International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/224449 Volume 11, Issue 2, March – April (2024)

RESEARCH ARTICLE

Dynamic Integration of Fast Furious Cheetah Optimization for Efficient and Secure Routing in Vehicular Ad Hoc Networks

A. Sheela Rini
Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamil Nadu, India.
sheelarini.a@gmail.com

C. Meena
Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamil Nadu, India.
cccmeena@gmail.com

Received: 31 January 2024 / Revised: 25 March 2024 / Accepted: 30 March 2024 / Published: 30 April 2024

Abstract – This research addresses the intertwined challenges of routing efficiency and data security in Vehicular Ad Hoc Networks (VANETs), characterized by dynamic Vehicle-to-Vehicle (V2V) communication. To bolster the Ad Hoc On-Demand Distance Vector (AODV) protocol, Route Life Time Enhanced AODV (RLE-AODV) is introduced, integrating Fast Furious Cheetah Optimization (FFCO) at each protocol step for comprehensive optimization. The robust security measures are concurrently incorporated using an enhanced iteration of Elliptic Curve Cryptography (ECC), which is seamlessly integrated into the secure routing framework. The study meticulously explores the synergistic integration of FFCO with RLE-AODV and ECC, optimizing routing efficiency while fortifying data security. After integration with ECC, the framework transforms into Fast Furious Cheetah Optimization-Based Secured Routing (FFCOSR), ensuring the integrity and confidentiality of data exchanged between vehicles. Through extensive simulations, the FFCOSR framework demonstrates superior performance and heightened security compared to conventional approaches in V2V VANETs. By orchestrating FFCO within RLE-AODV, the approach dynamically adjusts routing parameters to adapt to changing network conditions, prolonging route stability and enhancing overall network performance. This research significantly advances state-of-the-art efficient and secure vehicular communication, offering valuable insights into the synergy of optimization techniques for addressing multifaceted network challenges. The proposed FFCOSR framework represents a promising avenue for improving the reliability and security of V2V communication in VANETs, with potential applications in real-world scenarios where robustness and efficiency are paramount.

Index Terms – Ad Hoc On-Demand Distance Vector Routing, Particle Swarm Optimization, Machine Learning, Network Lifespan, Energy Balancing, Localization, Clustering, Routing Overhead, Throughput, End-to-End Delay.

1. INTRODUCTION

Vehicular Ad hoc Networks (VANETs) are specialized wireless communication networks that enable vehicles to communicate with each other and with roadside infrastructure. The primary objective of VANETs is to enhance road safety and optimize traffic flow through real-time information exchange among vehicles [1].

These networks use technologies like Dedicated Short Range Communication (DSRC) or Cellular Vehicle-to-Everything (C-V2X) to enable direct communication between vehicles and infrastructure. One key feature of VANETs is their dynamic and decentralized nature. Vehicles within the network form temporary connections based on their proximity, allowing them to share information about their speed, position, and other relevant data [2].

This information exchange helps vehicles make informed decisions, such as adjusting speed to avoid congestion or potential hazards. VANETs contribute to developing Intelligent Transportation Systems (ITS), offering the potential for improved transportation efficiency and reduced traffic congestion. As vehicles become increasingly equipped with communication capabilities, VANETs pave the way for innovative transportation solutions that leverage the connectivity of vehicles for more responsive and adaptive traffic management [3].

VANETs employ various types of communication to facilitate interaction among vehicles and between vehicles and roadside infrastructure. The primary communication types in VANETs include [4]–[6]:

RESEARCH ARTICLE

- Vehicle-to-Vehicle (V2V) Communication: In V2V communication, vehicles directly exchange information. This can include data about speed, position, acceleration, and other relevant parameters. V2V communication enables real-time collaboration among vehicles, helping them react to changes in the road environment, such as sudden braking or the presence of obstacles.
- Vehicle-to-Infrastructure (V2I) Communication: V2I communication involves data exchange between vehicles and roadside infrastructure, such as traffic lights, road signs, or toll booths. This type of communication enhances overall traffic management and allows vehicles to receive information about traffic conditions, road hazards, or updates on traffic signal timings.
- Vehicle-to-Everything (V2X) Communication: V2X is a comprehensive term that encompasses communication not only between vehicles (V2V) and between vehicles and infrastructure (V2I) but also includes communication with pedestrians (V2P) and other elements of the environment (V2E). V2X communication aims to create a holistic network where vehicles interact with various entities, enhancing safety and efficiency in diverse traffic scenarios.
- Vehicle-to-Roadside (V2R) Communication: Similar to V2I, V2R communication refers explicitly to vehicle and roadside infrastructure elements. This can include data exchange with fixed sensors, cameras, or other devices installed along the road.
- Vehicle-to-Network (V2N) Communication: V2N communication involves vehicles communicating with a central network or cloud-based system. This allows for broader data analytics, monitoring traffic, and disseminating traffic-related information to a larger audience.

In VANETs, vehicles must exchange information about their current positions, speeds, and other relevant data to enable efficient and reliable routing [7], [8]. Routing in VANETs involves determining the best paths for data transmission between vehicles, considering factors like traffic conditions, road topology, and potential obstacles. V2V communication plays a crucial role in this process by allowing vehicles to share this information with nearby vehicles. The exchanged data can be used to make informed decisions about the optimal routes and to avoid congested or unsafe areas [9]. Routing protocols in VANETs often leverage the dynamic nature of V2V communication to establish and update routes in real-time. Examples of routing protocols used in VANETs include Geographic Routing, Beacon-Based Routing, and Position-Based Routing. Routing in V2V communication involves determining efficient paths for data transmission between vehicles, and several challenges are associated with this process, as outlined below [10], [11]:

- Dynamic Network Topology: Rapid changes in vehicle positions and connectivity create a vibrant network topology, posing difficulties in establishing and maintaining stable and reliable V2V communication links.
- Interference and Signal Attenuation: Wireless signal interference and attenuation due to obstacles in urban environments can lead to communication disruptions and reduced effectiveness of V2V communication.
- Scalability Issues: As the number of vehicles on the road increases, scalability becomes a concern, impacting the efficiency of V2V communication protocols and the ability to handle many simultaneous connections.
- Quality of Service (QoS) Requirements: Meeting stringent QoS requirements, such as low latency and high reliability, is challenging in vehicular movement's dynamic and unpredictable nature.
- Security and Privacy Concerns: Ensuring the security and privacy of V2V communication is critical, as the exchange of sensitive information between vehicles could be vulnerable to malicious attacks, posing risks to safety and confidentiality.

Bio-inspired optimization for ad hoc networks leverages nature's principles to tackle challenges. Mimicking biological systems, these methods optimize network performance, energy efficiency, and resource allocation [12], [13]. Drawing from natural phenomena like self-organization, adaptation, and cooperation, bio-inspired approaches offer innovative solutions tailored to the dynamic and decentralized nature of ad hoc networks [14]. Such techniques promote robustness, scalability, and adaptability, enhancing the network's ability to handle mobility, limited resources, and changing environmental conditions [15].

1.1. Problem Statement

In the rapidly evolving landscape of VANETs, a significant problem arises in ensuring seamless communication and collaboration among vehicles. The challenge lies in the diverse nature of vehicles on the road, ranging from various makes and models to differing communication capabilities. Establishing a standardized communication protocol that caters to this diversity is a complex puzzle, leading to interoperability issues and communication gaps. Given the resource constraints inherent in vehicular devices, efficient real-time data processing at the network edge becomes a critical concern. The struggle to balance the need for rapid data handling with the limitations of available resources poses a significant hurdle in achieving optimal VANET performance. Energy efficiency emerges as a pressing

RESEARCH ARTICLE

challenge as the quest to minimize power consumption for prolonged device operation encounters difficulties. This challenge impacts the operational lifespan of V2V communication modules and raises questions about the sustainability of energy usage in a connected vehicular environment. In navigating these challenges, the seamless integration of VANETs with autonomous vehicles and intelligent infrastructure faces obstacles in establishing consistent communication protocols across diverse technological domains. The resulting lack of a harmonized system impedes the full potential of these networks in enhancing road safety and traffic optimization. As we address these challenges, the ultimate goal is to create a robust and adaptive vehicular communication framework that ensures the reliability, efficiency, and security of V2V communication in VANETs.

1.2. Motivation

The challenges inherent in VANETs form a compelling motivation for research endeavors. In a context where vehicles exhibit diverse characteristics, there is a clear imperative to establish a standardized communication protocol. This necessity propels the exploration of solutions that can effectively bridge interoperability gaps, enhancing the reliability of V2V communication. The urgency to process real-time data efficiently at the network edge serves as a driving force for innovative research approaches. The goal is to optimize VANET performance within the inherent resource constraints of vehicular devices. Energy efficiency becomes a pivotal focus, motivating the investigation of sustainable practices that minimize power consumption while ensuring the extended and reliable operation of V2V communication modules. This motivation extends to overcoming challenges associated with seamlessly integrating VANETs with autonomous vehicles and intelligent infrastructure. The ultimate objective is to unlock the full potential of these networks, fostering revolutionary advancements in road safety and traffic optimization. This research is inspired by a commitment to address these challenges to advance V2V communication in VANETs and contribute meaningfully to the broader landscape of intelligent transportation systems. By delving into these complexities, the aim is to pave the way for a robust, adaptive, and secure vehicular communication framework that can shape the future of transportation technologies.

1.3. Research Objective

The primary objective of this research is to introduce and develop a bio-inspired optimization-based routing protocol as an effective solution to address the challenges prevalent in VANETs. This involves designing a robust routing framework inspired by biological systems to enhance communication paths' adaptability, efficiency, and resilience in dynamic vehicular environments.

This research seeks to rigorously assess and enhance its adaptability, performance, security, real-world applicability, and resource efficiency as its secondary objectives, which are given below:

- Routing Protocol Development: Develop bio-inspired optimization algorithms, drawing inspiration from biological systems like swarm intelligence or genetic algorithms, to form the foundation of the proposed routing protocol.
- Adaptability Assessment: Evaluate the adaptability of the bio-inspired routing protocol to dynamic changes in VANET topologies, ensuring that it can efficiently respond to variations in vehicle positions and network connectivity.
- Performance Comparison: Conduct comprehensive simulations to compare the performance of the proposed bio-inspired routing protocol against existing VANET routing protocols. Assess critical metrics such as communication reliability, latency, scalability, and adaptability to varying traffic conditions.
- Real-world Validation: Validate the effectiveness of the bio-inspired routing protocol through real-world scenarios, considering factors like urban and rural environments, diverse weather conditions, and fluctuating traffic densities.
- Security Integration: Integrate security measures into the bio-inspired routing protocol to ensure the robustness of communication paths against potential threats or malicious activities, contributing to the overall reliability of V2V communication.
- Resource Efficiency Analysis: Analyze the resource efficiency of the proposed routing protocol, mainly focusing on power consumption and computational requirements. Strive to minimize energy usage while maintaining optimal communication performance.

1.4. Organization of the Paper

The paper is organized into several sections to comprehensively analyse the dynamic integration of Fast Furious Cheetah Optimization for efficient and secure routing in Vehicular Ad Hoc Networks (VANETs). The Introduction (Section 1) outlines the problem statement, research motivation, and objectives (1.1, 1.2, 1.3). The Literature Review (Section 2) delves into existing research gaps (2.1) to establish the study's context. The paper's core lies in the Fast Furious Cheetah Optimization-Based Secured Routing Protocol (Section 3), which details the proposed approach. Results and Discussion (Section 4) analyze simulation settings, energy consumption, packet delay, loss, and throughput (4.1-4.5). Finally, the Conclusion (Section 5)

RESEARCH ARTICLE

summarizes the findings and suggests future research directions. This structured organization ensures a logical progression, facilitating a clear understanding of the research methodology, results, and implications.

