-1} and v_k is r_{k-1}. Since the distance plays a vital role in forwarding the packet from v_0 to v_k, it is also taken into consideration for computation of the path cost. Therefore the energy path cost function based on the Euclidean distance is as follows as:

$$E_k(v_i) = E_a(v_i) - \left(d^3 * .001\right), \quad 0 \le i < k \tag{15}$$

where d is the actual distance between the source and the destination and is calculated as:

$$d = \sum_{i=0}^{k-1} r_i, \quad d = r_0 + r_1 + \cdots + r_{k-1} \tag{16}$$

Thus, the robustness of the path cost function is calculated as:

$$ep_c(f(v_i, v_k)) = \min(E_k(v_i))/\sum_{i=0}^{k-1} r_i, \quad 0 \leq i < k \tag{17}$$

The robust energy path cost function ep_c. is computed by the *forward ants* and it is maintained in pheromone table for any further update. This energy path cost function is also used by the *backward ants* as they traverse backwards from destination to the source node. The Table 1 describes the steps involved in *AntHocMMP* algorithm.

The procedure *forward-ant* in Table 2 is executed for finding the best robust path after the energy path cost functions of the nodes are identified by the forward ants. If the energy path cost of the current node is zero, then the path is dropped for not transmitting the data packets. If the stack routing table entry for the current node is the same as the destination, then the stack table id is labeled as "SUCCESS", and the data packet is forwarded to the next hop neighbor that is stored in the stack routing table, otherwise it is buffered at the current node. If the stack routing table id is labeled as "IN-REPAIR", it means that the ants are looking for the robust path to reach the destination with a high energy path cost function. If the stack table id is set to "FAILURE", then it is set to "IN-REPAIR" and the ants are tend to call the procedure *traverse-request* as in Table 3.

Table 1 Algorithm of AntHocMMP

```
Algorithm AntHocMMP (MMP, Rpath, $ep_c$, phr-table, st-rt-tab, dest-node, srce-node, neigh-node, current-node)
Begin
//MMP: Max-Min-Path routing algorithm, Phr-table: Pheromone table, st-rt-tab: stack routing table, dest-node: destination node, srce-node: source node, RPath: Relative path, drop: remove the path from usage, robust-path: robust path, link-disconn: link disconnections in path, optimal-path: optimum minimum energy path//
Step 1: Find the relative paths from MMP algorithm
            RPath = call Max-Min-Path (Energy, S, D, EG)
                if (RPath = 0)
                    drop (RPath);
                    return;
                else goto step 2;
                end if
Step 2: Generation of Forward ants by the source node
            robust-path = call Forward-ant (RPath, $ep_c$, neigh-node, current-node, dest-node)
Step 3: Update of Pheromones in the pheromone table by the visited nodes through the relative paths
            if (robust-path $\neq$ 0)
              update -phr-table = phr-table (current-node, neigh-node)
              stack-routing-table $\leftarrow$ update-stack-routing-table(current-node)
            end if
Step 4: Generation of Backward ants by the destination nodes
            if (forward ants = dest-node)
              call Backward-ant (RPath, $ep_c$,srce-node, dest-node, neigh-node, VLEPath-cost-nodes)
              update phr-table = call phr-table (current-node, neigh-node)
              call Neigh-reply(srce-node, dest-node))
            end if
Step 5: Detection and maintenance of link disconnections during traversal of the ants
            if (robust-path = 0)
              link-disconn = call link-disconnection (current-node, $ep_c$, neigh-node, RPath)
            end if
Step 6: Select the optimized shortest path between the source and destination
            if (link-disconn = 0)
              optimal-path = robust-path
            else
              RPath = call Max-Min-Path(Energy,S,D,EG)
            end if
End
```

Table 2 Traversal of forward Ant

```
Procedure Forward-ant (RPath, $ep_c$ ,neighnode, current-node, dest-node)
Begin
// $ep_c$ : the list of the energy path costs of the $N_{re}$ nodes, st-rt-id: stack routing table identifier, st-rt-tab : the stack  table, $HP_c$ : the hop count, drop: remove the path from usage//
Step 1: Check if the energy path cost of the current node is zero, then drop the packet
        if ($ep_c$ (current-node) == 0)
          drop ($ep_c$);
         return;
        end if
Step 2: Check if the current node is the destination node then restore its energy path cost
        dest-node := $ep_c$ (current-node) ← dest-node;
Step 3: Update the Stack routing table
        update-st-rt-tab ← stack-routing-table(current-node, dest-node);
Step 4: Check the three possible states in the stack table to route a packet to the destination
        case (st-rt-tab ←  st-rt-id):
           SUCCESS
                current-node := st-rt-tab-next-hop;
                update-stack-routing-table ← stack-routing-table(current-node);
                 $HP_c := HP_c +1$;
Step 5: Check if the current node does not find a robust path then store the packet in the stack
           FAILURE
                Enqueue(packet, current-node);
                $HP_c := HP_c +1$;
           IN-REPAIR
                st-rt-id := FAILURE;
                Enqueue(packet, current-node);
Step 6: Procedure traverse-request is invoked if the current node is labeled as FAILURE or IN-REPAIR
                call traverse-request (current-node, dest-node);
        end case
end
end procedure Forward-ant
```

The procedure *traverse-request* is active from the source node and the ants flow through the relative paths of effective path cost function. A list of visited energy path cost nodes are maintained in the stack routing table for adaptive flow of ants through that path. The *forward ants* chooses the next hop neighbour according to the transition probability function as in Eq. (8) of the energy path cost associated with the link between the nodes in the network. An array of nodes visited by the ants during traversal are stored as HP_c list (hop count) along the relative paths. This array is used by the *forward ants* to avoid loops created when choosing the next hop neighbour among the nodes that are not visited. This array list is stored in the stack routing table. This list also helps the *backward ants* to traverse back in finding the path back to the source node. Each node in the network is associated with the timer called as evaporation time which is increased by the evaporation factor as in Eq. (9) for every hop of the data packet in the relative path. When a data packet is received by the node, it is buffered in the node stack for limited period and it evaporates if it does not take a hop to its neighbour with in a pre-defined period of time.

