TCORRA: Termite Colony Optimization based Route Repair Algorithm for Auto-configuration of Broken Links in Mobile Ad hoc Networks

Ramanpreet Kaur, Kavita Taneja, *Member, IAENG*, Harmunish Taneja, *Member, IAENG*

Abstract— **With the increasing surge in user density and delay-sensitive applications such as remote healthcare, rescue operations, and multimedia transmission, Mobile Ad hoc Networks (MANETs) face substantial issues in ensuring seamless connectivity. MANET's inherent mobility and unpredictable topology have increased the likelihood of network link breakage. Furthermore, these disruptions significantly impact network efficiency related to increased transmission delay and packet loss. The ever-growing demand for seamless connectivity has highlighted the need for a robust link repair mechanism in the MANET environment. This paper presents the Termite Colony Optimization-Based Route Repair Algorithm (TCORRA) for auto-configuration of broken links in MANET. The fitness function of the proposed algorithm is designed by focusing on dual pivotal factors: community time and total packet loss. TCORRA leverages the Auto-Regressive Integrated Moving Average (ARIMA) model to envisage the movement patterns of network nodes. To effectively segregate routes into clusters of broken and active links, K-Means clustering is integrated into the algorithm. In the event of link failure, the cluster of active links is prioritized to select the optimal path. This approach helps in reducing overall path computation time. The performance of the proposed algorithm is validated regarding throughput, packet delivery ratio, total packets lost, delay, and control overhead. The simulation results and performance analysis indicate that TCORRA outperforms other competitive link repair algorithms. TCORRA enhances overall network connectivity owing to the swift reconfiguration of broken links in MANET.**

Index Terms— **ARIMA model, auto-configuration, link repair, mobile ad hoc networks, termite colony optimization.**

I. Introduction

Mobile ad hoc networks (MANETs) are versatile networks characterized by their inherent decentralized ability. These networks autonomously configure themselves without the reliance on a central control system. Each node within MANET is endowed with both a wireless transmitter and receiver, empowering them to serve as active routers for network-wide communication [1].

Manuscript received August 9, 2024; revised August 1, 2025.

R. Kaur is a PhD candidate of Computer Science and Applications Department, Panjab University, Chandigarh, India (corresponding author: 91-9041588036; e-mail: ramanvohra82@gmail.com).

K. Taneja is an assistant professor of Computer Science and Applications Department, Panjab University, Chandigarh, India (e-mail: kavitatane@gmail.com).

H. Taneja is an assistant professor of Computer Science and Applications Department, DAV College, Chandigarh, India (e-mail: harmunish.taneja@gmail.com).

The orchestration of network communication is managed through routing protocols. These protocols discover multi-hop paths between nodes and efficiently forward data packets to their destinations. The dynamic landscape of MANETs is characterized by mobility, which allows nodes to move in any direction. This continuous movement results in an unpredictable topology, leading to frequent network connection failures.

However, many other factors contribute to link breakage, including wireless media limitations, restricted node energy, traffic overload, and network partitioning. The process of re-establishing disrupted routes significantly reduces overall network performance. As MANETs progress, they become essential for applications such as rescue operations, remote healthcare, and military activities. The interruptions due to link failure in these delay-sensitive applications can have devastating consequences. Therefore, maintaining continuous connectivity is paramount in addressing the challenge of link failure in MANETs. Nature-inspired algorithms (NIAs) have been effectively employed to design innovative link repair mechanisms for MANET routing. These algorithms exploit natural organism's robustness, flexibility, and adaptability. These features work effectively in a MANET environment to auto-configure broken links and sustain uninterrupted connectivity.

In the existing literature, various local link repair methods [7], [12-15] have been proposed to tackle the issue of recurrent failure of network links in MANETs. A major limitation of these route repair mechanisms is the absence of forecast-driven analysis of link failure. These algorithms often employ reactive routing approaches for route repair, resulting in extended processing times and delays. To overcome these challenges, this paper presents an innovative route repair technique for auto-configuring broken links in MANETs. This approach forecasts link failure by predicting the past mobility patterns of mobile nodes across various time intervals. Therefore, predicting link failures can be framed as a Temporal Sequence Forecasting (TSF) problem. The Auto-Regressive Integrated Moving Average (ARIMA) model is the most widely used temporal sequence forecasting model. The proposed approach utilizes the ARIMA model to predict link failures before they occur. Additionally, this technique includes a fitness function that consists of multiple objectives based on essential parameters like community time and packet loss. The termite colony optimization algorithm inspires the design of the proposed link repair mechanism. This proactive strategy aids in identifying alternative paths for data transmission in advance to mitigate link failures. The key contribution of this study is as follows:

- To predict the mobility patterns of MANET nodes by employing the ARIMA model.
- To formulate a fitness function based on multiple parameters for exploring an optimal routing path.
- To design and develop a technique for auto-configuration of broken links in MANETs, Termite Colony Optimization Based Route Repair (TCORRA) strives to enhance throughput and diminish delay and packet loss.
- To substantiate the efficacy of the proposed methodology using a simulation environment and performance analysis with other state-of-the-art routing algorithms.

The paper is organized as follows: Section II reviews related work on link repair techniques. Section III discusses the problem statement. Section IV presents the detailed methodology of the proposed approach. Section V describes the simulation results and performance analysis, whereas Section VI provides the conclusion of the proposed work.

II. Related Work

The Cuckoo Search (CS) algorithm is integrated with the hop count metric to design a fitness function for optimizing the route selection process. A significant drawback of the study is that it does not consider mobility prediction during link repairs. Moreover, the performance analysis relies solely on conventional metrics [11]. A termite colony-inspired routing algorithm is introduced to manage network traffic. The suggested study is limited by the lack of a link repair mechanism [10]. A multicast routing strategy incorporating link repair using Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and a dynamic queuing mechanism is suggested to enhance QoS. ACO assists users in selecting the optimal path through its trust-based path feature. PSO is employed to identify alternative paths in the event of link failure. However, this approach lacks a prediction-based link repair approach [12]. This study presents a local link repair technique inspired by ACO. This repair mechanism recommends an alternative route by utilizing local node information as a heuristic function to discover the next active node. The disadvantage of the study is the absence of performance-based analysis [13]. A node-disjoint multi-path routing protocol leveraging ACO is introduced. The approach aims to improve QoS parameters while reducing routing overhead. A restoration mechanism is also proposed to redirect current traffic towards another available path for handling link failure. The primary drawback of this approach is the absence of a forecast-driven investigation for link failure [14]. A priority-based route repair approach is introduced for selecting intermediate nodes based on the traffic load handled by each node. However, this study utilizes a reactive technique for managing link failures, which is a disadvantage [15]. A routing mechanism inspired by ACO and a 2-opt heuristic for route exploration and maintenance is proposed. The author employs a Genetic Algorithm (GA) for refining ACO parameters. A notable limitation of this methodology is its inadequacy for complex and highly dynamic environments [16]. A termite colony optimization-based routing algorithm is presented in this study. The proposed study utilized a local monitoring technique and a cross-layer approach to identify stable nodes. The disadvantage of this method is the absence of a prediction-based maintenance procedure [17]. A multicast routing strategy utilizes fuzzy logic, and a route repair mechanism is proposed to enhance network lifetime. This approach considers multiple parameters, including bandwidth, residual energy, delay, and link stability as fuzzy inputs. The limitation of this approach is that it lacks a prediction-based analysis of link failure. [18]. To optimize the routing process, the proposed protocol seamlessly integrates the functionalities of AntOR-Disjoint-node and Parallel-AntOR. To tackle link failures, network nodes send ants through independent threads to explore alternative routes. However, the protocol lacks a mechanism for predicting the mobility of network nodes [19]. The approach discussed in this study introduces a route maintenance strategy inspired by ACO. In this method, pheromone information is diffusely shared among nodes to identify alternative routes. However, this approach primarily depends on information from neighboring nodes alone, which may limit its applicability in certain network scenarios [7]. A novel link repair algorithm is introduced in this paper. This approach aims to reduce routing costs in MANETs, leveraging fungi colonies. A specific repair time limit governs the repair process. If this time limit is surpassed without achieving a successful repair, a fault message is sent to the originating node. Importantly, the performance evaluation for this methodology is largely confined to low-traffic scenarios [20]. A routing approach is proposed for route discovery and maintenance, leveraging the Group Teaching Optimization Algorithm (GTA). The first phase of this approach addresses node trust evaluation using the Adaptive Neuro-Fuzzy Inference System model. The second phase deals with the discovery of multiple routes using GTA. The final phase selects the ideal path aided by an adaptive equilibrium optimizer. The route maintenance phase is also integrated into the system to explore alternate paths in case of link failure. However, this proposed approach has room for improvement, particularly for real-time application scenarios [21]. This work examines the Eagle-Based Density Clustering (EBDC) technique for routing in MANETs, focusing on prolonging network lifetime. This approach forecasts broken network links based on energy consumption by network nodes. Notably, this study overlooks the impact of node mobility [1]. A link repair technique grounded in a d-hop graph kernel and clustering scheme is introduced for MANETs. This technique incorporates node mobility and position for link status prediction. The performance of this approach is assessed based on ratio of packet loss and control packets. The major drawback of this algorithm is its inflexibility in complex, dynamic MANET environments [22]. A routing strategy inspired by ACO is recommended to enhance throughput in MANETs. This routing approach does not include a link repair mechanism [37]. A routing solution inspired by ACO and PSO is proposed for sensor networks to enhance the network life cycle. The pheromone model of this approach is designed using various factors, including energy, hop count, and node distance. The PSO's fitness evaluation function ensures rapid exploration of alternative paths in link failures. However, a limitation of this strategy is its lack of suitability for real-world applications [38].