2. LITERATURE REVIEW

"EdgeVehRoute" [16] integrates edge computing into multi-connected vehicles for cooperative route planning across diverse domains. The system optimizes route decisions by leveraging edge resources for real-time data processing. It establishes seamless vehicle communication, utilizing edge servers to exchange information about traffic, road conditions, and preferences. "CAVRouteControl" [17] evaluates the significance of optimal routing and signal timing control strategies in the context of connected autonomous vehicles (CAVs). The system employs advanced algorithms to optimize route selection and signal timing, considering real-time data from CAVs. It maximizes traffic flow efficiency by dynamically adjusting routes and signal timings based on vehicle connectivity and autonomy. "6GAutoRoute" [18] presents an autonomous vehicle routing protocol tailored for 6G networks, incorporating computational intelligence for efficient and trusted navigation. The protocol harnesses advanced algorithms to dynamically allocate resources, ensuring optimal routing for autonomous vehicles. It adapts to real-time traffic conditions using computational intelligence, enhancing route efficiency.

"AutoDynAlloc" [19] introduces an Automatic Dynamic User Allocation system coupled with opportunistic routing over the vehicular network for Intelligent Transport Systems (ITS). The system dynamically allocates resources, optimizing user assignments in real-time. Opportunistic routing leverages the mobility of vehicles to enhance data transmission efficiency. "K-MORP-UAV" [20] introduces a K-means online-learning routing protocol for Unmanned Aerial Vehicles (UAV) ad-hoc networks. This protocol leverages K-means clustering and online learning to dynamically adapt to changing network conditions. It optimizes routing for UAVs, ensuring efficient communication in ad-hoc scenarios. "TS-CAGR-IoV" [21] presents a Traffic Sensitive Connectivity-Aware Geocast Routing (TS-CAGR) protocol tailored for the Internet of Vehicles (IoV). The protocol prioritizes traffic sensitivity and connectivity awareness in geocast routing decisions, optimizing communication within the IoV ecosystem. By considering real-time traffic conditions and connectivity status, TS-CAGR improves the efficiency of geocast communication in vehicular networks.

"ReliableClusterRoute" [22] addresses cybersecurity threats in the Internet of Vehicles (IoV) communication system by implementing a robust clustering and routing approach. The protocol employs reliable clustering mechanisms to enhance network security, creating resilient clusters that mitigate potential cyber threats. "MMWaveITSRoute" [23] introduces an energy-efficient data transmission solution for Intelligent Transportation Systems (ITS) utilizing millimeter-wave (mmWave) based routing algorithms for connected vehicles. The protocol optimizes data transmission by leveraging mmWave technology, which offers high bandwidth and low latency.

"Evolutionary Algorithm Vehicular Clustering" [24] introduces an Evolutionary Algorithm-Based Vehicular Clustering Technique for VANETs. This technique utilizes evolutionary algorithms to dynamically form and optimize clusters in VANETs, addressing the complexities of vehicular environments. Adapting to the changing conditions and communication requirements of vehicular scenarios, the evolutionary algorithm enhances the effectiveness of clustering strategies. "Connectivity Enhance" [25] presents an innovative approach to improving the connectivity of Electric VANETs (E-VANETs) through the QL-mRSU Self-Learning Energy-Saving Algorithm. This solution optimizes connectivity in E-VANETs by leveraging a self-learning algorithm within Quick Learning (QL) mechanisms and Mobile Roadside Units (mRSUs). "SOMACA-SwarmOpt-Mobility Clustering-IoV" [26] introduces SOMACA, a novel Swarm Optimization-Based and Mobility-Aware Clustering Approach designed for the Internet of Vehicles (IoV). This approach leverages swarm optimization techniques to form IoV clusters, considering vehicle mobility patterns dynamically. "Bio-inspired Optimization-based Routing Protocols" [27], [28] have significant performances in terms of saving energy consumption in ad hoc networks.

"Dynamic Topo-Reliable Route (DTE-RR)" [29] significantly contributes to VANETs by introducing a hybrid routing algorithm that dynamically adapts to the changing network topology. The critical innovation involves integrating Genetic Algorithms (GAs) and the Firefly Algorithm to optimize communication paths. DTE-RR excels in real-time monitoring and predictive modelling, allowing anticipatory adaptations for efficient route optimization. Despite challenges such as computational complexity and parameter sensitivity, the algorithm proves valuable in its ability to adapt to varying network sizes, anticipate changes, and enhance reliability in communication. The Pseudocode of DTE-RR is provided in Algorithm 1.

Step 1: Initialize: $\alpha i = 0$, $fi = -yi$

Step 2: Compute: b_{high}, I_{high}, b_{low}, I_{low}

Step 3: Update α_{Ihigh} and α_{Illow}

Step 4: while True:

Step 5: Update fi

Step 6: Compute: b_{high}, I_{high}, b_{low}, I_{low}

Step 7: Update α_{Ihigh} and α_{Illow}

RESEARCH ARTICLE

Step 8: if $b_{low} <= b_{up} + 2 * \tau$:

Step 9: break

Step 10: Store the new α1 and α2 values

Step 11: Update weight vector w if SVM is linear

Step 12: Update the threshold b

Algorithm 1 DTE-RR

"Hybrid Genetic-Firefly Routing (HGFR)" [30] stands out in routing algorithms due to its pragmatic integration of Genetic Algorithms (GA) and Firefly Algorithms. The core contribution lies in the algorithm's ability to efficiently explore solution spaces using genetic operators and dynamically adapt to changing network conditions inspired by firefly behavior. HGFR excels in optimizing routes within complex networks, showcasing a balanced and practical approach to exploration and exploitation. Its practicality lies in providing robust solutions for dynamic routing scenarios, catering to the challenges posed by unpredictable network environments. The Pseudocode of HGFR is provided in Algorithm 2.

Step 1: $initialize_{parameters}()$;

Step 2: initialize population(routes, $genetic_{parameters}$, $firefly_{parameters}$);

Step 3: $evaluate_{fitness}$(routes);

Step 4: $convergence_{criteria\_met}$:= False;

Step 5: while not $convergence_{criteria\_met}$ do

Step 6: $selected_{routes}$:= $select_{parents}$(routes);

Step 7: recombine($selected_{routes}$);

Step 8: mutate($selected_{routes}$);

Step 9: $adapt_{routes\_firefly}$($selected_{routes}$, $firefly_{parameters}$);

Step 10: $evaluate_{fitness}$($selected_{routes}$);

Step 11: routes := $select_{survivors}$(routes, $selected_{routes}$);

Step 12: $convergence_{criteria\_met}$:= $check_{convergence\_criteria}()$;

Step 13: end while

Algorithm 2 HGFR

2.1. Research Gap

The literature reviewed in the context of vehicular communication networks highlights various routing protocols, including adaptive routing with genetic algorithms, firefly-inspired approaches, and multi-objective optimization. However, a distinct research gap emerges concerning the absence of a new, dynamic optimization-based secure routing technique in the existing work. The identified research gap underscores the pressing need for a routing paradigm that integrates dynamic optimization principles with robust security measures. The current literature lacks a comprehensive exploration of a routing solution that optimizes routes in response to changing conditions and prioritizes security in the inherently vulnerable VANET environment. As intelligent transportation systems increasingly rely on secure and efficient vehicle communication, a novel approach that addresses the dual challenges of adaptability and security becomes imperative. A new cum dynamic optimization-based secure routing technique would not only fill this research gap but also respond to the evolving demands of VANETs. Such a technique could significantly enhance the resilience and security of vehicular communication, providing a crucial foundation for the reliability and efficiency of future intelligent transportation systems.

3. FAST FURIOUS CHEETAH OPTIMIZATION-BASED SECURED ROUTING (FFCOSR)

FFCOSR is a cutting-edge framework tailored for Vehicular Ad Hoc Networks (VANETs). It integrates Fast Furious Cheetah Optimization (FFCO) with Route Life Time Enhanced AODV (RLE-AODV) and enhanced Elliptic Curve Cryptography (ECC), ensuring both optimal routing efficiency and robust data security. By dynamically adjusting routing parameters and fortifying data exchanges, FFCOSR offers superior performance and heightened security compared to traditional VANET approaches. Through meticulous simulations, FFCOSR showcases its ability to adapt to changing network conditions, prolong route stability, and enhance overall communication reliability in V2V scenarios, promising advancements in secure and efficient vehicular communication.

3.1. Route Lifetime Enhanced AODV (RLE-AODV)

Route Lifetime Enhanced AODV (RLE-AODV) is a refined version of AODV designed for VANET. It prioritizes route stability by extending the lifetime of established routes. By optimizing route discovery and maintenance, RLE-AODV minimizes disruptions, enhancing overall network efficiency. This improvement ensures more reliable communication in dynamic environments, making RLE-AODV a valuable advancement in ad-hoc network routing protocols.

3.1.1. Adaptive Route Expiration Time

Adaptive Route Expiration Time (ARET) is a critical aspect of the Route Lifetime Enhanced AODV (RLE-AODV) protocol, aiming to dynamically adjust the expiration time of routes based on observed stability and reliability. The rationale behind ARET lies in adapting the route lifetime to the changing conditions of a mobile ad-hoc network (MANET), where the stability of routes can vary due to node mobility, link quality, and other dynamic factors. To

RESEARCH ARTICLE

mathematically describe ARET, let's establish a framework that considers the stability metrics of a route. Let R_i represent a route between a source node and a destination node in the network, and let $S_i(t)$ denote the stability metric of R_i at time t. Link quality, packet loss rate, or historical stability observations can influence the metric. The adaptive expiration time (E_i) of R_i at time t, denoted by $E_i(t)$, is determined as Eq.(1).

$$E_i(t) = \alpha.S_i(t) + \beta.\big(1 - S_i(t)\big) \quad (1)$$

Where α and β are configurable parameters that weigh the stability and instability components, respectively. The term $\alpha.S_i(t)$ represents the contribution of the stability metric to the expiration time, while $\beta.\big(1 - S_i(t)\big)$ represents the contribution of the instability component. The combination of these components allows ARET to adapt dynamically to the observed route stability.

To further enhance the adaptability of ARET, a temporal smoothing factor (γ) can be introduced to consider the historical stability trends of the route. The updated stability metric $(S_i'(t))$ is then given by Eq.(2).

$$(S_i'(t)) = \gamma.S_i'(t-1) + (1-\gamma).S_i(t) \quad (2)$$

Where $S_i'(t-1)$ is the previous stability metric, and $S_i'(t)$is the updated stability metric at time t. The temporal smoothing factor γ determines the influence of historical stability on the current stability metric.

The adaptive expiration time $(E_i(t))$ can now be modified to incorporate the historical stability component, expressed as Eq.(3).

$$E_i(t) = \alpha.S_i'(t) + \beta.\big(1 - S_i'(t)\big) \quad (3)$$

3.1.2. Predictive Route Lifetime

Predictive Route Lifetime (PRL) builds upon the foundation laid by ARET. While ARET dynamically adjusts route expiration times based on observed stability, PRL introduces predictive analytics to estimate the future stability of routes, further enhancing the adaptability and efficiency of route management in MANETs. Let's denote the predictive stability metric of a route R_i at time t as $P_i(t)$.

This predictive metric is influenced by historical stability observations and is projected into the future. The PRL mechanism uses this predictive stability to estimate the route's future reliability and adjust its expiration time accordingly. The predictive stability $\big(P_i(t)\big)$ can be formulated as a function of the historical stability $\big(S_i'(t)\big)$ and other relevant parameters:

$$P_i(t) = f.(S_i'(t), \dots) \quad (4)$$

Where $f(\dots)$ represents the predictive function that considers historical stability and potentially other factors impacting future stability.