When the *forward ant* reaches the destination through the relative energy path cost function, the procedure *Backward-ant* is implemented as in Table 4. In this procedure, the *forward ant* is killed on reaching the destination and the *backward ant* is created. The stack routing table is updated for the local movement of the *backward ant* to find its path back to reach the source node. This ant uses the visited nodes array list to traverse backwards. This procedure in turn calls the procedure *Neigh-reply* as in Table 5. When a *backward ant* traverses to find a path backward and if it senses there is only one neighbor or if the energy path

Table 3 Maintaining the pheromone evaporation factor of the visited nodes

```
Procedure traverse-request (current-node, dest-node)
Begin
// $ep_c$ : the list of the energy path costs of the traversed paths, st-rt-tab: the stack table, $HP_c$ : the hop count, VLEPath-cost: list of visited energy path cost, neigh-node: neighbor node, current-node: current node//
Step 1: Check if the current node is not the destination node and energy path cost is greater than zero, then
        route the packet
              if (current-node ≠ dest-node) && ($ep_c$ (current-node) > 0)
Step 2: Update the stack routing table of the current node
              current-node ← st-rt-tab-next-hop-node;
Step 3: Calculate an array list of energy path cost of visited nodes
              for (i=0;i<= $HP_c$ ;i++)
              VLEPath-cost := current-node(i);
Step 4: Update the pheromone evaporation timer for the nodes in the array list
              update-pheromone-evaporation-timer(neigh-node) < = 1;
          end if
Step 5: Go to step 1, if the condition in step 1 is not true then drop the path
              drop($ep_c$);
end
end procedure traverse-request
```

Table 4 Traversal of backward Ant

```
Procedure Backward-ant (VLEPath-cost-node, RPath, $ep_c$, srce-node, dest-node, neigh-node)
Begin
// $ep_c$ : the list of the energy path costs of the traversed paths, st-rt-tab: the stack routing table, $HP_c$ : the hop count, VLEPath-cost: list of visited energy path cost, dest-node: destination node,  kill-forward-ant : destroys ants reaching destination nodes//
Step 1: Check if the source node is not the current node, then update the stack routing table
         if (srce-node ≠ current-node)
        update(stack-routing-table(current-node, neigh-node, dest-node));
Step 2: Find the next hop neighbor to traverse to source by using the array list of visited energy path cost
        nodes
              for (i=0;i<= $HP_c$ ;i++)
              next-hop-node := VLEpath-cost-nodes(i);
              current-node ← next-hop-node;
Step 3: Go to step 1, if the condition in step 1 is not true then dequeue the node from the stack, if source node
        does not have its neighbor to traverse
              dequeue(current-node);
              stop-creating-ants; // stops source node generating the forward ants//
              kill-forward-ant; //destroys ants reaching source nodes//
            end if
Step 4: call the procedure Neigh-reply for the source node having degree less than one for success traversal
         call Neigh-reply(srce-node, dest-node);
end
 end procedure Backward-ant
```

cost function of its neighbour is greater than the residual energy of the destination node, the stack routing table id is labeled as “SUCCESS”, otherwise the stack routing table id is labeled as “FAILURE” to buffer the data of the current node.

4.3 Stage 3: reconstruction of paths on link disconnections

To reduce the packet loss due to link disconnections, each node in the relative path must be within the communication

Table 5 Update of neighbor node reply

```
Procedure Neigh-reply (srce-node, dest-node)
Begin
// $ep_c$ : the list of the energy path costs of the traversed paths, st-rt-tab: the stack table, $HP_c$ : the hop count, $E_r$: Residual energy, srce-node: source node, deg: degree of the source node, dest-node: destination node//
Step 1: Check if the condition for degree of source node is one or energy path cost is greater than residual
        energy of source or destination node, update stack identifier of that node as success
          if (($ep_c$(srce-node || dest-node) > $E_r$ (dest-node)) || (deg(srce-node) = 1))
                    update-st-rt-tab = st-rt-tab-id:= “SUCCESS”;
Step 2: Remove the current node from the stack, as there is no path for traversal
                    dequeue(current-node);
                    srce-node:=stop-creating-ants; // stops source node generating the forward ants//
Step 3: Check if condition of step 1 is not true then, label the stack as FAILURE, and check for the
        next relative path using MMP algorithm
                    update-st-rt-tab = st-rt-id:=”FAILURE”;
          end if
end
end procedure Neigh-reply
```

Table 6 Link disconnection

```
Procedure link-disconnection (current-node, neigh-node, RPath)
Begin
// $ep_c$ : the list of the energy path costs of the traversed paths, st-rt-tab : the stack table, $HP_c$ : the hop count, MMP: Max-Min-Path algorithm, phr-table: pheromone table, current-node: current node, dest-node: destination Node, Rpath: relative path, srce-node: source-node //
Step 1: Check if the energy path cost of the current node is zero or there are no neighbors for the source node,
        then choose an alternate relative path from MMP algorithm
          if (($ep_c$(current-node) == 0) || (Neigh-reply(current-node) == null)
          RPath = call MMP (Energy, S, D, EG);
Step 2: An alternate relative path from MMP algorithm is utilized by AntHocMMP during link disconnections
          call AntHocMMP (MMP, Rpath, $ep_c$, phr-table, st-rt-tab, dest-node, srce-node, neigh-node, current-node);
end
end procedure link-disconnection
```

range. When node in the relative path reaches out of its communication range then there is a problem of link disconnections in the network. Most ACO routing algorithms use the “hello packets” to identify the link disconnections. In AntHocNet, the link disconnections are predicted using GPRS techniques [21]. The proposed AntHocMMP algorithm uses an adaptive re-transmission approach to detect the link disconnections and selects a new relative path in dynamic network behavior. Therefore, the *forward ant* updates the pheromone trails to select the next hop neighbor as in Eq. (10). If a node is found to be less likely visited by the ants the evaporation factor is computed as in Eq. (9) for that node and it is updated in the pheromone table. If a *forward ant* or a *backward ant* loses its relative path or do not find its next-hop neighbour for a pre-defined interval of time, the data packet is buffered in the stack routing table of that current node and an alternate path is immediately reconstructed from other relative paths discovered through MMP algorithm. The link failure notification messages are alarmed to all the nodes in that path. Therefore, this approach achieves an efficient utilization of energy nodes in the network. The Table 6 list the procedure *link-disconnection* used by the proposed AntHocMMP algorithm.

5 Simulation results

The proposed AntHocMMP algorithm has been simulated using NS2.34 simulator. The crucial performance metrics viz. the packet delivery ratio, the average end-to-end delay, the hop-count and the control overhead have been analyzed to evaluate the performance of the proposed AntHocMMP algorithm. The performance metrics are analyzed by varying node speed, communication range and the number of nodes. Initially, nodes are placed in a field of 3000 × 3000 m^2. The total number of mobile nodes is varied from 50 to 150 nodes. The transmission range is varied from 50 to 400 m and the maximum speed of the mobile nodes is varied from 5 to 50 m/s. The Table 7 shows the simulation parameters used for the simulation of AntHocMMP algorithm. In simulations, the evaporation factor of pheromone is1 ms, evaporation rate is set to 0.2 ms.