III. Problem Statement

MANET is a self-organizing network of mobile nodes that operates devoid of a fixed structure. Due to the rapid mobility of network nodes and recurrent topological variations, the likelihood of network link failure increases. In MANETs, link failures can have detrimental effects, including increased packet loss, prolonged delays, and reduced data throughput. Therefore, maintaining seamless communication with stable links is a significant challenge in MANET. Traditionally, when a network link fails, the routing protocol generates error messages and triggers a route discovery process. This conventional approach often results in increased network delays and packet loss. To address these issues, an innovative route repair solution is needed to predict network link failures for uninterrupted communication.

IV. Proposed Approach

A. System model

The MANET model facilitates the implementation of TCORRA, where nodes are randomly distributed within a clearly defined space. The network at time t is signified by a graph G = (V, E), with V symbolizing the set of network nodes and E representing the set of links. Each node in V can move independently, contributing to the network's dynamic nature. Each link in E establishes a direct connection between network nodes, m and n, in the set V. All nodes in V operate within the same transmission range r. When node m enters the coverage area of node n, a new link (m, n) is created and added to E. In MANETs, nodes are characterized by their mobility, wireless connectivity, and dynamic topology. This unpredictable behavior of MANET nodes can result in network link breakage. The proposed approach effectively manages link failures while minimizing packet loss and enhancing network performance. Fig.1 illustrates the block diagram of TCORRA. This predictive auto-configuration strategy helps to alleviate the effects of link failures, ensuring efficient communication in MANETs.

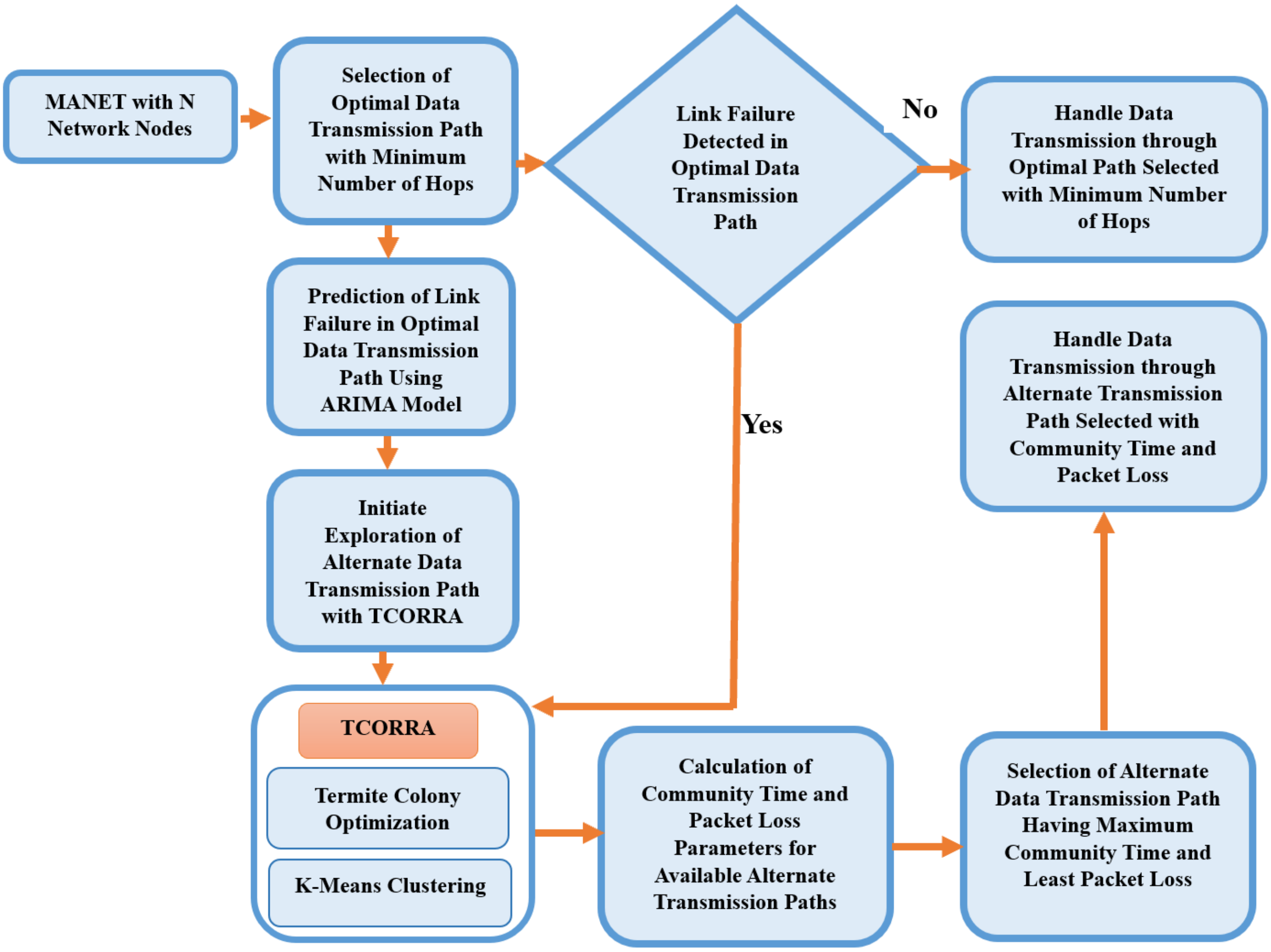

Fig. 1. System model of TCORRA

B. Mobility prediction using ARIMA model

In MANETs, the mobility of nodes is a central factor in shaping data transmission paths. The continuous movement of network nodes introduces substantial disruptions to wireless connectivity, leading to packet drops and prolonged delays [23]. These disruptions become more evident when high-speed mobile nodes contribute to routing activities. These fast-moving nodes provide a crucial challenge in upholding stable network connectivity. Therefore, predicting node mobility helps in seamless network communication. Forecasting a node's mobility in the dynamic MANET environment is critical for effective routing. Mobility prediction provides foresight into the forthcoming movements of mobile nodes by estimating their future locations. This information can be effectively utilized in optimizing network performance [24]. In addition, mobility prediction provides a proactive selection of stable network nodes for data transmission to streamline the complete routing process [25].

The Auto-Regressive Integrated Moving Average (ARIMA) model, presented by Box and Jenkins, is utilized for future predictions based on historical data. ARIMA aims to forecast future values precisely by integrating moving average and auto-regression practices [26]. ARIMA boasts various advantages such as a lower root mean error, reduced time of processing, and overall enhanced performance [27]. This model is proficient in estimating future speed values for mobile nodes, treating the task of prediction [28] as a nonlinear autoregressive problem [29]. The ARIMA model can be defined by three parameters (p, d, q). In this tuple, 'p' implies total time lags, implementing the autoregressive feature of ARIMA. This aspect of ARIMA approximates the present value from a linear-weighted sum of its earlier values [30]. The parameter 'd' implies the number of differences from the preceding data in a temporal context. This symbolizes the unified nature of ARIMA, capturing the relationship between successive values. It involves subtracting past values to attain steadiness in the temporal records [31]. Lastly, 'q' indicates the direction of the moving average method. This embodies the moving average component of ARIMA, where the model explores the relationship between prediction errors.