The predictive expiration time $\Big(P_{E_i}(t)\Big)$ of R_i at time t is then determined using Eq.(5) which is the predicted stability:

$$P_{E_i}(t) = \alpha.P_i(t) + \beta.\big(1 - P_i(t)\big) \quad (5)$$

Where parameters α and β weigh the contribution of predicted stability and its complement to the expiration time, similar to the ARET mechanism. This predictive approach allows RLE-AODV to anticipate potential variations in route stability, enabling proactive adjustments to expiration times before actual instability occurs.

Machine learning or statistical models can be integrated into the predictive function to refine the prediction process further, leveraging historical stability data to make more accurate forecasts. This introduces a learning factor (λ) that influences the impact of historical data on predictive stability:

$$P_i(t) = \lambda.P_{i-1}(t-1) + (1-\lambda).S_i'(t) \quad (6)$$

Where $P_{i-1}(t-1)$ is the previous predictive stability and $S_i'(t)$ is the updated stability metric at time t.

3.1.3. Traffic Load-based Lifetime Extension

Traffic Load-based Lifetime Extension (TLLE) introduces an innovative approach to dynamically adjust route expiration times based on the real-time traffic load experienced by individual routes. Let's denote the traffic load on a route R_i at time t as $L_i(t)$. Packet transmission rates, data volume, and congestion levels along the route can influence this traffic load metric. The adaptive expiration time $(E_i(t))$ of R_i at time t, incorporating the TLLE mechanism, is given by Eq.(7).

$$E_i(t) = \alpha.S_i'(t) + \beta.(1 - S_i'(t)) + \gamma.L_i(t) \quad (7)$$

Where γ represents the weight assigned to the traffic load component in the overall expiration time calculation. The TLLE mechanism recognizes that routes experiencing higher traffic loads may need more frequent updates to prevent premature expiration due to increased data transmission demands.

To avoid abrupt changes in expiration times caused by instantaneous traffic fluctuations, a smoothing factor (δ) can be introduced to account for the average traffic load over a specific time window using Eq.(8).

RESEARCH ARTICLE

$$\bar{L}_i(t) = \delta.\bar{L}_i(t-1) + (1-\delta).L_i(t) \quad (8)$$

Where $\bar{L}_i(t)$ represents the smoothed or averaged traffic load and $\bar{L}_i(t-1)$ is the previous smoothed traffic load. The parameter δ determines the influence of historical traffic load on the current smoothed value.

Considering the smoothed traffic load, the adjusted expiration time with TLLE is expressed in Eq.(9).

$$E_i(t) = \alpha.S_i'(t) + \beta.(1-S_i'(t)) + \gamma.\bar{L}_i(t) \quad (9)$$

Incorporating TLLE in RLE-AODV acknowledges the impact of varying network traffic loads on route stability.

3.1.4. Link Stability Monitoring

Link Stability Monitoring (LSM) aims to fortify the protocol by incorporating a real-time assessment of the stability of individual links within a route. By dynamically considering the link stability, RLE-AODV becomes more adept at predicting and responding to potential disruptions, contributing to further refinements in route lifetime management. Let denote the strength of the link between nodes i and j at time t as $L_{ij}(t)$. This link stability metric may encapsulate signal quality, error rates, and historical observations. The link stability for a given link is influenced by the stability metrics of its constituent nodes and the quality of the communication channel between them. The link stability monitoring mechanism adjusts the adaptive expiration time $(E_i(t))$ to consider the stability of each link $(L_{ij}(t))$ along the route. The modified expiration time is expressed as Eq.(10).

$$E_i(t) = \alpha.S_i'(t) + \beta.(1-S_i'(t)) + \gamma.\sum_j L_{ij}(t) \quad (10)$$

Where γ represents the weight assigned to the link stability component in the overall expiration time calculation, the summation is taken over all links (j) in the route.

To enhance the robustness of LSM, a smoothing factor (ϵ) is introduced to account for the historical link stability over time, expressed in Eq.(11).

$$\bar{L}_{ij}(t) = \epsilon.\bar{L}_{ij}(t-1) + (1-\epsilon).L_{ij}(t) \quad (11)$$

Where $\bar{L}_{ij}(t)$ represents the smoothed or averaged link stability for the link between nodes i and j, and $\bar{L}_{ij}(t-1)$is the previous smoothed link stability. The parameter ϵ controls the influence of historical link stability on the current smoothed value.

The adjusted expiration time with LSM, considering the smoothed link stability is then given by Eq.(12).

$$E_i(t) = \alpha.S_i'(t) + \beta.(1-S_i'(t)) + \gamma.\sum_j L_{ij}(t) \quad (12)$$

Integrating LSM into RLE-AODV allows the protocol to dynamically adapt route expiration times based on the stability of individual links.

3.1.5. Proactive Route Refresh

Proactive Route Refresh (PRR) aims to optimize route maintenance by anticipating potential route expiration and initiating timely updates to ensure continuous connectivity in MANETs. The proactive nature of PRR distinguishes it from traditional reactive route maintenance approaches, where route adjustments are triggered only when a route is about to expire. PRR, on the other hand, takes a preemptive stance by periodically refreshing routes before they expire. This proactive approach helps to minimize the delay and potential disruptions associated with reactive route discovery processes. Let R_i be a route between a source and a destination node, and $E_i(t)$ represent the adaptive expiration time of R_i at time t. Eq.(13) adjusts the expiration time based on a proactive factor (ρ).

$$E_i(t) = \alpha.S_i'(t) + \beta.(1-S_i'(t)) + \gamma.\sum_j L_{ij}(t) + \rho \quad (13)$$

Where ρ represents the proactive factor added to the expiration time to initiate proactive route refreshment.

The proactive factor (ρ) is determined by considering factors such as historical stability trends, network conditions, and the desired level of proactiveness. A balance must be struck to avoid unnecessary overhead caused by frequent proactive refreshments while ensuring that routes remain up-to-date and reliable.

3.1.6. Energy-Aware Route Maintenance

Energy-Aware Route Maintenance (EARM) recognizes the critical role of node energy levels in VANETs and aims to optimize route maintenance based on the energy efficiency of individual nodes. Let's denote the energy level of node i at time t as $E_i(t)$. This energy metric reflects the available energy resources at the node, which can be crucial in determining the route's effectiveness and reliability. Building upon the existing route expiration time calculation, the EARM mechanism adjusts the adaptive expiration time $(E_i(t))$ to account for the energy level of the nodes along the route. The modified expiration time is expressed as Eq.(14).

$$E_i(t) = \alpha.S_i'(t) + \beta.(1-S_i'(t)) + \gamma.\sum_j \bar{L}_{ij}(t) + \rho + \delta.E_i(t) \quad (14)$$

Where δ represents the weight assigned to the energy level component in the overall expiration time calculation.

RESEARCH ARTICLE

A dynamic threshold (ϕ) can be introduced to determine the necessity of proactive adjustments based on the node's energy level and expressed as Eq.(15) to ensure that the energy-aware adjustments are adequate.

$$Proactive\ Refresh\ Decision: if\ E_i(t) < \phi, \quad then\ initiate\ a\ proactive\ refresh \tag{15}$$

This decision-making process ensures that proactive refreshments are triggered when a node's energy level falls below a certain threshold, balancing the need for energy-aware maintenance with the potential overhead of frequent route updates. To enhance the EARM mechanism, an energy smoothing factor (ζ) can be introduced as Eq.(16) to account for the historical energy level over time.

$$\bar{E}_i(t) = \zeta . \bar{E}_i(t-1) + (1-\zeta) . E_i(t) \tag{16}$$

Where $\bar{E}_i(t)$ represents the smoothed or averaged energy level for node i, and $\bar{E}_i(t-1)$ is the previous smoothed energy level. The parameter ζ controls the influence of historical energy levels on the current smoothed value.

The adjusted expiration time with EARM, considering the smoothed energy level is then given by Eq.(17).

$$E_i(t) = \alpha . S_i'(t) + \beta . \left(1 - S_i'(t)\right) + \gamma . \sum_j \bar{L}_{ij}(t) + \rho + \delta . \bar{E}_i(t) \tag{17}$$

3.1.7. Cross-Layer Optimization for Lifetime

The Cross-Layer Optimization for Lifetime (CLOL) step aims to leverage interactions between different protocol stack layers, enhancing route maintenance's adaptability and efficiency in MANETs. Let's denote the cross-layer optimization factor at time t as $CLOL(t)$. The cross-layer optimization factor represents the collective influence of various parameters from different protocol layers, such as network, transport, and physical layers, on the route maintenance decisions. Building upon the existing route expiration time calculation, the CLOL mechanism adjusts the adaptive expiration time $\left(E_i(t)\right)$ by incorporating the cross-layer optimization factor as specified in Eq.(18).

$$E_i(t) = \alpha . S_i'(t) + \beta . \left(1 - S_i'(t)\right) + \gamma . \sum_j \bar{L}_{ij}(t) + \rho + \delta . \bar{E}_i(t) + \omega . CLOL(t) \tag{18}$$

Where ω represents the weight assigned to the cross-layer optimization factor in the overall expiration time calculation.

Specific parameters influencing route stability, such as signal strength, interference levels, and available bandwidth, may be considered to enable effective cross-layer interactions. These parameters contribute to cross-layer optimization and facilitate a more comprehensive and context-aware approach to route lifetime management. To ensure smooth integration of cross-layer optimization, a smoothing factor (η) can be introduced to account for the historical cross-layer optimization factor over time, expressed as Eq.(19).

$$smoothed\ CLOL(t):\ CLOL(t) = \eta . CLOL(t-1) + (1-\eta) . CLOL(t) \tag{19}$$

Where $C\bar{L}OL(t)$ represents the smoothed or averaged cross-layer optimization factor, and $C\bar{L}OL(t-1)$ is the previous smoothed value. The parameter η controls the influence of historical cross-layer optimization on the current smoothed value.

The adjusted expiration time with CLOL, considering the smoothed cross-layer optimization factor is given by Eq.(20).

$$E_i(t) = \alpha . S_i'(t) + \beta . \left(1 - S_i'(t)\right) + \gamma . \sum_j \bar{L}_{ij}(t) + \rho + \delta . \bar{E}_i(t) + \omega . CLOL(t) \tag{20}$$

3.1.8. Load Balancing for Lifetime Extension

Load Balancing for Lifetime Extension (LBLE) focuses on optimizing route maintenance by considering the distribution of traffic across available routes. Integrating LBLE into RLE-AODV enhances the protocol's ability to adapt to varying traffic loads and network conditions. Let's denote the load balancing factor at time t as $LBLE(t)$. The load balancing factor represents the degree of load balancing in the network, considering factors such as the current traffic distribution and congestion levels. Building upon the existing route expiration time calculation, the LBLE mechanism adjusts the adaptive expiration time $\left(E_i(t)\right)$ By incorporating the load balancing factor, Eq.(21) applies.

$$E_i(t) = \alpha . S_i'(t) + \beta . \left(1 - S_i'(t)\right) + \gamma . \sum_j \bar{L}_{ij}(t) + \rho + \delta . \bar{E}_i(t) + \omega . C\bar{L}OL(t) + \theta . LBLE(t) \tag{21}$$

Where θ represents the weight assigned to the load balancing factor in the overall expiration time calculation.

To calculate the load balancing factor, one approach is to consider the ratio of the traffic on the current route $(L_{current})$ to the average traffic load across all routes $\left(\bar{L}_{average}\right)$.

$$LBLE(t) = \frac{L_{current}(t)}{\bar{L}_{average}(t)} \tag{22}$$

Eq.(22) measures how evenly the traffic is distributed across routes. If the load balancing factor is close to 1, it indicates a balanced traffic distribution, while values significantly deviating from 1 suggest a load imbalance.