Table 7 Simulation parameters

Parameter	Value
Area dimension	3000 × 3000 (m^2)
Number of nodes	50,100 and 150 nodes
Maximum speed of mobile nodes	5–50 (m/s)
Transmission range of mobile nodes	50–400 (m)
Mobility model	Random way point (RWP)
Simulation time	1800 (s)
Hello packet broadcast interval	1 (s)

The performance of the AntHocMMP algorithm has been compared with AntHocNet, R-ACO1, LAR and MMP algorithms and the simulations results are shown in Sect. 5.1. In AntHocMMP, the forward and backward ants behave proactively like simple ACO routing algorithm to maintain the pheromone diffusion and reconstruction of paths during link disconnections. Each node in AntHocMMP periodically broadcasts *hello packets* to all its neighbors for detection of link disconnection or energy exhaustions of nodes. The link failure notification messages are alarmed to all nodes in the link so as to update the information in the pheromone table. Similar to the AntHocMMP, the AntHocNet also periodically broadcasts the *hello packets* to all its neighbours to detect link disconnections. The link failure messages are forwarded to all its neighbour nodes to update the failure notifications. The drawback of the AntHocNet algorithm lacks in considering the individual lifetime of the nodes in the network. The AntHocMMP algorithm enhances the lifetime of the individual nodes by sustaining the energy level to its threshold. Therefore, this algorithm reduces the cost of communication between the nodes in comparison with the AntHocNet algorithm. The packet delivery ratio is computed as the ratio of the total number of packets received

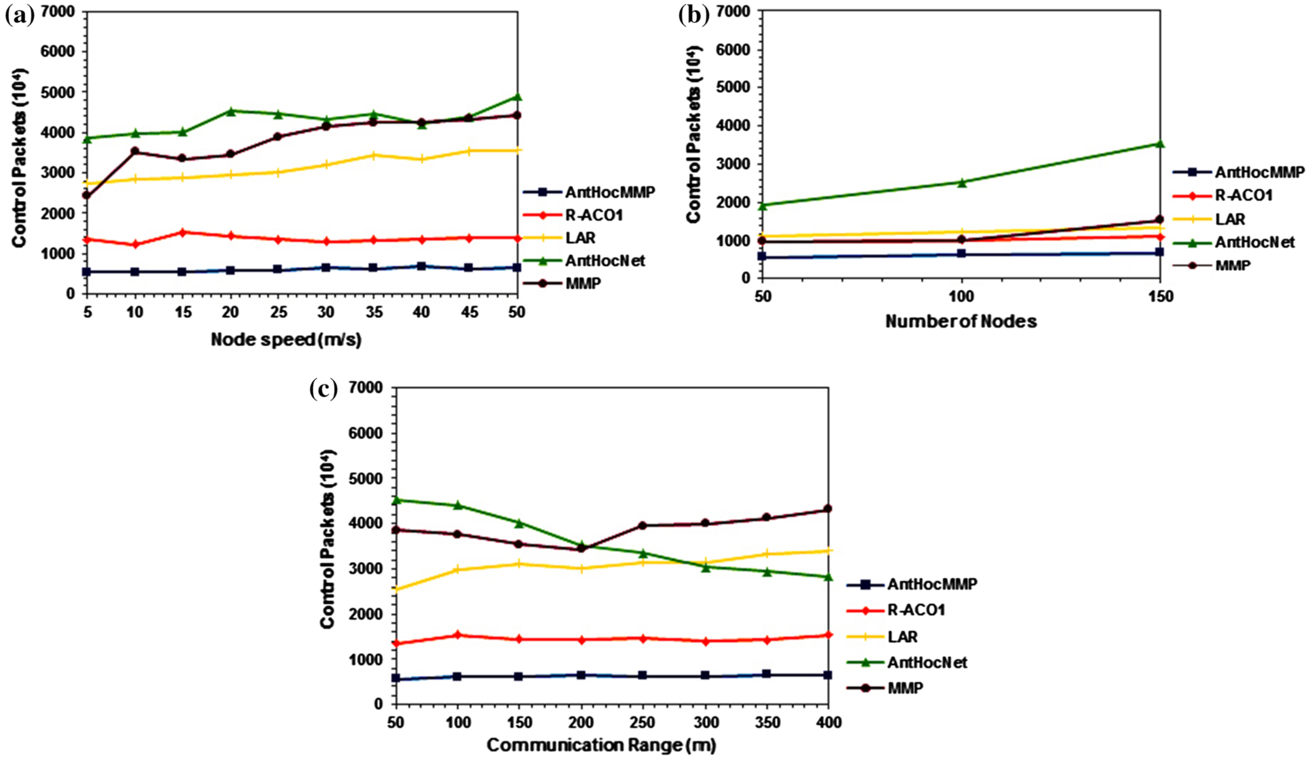

Fig. 1 The Analysis of control overhead. **a** The node speed (m/s). **b** The number of nodes. **c** The communication range (m)

by the destination node. The average end-to-end delay of a packet is the time taken by the packet to reach the destination. The hop-count is computed as the packets are received at the destinations.

5.1 Simulation results of ACO routing algorithms

The pheromone update by the forward ants tends to highly reduce the cost function of the path in the network. A simple ACO routing algorithm like R-ACO1 uses only backward ants to update pheromone whereas the AntHocMMP algorithm uses both the forward and the backward ants to update pheromones. This reduces the cost metrics and also detects energy exhaustion and link disconnections during routing process.

5.1.1 Performance evaluation AntHocMMP algorithm

The proposed AntHocMMP algorithm has been simulated and the experimental results are obtained. The experiments has been conducted to analyze the control overhead, the packet delivery ratio, the average end-to-end delay and the hop-count by varying the node speed, the number of nodes and the communication delay. The performance of the proposed AntHocMMP algorithm was also compared with the existing algorithms viz. AntHocNet, R-ACO1, LAR and MMP.

Analysis on control overhead The Fig. 1a shows the variation of the control overhead by varying the max-speed of the mobile nodes from 5 to 50 m/s. The number of nodes taken for this experiment is 150 nodes with the transmission range set as 400 m The proposed AntHocMMP algorithm incurs less control overhead than the existing algorithms viz. AntHocNet, MMP, R-ACO1 and LAR. This is due to the fact that the AntHocMMP algorithm finds the optimized paths among the relative paths discovered by the MMP algorithm. During frequent link disconnections in the dynamic network, the AntHocMMP outperforms by transferring low number of control packets across the network. Therefore, it is found that there is no significant change in the control overhead by varying the node speed from 5 to 50 m/s.