Consider x_t presents a sequence of values at various time intervals t. The auto-regressive (AR) component of this model can be articulated as follows:

$$x_t = \sum_{i=1}^{p} \alpha_i \, x_{t-i} \tag{1}$$

This equation demonstrates that x_t is dependent on its previous p values, where $\alpha_i (i = 1, \dots . p)$ represents the coefficient of AR component. Additionally, considering ϵ_t as an error term at the time t exhibiting white noise, the Moving Average (MA) component is computed as:

$$x_t = \gamma + \epsilon_t + \sum_{j=1}^{q} \beta_j \, \epsilon_{t-j} \tag{2}$$

This expression signifies x_t varies linearly with the present and the preceding q terms with white noise errors. Here, γ represents the expected value of x_t, and $\beta_j \,(j = 1, \dots q)$ denotes the parameter of the model associated with the j^{th} lag of the moving average component. The integration of the AR and MA components is represented as follows:

$$x_t = \sum_{i=1}^{p} \alpha_i \, x_{t-i} + \sum_{j=1}^{q} \beta_j \, \epsilon_{t-j} + \epsilon_t \tag{3}$$

In practical scenarios, time series data often exhibits dynamic behaviors. To address and attain the steadiness characteristic of a temporal series, the subtracting technique is employed through the Integrated (I) component in ARIMA. Denoting $\Delta^d x_t$ as the d-order difference of x_t, the ARIMA (p, d, q) can be represented as:

$$\Delta^d x_t = \sum_{i=1}^{p} \alpha_i \Delta^d x_{t-i} + \sum_{j=1}^{q} \beta_j \, \epsilon_{t-j} + \epsilon_t \tag{4}$$

Here α_i and β_j represent the factors of AR and MA, and ϵ_{t-j} reflects the stochastic fault terms of the previous q observations. This value is characterized as an uncorrelated variable with a zero mean and sharing the same distribution. ϵ_t signifies the forecasting error at the present stamp.

C. Termite colony optimization

Termite Colony Optimization (TCO) is a nature-inspired algorithm that draws its inspiration from the collective traits of termites. This algorithm mimics the decision-making seen in termite societies to adjust their movement trajectories [32,33]. TCO has emerged as a powerful tool in the realm of computational intelligence [34]. Termites navigate randomly in their search environment. Termite movements are inclined toward areas with higher pheromone concentrations. TCO utilizes a stochastic process to discover optimal solutions from the search space. It employs a population of N artificial termites to navigate through a search space denoted as Se. This search space is a subclass of the E-dimensional Euclidean space R^E. Each termite, denoted as k within this population, characterized by a position vector $z_k = (z_{k1}, z_{k2}, \dots \dots . z_{kE})$. Here, each position vector z_k, analogous to a hill in a landscape. The quality of z_k is represented by a fitness value denoted as $fv(z_k)$. This fitness value indicates the amount of pheromone associated with the hill. The initial step in TCO involves determining the number of termites (N) and setting the maximum iteration limit ($Itermax$). Subsequently, all termites are randomly positioned within the designated search space [33] to start the optimization process.

$$z_k(0) = Init(k, Se) \;\; 1 \le k \le N \tag{5}$$

Here, the initialization function $Init(k, Se)$, is responsible for randomly assigning a position to each termite k in the search environment Se. During each iteration, the termites dynamically modify their positions based on the optimization process [10]. The value of pheromone at the i^{th} position is computed as follows:

$$\varphi_k(t) = (1 - \rho)\varphi_k(t - 1) + 1/fv(z_k) + 1 \tag{6}$$

In the given equation, ρ indicates the fading rate within the range of [0 to 1]. The terms $\varphi_k(t - 1)$ and $\varphi_k(t)$ correspond to the pheromone concentrations at the present and prior locations of the k^{th} termite, respectively. Termites adopt a random trajectory to explore potentially more advantageous areas. This stochastic walk is implemented in a region with a range denoted as τ. This region is centered around the current position of termite. The next position of a termite is then recomputed as follows:

$$z_k(t) = z_k(t-1) + Rw\big(\varphi, z_k(t-1)\big) \tag{7}$$

The starting index of the radius τ is expressed as a difference of $|X_{max} - X_{min}|$. This difference denotes the uppermost and lowermost values within a dimension, X_{max} and X_{min} respectively. In the given equation, $z_k(t-1)$ denotes the previous location of the termite and $z_k(t)$ represents the new location of termites. Rw represents a stochastic walk function based on the present location of the termite and the exploration range τ. This function is represented by $|X_{max} - X_{min}|$ and φ denotes the step size. The termite assesses the existing pheromone levels and alters its movement to a location with the peak concentration of pheromones. Each k^{th} termite regards the finest position within its vicinity (symbolised as bp_k) as its favorable location. It then moves to a favorable position if its present location exhibits a lower concentration of pheromones than the best position within its vicinity. The movement path of k^{th} termite is computed as:

$$z_k(t) = z_k(t-1) + \omega_b r_b\big(bp_k - z_k(t-1)\big) \tag{8}$$
$$if\ \Big(\varphi_k(t-1) < \varphi_{b_k}(t-1)\Big)$$

where $1 < \omega_{bp} < 2$ & $0 < r_{bp} < 1$
These parameters probabilistically regulate the termite's attraction towards the local best position. The term ω_{bp} influences the weight assigned to the attraction towards the local best position. r_{bp} introduces a random factor that influences the probability of the termite adjusting its trajectory toward this position.

D. Multi-objective fitness function

The fitness function of TCORRA guides the optimization process by quantifying the quality of potential solutions. The fitness function of the proposed algorithm evaluates the performance of candidate solutions and selects the optimal solution. The fitness function of TCORRA is crafted using multi-objective optimization, considering two major indicators: community time and packet loss. This optimization technique [35], involves amalgamating distinct objective functions into an integrated objective function. The integration is achieved using a weighted sum method. Employing the weighted sum technique, the fitness value for the proposed algorithm is computed as follows:

$$\text{f(x)} \quad = \quad w_1 * \Big(\textstyle\sum_{j=1}^{n}(CTime_n)\Big) + w_2 * \left(\frac{1}{\sum_{j=1}^{n} PLoss_n}\right) \tag{9}$$

with $\sum_{i=1}^{2} w_i = 1$ and $w_i \,\epsilon\, [0,1]$

Here, f(x) signifies the unified objective function, n is the number of nodes, $CTime_n$ is community time, $PLoss_n$ is packet loss. w_1 and w_2 are weighing factors, stochastically generated from a uniform distribution. This formulation allows for a comprehensive evaluation of multiple objectives. It effectively accommodates the diverse considerations essential in multi-objective optimization scenarios. The solution that maximizes community time and minimizes packet loss yields a high fitness function value.

1) Community time

The most significant challenge in MANET is frequent link failures triggered by unpredictable topology and mobility. To handle this issue, the community time indicator is incorporated into the optimization criteria of TCORRA. Every MANET node possesses a predefined transmission range. For successful packet transmission, the sender and receiver nodes must be within range of each other. However, link failure occurs if they exceed the transmission range, resulting in increased packet loss. The community of the node v_k consists of the complete set of neighboring nodes δ of v_k. These neighboring nodes may be situated either close to v_k or beyond, but inside the communication radius of v_k. For every adjoining node v_l of node v_k, the radius identifier $Rg_{spec}(v_k, v_l)$ concerning v_k is computed as follows:

$$Rg_{spec}(v_k, v_l) = \begin{cases} 1, & if\ dis(k,l) < 250 \\ 0, & otherwise \end{cases} \tag{10}$$

To choose the most favourable adjacent node of v_k from its range, the Euclidean distance between v_k and v_l, and the forecasted mobility value of v_k and v_l are considered. ARIMA model is utilized to compute the predicted values of mobility SV_k and SV_l respectively. To determine the total distance v_l must traverse to exit the community of v_k, denoted as $(\theta\text{d})_l$ computed as follows:

$$(\theta\text{d})_l = v_k^r - D^m(v_k, v_l) \tag{11}$$

where $(\theta\text{d})_l$ represents the distance covered by v_k, v_k^r is the communication radius of v_k, and $D^m(v_k, v_l)$ is calculated using the following formula:

$$D^m(v_k, v_l) = \sqrt{(X_l - X_k)^2 + (Y_l - Y_k)^2} \tag{12}$$

where (X_k, Y_k) and (X_l, Y_l) are the X and Y co-ordinates of the node v_k and v_l correspondingly. Consequently, the calculation of community radius indicator is represented as follows:

$$\text{Community time } (CTime_n) \quad = \quad \frac{(\theta\text{d})_l}{|SV_k - SV_l|} \tag{13}$$

where SV_k and SV_l is the predicted velocity of the node V_k and V_l respectively.

2) Packet loss

Packet loss occurs in a network when certain packets fail to reach their intended destination. This phenomenon leads to data loss when packets are either dropped or discarded during network transmission. A key factor contributing to packet loss is link failure, where the connection between two nodes experiences disruptions. Packet loss is a critical issue with substantial consequences for MANET performance. This parameter contributes to challenges, including diminished throughput, interruptions in communication, deferred operations, and increased retransmissions. To reduce overall network delay, the fitness function of TCORRA integrates the packet loss indicator. The calculation of the packet loss indicator is done with following equation:

$$\text{Packet loss } (PLoss_n) = \sum(P\_Sent)_n - \sum(P\_Rec)_n \quad (14)$$

where P_Sent_n indicates the total packets sent and P_Rec_n specifies the total packets accepted during packet broadcasting. To get the values of P_Sent_n and P_Rec_n, a Network Animation (NAM) file is used. NAM file is automatically generated by the network simulator. This file contains the pictorial depiction of network topology and its activities during complete simulation process. NAM file acts as a complete record of the simulation process, providing vital understandings of network nodes, associations, packets, actions and traffic.