RESEARCH ARTICLE

To calculate the average traffic load $\bar{L}_{average}$ over time, a smoothing factor (λ) can be introduced using Eq.(23).

$$\bar{L}_{average}(t) = \lambda.\bar{L}_{average}(t-1) + (1-\lambda).L_{current}(t) \quad (23)$$

Where $\bar{L}_{average}(t)$ represents the smoothed or averaged load balancing factor and $\bar{L}_{average}(t-1)$ is the previous smoothed value. The parameter λ controls the influence of historical load balancing factors on the current smoothed value.

The adjusted expiration time with LBLE, considering the smoothed load balancing factor, is then given by Eq.(24).

$$E_i(t) = \alpha.S_i'(t) + \beta.\left(1 - S_i'(t)\right) + \gamma.\sum_j \bar{L}_{ij}(t) + \rho + \delta.\bar{E}_i(t) + \omega.C\bar{L}OL(t) + \theta.L\bar{B}LE(t) \quad (24)$$

3.1.9. Dynamic Threshold Adjustment

Dynamic Threshold Adjustment (DTA) enhances the protocol's adaptability by dynamically adjusting thresholds based on the observed stability and reliability of the network, ensuring optimal route maintenance and lifetime extension. Let's denote the dynamic threshold adjustment factor at time t as $DTA(t)$. The dynamic threshold adjustment factor represents the degree of adjustment required based on the current network conditions. Building upon the existing route expiration time calculation, the DTA mechanism adjusts the adaptive expiration time $\left(E_i(t)\right)$ by incorporating Eq.(25).

$$E_i(t) = \alpha.S_i'(t) + \beta.\left(1 - S_i'(t)\right) + \gamma.\sum_j \bar{L}_{ij}(t) + \rho + \delta.\bar{E}_i(t) + \omega.C\bar{L}OL(t) + \theta.L\bar{B}LE(t) + \phi.DTA(t) \quad (25)$$

Where ϕ represents the weight assigned to the dynamic threshold adjustment factor in the overall expiration time calculation.

The dynamic threshold adjustment factor ($DTA(t)$) can be calculated based on the observed stability and reliability metrics of the network. For instance, it may be determined by analyzing the packet delivery ratio (PDR), link quality, or other relevant metrics.

A higher $DTA(t)$ value signifies a need for more conservative threshold settings, while a lower value suggests that thresholds can be adjusted more aggressively. To adapt thresholds based on historical observations, a smoothing factor (μ) is introduced using Eq.(26).

$$D\bar{T}A(t) = \mu.D\bar{T}A(t-1) + (1-\mu).DTA(t) \quad (26)$$

Where $D\bar{T}A(t)$ represents the smoothed or averaged dynamic threshold adjustment factor, and $D\bar{T}A(t-1)$ is the previous smoothed value. The parameter μ controls the influence of historical adjustments on the current smoothed value.

3.2. Fast Furious Cheetah Optimization

Fast Furious Cheetah Optimization (FFCO) is an innovative algorithm inspired by the agile behavior of cheetahs. It enhances decision-making processes in various applications, including optimization problems. Mimicking cheetahs' swift and adaptive nature, this algorithm dynamically adjusts parameters to achieve optimal solutions efficiently.

Whether applied in routing protocols or other optimization domains, FFCO contributes to faster and more adaptive decision-making. It is a promising approach in fields that demand swift and efficient solutions to complex problems.

3.2.1. Initialization

The initial step in the optimization process involves the creation of a population of candidate solutions denoted as $X^{(0)}$. Let N represent the population size, D the dimensionality of the solution space, and x_{ij} denote the j-th parameter of the i-th solution in the population. The initialization process can be expressed as Eq.(27).

$$X^{(0)} = \left\{x_{ij}^{(0)}\right\},\ \ i = 1,2,\dots,N;\ \ j = 1,2,\dots..,D \quad (27)$$

Where $X^{(0)}$ represents the initial value of the j-th parameter for the i-th solution in the population.

The optimization process is guided by an objective function $f(X)$ that evaluates the performance of each candidate solution. The objective function is the mapping from the solution space to a scalar fitness value as specified in Eq.(28) and Eq.(29).

$$f(X) = \{f(x_i)\},\ \ i = 1,2,\dots..,N \quad (28)$$

$$f(x_i) = \frac{g(x_i) - min\left(g(X)\right)}{max\left(g(X)\right) - min\left(g(X)\right)} \quad (29)$$

Where $f(x_i)$ represents the fitness value of the i-th solution in the population.

Compute the gradient of the objective function concerning the parameters of each solution. Let $\nabla f\left(x_{ij}\right)$ denote the gradient vector for the j-th parameter of the i-th solution in Eq.(30)

$$\nabla f(X) = \{\nabla f(x_i)\},\ \ i = 1,2,\dots..,N;\ \ j = 1,2,\dots..,D \quad (30)$$

The gradient provides information about the direction and magnitude of the steepest ascent in the objective function landscape.

RESEARCH ARTICLE

3.2.2. Fast Fourier Transform Analysis for Frequency Components

Perform a Fast Fourier Transform (FFT) analysis on the objective function values $f(x_i)$ to decompose the function into its frequency components. Let $F(k)$ in Eq.(31) represent the Fourier Transform of $f(x_i)$ at frequency k.

$$F(k) = FFT\{f(x_i)\}, \quad k = 1,2,\dots,K \tag{31}$$

Where K is the number of frequency components obtained from the FFT analysis.

Identify the dominant frequencies by examining the magnitude spectrum of the Fourier Transform. Let k_{dom} denote the index of the dominant frequency component in Eq.(32).

$$k_{dom} = \arg max_k |F(k)|; \; k_{dom} \in [1, K] \tag{32}$$

The dominant frequency in Eq.(32) indicates the most prominent periodicity in the objective function landscape.

Modify the candidate solutions based on the identified dominant frequencies in the frequency domain. Introduce a frequency-dependent modification factor M_{ij} for each parameter by applying Eq.(33)

$$x_{ij}^{(mod)} = x_{ij} + M_{ij}.sin(2\pi k_{dom}), \quad i = 1,2,\dots,N; \; j = 1,2,\dots,D \tag{33}$$

Where $x_{ij}^{(mod)}$ represents the modified value of the j-th parameter for the i-th solution in the population and M_{ij} is the modification factor.

Perform an inverse FFT to transform the modified solutions to the original parameter space, expressed as Eq.(34).

$$x_{ij}^{(mod)} = IFFT\left\{X_{ij}^{(mod)}\right\}, \quad i = 1,2,\dots,N; \; j = 1,2,\dots,D \tag{34}$$

Where $X_{ij}^{(mod)}$ represents the modified solution in the frequency domain.

3.2.3. Evaluation of Frequency-Adapted Solutions

Evaluate the performance of the frequency-adapted solutions using the objective function. The fitness values $f\left(x_i^{(mod)}\right)$ are obtained for each modified solution using Eq.(35).

$$f\left(X^{(mod)}\right) = \left\{f\left(x_i^{(mod)}\right)\right\}, \quad i = 1,2,\dots,N; \tag{35}$$

Where $f\left(x_i^{(mod)}\right)$ represents the fitness value of the i-th solution after the frequency-based modification.

Compute the gradients of the objective function concerning the parameters of the frequency-adapted solutions. Let $\nabla f\left(x_{ij}^{(mod)}\right)$ denote the gradient vector for the j-th parameter of the i-th modified solution, and it is calculated using Eq.(36).

$$\nabla f\left(X^{(mod)}\right) = \left\{\nabla f\left(x_{ij}^{(mod)}\right)\right\}, \quad i = 1,2,\dots,N; \tag{36}$$

The gradient information for the modified solutions guides combined evaluation and frequency adaptions.

Combine the objective function values and gradient information for the frequency-adapted solutions, which is mathematically expressed as Eq.(37), and it ensures that the optimization algorithm considers both the fitness values and the modified frequency-dependent characteristics of the solutions.

$$f\left(X^{(mod)}\right) = \left\{f\left(x_i^{(mod)}\right)\right\}, \nabla f\left(X^{(mod)}\right) = \left\{\nabla f\left(x_{ij}^{(mod)}\right)\right\} \tag{37}$$

Select the top-performing frequency-adapted solutions based on their evaluation scores. The selection process involves identifying solutions with improved fitness values and favorable gradients where Eq.(38) expresses the same.

$$X^{(selected)} = \left\{x_{ij}^{(mod)}\right\}, \; i = 1,2,\dots,N; \; j = 1,2,\dots D \tag{38}$$

The selected solutions form the basis for generating offspring in subsequent steps.

3.2.4. Frequency-Adapted Recombination (Crossover) and Mutation

The features are combined from the selected frequency-adapted solutions using recombination (crossover). The crossover process incorporates frequency-based adaptations using Eq.(39) to guide the generation of offspring solutions.

$$x_{ij}^{(offspring)} = \alpha.x_{ij}^{(selected)} + (1-\alpha).x_{ij}^{(random)} \tag{39}$$

Where $x_{ij}^{(offspring)}$ represents the j-th parameter of the offspring, α is a recombination parameter, and $x_{ij}^{(random)}$ is a randomly selected parameter from the population.

By introducing a small random change to the parameters of some offspring solutions, considering the frequency-based adaptations. The mutation process aims to promote exploration in both the parameter space and the modified frequency domain which is expressed in Eq.(40).

$$x_{ij}^{(mutated)} = x_{ij}^{(offspring)} + \beta.rand() \tag{40}$$

RESEARCH ARTICLE

Where $x_{ij}^{(mutated)}$ represents the j-th parameter of the mutated offspring, β is a mutation parameter, and rand() is a random value from a specified distribution.

By combining the offspring solutions generated through recombination and mutation, $x_{ij}^{(offspring)}$ will be calculated using Eq.(41). The offspring solutions represent a combination of recombined and mutated solutions, with consideration given to the frequency-adapted characteristics.

$$X^{(offspring)} = \left\{x_{ij}^{(offspring)}, x_{ij}^{(mutated)}\right\}, \quad i = 1,2,\ldots.N; \quad j = 1,2,\ldots,D \tag{41}$$

Evaluate the performance of the generated offspring solutions using the objective function. Obtain the fitness values and gradients for the offspring using Eq.(42). The evaluation phase ensures that the offspring solutions' performance is assessed in both recombination and mutation scenarios.

$$\begin{aligned} &f\left(X^{(offspring)}\right) \\ &= \left\{f\left(x_i^{(offspring)}\right), f\left(x_i^{(mutated)}\right)\right\}, \\ &\nabla f\left(X^{(offspring)}\right) \\ &= \left\{\nabla f\left(x_{ij}^{(offspring)}\right), \nabla f\left(x_{ij}^{(mutated)}\right)\right\} \end{aligned} \tag{42}$$

3.2.5. Replacement and Convergence Check

Replace the old population with a combination of the original solutions and the newly generated offspring based on their fitness values. Implement a selection mechanism that considers both parent and offspring solutions using Eq.(43) to maintain a diverse population while favouring solutions with improved fitness.

$$X^{(new)} = \left\{x_{ij}^{(selected)}, x_{ij}^{(offspring)}, x_{ij}^{(mutated)}\right\}, \quad i = 1,2,\ldots N; \quad j = 1,2,..,D \tag{43}$$

Check for convergence by assessing whether the optimization process has met predefined criteria. Convergence criteria may include a satisfactory level of fitness, stability in the population, or a specified number of iterations. The decision to converge is influenced by factors such as the fitness values, gradients, and modifications introduced during the optimization process, which is Eq.(44).

$$\begin{aligned} &Converged = \\ &CheckConvergence(fitness\_values, \\ &gradients, modifications, \ldots) \end{aligned} \tag{44}$$

Where $Converged$ is a binary variable indicating whether the optimization process has converged, and $CheckConvergence$ is a function that evaluates convergence criteria.