The Fig. 1b shows that the variation of the control overhead by varying the number of nodes from 50 to 150 nodes. The Max-speed of the mobile nodes has been set as 30 m/s and its transmission range as 350 m. The AntHocMMP incurs less control overhead than the other existing algorithms. This due to the fact that the AntHocMMP finds more alternative paths between the source and destination during the link disconnection as the number of nodes increases. Since additional alternate paths are available, the source node reconstructs the path using the broadcast message with less control overhead. But, the

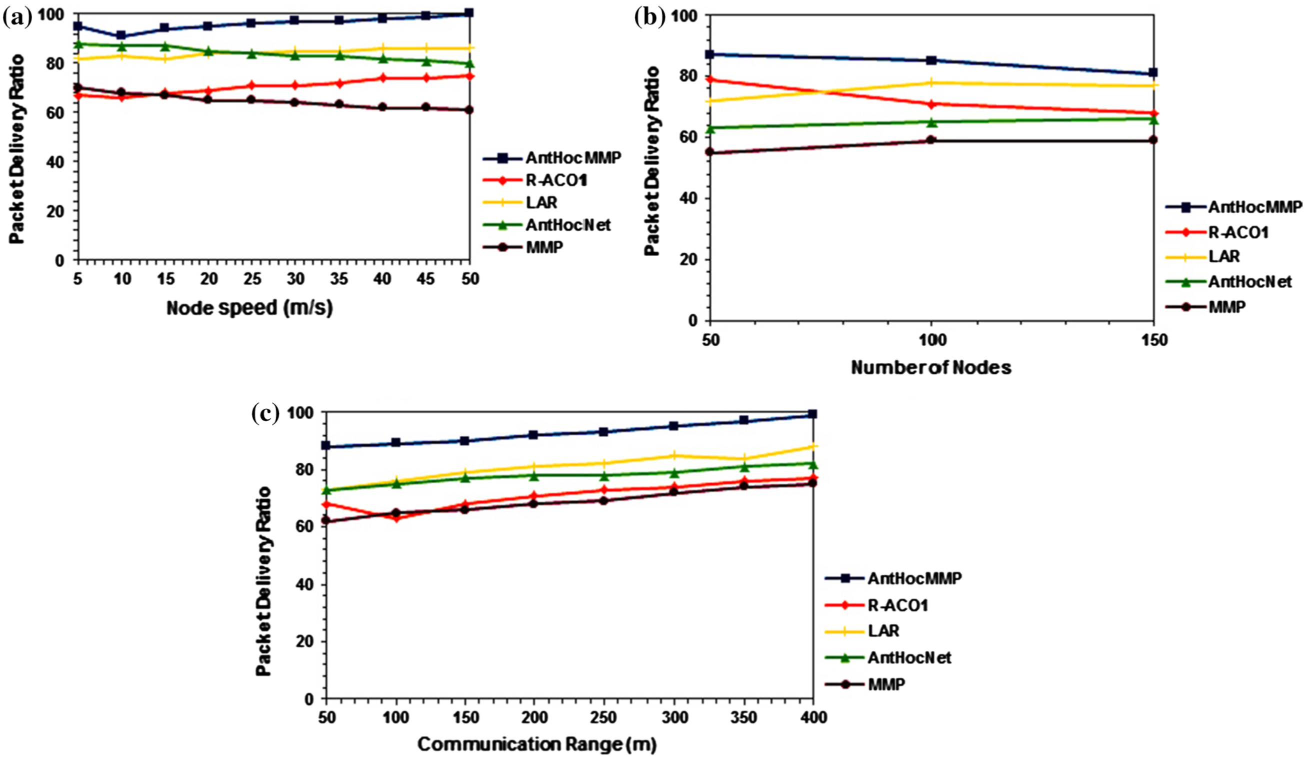

Fig. 2 Analysis of packet delivery ratio. **a** The node speed (m/s). **b** The number of nodes. **c** The communication range (m)

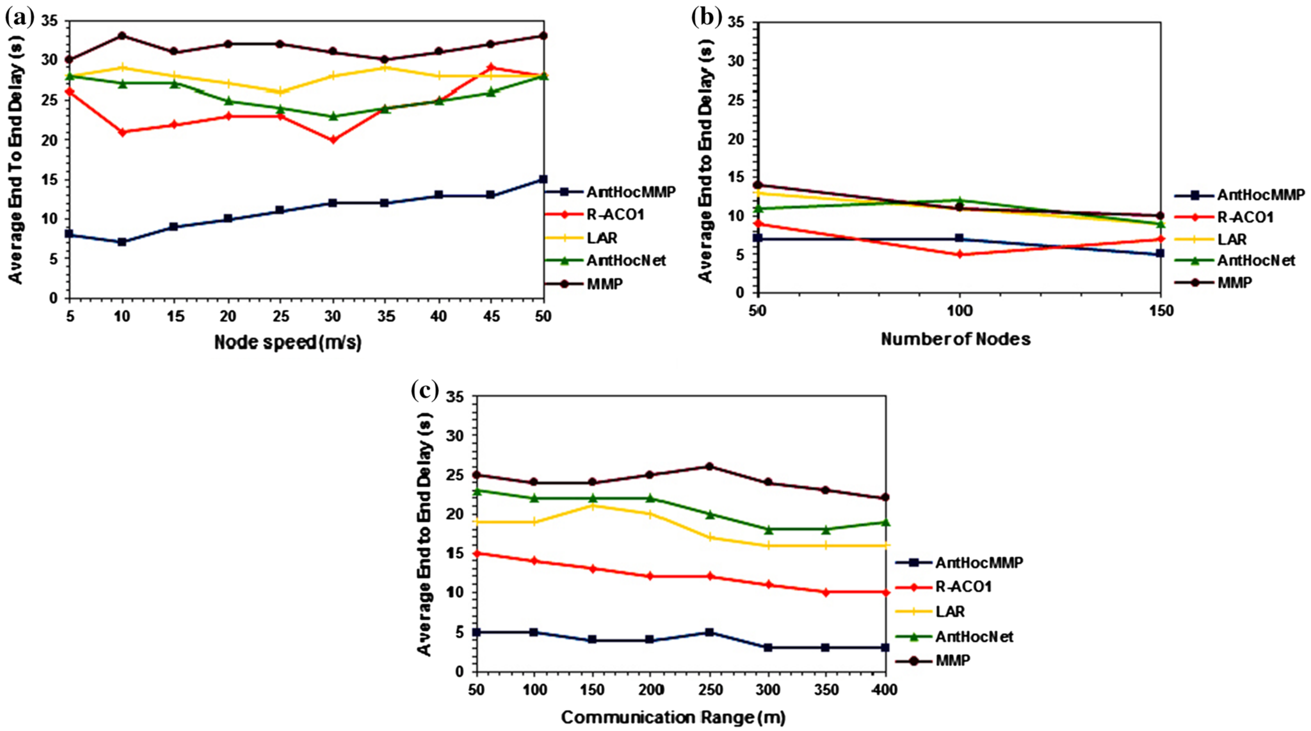

Fig. 3 Analysis of average end-to-end delay. **a** The node speed (m/s). **b** The number of nodes. **c** The communication range (m)