E. K-Means Clustering

Clustering is a versatile tool for categorizing groups or clusters in multivariate data. This process includes grouping objects into subgroups, referred to as clusters. This grouping is done by identifying intrinsic similarities among the objects within each group [36]. K-Means clustering is the most widely utilized technique for pattern recognition, data exploration, and simplification of complex datasets. The algorithm operates by categorizing a given dataset $X = \{X_1, X_2 \ldots\ldots X_n\}$ into T clusters $C_1, C_2 \ldots\ldots C_T$. This approach ensures the items within the same cluster share more similarities than those in other clusters. K-Means clustering is characterized as a point-based approach. This algorithm iteratively refines the grouping of data points within a given dataset. In this process, each cluster is associated with a central point, commonly referred to as a centroid [36]. The cluster centroids are denoted as $\{m_1, m_2, m_3 \ldots\ldots, m_k, m_T\}$. The algorithm iteratively adjusts these centroids until a stable configuration is reached. Each data point is allocated to the cluster whose centroid is adjacent to that cluster. K-Means clustering aims to minimize the calculated distance between individual data point and its respective cluster centers. The sum of the squared Euclidian distances between each data point X_i and the centroid m_k of the subset C_k which comprises X_i is computed as:

$$E(m_1, m_2, m_3 \ldots\ldots, m_k, m_T) = \sum_{i=1}^{N} \sum_{k=1}^{T} I(X_i \in C_k) |X_i - m_k|^2 \quad (15)$$

Here $I(X)$ =1 if X is true and 0 otherwise.

F. Termite Colony Optimization-Based Route Repair Algorithm

This paper introduces a route repair mechanism, TCORRA, to resolve frequent link failures in MANETs. This proactive approach is inspired by the synchronized behavior unveiled by the termites during the construction of hills. The termite hill-building process involves collecting pebbles scattered across an area and consolidating them in a single location. Every termite carries only one pebble and moves by following the pheromone trail left by other termites [17].

In the context of MANET, every node is metaphorically characterized as a termite hill. The pheromones are deposited along the links connecting nodes engaged in communication. This pheromone is a guiding factor for packets traversing the network, exhibiting a positive feedback mechanism. The determination of the succeeding hop is always probabilistically made considering the concentration of pheromone using random walks. Adaptations to variations in the network topology or path quality are accommodated by letting pheromones evaporate over time. This process introduces a negative response mechanism. The algorithm initiates its process by conducting a route exploration phase. TCORRA utilizes the Ad Hoc On-Demand Multipath Distance Vector Routing Protocol (AOMDV) to select an optimal route for data transmission. This methodology also offers a resilient route repair solution for situations where the optimal transmission path encounters failure. The primary objective of TCORRA is to enhance network throughput while minimizing packet loss and delay. This approach employs a multi-objective fitness function derived from community time and packet loss. To forecast the mobility patterns of network nodes, TCORRA monitors the position of each node relative to its broadcast range. This process recognizes the nodes unsuitable for the routing path and is on the verge of exiting their transmission range. Consequently, this rapid repair mechanism simplifies the establishment of alternative links beforehand. This proactive technique handles the impact of potential link failures and contributes to the algorithm's efficacy in maintaining robust network connectivity.

V. RESULTS AND ANALYSIS

The performance analysis of TCORRA is evaluated with the help of a network simulator. Three routing algorithms: Termite [10], Mobility Aware Termite (MA-Termite) [17] and Priority-based route maintenance (P-AODV) [15] are selected for performance comparison. In the simulation setup, 200 nodes are homogeneously distributed over a 1200 x 1200 sq. meter area. The random waypoint model of mobility is employed to imitate the mobility sequence of nodes in the network. This model selects a random route towards the target, with a random velocity taken from a pre-defined range (5 to 30 m/s). After reaching its intended destination, each node pauses for a defined duration before resuming its movement. This model introduces randomness into destination selection, pause durations, and node speeds. It adds a stochastic component to node movement to enhance the realism of the simulation environment. Table I presents the details of the simulation parameters used for implementing TCORRA.

In today's swiftly evolving real-time application scenarios, MANETs operate in diverse environments. Their performance is profoundly influenced by two major factors: node mobility and node density. Evaluating MANET performance under varying settings of these factors ensures the proposed approach is effectively applicable to tackle connectivity, delay, packet loss, and scalability issues in various real-time application areas.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Simulator	Network Simulator
Area of Simulation	1200*1200 sq. m.
Network Topology	Arbitrary Network Node Placement
Simulation Time	50 seconds
Channel Type	Wireless Channel
Antenna Type	Omnidirectional Antenna
Total Network Nodes	50 to 200 Network Nodes
Network Mobility Model	Random Way Point Mobility Model
Transmission Range	250 m
Node Mobility Speed	5 to 30 m/s
Routing Protocols	Termite, MA-Termite, P-AODV, TCORRA

A. Termite colony optimization route repair algorithm

Initialize N as total nodes in the network, P as total paths discovered in the broadcasting phase, Mei_n as a set of neighbouring nodes $(MN_1, MN_2, \dots MN_k, MN_N)$, CL_{active}, CL_{broken} as a set of paths in a group of active links and broken links, $lbest$ is the best solution

1. For *K*=1 to *N*
2. Broadcast RREQ packets from source to destination
3. Navigate the reverse paths through multiple RREP packets
4. End For *K*
5. Select the optimal path with minimal hop count and forward the data
6. For *I*=1 to *P*
7. Predict mobility patterns of the nodes using Eq. (4)
8. Compute the fitness and pheromone level of all paths using Eq. (9) and Eq. (6)
9. Segregate P paths into two different clusters using Eq. (15)
10. End For *I*
11. For *J*=1 to CL_{active}
12. Apply random walk to solutions available in CL_{active} using Eq. (7)
13. Select initial solution MN_k
14. Designate MN_k as $lbest$ solution
15. If (fitness ((MN_{k+1}) > fitness ($lbest$))
16. Update $lbest$ solution using Eq. (8)
17. End If
18. End For *J*
19. Send data to the destination using the new path

B. Performance evaluation metrics

The performance of TCORRA is assessed utilizing five metrics: throughput, packet loss, end-to-end delay, packet delivery ratio, and control packets overhead. The network performance of the proposed approach is assessed regarding varying node mobility velocity and node distribution. The explanation of various metrics is as follows:

Throughput: Throughput can be defined as the rate at which data effectively reaches from the origin to the target node in a network. Conquering higher throughput is desirable for an efficient routing process.

$$Throughput = \frac{\sum Packs\_rec*8}{1024*(Simulation\ time)} \quad (16)$$

Packet Loss: Packet loss can be described as the variation between the overall packets transmitted and effectively attained within the network. This metric is crucial in evaluating the reliability and performance of network connections. Reducing the total packets lost during data transmission boosts network efficiency.

$$Packet\ loss = \sum(P\_Sent)_n - \sum(P\_Rec)_n \quad (17)$$

End-to-End Delay (E2E): End-to-end delay signifies the cumulative period from packet dispatch until it is reliably acknowledged at the destination. It is computed by dividing the total time expended until all data packets are distributed to the target by the total packets communicated from the originating node. Minimizing E2E helps ensure prompt and reliable delivery of data packets. This metric enhances the overall quality and responsiveness of network-based applications and services.

$$E2E = \frac{Time_{rec} - Time_{sent}}{\sum Packets\_sent} \quad (18)$$

Packet Delivery Ratio (PDR): The packet delivery ratio indicates the proportion of total packets that arrive at their targets successfully to the total packets forwarded by source nodes. PDR is used to measure the network's efficiency.

$$PDR = \frac{\text{Total packets delivered}}{\text{Total packets sent}} \times 100 \quad (19)$$

Control Packets Overhead (CO): Control packets overhead can be elaborated as the ratio of control data bytes to total application data bytes sent during transmission. To enhance overall network performance, reducing control packet overhead is desirable.

$$CO = \frac{Total\ control\ data\ bytes\ sent}{Total\ application\ data\ bytes\ sent} \quad (20)$$