Eq.(45) is applied to combine the replacement and convergence check steps. Replacing the old population with a diverse set of solutions, combined with a convergence check, ensures the optimization process evolves towards satisfactory solutions.

$$X^{(new)} = \left\{x_{ij}^{(selected)}, x_{ij}^{(offspring)}, x_{ij}^{(mutated)}\right\} \tag{45}$$

If convergence has not been achieved, repeat the optimization steps from 3.2.2 to 3.2.5 until the convergence criteria are met or a predefined number of iterations is reached. Iteration control ensures the optimization continues until convergence or a specified limit is reached. The integration of replacement strategies and a convergence check in this step contributes to the dynamic evolution of the population, with periodic assessments of the optimization progress.

3.2.6. Iterative Frequency Adaptation

Continuing the iterative process, another Fast Fourier Transform (FFT) analysis on the objective function values of the current population will be performed. Eq.(46) aims to capture any evolving frequency patterns in the accurate function landscape as the optimization progresses.

$$F^{(t)}(k) = FFT\left\{f\left(x_i^{(t)}\right)\right\}, \quad k = 1,2,\ldots K^{(t)} \tag{46}$$

Where t denotes the current iteration, and $K^{(t)}$ is the number of frequency components at iteration t. The dominant frequency is updated by identifying the index of the maximum magnitude in the frequency spectrum obtained from the FFT analysis at the current iteration. Eq.(47) shifts or changes the frequency characteristics of the objective function landscape.

$$k_{dom}^{(t)} = \arg max_k \left|F^{(t)}(k)\right|, \qquad k_{dom}^{(t)} \in \left[1, K^{(t)}\right] \tag{47}$$

The candidate solutions are modified in the frequency domain based on the updated dominant frequency. Utilize the frequency-dependent modification factor $M_{ij}^{(t)}$ for each parameter as specified in Eq.(48).

$$x_{ij}^{(mod,t)} = x_{ij}^{(t)} + M_{ij}^{(t)}.sin\left(2\pi k_{dom}^{(t)}\right) \tag{48}$$

Where $x_{ij}^{(mod,t)}$ represents the modified value of the j-th parameter for the i-th solution in the population at iteration t. An inverse FFT is performed to transform the modified solutions back to the original parameter space for the current iteration, and it is expressed as Eq.(49).

$$x_{ij}^{(mod,t)} = IFFT\left\{X_{ij}^{(mod,t)}\right\}, \quad i = 1,2,\ldots,N; \quad j = 1,2,\ldots,D \tag{49}$$

3.2.7. Enhanced Recombination and Mutation with Frequency Adaptation

This step enhances the recombination (crossover) process by incorporating the iterative frequency-adapted information.

RESEARCH ARTICLE

The features are combined from the selected frequency-adapted solutions using recombination, emphasizing the updated frequency characteristics. Mathematically, it is expressed as Eq.(50)

$$x_{ij}^{(offspring,t)} = \alpha . x_{ij}^{(selected,t)} + (1-\alpha) . x_{ij}^{(random,t)} \quad (50)$$

Where $x_{ij}^{(offspring,t)}$ represents the j-th parameter of the offspring at iteration t, α is a recombination parameter, and $x_{ij}^{(random,t)}$ is a randomly selected parameter from the population at iteration t.

Introducing small random changes to the parameters of some offspring solutions considers the evolving frequency-based adaptations. The mutation process promotes exploration in both the parameter space and the modified frequency domain as Eq.(51)

$$x_{ij}^{(mutated,t)} = x_{ij}^{(offspring,t)} + \beta . rad() \quad (51)$$

Where $x_{ij}^{(mutated,t)}$ represents the j-th parameter of the mutated offspring at iteration t, β is a mutation parameter, and $rand()$ is a random value from a specified distribution.

The offspring solutions are combined by generating through enhanced recombination and mutation with frequency adaptation, as expressed in Eq.(52). The offspring solutions at iteration t now reflect an improved combination of recombined and mutated solutions, guided by the evolving frequency characteristics.

$$X^{(offspring,t)} = \left\{ x_{ij}^{(offspring,t)}, x_{ij}^{(mutated,t)} \right\}, \quad i = 1,2,\dots,N; \;\; j = 1,2,\dots,D \quad (52)$$

Evaluate the generated offspring solutions' performance using the current iteration's objective function. Obtain the fitness values and gradients for the offspring using Eq.(53). The iterative evaluation phase ensures that the performance of the progeny is assessed considering both recombination and mutation scenarios.

$$\begin{aligned} & f\left(X^{(offspring,t)}\right) \\ & = \left\{ f\left(x_i^{(offspring,t)}\right), f\left(x_i^{(mutated,t)}\right) \right\} \\ & \nabla f\left(X^{(offspring,t)}\right) \\ & = \left\{ \nabla f\left(x_{ij}^{(offspring,t)}\right), \nabla f\left(x_{ij}^{(mutated,t)}\right) \right\} \end{aligned} \quad (53)$$

3.2.8. Iterative Replacement and Adaptive Convergence Check

The replacement strategy is refined by adapting the process based on the evolving frequency-adapted solutions. Consider a combination of parent solutions, frequency-adapted offspring, and potentially mutated offspring in the replacement phase.

$$X^{(new,t)} = \left\{ x_{ij}^{(selected,t)}, x_{ij}^{(offspring,t)}, x_{ij}^{(mutated,t)} \right\}, \quad i = 1,2,\dots,N; \;\; j = 1,2,\dots,D \quad (54)$$

Eq.(54) aims to maintain diversity while favoring solutions with improved fitness and incorporating information from the frequency-adapted offspring.

The convergence check is enhanced by considering the frequency-adapted solutions and potentially updating the convergence criteria dynamically. Eq.(55) evaluates whether the optimization process has met adaptive convergence criteria considering the evolving frequency characteristics, fitness values, and gradient information.

$$CheckAdaptiveConvergence \left(fitness\_values^{(t)}, gradients^{(t)}, mod\right) \quad (55)$$

Where $Converged^{(t)}$ is a binary variable indicating whether the optimization process has converged at iteration t, and $CheckAdaptiveConvergence$ is a function that dynamically evaluates adaptive convergence criteria.

If convergence has not been achieved at iteration t, repeat the optimization steps from section 3.2.2 to 3.2.8 until the adaptive convergence criteria are met or a predefined number of iterations is reached. The iterative nature of the optimization process, coupled with adaptive replacement and convergence checks, ensures the algorithm dynamically responds to changes in the optimization landscape. By integrating adaptive strategies, the algorithm can navigate challenging scenarios where the characteristics of the objective function may vary across different phases of the optimization process.

3.2.9. Fine-Tuning and Solution Refinement

Introduce a fine-tuning mechanism to refine solutions in the vicinity of the current population. Incorporate local search techniques to exploit promising solutions in the neighborhood. Fine-tuning involves minor adjustments to the parameters of selected solutions, focusing on controlling local improvements expressed as Eq. (56).

$$x_{ij}^{(fine-tuned,t)} = x_{ij}^{(t)} + \gamma . local\_search\left(x_{ij}^{(t)}\right) \quad (56)$$

Where $x_{ij}^{(fine-tuned,t)}$ represents the fine-tuned version of the j-th parameter for the i-th solution at iteration t, and γ is a fine-tuning parameter.

Evaluate the performance of the fine-tuned solutions using the objective function. Adjust the evaluation strategy using Eq.(57) to dynamically consider the fine-tuned solutions in assessing fitness values and gradients. The adaptive

RESEARCH ARTICLE

evaluation ensures that the fine-tuned solutions contribute to the optimization process based on their adjusted fitness and gradient information.

$$f\left(X^{(fine-tuned,t)}\right) = \left\{f\left(x_i^{(fine-tined,t)}\right)\right\}$$
$$\nabla f\left(X^{(fine-tuned,t)}\right) = \left\{\nabla f\left(x_{ij}^{(fine-tined,t)}\right)\right\} \quad (57)$$

Combine the fine-tuned solutions with the current population and frequency-adapted offspring expressed as Eq.(58).

$$X^{(final,t)} = \left\{x_{ij}^{(selected,t)}, x_{ij}^{(offspring,t)}, x_{ij}^{(mutated,t)}, x_{ij}^{(fine-tuned,t)}\right\} \quad (58)$$

The combined set of solutions represents the final population at iteration t, including the initially selected solutions, frequency-adapted offspring, potentially mutated solutions, and fine-tuned solutions.

The convergence check is adapted to include the fine-tuned solutions in the assessment of convergence criteria. The iterative convergence check, now accounting for fine-tuned solutions, ensures that the optimization process responds to local improvements and potential refinement opportunities.

Introduce an adaptive step size control mechanism to dynamically adjust the exploration-exploitation balance during optimization. The step size control influences the magnitude of parameter updates in the population and fine-tuned solutions. Mathematically, it is expressed as Eq.(59).

$$x_{ij}^{(adaptive,t)} = x_{ij}^{(final,t)} + \delta.step\_size\left(x_{ij}^{(final,t)}\right) \quad (59)$$

Where $x_{ij}^{(adaptive,t)}$ represents the parameter with an adaptive step size at iteration t, and δ is the adaptive step size factor.

Promote global exploration by applying the adaptive step size control to all solutions in the population, fine-tuned solutions, and any additional exploration strategies, which uses Eq.(60). The global exploration set includes solutions with adaptive step sizes and any other exploration strategies designed to enhance the search process.

$$X^{(exploration,t)} = \left\{x_{ij}^{(adaptive,t}, x_{ij}^{(additional\_exploration,t}\right\} \quad (60)$$

(a). Iterative Evaluation of Globally Explored Solutions

Evaluate the performance of the globally explored solutions using the objective function. Dynamically consider the fitness values and gradients of the solutions with adaptive step sizes and any additional exploration strategies.

$$f\left(X^{(exploration,t)}\right) = \left\{f\left(x_i^{(adaptive,t}\right), f\left(x_i^{(additional\_exploration,t}\right)\right\}$$
$$\nabla f\left(X^{(exploration,t)}\right) = \left\{\nabla f\left(x_{ij}^{(adaptive,t}\right), \nabla f\left(x_{ij}^{(additional\_exploration,t}\right)\right\} \quad (61)$$

Eq.(61) ensures that the performance of globally explored solutions is assessed, considering both adaptive step sizes and any additional exploration strategies.

(b). Combined Globally Explored Equations

Combine the globally explored solutions using Eq.(62) with the current population, frequency-adapted offspring, potentially mutated solutions, and fine-tuned solutions. It represents the global exploration set at iteration t, incorporating solutions with adaptive step sizes and any additional exploration strategies.

$$X^{(gloabl,t)} = \left\{x_{ij}^{(selected,t)}, x_{ij}^{(offspring,t)}, x_{ij}^{(mutated,t)}, x_{ij}^{(fine-tuned,t)}, x_{ij}^{(exploration,t)}\right\} \quad (62)$$

(c). Iterative Convergence Check with Global Exploration

Adapt the convergence check using Eq.(63) to include the globally explored solutions in the assessment of convergence criteria:

$$Converged^{(t)} = CheckGlobalConvergence(fitness\_values^{(t)}, gradients^{(t)}, modific) \quad (63)$$

Where $Converged^{(t)}$ is a binary variable indicating whether the optimization process has converged at iteration t, and $CheckGlobalConvergence$ is a function that dynamically evaluates convergence criteria considering globally explored solutions.