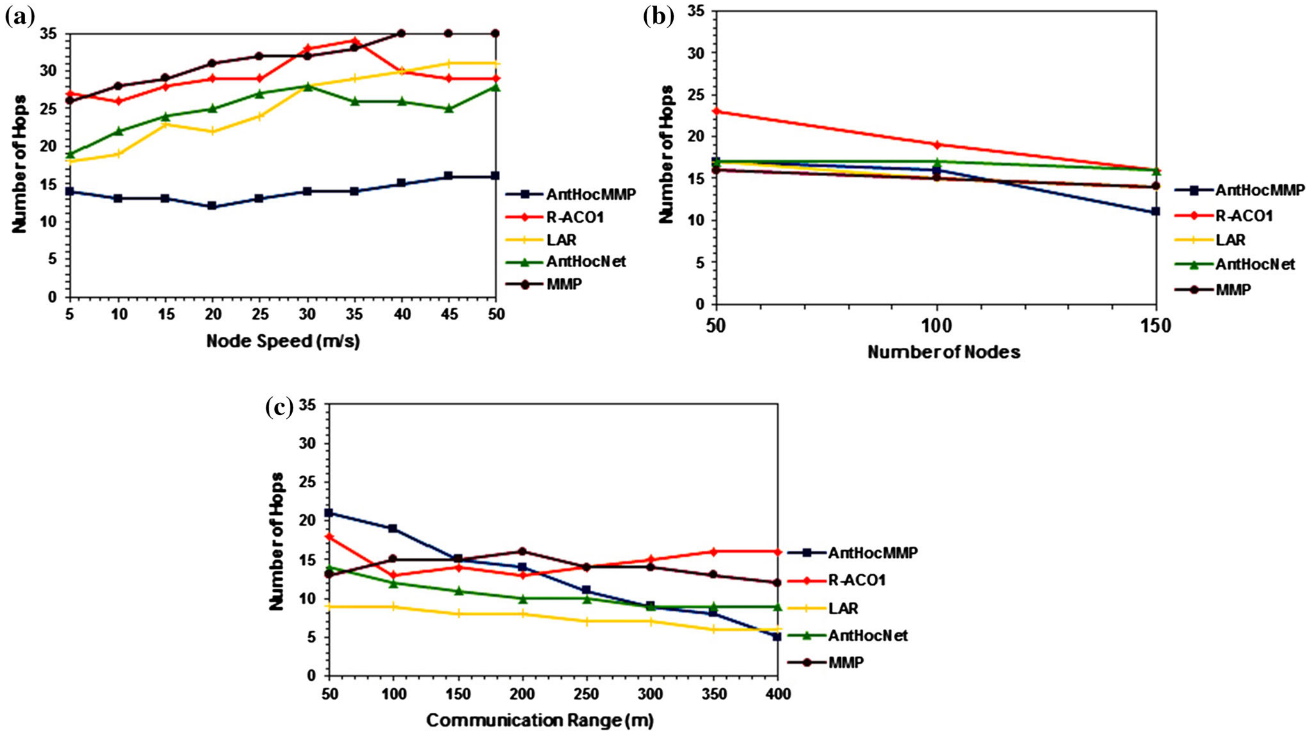

Fig. 4 Analysis of Hop-count. **a** The node speed (m/s). **b** The number of nodes. **c** The communication range (m)

LAR and AntHocNet algorithms incur high overhead as the time taken to repair the broken path becomes high in a dense network.

The Fig. 1c shows the variation of the control overhead by varying the communication range from 50 to 400 m. The number of nodes considered for this experiment is 100 nodes. The Max-speed of the mobile nodes has been set as 40 m/s. It is observed that the AnthocMMP algorithm incurs less control overhead than the other existing algorithms. It is found that as the communication range increases, there is a less occurrence of link disconnections thereby reducing the control overhead in the network.

Analysis on packet delivery ratio The Fig. 2a shows the comparison of packet delivery ratio of AntHocMMP algorithm with the existing algorithms viz. AntHocNet, R-ACO1, LAR and MMP by varying the max-speed of the mobile nodes from 5 to 50 m/s. The number of nodes taken for this experiment is 150 nodes with the transmission range as 300 m.

The proposed AntHocMMP algorithm achieves a high packet delivery ratio despite of increasing the node speed from 5 to 50 m/s. In a dynamic network behaviour, the AntHocNet and R-ACO1 routing algorithms takes longer time to construct and maintain the paths between the source and the destination nodes during the link failures. This leads to high packet loss in the dynamic network. But, the proposed AntHocMMP algorithm takes less time to find an alternate path during the link failure and thereby achieves a high packet delivery ratio in high dynamic network.

The Fig. 2b shows the variation of the packet delivery ratio by varying the number of nodes from 50 to 150 nodes. The max-speed of the mobile nodes has been set as 40 m/s. and its transmission range as 400 m. It has been observed that as more the number of nodes is added to the network, there is the possibility of having less packet loss thereby achieving high packet delivery ratio. Therefore, the proposed AntHocMMP algorithm outperforms with high packet delivery ratio than the existing algorithms viz. AntHocNet, MMP, R-ACO and LAR algorithms.

The Fig. 2c shows the variation of the packet delivery ratio of the AntHocMMP algorithm by varying the communication range from 50 to 400 m. The number of nodes considered for this experiment is 100 nodes. The max-speed of the mobile nodes has been set as 40 m/s. As the communication range increases there are fewer occurrences of link disconnections thereby reducing the packet loss to a great extent. Therefore, the AntHocMMP achieves high packet delivery ratio by varying the communication range of the nodes in the network.