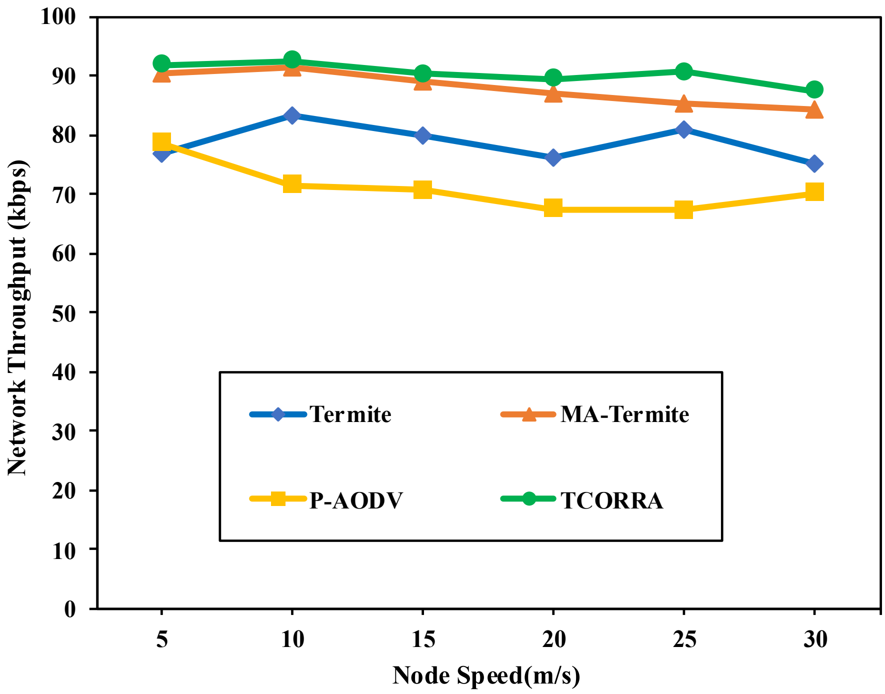

Fig. 2. Performance evaluation of network throughput under varying node speeds

TABLE II

PERFORMANCE ANALYSIS OF VARIOUS ALGORITHMS WITH RESPECT TO NODE MOBILITY SPEED

Node Speed	Comparative Algorithms	Network Throughput	Total Packets Lost in the Network	Packets Delivery Rate in the Network	Total Control Packets Used in the Network	Total Network Delay
5m/s	Termite	76.9154	1851	84.2319	3249	0.2580
	MA-Termite	90.4328	1478	94.3243	2875	0.1894
	P-AODV	78.5553	2050	82.7655	3550	0.2695
	TCORRA	91.8234	1400	96.8382	2666	0.1556
10m/s	Termite	83.0845	2201	82.0845	3585	0.3020
	MA-Termite	91.3233	1470	94.7783	3100	0.2084
	P-AODV	71.5345	2300	79.9875	4566	0.3074
	TCORRA	92.54352	1425	96.9454	3010	0.1811
15m/s	Termite	79.7014	2373	81.6770	4093	0.3344
	MA-Termite	88.8606	1815	93.1658	3677	0.2457
	P-AODV	70.67008	2445	77.7256	5074	0.3491
	TCORRA	90.25234	1523	95.5625	3498	0.2285
20m/s	Termite	76.1194	2202	79.8761	4502	0.3235
	MA-Termite	86.8955	1725	91.8679	3190	0.2744
	P-AODV	67.3401	2500	77.6401	5220	0.3291
	TCORRA	89.3534	1300	94.9835	3010	0.2312
25m/s	Termite	80.9950	2579	80.3344	4765	0.3273
	MA-Termite	85.2741	1710	91.1232	4266	0.2683
	P-AODV	67.2378	2580	74.2678	5498	0.3129
	TCORRA	90.5544	1345	95.1575	3967	0.2412
30m/s	Termite	75.2341	2604	75.5623	5319	0.3145
	MA-Termite	84.1216	1654	89.2761	4701	0.2353
	P-AODV	70.1688	2700	71.8791	5623	0.3273
	TCORRA	87.3243	1410	92.8973	4445	0.2135

C. Performance analysis concerning varying mobility speed

Fig.2 shows a performance comparison of throughput across various speed values. The curve of the throughput graph reveals a downward trend with increasing node speed. TCORRA attains a throughput of 87.3243 kbps, whereas MA-Termite has 84.1216 kbps. Termite has a throughput of 75.2341 kbps, and P-AODV has 70.1688 kbps. The findings validate TCORRA as the top-performing algorithm among the four compared due to reduced link failures. The performance comparison concerning the total number of packets lost under different velocity conditions is demonstrated in fig. 3. TCORRA experiences a loss of 1410 packets while MA-termite reports 1654 lost packets. On the other hand, P-AODV and Termite show losses of 2700 and 2604 packets, respectively. The results highlight TCORRA's superiority over other routing approaches in mitigating packet loss. The performance comparison of delays at varying speed values is presented in fig. 4. TCORRA reports the least delay with 0.2135 secs, while MA-Termite with 0.2353 secs. P-AODV has a delay of 0.32732 secs, and Termite has 0.31453 secs owing to rapid recovery from kink failures. The results underline the efficiency of TCORRA in attaining minimized end-to-end delays in comparison to the other algorithms. The performance comparison regarding PDR is presented in fig. 5. As shown in the graph, TCORRA attains a maximum PDR of 92.8973, followed by MA-Termite at 89.2761. In contrast, Termite achieves a PDR of 75.5623 and P-AODV attains 71.8791. The evaluation analysis regarding control packets overhead is presented in fig. 6. The minimum control packets overhead is generated by TCORRA with 4445 packets, whereas MA-Termite with 4701 packets. On the other hand, Termite has reported 5319 and P-AODV with 5623 control packets. TCORRA incurs less control overhead due to a reduction in the number of route rediscoveries. Table II presents the comparative performance analysis of various routing algorithms concerning varying node mobility speeds.

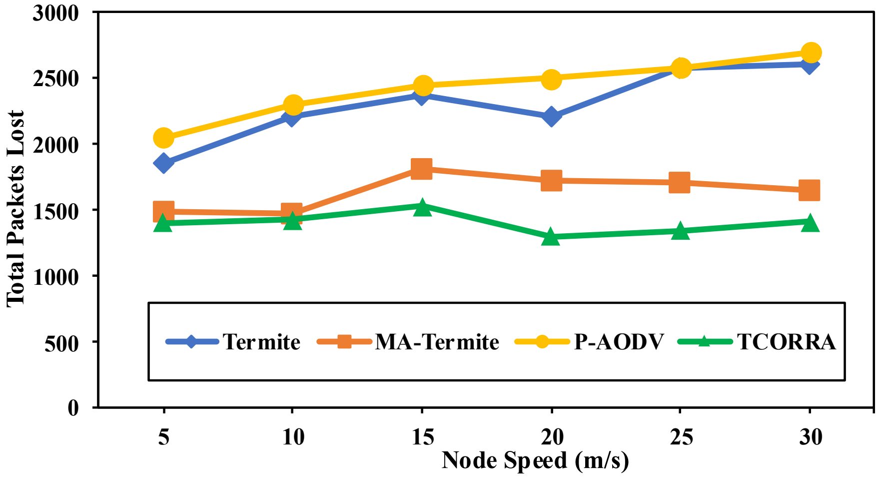

Fig. 3. Performance evaluation of total packets lost under varying node speeds

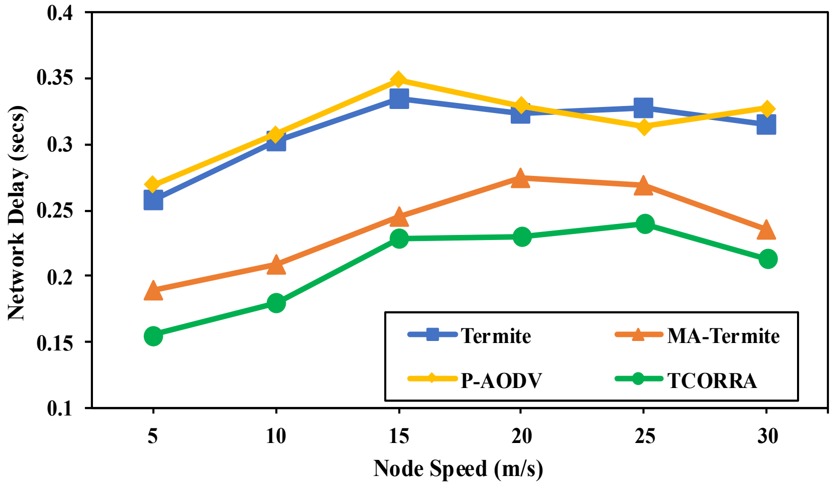

Fig.4 Performance evaluation of network delay under varying node speeds

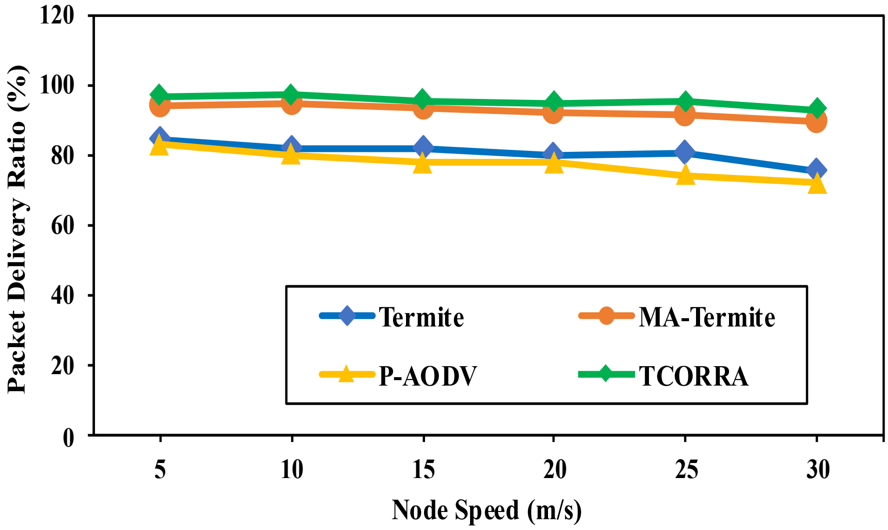

Fig. 5. Performance evaluation of packet delivery ratio under varying node speeds