3.3. Elliptic Curve Cryptography (ECC)

Integrating Elliptic Curve Cryptography (ECC) within the Fast Furious Cheetah Optimization (FFCO) framework results in Fast Furious Cheetah Optimization-Based Secured Routing (FFCOSR). This integration enhances the security of routing protocols within the FFCOSR paradigm, bolstering the confidentiality, integrity, and authentication of transmitted data. Leveraging ECC's efficiency and effectiveness, FFCOSR reinforces routing optimization with robust security measures, rendering it well-suited for dynamic and resource-constrained environments like vehicular ad hoc networks (VANETs). The amalgamation of ECC within FFCO signifies

RESEARCH ARTICLE

a pivotal advancement, offering a holistic solution for secure and efficient routing in VANETs.

Elliptic Curve Cryptography (ECC) is a robust cryptographic framework widely employed for secure data transmission and digital signatures. Rooted in elliptic curves' mathematical properties, ECC provides stronger security with shorter critical lengths than traditional methods. ECC's strength lies in its ability to resist various attacks, including those anticipated with the advent of quantum computing. The core principle involves leveraging elliptic curve algebraic structures, where points on the curve define keys for encryption, decryption, and digital signatures. ECC's computational efficiency makes it suitable for resource-constrained environments, such as mobile devices and embedded systems, while maintaining high levels of security. The security of ECC relies on the difficulty of the elliptic curve discrete logarithm problem, contributing to its resilience against potential threats. Widely adopted in contemporary cryptographic protocols like TLS and PGP, ECC is instrumental in ensuring the confidentiality and integrity of sensitive information. Its efficiency and formidable security features position ECC as a cornerstone in modern cryptographic applications, navigating the evolving landscape of information security challenges. The functionality of ECC can be classified into 7 phases, which are as follows.

3.3.1. Key Generation:

Key generation is the foundational step in ECC and is crucial for securing communications and data integrity. This process involves the computation of a public key (Q) and a corresponding private key (d) using elliptic curve mathematics. The elliptic curve equation Eq.(64) defines the curve's points over a finite field F_p, where a and b are constants, and p is a prime number.

$$y^2 = x^3 + ax + b \tag{64}$$

The public key (Q) is derived by performing scalar multiplication of a chosen base point (G) on the curve by the private key (d), which is represented mathematically in Eq.(65).

$$\mathrm{Q} = d \times G \tag{65}$$

This scalar multiplication is efficiently achieved through repeated point addition and doubling operations. The resulting Q becomes the user's public key, intended for distribution.

The private key (d) is a randomly selected scalar within the range $[1, n-1]$, where n is the order of the base point (G).

$$\mathrm{n} \times \mathrm{G} = \mathrm{O} \tag{66}$$

In Eq.(66) as represented n is the smallest positive integer, where O is the point at infinity. The private key is kept confidential and is fundamental for cryptographic operations.

The essential generation process in ECC, as expressed by equations, involves the selection of a private key (d) from a finite range and the computation of the corresponding public key (Q) by scalar multiplication of the base point (G) on the elliptic curve.

3.3.2. Key Exchange:

The Key Exchange phase involves the computation of a shared secret S between two parties using their respective private and public keys. The sender's private key $dsender$ and the receiver's public key $Qreceiver$ are utilized in this process. The shared secret S is calculated through scalar multiplication mathematically represented in Eq.(67).

$$\mathrm{S} = d_{sender} \times Q_{receiver} \tag{67}$$

Eq.(67) signifies the combination of the sender's secret scalar with the receiver's public key point on the elliptic curve. The resulting S is a shared secret that remains confidential between the communicating entities.

To achieve secure key exchange, both parties must ensure the validity of each other's keys and employ the same elliptic curve parameters. The security of this process hinges on the difficulty of the elliptic curve discrete logarithm problem, wherein deriving the private key from the public key is computationally infeasible.

This process is robust due to the inherent complexity of elliptic curve arithmetic. The scalar multiplication involves a series of point additions and doublings, making it resistant to traditional cryptographic attacks. The security of the shared secret S lies in the strength of the private key $dsender$, ensuring confidentiality in the communication channel. The Key Exchange step establishes a shared secret S between two entities, paving the way for secure communication and cryptographic operations.

3.3.3. Digital Signatures

Digital Signatures play a pivotal role in ensuring data integrity and authentication. This process involves the generation and verification of a signature using the private key d and the corresponding public key Q of the signer. For a message m, the signer computes a pair of integers r, s as depicted in Eq.(68) to Eq.(70).

$$(x_1, y_1) = k \times G \tag{68}$$

$$\mathrm{r} = x_1 \bmod n \tag{69}$$

$$\mathrm{s} = k^{-1} \cdot (H(m) + d \cdot r) \bmod n \tag{70}$$

Where G represents the base point on the elliptic curve, k is a randomly chosen integer $H(m)$ is the hash of the message m, and n is the order of the base point. The resulting (r, s) pair constitutes the digital signature.

RESEARCH ARTICLE

The verification process involves the receiver, who possesses the sender's public key Q, the received message m, and the received signature r, s. The receiver checks the validity of the signature through the following equations from Eq.(71) to Eq.(75).

$$\mathrm{w} = s^{-1} \, mod \, n \tag{71}$$

$$u_1 = H(m) \cdot \mathrm{w} \, mod \, n \tag{72}$$

$$u_2 = \mathrm{r} \cdot \mathrm{w} \, mod \, n \tag{73}$$

$$(x_1, y_1) = u_1 \times G + u_2 \times Q \tag{74}$$

$$v = x_1 mod \, n \tag{75}$$

The signature is considered valid if v is equal to r. The use of a nonce k prevents deterministic signatures and enhances security. The digital signatures step in ECC involves the generation and verification of signatures using private and public keys, ensuring the authenticity and integrity of transmitted data.

3.3.4. Encryption

The encryption process involves transforming plaintext into ciphertext using the recipient's public key Q and a randomly chosen integer k. For a given plaintext message m, Eq.(76) and Eq.(77) represent the encryption process mathematically.

$$(x_{k,} y_k) = k \times G \tag{76}$$

$$(x_c, y_c) = (x_m + x_k, y_m + y_k) \tag{77}$$

Where G is the base point on the elliptic curve, and (x_m, y_m) represents the coordinates of the plaintext message. The ciphertext (x_c, y_c) is obtained by adding the result of the scalar multiplication of the base point G with k to the coordinates of the plaintext message.

The decryption process, performed by the recipient possessing the private key d, involves reversing the encryption to retrieve the original plaintext coordinates.

$$(x_m, y_m) = (x_c, y_c) - \mathrm{d} \times (x_k, y_k) \tag{78}$$

Eq.(78) employs scalar multiplication and subtraction on the elliptic curve to obtain the original coordinates of the plaintext message. The security of the Encryption step relies on the computational infeasibility of reversing the scalar multiplication process without knowledge of the private key d. The elliptic curve properties ensure that only the recipient, possessing the private key, can successfully decrypt the ciphertext to retrieve the original plaintext. The Encryption step in ECC leverages mathematical operations on elliptic curves to achieve data confidentiality.

3.3.5. Key Storage and Management:

Key Storage and Management involve the secure handling of private keys d to maintain the confidentiality and integrity of cryptographic operations. The private key must be securely stored and managed to prevent unauthorized access and potential compromise. Key storage practices commonly include secure mechanisms such as Hardware Security Modules (HSMs) or key vaults. The process entails properly storing the private key d in a secure location. While the mathematical representation is abstract, the essence lies in safeguarding the private key from unauthorized access or disclosure, represented mathematically using Eq.(79).

$$\text{Secure Storage} : Store(d) \tag{79}$$

The stored private key is crucial for cryptographic operations like signature generation and decryption. Its protection is paramount to the overall security of the ECC system. The management of private keys involves considerations such as key rotation and periodic updates. The mathematical representation of key rotation might include generating a new private key (d_{new}) and securely updating the stored key.

$$\text{Key Rotation} : Store(d_{new}) \tag{80}$$

Eq.(80) represents the process of securely storing a new private key, ensuring the continuous security of cryptographic operations.

In ECC, the management of public keys is equally vital. The mathematical aspect involves securely distributing and verifying the authenticity of public keys. An equation capturing this concept is Eq.(81) as public key distribution.

$$Q_{receiver} = \text{Retrieve}(d_{receiver}) \times G \tag{81}$$

Eq.(81) symbolizes the distribution of a public key ($Q_{receiver}$) by scalar multiplication of the recipient's private key ($d_{receiver}$) with the base point (G). The Key Storage and Management step in ECC involves secure storage practices for private keys and effective management of private and public keys.

3.3.6. Post-Quantum Considerations

Post-quantum considerations address the evolving landscape of quantum computing and its potential threats to existing cryptographic systems. The emergence of powerful quantum computers has spurred the exploration of quantum-resistant algorithms to ensure data security in the post-quantum era. One such consideration involves transitioning from traditional elliptic curve-based cryptographic schemes to post-quantum algorithms. Choosing a post-quantum algorithm is crucial for maintaining security against quantum attacks. The post-quantum key exchange is mathematically represented mathematically with Eq.(82).

$$S_{PQ} = SK_{PQ} \times PK_{OtherParty} \tag{82}$$

Where SK_{PQ} the post-quantum private is key, and $PK_{OtherParty}$ is the public key of the other party.

RESEARCH ARTICLE

As quantum computers pose a threat to traditional public-key cryptography, the adoption of lattice-based or hash-based post-quantum cryptographic algorithms becomes imperative. A representative equation for a post-quantum digital signature might involve the use of a post-quantum private key SK_{PQ} .

$$\text{Post} - \text{Quantum Signature}: Sig_{PQ} = Sign_{PQ}(SK_{PQ}, Message) \quad (83)$$

Where, in Eq.(83) $Sign_{PQ}$ denotes the signing algorithm for the post-quantum scheme.

In anticipation of quantum adversaries, researchers explore mathematical structures that remain secure even in the face of quantum algorithms.

3.3.7. Randomness Considerations

Randomness Considerations play a crucial role in enhancing the security of cryptographic operations. Randomness is integral for generating unpredictable values, such as nonces or ephemeral keys, to thwart deterministic attacks.

Incorporating randomness is vital in various aspects of ECC, including key generation and signature schemes. Generating random nonces (k) is a common practice in ECC to prevent predictability in signatures. The mathematical representation of the nonce generation process involves obtaining a random value within a specified range.

$$\text{k} \in [1, n-1] \quad (84)$$

Where in Eq.(84) n denotes the order of the base point on the elliptic curve. The unpredictability of k contributes to the security of the digital signature.

Randomness is often employed in secure multiparty computation or distributed key generation. The mathematical representation of a secure joint key generation process might involve the combination of individual contributions with randomness, as shown in Eq.(85).

$$\text{Shared Key} = Combine(Key_1 + Random_1, Key_2 + Random_2, \dots) \quad (85)$$

In cryptographic protocols, especially those involving multiple parties, introducing randomness adds an extra layer of security. Randomness is crucial in the context of key diversification. The mathematical representation of diversification through randomness might involve combining a master key (MK) with a random value (R) to derive a diversified key (DK).

$$\text{Diversified Key} = Derive(MK, R) \quad (86)$$

Eq.(86) symbolizes the generation of a diversified key (DK) using a master key (MK) and a random value (R).