Analysis on Average end-to-end delay The Fig. 3a shows the variation of the average end-to-end delay by varying the max-speed of the mobile nodes from 5 to 50 m/s. The number of nodes taken for this experiment is 100 nodes with the transmission range as 400 m.

The proposed AntHocMMP algorithm attains low end-to-end delay than the existing algorithms viz. AntHocNet, LAR, R-ACO1 and MMP. The AntHocNet incurs longer delay due to restoration of the broken paths between the source and the destination pairs. But, the proposed AntHocMMP restores its path amongst the relative paths that are already discovered by the MMP algorithm which in turn causes less delay for a small period of time. Therefore, the AntHocMMP algorithm outperforms the other algorithms with minimum end-to-end delay.

The Fig. 3b shows the comparison of average end-to-end delay of the proposed AntHocMMP algorithm with the other existing algorithms viz. AntHocNet, MMP, R-ACO1 and LAR by varying the number of mobile nodes from 50 to 150 nodes in the network. The max-speed of the mobile nodes has been set as 40 m/s and its transmission range as 400 m. It has been observed that the average end-to-end delay decreases as the number of nodes increases in the network. This is due to the fact that, as the number of nodes increases there is a high possibility of having additional alternate paths to transmit the packets between the source and the destination node during the link failures. Therefore, the AntHocMMP algorithm incurs less end-to-end delay than the other existing algorithms.

The Fig. 3c shows that variation of the average end-to-end delay of the proposed AntHocMMP algorithm by varying the communication from 50 to 400 m. The number of nodes considered for this experiment is 150. The Max-speed of the mobile nodes has been set as 50 m/s. It is observed that the AntHocMMP algorithm attains less end-to-end delay as less link disconnections

Table 8 Paired samples test

Factors(average end-to-end delay)	Paired differences			
	Mean	SD (s)	Std. error mean (SE)	95 % Confidence interval of the difference Lower
Pair 1				
ANHOCMMPFACTOR ANTHOCNETFACTOR	5.650	3.23879	.51210	4.61419

Table 9 Two tailed *t* test

Factors (average end-to-end delay)	Paired differences 95 % Confidence interval of the difference Upper	t	df	Sig. (2-tailed)
Pair 1				
ANHOCMMPFACTOR	6.68581	11.033	39	.000
ANTHOCNETFACTOR				

Table 10 ANOVA test I for different methods

Methods	N	Mean	SD	Std. error	95 % Confidence interval for mean	
					Lower bound	Upper bound
1	40	11.0750	2.61541	.41353	10.2386	11.9114
2	40	25.7500	1.75046	.27677	25.1902	26.3098
3	40	24.8750	2.95425	.46711	23.9302	25.8198
4	40	27.3500	1.36907	.21647	26.9121	27.7879
5	40	31.2000	1.11401	.17614	30.8437	31.5563
Total	200	24.0500	7.16128	.50638	23.0514	25.0486

occurs by increasing the communication range of the nodes in the network.

Analysis on Hop-count The Fig. 4a shows the variation of average number of hops of AntHocMMP algorithm with existing algorithms viz. AntHocNet, R-ACO1, LAR and MMP by varying the Max-speed of the mobile nodes from 5 to 50 m/s. The number of nodes considered for this experiment is 150 nodes with the transmission range of 350 m.

The proposed AntHocMMP algorithm requires less number of hops between the source and the destination. This is due to the fact that the packets are transferred through the optimized path that has less energy path cost

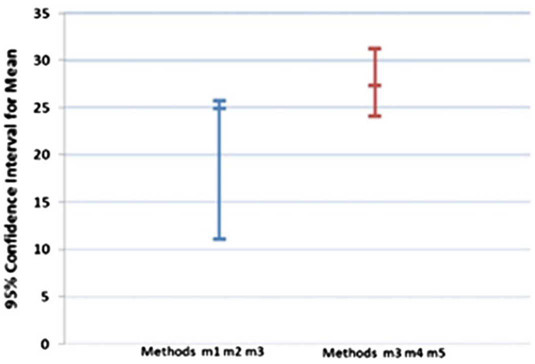

Fig. 5 ANOVA test I for 95 % confidence interval for mean

Table 11 ANOVA test II with its significance between the methods

	Sum of squares	Df	Mean square	F	Sig.
Between Groups	9357.350	4	2339.338	537.842	.000
Within Groups	848.150	195	4.349		
Total	10205.500	199			

Table 12 Anova test II

Factors	N	Mean	SD	Std. error	95 % Confidence interval for mean	
					Lower bound	Upper bound
1.00	40	95.8750	2.88397	.45600	94.9527	96.7973
2.00	40	92.7500	4.53335	.71679	91.3002	94.1998
3.00	40	84.9250	2.41138	.38127	84.1538	85.6962
Total	120	91.1833	5.72329	.52246	90.1488	92.2179

Table 13 Anova test II with significance between the factors

	Sum of squares	df	Mean square	F	Sig.
Between Groups	2545.317	2	1272.658	110.081	.000
Within Groups	1352.650	117	11.561		
Total	3897.967	119			

thereby reducing the hop-count. However, the AntHocNet requires more number of hops since the energy path cost is not considered to identify the shortest path between the source and the destination nodes. Therefore, the AntHocMMP algorithm outperforms the other existing algorithms with minimum hop count by varying the speed of nodes from 5 to 50 m/s.

The Fig. 4b shows the variation of hop-count of the proposed AntHocMMP algorithm by varying the number of nodes from 50 to 150. The max-speed of the nodes has been set as 50 m/s with the transmission range as 350 m. The proposed AntHocMMP requires less number of hops to reach destination node as the shortest path is chosen amongst the more number of nodes in the network. This due to the fact that as the number of nodes increases, the availability of additional relative paths also increases. Therefore, the proposed AntHocMMP incurs less end-to-end delay by selecting the appropriate shortest path among the additional relative paths available in the network.

The Fig. 4c shows the variation of hop-count by varying the communication range from 50 m to 400 m respectively. The max-speed of the mobile nodes has been set as 30 m/s and the number of nodes taken for simulation is 150. As the communication range increases, the number of hops between the source and the destination node also decreases in the network.

5.2 Statistical analysis of *t* test and ANOVA test

The two statistical tests viz. *t* test and ANOVA test have been performed to evaluate the performance of the proposed AntHocMMP algorithm against the existing algorithms viz. AntHocNet, R-ACO1, LAR and MMP.

***t* test**

The *t* test has been performed to find the significant difference between the methods: AntHocMMP m_1 over AntHocNet m_2 and it is carried out with the following steps.