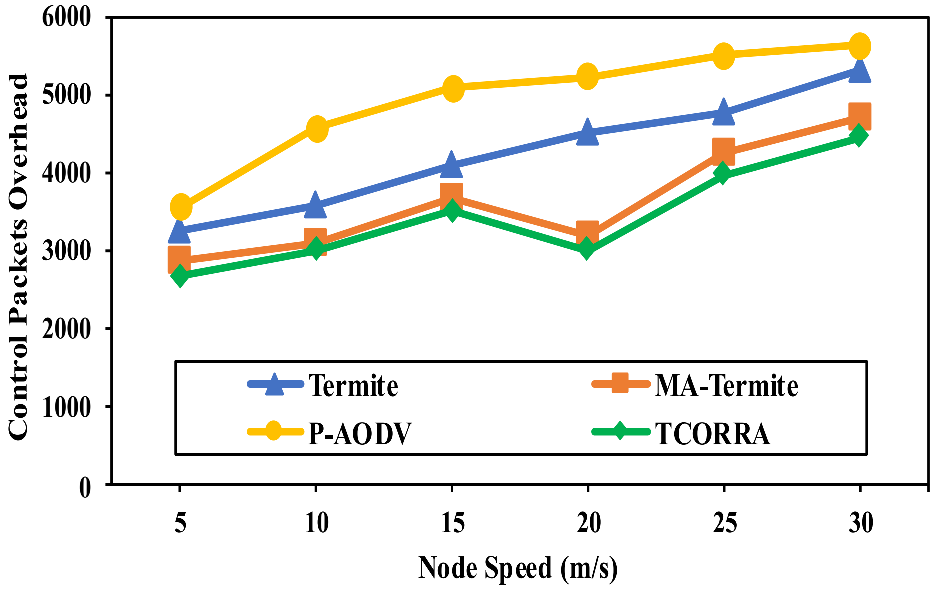

Fig. 6. Performance evaluation of control packets overhead under varying node speeds

D. Performance analysis concerning varying number of network nodes

The network performance comparison regarding throughput based on varying node density is illustrated in fig. 7. The graph indicates that the throughput value decreases with increased network nodes. TCORRA achieves a maximum throughput of 92.5241 kbps, whereas MA-Termite has 90.7875 kbps. Termite has a throughput value of 74.7852 kbps, and P-AODV has 69.3427 kbps. The findings indicate that TCORRA is the top-performing algorithm among the four compared.

The performance comparison concerning the number of lost packets under a variable node density is presented in fig. 8. TCORRA attains a minimum packet loss of 1395 packets, followed by MA-Termite with 1623 packets. On the other hand, Termite has reported packet loss of 2167 packets and P-AODV with 2528 packets. The results indicate that TCORRA outperforms other competitive algorithms in mitigating packet loss during data transmission. Table III presents the comparative analysis of different routing algorithms concerning the total number of network nodes.

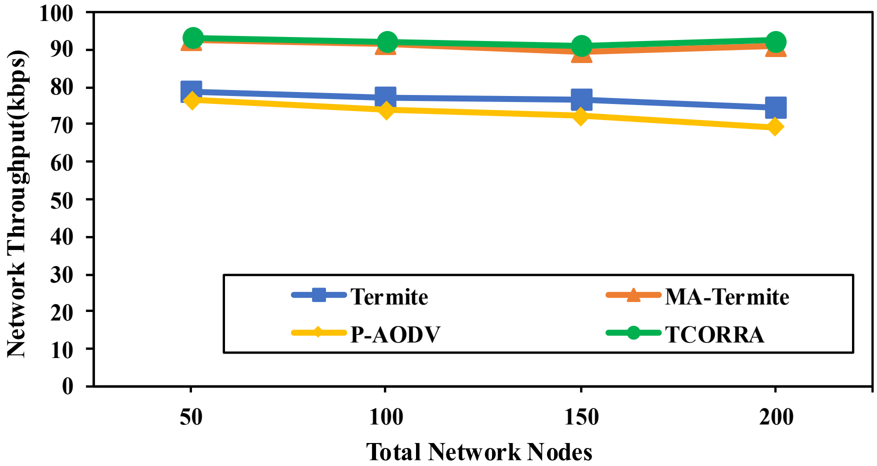

Fig. 7. Performance evaluation of network throughput under varying node density

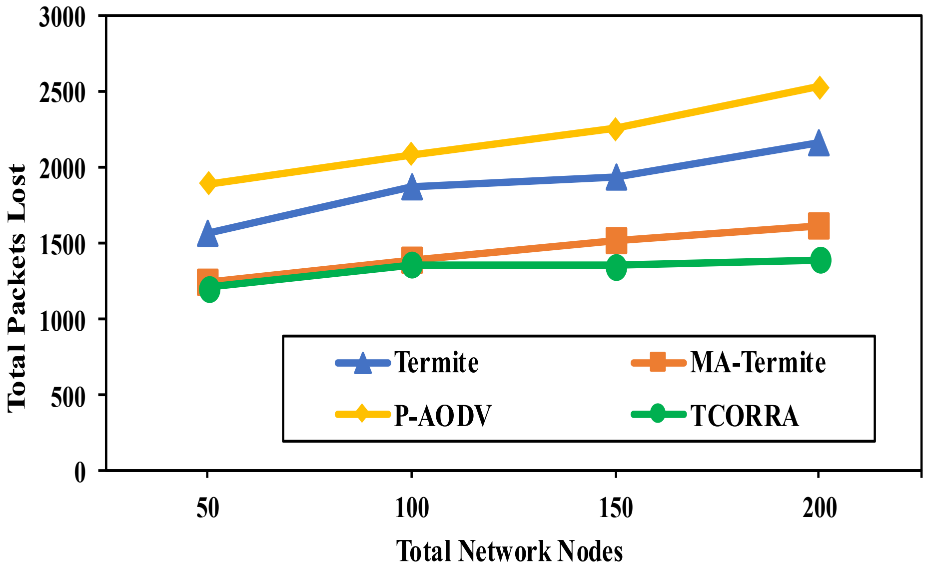

Fig. 8. Performance evaluation of total packets lost under varying node density

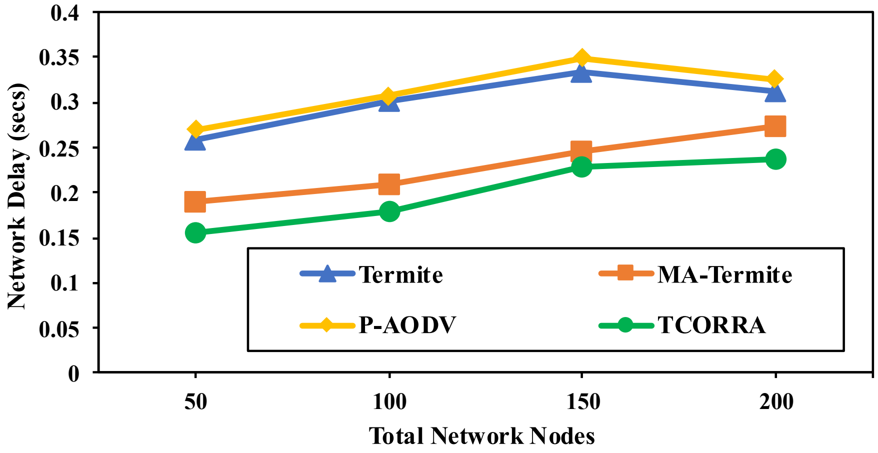

Fig. 9. Performance evaluation of network delay under varying node density

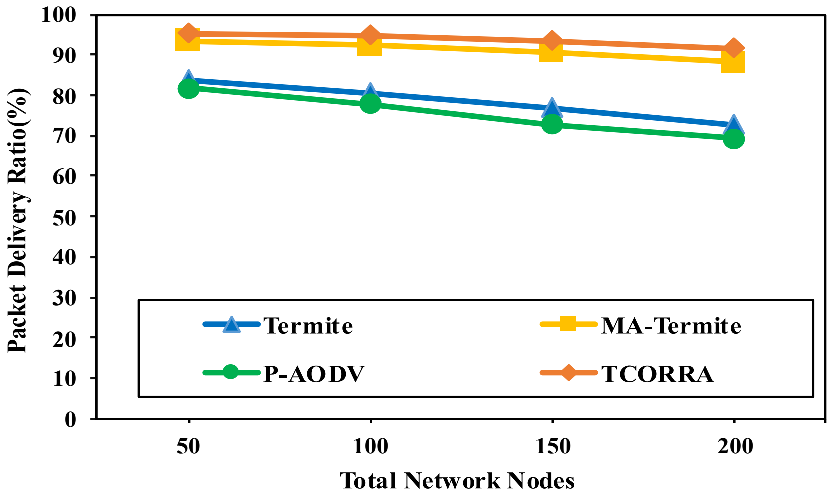

Fig. 10. Performance evaluation of packet delivery ratio under varying node density