3.3.8. Error Handling and Robust Implementations

Error detection is a fundamental aspect of ECC implementations, often involving the validation of inputs. The mathematical expression for input validation may include conditions and constraints, ensuring that the input adheres to specified criteria. Detection of error signaling mechanisms is crucial for communicating the nature of the error. Method representations for signalling may involve error codes or messages. Error recovery strategies are implemented to mitigate the impact of errors. The depiction of error recovery might involve retrying an operation or switching to an alternative method. Logging and auditing error information contribute to the analysis and improvement of cryptographic implementations. A method of representing logging error information may involve recording relevant details. Exception-handling mechanisms in programming languages are critical for controlled error management.

$Step\ 1$: $procedure\ ECCAlgorithm(message, privateKey, curveParameters)$

$Step\ 2$: $publicKey = GeneratePublicKey(privateKey, curveParameters)$

$Step\ 3$: $ciphertext = Encrypt(message, publicKey, curveParameters)$

$Step\ 4$: $signature = Sign(message, privateKey, curveParameters)$

$Step\ 5$: $decryptedMessage = Decrypt(ciphertext, privateKey, curveParameters)$

$Step\ 6$: $isVerified = VerifySignature(message, signature, publicKey, curveParameters)$

$Step\ 7$: $return\ ciphertext, signature, decryptedMessage, isVerified$

$Step\ 8$: $end\ procedure$

Algorithm 3 Error Handling and Robust Implementations

Algorithm 3 depicts the steps of Error Handling and Robust Implementations. Error handling is an integral part of maintaining system integrity and security. While specific mathematical equations may not directly represent error-handling procedures, the implementation involves logical and conditional statements within the code. The systematic approach to detecting, signalling, and recovering from errors ensures that ECC implementations remain robust in the face of unforeseen challenges.

Step 1: Structure EllipticCurve:

RESEARCH ARTICLE

a) F_p: Finite field

b) E: Elliptic curve defined over F_p

c) G: Base point on E

d) n: Order of G

Step 2: Structure ECCKeyPair:

a) PrivateKey: Secret scalar (d)

b) PublicKey: Public key point (Q)

Step 3: Function ECCKeyGeneration() → ECCKeyPair:

a) Curve ← EllipticCurve(F_p, E, G, n)

b) PrivateKey ← RandomNumberInRange(1, n − 1)

c) PublicKey ← ScalarMultiplication(PrivateKey, G)

d) Return ECCKeyPair(PrivateKey, PublicKey)

Step 4: Function ECCEncryption(plaintext, PublicKey) → Ciphertext:

a) k ← RandomNumberInRange(1, n − 1)

b) K ← ScalarMultiplication(k, G)

c) c1 ← K.x

d) s ← ScalarMultiplication(k, PublicKey)

e) c2 ← plaintext XOR Hash(s)

f) Return Ciphertext(c1, c2)

Algorithm 4 ECC

Algorithm 4 outlines the significant steps involved in ECC, incorporating key generation, key exchange, digital signatures, encryption, key storage and management, post-quantum considerations, randomness considerations, and error handling.

3.4. Fusion of FFCO and RLE-AODV

Integrating the RLE-AODV and FFCO presents a synergistic approach to addressing the challenges in vehicular communication networks. RLE-AODV, a robust routing protocol, forms the foundation for establishing reliable communication paths within dynamic vehicular environments. By enhancing communication routes' adaptability, efficiency, and resilience, RLE-AODV responds effectively to variations in vehicle positions and network connectivity. Fast Furious Cheetah Optimization complements this routing framework with its unique frequency-based adaptation strategy. Through a Fast Fourier Transform (FFT) analysis, the optimization process identifies dominant frequencies in the objective function landscape, offering insights into the dynamic nature of the vehicular network. The subsequent modification of candidate solutions in the frequency domain, guided by dominant frequencies, introduces a novel dimension to the routing optimization process. This frequency-dependent modification, expressed through sine functions, aims to enhance the overall performance of the vehicular communication network.

The iterative nature of FFCO ensures continuous adaptation to evolving frequency patterns, aligning with the dynamic changes in the vehicular environment. This iterative frequency adaptation and enhanced recombination and mutation processes generate offspring solutions that exhibit improved fitness values and favorable gradients. The selection of top-performing frequency-adapted solutions forms the basis for subsequent iterations, driving the optimization process towards more satisfactory outcomes. Algorithm 5 establishes a comprehensive routing protocol that leverages the strengths of both methodologies. This integration offers a promising avenue for creating a robust, adaptive, and secure vehicular communication framework, enhancing communication paths' overall reliability and efficiency within dynamic vehicular environments.

Step 1: Dynamic Adaptive Networking

Step 2: Begin

Step 3: DAN_Setup InitializeDANSetup():

a) MaxNodes ← 100

b) NetworkTopology ← GenerateRandomTopology(MaxNodes)

c) InitialRoutingTable ← CreateInitialRoutingTable(NetworkTopology)

d) Return DAN_Setup { MaxNodes, NetworkTopology, InitialRoutingTable }

Step 4: FFO_Parameters InitializeFFOParameters():

a) MaxIterations ← 500

b) CheetahPopulation ← 30

c) ConvergenceCriteria ← 1e − 5

RESEARCH ARTICLE

d) Return FFO_Parameters { MaxIterations, CheetahPopulation, ConvergenceCriteria }

Step 5: Solution FastFuriousOptimization(FFO_Parameters parameters):

a) Population ← InitializeCheetahPopulation(parameters. CheetahPopulation)

b) For iteration in 1 to parameters. MaxIterations:

c) EvaluateFitness(Population, DAN_Setup. InitialRoutingTable)

d) UpdateCheetahPositions(Population)

e) If Converged(Population, parameters. ConvergenceCriteria):

a. Break

f) Return GetBestSolution(Population)

Step 6: DynamicRoutingTable UpdateDynamicRoutingTable
(Solution optimizedSolution):

a) UpdatedRoutingInfo ← ExtractRoutingInfo(optimizedSolution)

b) UpdatedRoutingTable ← DAN_Setup. InitialRoutingTable. Merge(UpdatedRoutingInfo)

c) Return UpdatedRoutingTable

Step 7: EncryptedData $FFO_{Encryption}$
(OriginalData data, Solution optimizedSolution):

a) HashedSolution ← SHA_256. Hash(optimizedSolution)

b) EncryptionKey ← DeriveSymmetricKey(HashedSolution)

c) EncryptedData ← AES. Encrypt(data, EncryptionKey)

d) Return EncryptedData

Step 8: EncryptedData DAN_FFO_FusionAlgorithm(OriginalData data):

a) DANSetup ← InitializeDANSetup()

b) FFOParameters ← InitializeFFOParameters()

c) For iteration in 1 to FFOParameters. MaxIterations:

d) FFOBestSolution ← FastFuriousOptimization(FFOParameters)

e) UpdatedRoutingTable ← UpdateDynamicRoutingTable(FFOBestSolution)

f) EncryptedData ← FFO_Encryption(data, FFOBestSolution) XOR UpdatedRoutingTable

g) Return EncryptedData

Step 9: END

Algorithm 5 Dynamic Adaptive Networking with FFCO

3.4.1. Advantages of FFCO

The holistic approach to vehicular network optimization, combining the strengths of RLE-AODV and Fast Furious Cheetah Optimization, results in extended route lifetimes, real-time adaptability, and optimized traffic flow with reduced communication disruptions.

- Extended Route Life: DynamicAdaptNet uses a boosted AODV (RLE-AODV) to create dependable communication paths, reducing the need for constant rediscovery of routes.
- Smooth Traffic: Fast Furious Cheetah Optimization enhances traffic flow by considering fitness values and

RESEARCH ARTICLE

characteristics adapted to frequencies in vehicular networks.

- Diversity in Solutions: DynamicAdaptNet keeps things diverse by balancing parent and offspring solutions and tackling scale and service quality challenges in vehicular networks.
- Adaptable Progress Check: DynamicAdaptNet stays on its toes by responding to changes in the objective function landscape, updating its criteria for a robust optimization process.
- Fine-tuning for Local Touch-ups: DynamicAdaptNet fine-tunes solutions in the nearby population, making local adjustments and boosting overall network efficiency.

3.5. Fusion of FFCO and ECC Robust Security Solutions

The fusion of ECC and FFCO (Algorithm 6) presents a formidable alliance, combining advanced encryption techniques with dynamic optimization strategies. The integration of ECC and FFCO brings forth a cryptographic framework that combines the robustness of ECC's mathematical foundation with the adaptive optimization capabilities of FFCO.

- Optimized Cryptographic Operations: FFCO enhances ECC's operations, optimizing mathematical computations for more efficient and secure cryptographic processes.
- Resistance Against Attacks: The integration provides a multi-layered defense, combining FFCO's iterative frequency adaptation with ECC's inherent security measures against cryptographic threats.
- Efficient Resource Utilization: ECC's resource-efficient design, enhanced by FFCO, ensures minimal computational and energy resources for cryptographic operations, crucial in resource-constrained scenarios.

This cryptographic framework stands at the forefront of secure communication, offering a harmonious balance between robust mathematical security and dynamic optimization strategies. The resulting synergy enhances cryptographic operations' adaptability, efficiency, and overall security, paving the way for advanced and resilient cryptographic protocols in diverse applications.

Step 1: ECC-FFCO

Step 2: Begin

Step 3: ECC_KeyPair GenerateECCKeyPair():

a) Curve ← EllipticCurve.SelectCurve(NIST_P_256)

b) PrivateKey ← RandomNumberGenerator.GenerateRandomInteger(1, Curve.Order - 1)

c) PublicKey ← Curve.GeneratorPoint * PrivateKey

d) Return ECC_KeyPair { PrivateKey, PublicKey}

Step 4: FFCO_Parameters InitializeFFCOParameters():

a) MaxIterations ← 1000

b) CheetahPopulation ← 50

c) ConvergenceCriteria ← 1e-6

d) Return FFCO_Parameters { MaxIterations, CheetahPopulation, ConvergenceCriteria }

Step 5: Solution FastFuriousCheetahOptimization(FFCO_Parameters parameters):

a) Population ← InitializeCheetahPopulation(parameters.CheetahPopulation)

b) For iteration in 1 to parameters.MaxIterations:

c) EvaluateFitness(Population)

d) UpdateCheetahPositions(Population)

e) If Converged(Population, parameters.ConvergenceCriteria):

f) Break

g) Return GetBestSolution(Population)

Step 6: HashedData HashFunction(Solution data):

a) HashedData ← SHA_256.Hash(data)

b) Return HashedData

Step 7: EncryptedData ECCEncryption(OriginalData data, ECC_Key key):

a) SharedSecret ← ECDH.KeyExchange(data, key)

b) EncryptionKey ← DeriveSymmetricKey(SharedSecret)

c) EncryptedData ← AES.Encrypt(data, EncryptionKey)

d) Return EncryptedData

Step 8: EncryptedData ECC_FFCO_FusionAlgorithm(OriginalData data):

a) ECCKeyPair ← GenerateECCKeyPair()

b) FFCOParameters ← InitializeFFCOParameters()

c) For iteration in 1 to FFCOParameters.MaxIterations:

d) FFCOBestSolution ← FastFuriousCheetahOptimization(FFCOParameters)

RESEARCH ARTICLE

e) HashedSolution ← HashFunction(FFCOBestSolution)

f) EncryptedData ← ECCEncryption(data, ECCKeyPair.PublicKey XOR HashedSolution)

g) Return EncryptedData

Step 9: END

Algorithm 6 FFCO-ECC

4. RESULTS AND DISCUSSION

4.1. Simulation Settings

Table 1 Simulation Settings

Parameter	Values
Network simulator	NS-3
MAC Type	IEEE 802.11p
Radio Wave Propagation	Two-Ray Ground
Antenna	Omnidirectional
Simulation Time	600 seconds
Mobility Model	Random Waypoint
Area size	Urban road network
No. of Vehicles	100
Traffic Density	Medium
Communication Range	200 meters
Transmission Power	20 dBm
Routing Protocol	AODV
Vehicle Types	Cars
Interference Model	Path loss
Data Packet Size	200 bytes
Network Layers	MAC
Road Layout Complexity	Urban roads with intersections
Environmental Conditions	Normal weather conditions
Transmission Models	V2V

NS-3 is well-suited for simulating VANETs due to its flexibility, extensibility, and robust networking capabilities. With a modular architecture written in C++ and Python, NS-3 enables researchers to model and analyze complex VANET scenarios, including realistic mobility models, diverse communication protocols, and dynamic network conditions. Its open-source nature allows for the integration of custom protocols tailored to the specific needs of VANET research. As a powerful tool for networking experiments, NS-3 plays a crucial role in evaluating the performance, security, and efficiency of VANET-related technologies, contributing significantly to advancements in intelligent transportation systems. The simulation settings are shown in Table 1.