Step 1: state the hypothesis

The hypotheses concern a variable d, the difference between paired values v_1, v_2 from the methods m_1 and m_2.

$$d = v_1 - v_2$$

where v_1, and v_2 represents the average end–end delay in the first data set m_1 and the second data set m_2 respectively by varying the speed of node from 0 to 50 m/s.

Null Hypotheses $\mathbf{H}_0$: There is a no significance difference between AntHocMMP m_1 over AntHocNet m_2

$$\mu_{m1} = \mu_{m2}$$

Alternate Hypotheses $\mathbf{H}_\alpha$: There is an improvement of AntHocMMP m_1 over AntHocNet m_2

$$\mu_{m1} < \mu_{m2}$$

Step 2: formulate an analysis plan

In order to analyze the hypotheses H_0, a two-tailed *t* test has been conducted on the paired data set with the significance level of 0.05 as its upper bound.

Step 3: analysis of sample data

The paired data set is analyzed to find the mean, standard deviation of the differences (s), the standard error (SE) of the mean difference, the degrees of freedom (df), and the *t* score test statistic (t). The Table 8 shows the paired sample test for the factor (average end-to-end delay) of the methods m_1 and m_2.

Step 4: interpretation of the results

The two tailed *t* test proves that there is a significant difference between sample means for paired data. It is also found that there is a significant difference between the two methods m_1 and m_2 with respect to the factor (average end-to-end delay). Therefore, the null hypothesis H_0 is rejected. The Table 9 also shows that the significance values of the paired data is less than the upper bound of 0.05. Therefore, it is concluded that the proposed AntHocMMP algorithm performs better than the AntHocNet algorithm.

ANOVA-Test I

The ANOVA test I has been conducted to find the significance difference between the five methods viz. AntHocMMP m_1, AntHocNet m_2, R-ACO1,m_3, LAR m_4, MMP m_5 respectively and it is carried out with the following steps:

Step 1: statement of hypothesis

Null Hypothesis $\mathbf{H}_0$:

$$\mu_{m1} = \mu_{m2} = \mu_{m3} = \mu_{m4} = \mu_{m5}$$

Alternate Hypothesis $\mathbf{H}_\alpha$:

$$\mu_{m1} \neq \mu_{m2} \neq \mu_{m3} \neq \mu_{m4} \neq \mu_{m5}$$

Step 2: formulate an analysis plan

In order to analyze the hypotheses H_0, the ANOVA test I has been conducted on the paired data set $(v_1, v_2, v_3, v_4, v_5)$ that represents the packet delivery ratio of the methods m_1 m_2 m_3 m_4 m_5 respectively. The speed of the nodes is varied from 0 to 50 m/s for all the methods. The significance level of 0.05 is considered as upper bound on the paired data set of the null hypothesis.

Step 3: analysis of sample data

The paired data set is analyzed using Completely Random Design (CRD) method to find the Mean, the Standard Deviation, the Standard Error, the degrees of freedom (df), the sum of Squares, the Mean Square, and the F-score ANOVA statistic (F). The Table 10 and Fig. 5 shows the ANOVA test I for the paired data set $(v_{1,}v_{2,}v_3, v_4, v_5)$ of packet delivery ratio of the methods m_1 m_2 m_3 m_4 m_5 respectively.

Step 4: interpretation of the results

The ANOVA-test I proves that there is a significant difference between the methods m_1 m_2 m_3 m_4 m_5 with respect to the Packet delivery ratio. Therefore, the null hypothesis H_0 is rejected. The Table 11 shows the significance value is less than the upper bound 0.05. Therefore, it is concluded that the proposed AntHocMMP algorithm performs better than the other existing algorithms viz. AntHocNet, R-ACO1, LAR, and MMP algorithms.

ANOVA TEST II

The ANOVA test II has been performed to find the significance difference between the factors viz. average end–end delay $F_{1,}$ control overhead F_2, packet delivery ratio F_3 respectively and it is carried out with the following steps:

Step 1: statement of hypothesis

Null hypothesis $\mathbf{H}_0$

$$\mu_{F1} = \mu_{F2} = \mu_{F3}$$

Alternative hypothesis $\mathbf{H}_a$

$$\mu_{F1} \neq \mu_{F2} \neq \mu_{F3}$$

Step 2: formulate an analysis plan

In order to analyze the hypotheses H_0, the ANOVA test II has been conducted on the paired data set (average end–end delay v_1, control overhead $v_{2,}$ and the hop-count v_3) with respect to the packet delivery ratio. The significance level of 0.05 is considered as upper bound on the paired data set of the null hypothesis.

Step 3: analysis of sample data

The paired data set is analyzed using Completely Random Design (CRD) method to find the Mean, the Standard Deviation, the Standard Error, the degrees of freedom (df), the sum of Squares, the Mean Square, and the F-score ANOVA statistic (F). The Table 12 shows the ANOVA test II for the paired data set as average end–end delay v_1, control overhead $v_{2,}$ and the hop-count v_3 with respect to the packet delivery ratio.

Step 4: interpretation the results

The ANOVA test II proves that there is a significant difference between the factors viz. end–end delay $F_{1,}$ control overhead F_2, packet delivery ratio F_3 respectively. The Table 13 shows the significance values between the factors is less than the upper bound 0.05. Therefore, it is concluded that the average end–end delay, control overhead and Packet delivery ratio have high variance among one another.

6 Conclusion

The Ant Colony Optimization (ACO) based routing algorithm effectively utilizes the energy of the nodes and enhances the lifetime of the network. In this paper, an exhaustive literature survey has been carried out related to Ant Colony based routing algorithms in MANETs. A novel routing algorithm named AntHocMMP has been proposed. This algorithm highlights the ability to use ant as agents to perform foraging activities to communicate between the nodes and to achieve the robustness of an energy-efficient shortest path in MANETs.

The AntHocMMP algorithm is an enhanced version of the Max-Min-Path (MMP) [7] routing algorithm. This algorithm constructs the relative paths between the source and its destination nodes in the network by using energy path cost functions. The relative paths are utilized in the AntHocMMP algorithm to find the optimal path between the source and the destination node in the network. The performance of AntHocMMP has been investigated using the metrics viz. the packet delivery ratio, the control overhead, the average end-to-end delay and the hop-count by varying the node speed, the communication range and the number of nodes.

The simulation results show that the proposed AntHocMMP achieves a high packet delivery ratio with low energy path cost and less control overhead in high dynamic networks. The performance of AntHocMMP has also been compared with the other existing algorithms viz. AntHocNet, R-ACO1 and LAR and it is observed that the AntHocMMP outperforms the existing algorithms. In addition to the simulation results, the statistical analysis has also been carried out by performing *t* test, ANOVA test I and ANOVA test II. The statistical test proves that the proposed AntHocMMP algorithm dominates the existing algorithms in a more significant way.