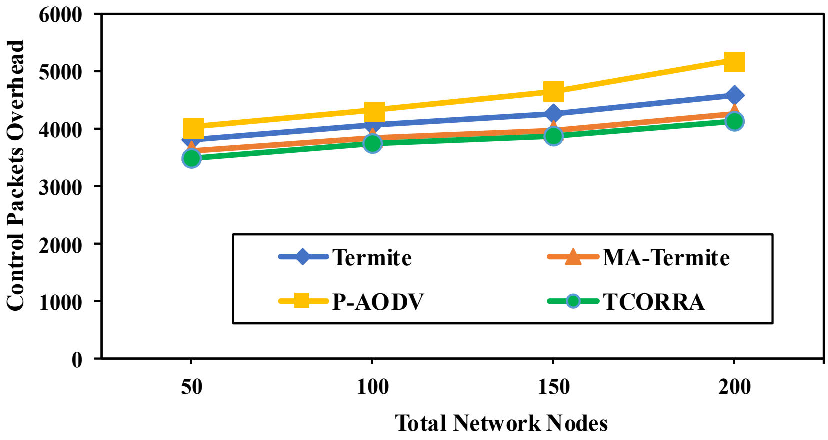

Fig.11. Performance evaluation of control packets overhead under varying node density

TABLE III
PERFORMANCE ANALYSIS OF VARIOUS ALGORITHMS WITH RESPECT TO VARYING NUMBER OF NETWORK NODES

Number of Network Nodes	Comparative Algorithms	Network Throughput	Total Packets Lost in the Network	Packet Delivery Rate in the Network	Total Control Packets Used in the Network	Total Network Delay
50	Termite	78.6623	1567	83.7699	3822	0.2580
	MA-Termite	92.4536	1240	93.5674	3610	0.1954
	P-AODV	76.4639	1897	81.7315	4027	0.2695
	TCORRA	93.4256	1212	95.3456	3498	0.1655
100	Termite	77.4425	1882	80.4824	4065	0.3020
	MA-Termite	91.5532	1389	92.3168	3846	0.2167
	P-AODV	73.8867	2086	77.8233	4323	0.3074
	TCORRA	92.3167	1367	94.8945	3753	0.1966
150	Termite	76.7856	1945	76.8892	4273	0.3344
	MA-Termite	89.2234	1525	90.6689	3963	0.2555
	P-AODV	72.3376	2256	72.6756	4654	0.3491
	TCORRA	91.2284	1355	93.5579	3876	0.2362
200	Termite	74.7852	2167	72.5314	4585	0.3123
	MA-Termite	90.7875	1623	88.2674	4267	0.2744
	P-AODV	69.3427	2528	69.2467	5189	0.3256
	TCORRA	92.5241	1395	91.3978	4125	0.2367

Fig. 9. highlights the performance comparison regarding E2E at varying node densities. P-AODV attains a maximum delay of 0.3256 secs, and TCORRA reports a minimum delay of 0.23679 secs. However, MA-Termite has a delay of 0.27442 secs, and P-AODV has 0.32569 secs. The analysis indicates the efficiency of TCORRA in conquering minimized end-to-end delays when compared with other competitive algorithms. The performance analysis concerning PDR is demonstrated in fig. 10. As shown in the graph, TCORRA attains a maximum PDR of 91.3978, followed by MA-Termite 88.2674. In contrast, Termite has a PDR value of 72.5314, and P-AODV has 69.2467. The comparative analysis based on control packets overhead is shown in fig. 11. As illustrated in the graph, control packet overhead increases with an increased number of nodes. The minimum control packet overhead is generated by TCORRA with 4125 packets, whereas MA-Termite with 4267 packets. On the other hand, Termite has reported an overhead of 4585 packets and P-AODV has 5189 packets. The overhead analysis signifies the efficient performance of TCORRA.

VI. CONCLUSION

The explosion of data-intensive mobile applications such as video conferencing, video streaming, and real-time communication have significantly contributed to the substantial increase in network traffic. This surge in network data flow has accentuated the demand for uninterrupted connectivity and higher transmission rates. Therefore, robust routing approaches are required to handle diverse users and increase data traffic efficiently. Hence, preserving constant connectivity with strong links in MANETs becomes imperative. The proposed algorithm advocates prompt addressing of frequent disruptions caused by link failures through auto-configuration of broken links in MANET. This autonomous link-healing process is inspired by the adaptive behavior observed in termite colonies. The prediction-based analysis of node mobility patterns is the novelty of this repair mechanism. TCORRA forecasts movement patterns of network nodes using the ARIMA model. This forecasted mobility information is utilized to find the fittest path based on community time and packet loss. The outcomes of simulation show that TCORRA outstrips other competitive routing algorithms in terms of performance. At a mobility speed of 30m/s, TCORRA achieved a maximum throughput of 87.324 kbps, with the least packet loss of 1410. However, the E2E delay of TCORRA is 0.2135 secs with a PDR of 92.89%. On the other hand, concerning node density, TCORRA attains a throughput of 92.5241 kbps with a loss of 1395 packets. However, TCORRA had reported a delay of 0.23679 secs with a PDR of 91.3978 and a control overhead of 4125 packets. TCORRA seeks to offer robust connectivity by acclimating smoothly to dynamic conditions. This seamless connectivity enhances the overall user experience in varying topological changes and swift computing growth.

REFERENCES

[1] R. Vatambeti, S. Sanshi, and D. P. Krishna, "An efficient clustering approach for optimized path selection and route maintenance in mobile ad hoc networks," *Journal of Ambient Intelligence and Humanized*

Computing, vol. 14, no. 1, pp. 305–319, 2023, doi: 10.1007/s12652-021-03298-3.

[2] P. Abhilash, S. Perur, and S. Iyer, “Router handoff: An approach for preemptive route repair in mobile Ad Hoc networks,” *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2552, pp. 347–357, 2002, doi: 10.1007/3-540-36265-7_33.

[3] M. Pan, S. Y. Chuang, and S. De Wang, “Local repair mechanisms for on-demand routing in mobile ad hoc networks,” in 2005 *Proc. PRDC Conf.*, pp. 317–324, doi: 10.1109/PRDC.2005.38.

[4] K. Taneja, H. Taneja, and R. Kaur, “Evolutionary Computation Techniques for Intelligent Computing in Commercial Mobile Adhoc Networks,” *International Journal of Next-Generation Computing*, vol. 12, no. 2, pp. 209–217, 2021, doi: https://doi.org/10.47164/ijngc.v12i2.205.

[5] H. Kumar, M. Malakar, D. Sourabh, and M. Rafi, “Study and Design of Route Repairing Mechanism in MANET,” in *Design Frameworks for Wireless Networks*, vol. 82, S. K. Das, S. Samanta, N. Dey, and R. Kumar, Eds., Singapore: Springer Nature, 2020, pp. 139–165.

[6] R. Kaur, K. Taneja, and H. Taneja, “Evolutionary Computation Techniques for Strengthening Performance of Commercial MANETs in Society 5.0,” in *Evolutionary Computation With Intelligent Systems*, C. R.S, T. Kavita, K. Rajiv, K. Vishal, and R. Rahul, Eds., New York: CRC Press Taylor & Francis Group, 2022, pp. 181–196.

[7] H. H. Choi, B. Roh, M. Han, and J. R. Lee, “Regional Route Maintenance Protocol Based on Local Pheromone Diffusion for Mobile Ad Hoc Networks,” *Mobile Networks and Applications*, vol. 23, no. 3, pp. 580–596, 2018, doi: 10.1007/s11036-017-0907-9.

[8] R. Kaur, K. Taneja, and H. Taneja, “Bio-inspired Routing: A Pedestal to Computational Intelligence for Enhancing MANET Performance in Human-Centered Society,” in *Algorithms for Intelligent Systems*, S. Kumar, S. Hiranwal, S. D. Purohit, and M. Prasad, Eds., Singapore: Springer Nature, 2021, pp. 161–174. doi: 10.1007/978-981-99-3485-0.

[9] R. Zhou, Y. Zhang, X. Sun, H. Liu, and Y. Cai, “MSWOA : Multi-strategy Whale Optimization Algorithm for Engineering Applications,” *Engineering Letters.*, vol. 32, no. 8, pp. 1603–1615, 2024.

[10] M. Roth and S. Wicker, “Termite: Ad-Hoc Networking with Stigmergy,” in 2003 *Proc. GLOBECOM*, vol. 5, pp. 2937–2941, doi: 10.1109/glocom.2003.1258772.

[11] A. Kout, S. Labed, S. Chikhi, and E. B. Bourennane, “AODVCS, a new bio-inspired routing protocol based on cuckoo search algorithm for mobile ad hoc networks,” *Wireless Networks*, vol. 24, no. 7, pp. 2509–2519, 2017, doi: 10.1007/s11276-017-1485-2.

[12] P. Kumari and S. K. Sahana, “Swarm Based Hybrid ACO-PSO Meta-Heuristic (HAPM) for QoS Multicast Routing Optimization in MANETs,” *Wireless Personal Communications*, vol. 123, no. 2, pp. 1145–1167, 2022, doi: 10.1007/s11277-021-09174-9.

[13] J. Jain, R. Gupta, and T. K. Bandhopadhyay, “Ant colony algorithm in MANET-local link repairing of AODV,” in 2011 *Proc. ICECT,* vol. 6, pp. 270–273, 2011, doi: 10.1109/ICECTECH.2011.5942096.

[14] A. Sharma and L. Tharani, “Ant Colony Based Node Disjoint Local Repair in Multipath Routing in MANET Network,” *Wireless Personal Communications*, no. 0123456789, 2021, doi: 10.1007/s11277-021-08098-8.

[15] D. Sharma, A. Patra, and C. Kumar, “P-AODV : A Priority Based Route Maintenance Process in Mobile Ad Hoc Networks,” *Wireless Personal Communications*, vol. 95, pp. 4381–4402, 2017, doi: 10.1007/s11277-017-4085-7.

[16] S. Labed, A. Kout, S. Chikhi, and E. B. Bourennane, “A hybrid bio-inspired approach to solving the routing problem in mobile ad hoc networks,” *International Journal of Systems, Control and Communications*, vol. 9, no. 3, pp. 185–209, 2018, doi: 10.1504/IJSCC.2018.093395.

[17] K. Manjappa and R. M. R. Guddeti, “ Mobility aware-Termite: a novel bio inspired routing protocol for mobile ad-hoc networks ,” *IET Networks*, vol. 2, no. 4, pp. 188–195, 2013, doi: 10.1049/iet-net.2012.0203.

[18] N. Fareena and S. Sharmila Kumari, “A distributed fuzzy multicast routing protocol (DFMCRP) for maximizing the network lifetime in mobile ad-hoc networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 5, pp. 4967–4978, 2021, doi: 10.1007/s12652-020-01936-w.

[19] L. J. Garcia Villalba, D. Ruperez Canas, A. L. Sandoval Orozco, and T. H. Kim, “Multiple interface parallel approach of bioinspired routing protocol for mobile Ad Hoc networks,” *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 1–5, 2012, doi: 10.1155/2012/532572.

[20] C. R. da C. Bento and E. C. G. Wille, “Bio-inspired routing algorithm for MANETs based on fungi networks,” *Ad Hoc Networks*, vol. 107, p. 102248, 2020, doi: 10.1016/j.adhoc.2020.102248.

[21] R. Hemalatha, R. Umamaheswari, and S. Jothi, “Optimal route maintenance based on adaptive equilibrium optimization and GTA based route discovery model in MANET,” *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, pp. 3416–3430, 2021, doi: 10.1007/s12083-021-01186-3.

[22] A. Gayathiri and S. Mohanapriya, “A Novel Link Failure Prediction in Cluster Based Routing Protocol for MANETs,” *International Journal of Intelligent Enggnering Systems*, vol. 17, no. 2, pp. 324–332, 2024, doi: 10.22266/ijies2024.0430.27.

[23] V. V Kohir and V. D. Mytri, “Mobility Prediction algorithm to improve the Routing performance in MANET,” *International Journal of Computer Science and Network Security*, vol. 14, no. 9, pp. 39–44, 2014.

[24] G. Singal, V. Laxmi, M. S. Gaur, and V. Rao, “Moralism: mobility prediction with link stability based multicast routing protocol in MANETs,” *Wireless Networks*, vol. 23, no. 3, pp. 663–679, 2017, doi: 10.1007/s11276-015-1186-7.

[25] U. Palani, K. C. Suresh, and A. Nachiappan, “Mobility prediction in mobile ad hoc networks using eye of coverage approach,” *Cluster Computing*, vol. 22, pp. 14991–14998, 2019, doi: 10.1007/s10586-018-2479-2.

[26] R. K. Yadav and M. Balakrishnan, “Comparative evaluation of ARIMA and ANFIS for modeling of wireless network traffic time series,” *Eurasip Journal on Wireless Communications and Networking*, vol. 2014, pp. 2–9, 2014, doi: 10.1186/1687-1499-2014-15.

[27] G. Li and Y. Wang, “Automatic ARIMA modeling-based data aggregation scheme in wireless sensor networks,” *Eurasip Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 1–13, 2013, doi: 10.1186/1687-1499-2013-85.

[28] P. Chen, H. Yuan, and X. Shu, “Forecasting crime using the ARIMA model,” in 2008 *Proc. FSKD,* vol. 5, no. November 2008, pp. 627–630, 2008, doi: 10.1109/FSKD.2008.222.

[29] A. Pal, J. P. Singh, and P. Dutta, “Path length prediction in MANET under AODV routing: Comparative analysis of ARIMA and MLP model,” *Egyptian Informatics Journal*, vol. 16, no. 1, pp. 103–111, 2015, doi: 10.1016/j.eij.2015.01.001.

[30] C. Yan, Y. Zhang, W. Zhong, C. Zhang, and B. Xin, “A truncated SVD-based ARIMA model for multiple QoS prediction in mobile edge computing,” *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 315–324, 2022, doi: 10.26599/TST.2021.9010040.

[31] Z. Guo and B. Malakooti, “Energy Aware Proactive MANET Routing with Prediction on Energy Consumption,” in *Conf. Rec. 2008, Int. Conf. Wirel. Algorithms, Syst. Appl. Energy*, pp. 287–293. doi: 10.1109/wasa.2007.20.

[32] A. M. Zungeru, L. M. Ang, and K. P. Seng, “Termite-hill: Performance optimized swarm intelligence based routing algorithm for wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1901–1917, 2012, doi: 10.1016/j.jnca.2012.07.014.

[33] R. Hedayatzadeh, F. A. Salmassi, M. Keshtgari, R. Akbari, and K. Ziarati, “Termite colony optimization: A novel approach for optimizing continuous problems,” in 2010 *Proc. ICEE 2010*, pp. 553–558, 2010, doi: 10.1109/IRANIANCEE.2010.5507009.

[34] M. Roth and S. Wicker, “Termite: A swarm intelligent routing algorithm for mobile wireless Ad-Hoc networks,” *Studies in Computational Intelligence*, vol. 31, no. 2006, pp. 155–184, 2006, doi: 10.1007/978-3-540-34690-6_7.

[35] R. T. Marler and J. S. Arora, “The weighted sum method for multi-objective optimization: New insights,” *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010, doi: 10.1007/s00158-009-0460-7.

[36] S. Monga, J. L. Rana, and J. Agarwal, “Clustering schemes in mobile ad-hoc network (MANET): A review,” *International Journal of Scientific and Technology Research*, vol. 8, no. 8, pp. 1168–1176, 2019.

[37] G. Singh, A. Kaur, R. Gupta, A. Singh, "Enhancing Throughput in Mobile Ad Hoc Networks Through Ant Colony Optimization Integration," in *Innovative Computing and Communications ICICC 2024, Lecture Notes in Networks and Systems*, A.E. Hassanie, S. Anand, A. Jaiswal, P. Kumar, Eds., Singapore :Springer, 2024, vol. 1038, pp. 461-470. https://doi.org/10.1007/978-981-97-4149-6_31

[38] C. Guo, M. Su, F. Cui, “A Route Self-Repair Method for Wireless Sensor Networks,” in 2004 *Proc. ICCMT*, ISBN:979-8-4007-1826-7, pp. 129-134, 2024.