Acknowledgments The author wishes to thank the Karunya Univesity for providing financial support and lab facilities to carry out the research work. The authors also wish to thank the editor, the reviewers and all the supporting higher officials for their continuous support and encouragement in the successful completion of the research work.

References

1. Camilo, T., Carreto, C., Silve, J. S., & Boavida, F. An energy-efficient ant-based routing algorithm for wireless sensor networks. *Ant Colony Optimization and Swarm Intelligence, LNCS, 4150*, 49–59.
2. Parsapoor, M., & Bilstrup, U. (2013). Ant colony optimization for channel assignment in a clustered mobile ad hoc network. In *4th international conference, ICSI 2013* (pp. 314–322). Springer: Berlin Heidelberg.
3. Kamali, S., & Opatrny, J. (2007). POSANT: A position based ant colony routing algorithm for mobile ad hoc networks. In *Proceedings of the third international conference on wireless and mobile communications* (ICWMC'07), pp. 21–28.
4. Dorigo, M., & Caro, G. D. (1999). The ant colony optimization meta-heuristic, New Ideas in Optimization, pp. 11–32.
5. Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization: Artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine, 1*(4), 28–39.
6. Guines, M., Sorges, U., Bouazizi, I. (2002). ARA—the ant-colony based routing algorithm for MANETs. In *ICPP workshops, IEEE computer society*, pp. 79–85.
7. Baskaran, R., Victer Paul, P., & Dhavachelvan, P. (2012). Ant colony optimization for data Cacje technique in MANET. In *Advances in intelligent systems and computing*, Springer, India, pp. 873–878.
8. Camara, D., & Loureiro, A. A. F. (2001). GPS/ant-like routing in ad hoc networks. *Telecommunication Systems, 18*(1–3), 85–100.
9. Guo, S., Oliver, W. W., & Yang, W. (2007). Energy-aware multicasting in wireless ad hoc networks: A survey and discussion. *Computer Communications, 30*(9), 2129–2148.
10. Di Caro, G., Ducatella, F., Gambardella, L. M. (2004). AntHocNet: An ant-based hybrid routing algorithm for mobile ad hoc networks. In *Parallel problem solving from nature*—PPSN VIII, Vol. 3242, LNCS, pp. 461–470.
11. Vijayalakshmi, P., Francis, S. J., & Abraham Dinakaran, J. (2014). *Max-min-path energy-efficient routing algorithm—A novel approach to enhance network lifetime of MANETs", ICDCN 2014*, Springer LNCS 8314, pp. 512–518.
12. Krzysztof, W. (2005). Ant algorithm for flow assignment in connection-oriented networks. *International Journal of Applied Mathematics and Computer Science, 15*(2), 205–220.
13. Abolhasan, M., Wysocki, T., & Dutkiewicz, E. (2004). A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks, 1*, 1–22.
14. Haas, Z. J., & Pearlman, M. R. (2001). The performance of query control schemes for the zone routing protocol. *ACM/IEEE Transactions on Networking, 9*(4), 427–438.
15. Giagkos, Alexandros, & Wilson, Myra S. (2014). BeeIP—A swarm intelligence based routing for wireless ad hoc networks. *Information Sciences, 265*, 23–35.
16. Matsuo, H., & Mori, K. (20014). Accelerated ants routing in dynamic networks. In *Proceedings of the international conference on software engineering, artificial intelligence, networking and parallel/distributed computing*, pp. 333–339.
17. Dorigo, M., & Stutzle, T. (2004). *Ant Colony Optimization*. Cambridge: MIT Press.
18. Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews, 2*(4), 353–373.
19. Perkins, C. E. (2001). *Ad hoc networking*. Reading: Addison-Wesley.
20. Toh, C. (2002). *Ad hoc mobile wireless networks: Protocols and systems*. Englewood Cliffs: Prentice-Hall.
21. Kadono, D., Izumi, T., Ooshita, F., & Kakugawa, H. (2010). An ant colony optimization routing based on robustness for ad hoc networks with GPSs. *Elsevier-Ad Hoc Networks, 8*, 63–76.
22. Correia, F., & Vazao, T. (2010). Simple ant routing algorithm strategies for a (Multipurpose) MANET model. *Elsevier-Ad Hoc Networks, 8*, 810–823.
23. Sardar, A. R., Singh, M., Sahoo, R. R., Majumder, K., Sing, J. K., & Sarkar, S. K. (2014). An efficient ant colony based routing algorithm for better quality of services for MANTETs, ICT and critical infrastructure. In *Proceeding of the 48*th *annual convention of CSI—Volume 1, advances in intelligent systems and computing,* Vol. 248, Springer International Publishing: Switzerland, pp. 233–240.
24. Sisodia, R. S., Manoj, B. S., & Murthy, C. R. (2002). A preferred link based routing protocol for wireless ad hoc networks. *Journal of Communication and Networks, 4*(1), 14–21.
25. Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *Computer Communications Review, 10*, 234–244.
26. Perkins, C. E., Royer, E. M., & Das, S. R. (1999). Ad hoc on demand distance vector (AODV) routing. In *IEEE workshop on mobile computing systems and applications*, pp. 90–100.
27. Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53–66.
28. Pandy, N. P. (2005). *Artificial intelligence and intelligent systems* (pp. 532–574). Oxford: Oxford University Press.

P. Vijayalakshmi is a research scholar in the Department of Computer Applications, School of Computer Science and Technology, Karunya University, India. She is postgraduate with M.C.A. from the Bharathidasan University, India. Her research interests include ad hoc routing/interoperability, wireless mobility support/management, energy conservation routing protocols and wireless sensor networks.

Sharmila Anand John Francis obtained her Doctorate of Philosophy in Computer Science and Engineering/Computer Applications from Karunya University, Coimbatore, India in 2010. Her specialized area includes mobile ad hoc networks, Sensor networks, Mesh networks, Cloud computing. She has published his research articles in many International Journals and also presented in many National/International Conferences. At present, she is working in the Department of Computer Applications, Karunya University and extending her guideship towards six Ph.D. scholars and Master degree students in the areas of computer networks.

J. Abraham Dinakaran Assistant Professor with the Department of Computer Applications, School of Computer Science and Technology, Karunya University, India. He is postgraduate with M.C.A. from the Bharathidasan University, India. His research focus includes Location Service in MANETs, Machine Learning, Artificial Intelligence and Data Mining.