4.2. Energy Consumption

Energy consumption in VANETs refers to the energy utilized by individual vehicles or nodes for communication, computation, and data transmission within the network. Managing energy resources is crucial in VANETs, where vehicles are equipped with communication devices, sensors, and computing units. Figure 1 and Figure 2 provide the analysis of FFCOSR's energy consumption against the state-of-the-art protocols.

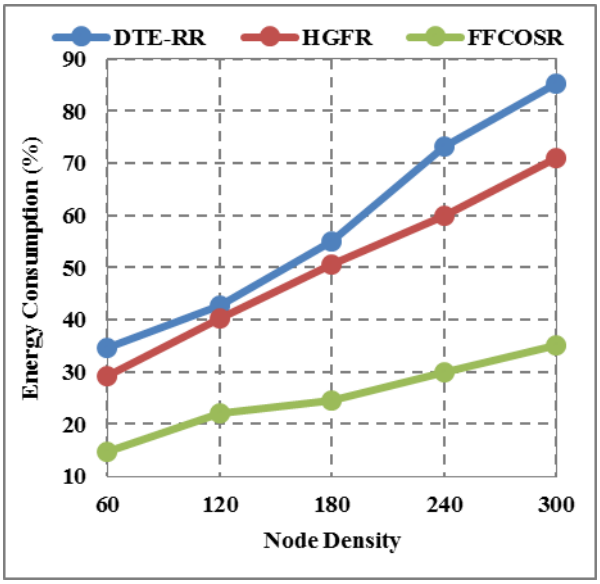

Figure 1 Energy Consumption

Figure 1 elucidates the intricate relationship between energy consumption and varying node densities, unraveling the operational characteristics of three prominent routing protocols in VANETs: DTE-RR, HGFR, and FFCOSR. At the initial node density of 60, DTE-RR exhibits a modest energy consumption of 34.662. This highlights its adeptness in optimizing energy utilization in relatively sparse vehicular presence scenarios. The protocol, relying on dynamic topology evolution, balances efficient communication and reasonable energy consumption, a crucial factor for the sustainability of VANETs. HGFR closely follows with an energy consumption of 29.155 at 60 nodes, showcasing its cooperative and adaptive nature. By leveraging a hybrid genetic-firefly strategy, HGFR optimizes routing decisions, resulting in reduced energy consumption. This efficiency is particularly evident in scenarios with lower node densities,

RESEARCH ARTICLE

underscoring HGFR's ability to enhance communication reliability while conserving energy resources. Remarkably, FFCOSR stands out with a significantly lower energy consumption of 14.610 at 60 nodes. This underscores FFCOSR's unique optimization inspired by Fast Furious Cheetah behavior, emphasizing its prowess in minimizing energy usage for secure routing in less congested VANET scenarios. The protocol introduces an innovative approach that aligns with energy conservation goals while ensuring effective data transmission.

As the node density escalates, DTE-RR adapts while maintaining competitiveness. At 240 nodes, it exhibits an energy consumption of 73.153, showcasing its ability to adjust to denser vehicular environments dynamically. HGFR and FFCOSR adapt, revealing energy consumptions of 59.831 and 30.011, respectively, at 240 nodes. Notably, FFCOSR consistently excels, emphasizing its secure routing approach inspired by cheetah optimization and its capacity to ensure minimal energy consumption even in denser VANET scenarios. Figure 1 provides a detailed snapshot of the energy consumption dynamics across node densities. DTE-RR, HGFR, and FFCOSR showcase distinctive profiles, with DTE-RR and HGFR demonstrating adaptability and FFCOSR excelling in energy efficiency. This nuanced understanding is vital for network planners and researchers aiming to deploy VANETs with routing protocols that align with specific density conditions and sustainability goals.

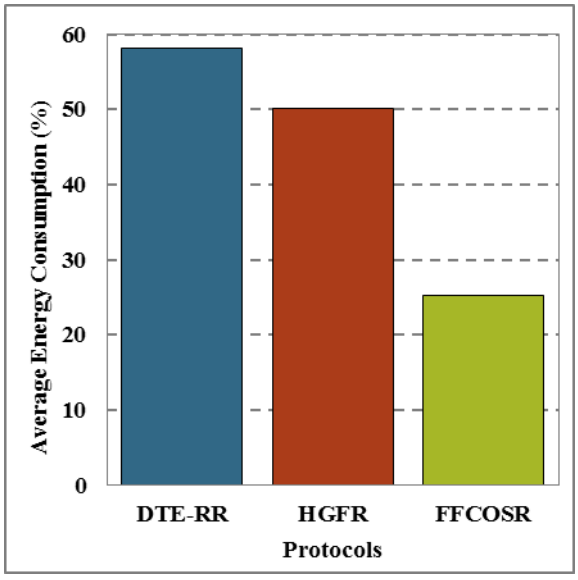

Figure 2 Average Energy Consumption

4.3. Packet Delay Analysis

Packet delay is a critical metric representing the time data packets traverse the network from source to destination, influencing communication efficiency and reliability.

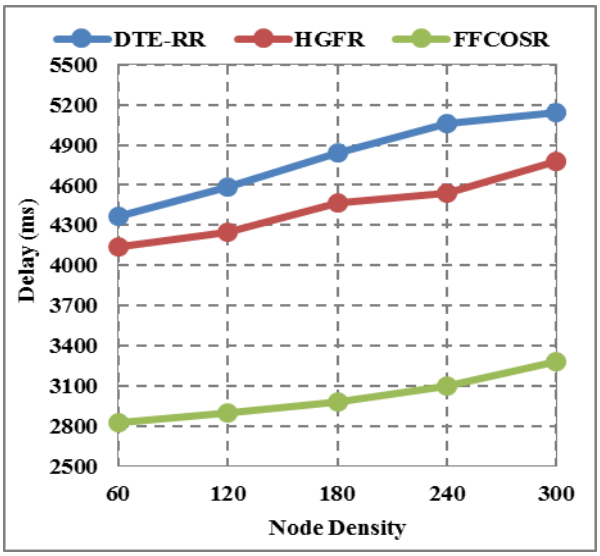

Figure 3 Packet Delay

Figure 3 unveils the intricate delay dynamics across varying node densities, providing insights into the operational behavior of DTE-RR, HGFR, and FFCOSR within VANETs. DTE-RR, leveraging dynamic topology evolution, exhibits robust adaptability across node densities. At 60 nodes, DTE-RR demonstrates a delay of 4364, emphasizing its efficiency in managing communication paths in scenarios with sparse vehicular presence.

This showcases DTE-RR's capacity to optimize data transmission in less congested environments, contributing to reduced communication latency. HGFR, employing a hybrid genetic-firefly strategy, closely follows with a delay of 4141 at 60 nodes. This highlights HGFR's cooperative and adaptive nature, where genetic algorithms and firefly optimization enhance routing decisions, reducing delay. The protocol's efficiency becomes evident, particularly in scenarios with lower node densities, showcasing its effectiveness in minimizing communication latency.

FFCOSR excels with a notably low delay of 2821 at 60 nodes, showcasing its unique optimization inspired by Fast Furious Cheetah behavior for secure routing. FFCOSR's ability to minimize delay in low-density environments underscores its effectiveness in swiftly and securely transmitting data packets, making it a standout performer in less congested VANET scenarios.

As node density increases, DTE-RR maintains competitiveness, showcasing adaptability with delays of 5056 at 240 nodes. HGFR and FFCOSR adapt, displaying delays of 4542 and 3102, respectively, at 240 nodes. FFCOSR consistently excels, emphasizing its secure routing approach inspired by cheetah optimization, ensuring minimal delay even in denser VANET scenarios.

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/224449 Volume 11, Issue 2, March – April (2024)

RESEARCH ARTICLE

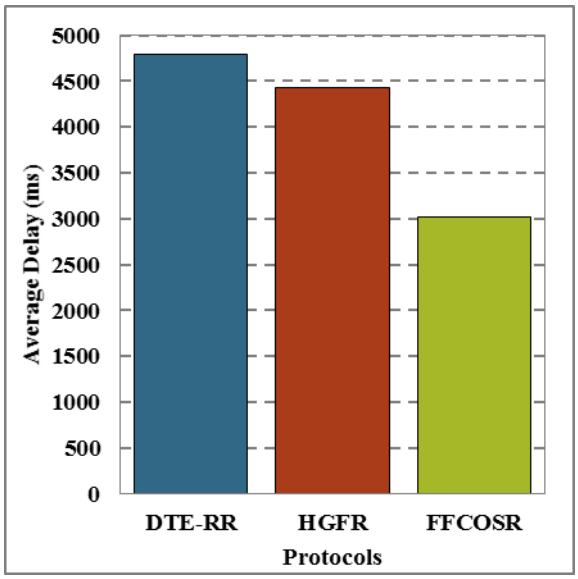

Figure 4 Average Packet Delay

Figure 4 provides a holistic perspective on the average delay landscape, offering a comprehensive view of the overall efficiency of DTE-RR, HGFR, and FFCOSR across diverse node densities in VANETs. DTE-RR maintains an average delay of 4797.6, showcasing balanced and consistent performance. This underscores DTE-RR's dynamic adaptability, where the dynamic topology evolution mechanism optimizes communication paths effectively, ensuring timely data packet delivery across varied scenarios. HGFR closely follows with an average delay of 4434.8, highlighting the efficiency of its hybrid genetic-firefly approach. Integrating genetic algorithms and firefly optimization contributes to a notable reduction in average delay. HGFR's adaptability and cooperation make it particularly efficient in scenarios with diverse node densities, showcasing its effectiveness as a routing solution in VANETs.

FFCOSR stands out with an average delay of 3015.0, establishing itself as remarkably efficient, especially in scenarios characterized by high node densities. FFCOSR's success can be attributed to its unique optimization approach inspired by Fast Furious Cheetah behavior, ensuring swift and secure routing decisions. The protocol's ability to minimize average delay signifies its suitability for resource-intensive, densely populated vehicular environments, emphasizing its role as a safe and efficient routing solution in VANETs. Figures 3 and 4 collectively provide a nuanced understanding of how each routing protocol manages delay in VANETs. DTE-RR exhibits resilience and adaptability, and HGFR proves consistently efficient. At the same time, FFCOSR excels, particularly in dense network environments, showcasing its unique approach to secure and swift routing inspired by Fast Furious Cheetah Optimization. Researchers and practitioners can leverage these distinctions for protocol selection based on specific VANET deployment scenarios and node density conditions.

4.4. Packet Loss

Packet Loss is when data packets transmitted across a network fail to reach their destination. This phenomenon can occur due to network congestion, errors in transmission, or limitations in network resources, impacting the reliability and completeness of data delivery.

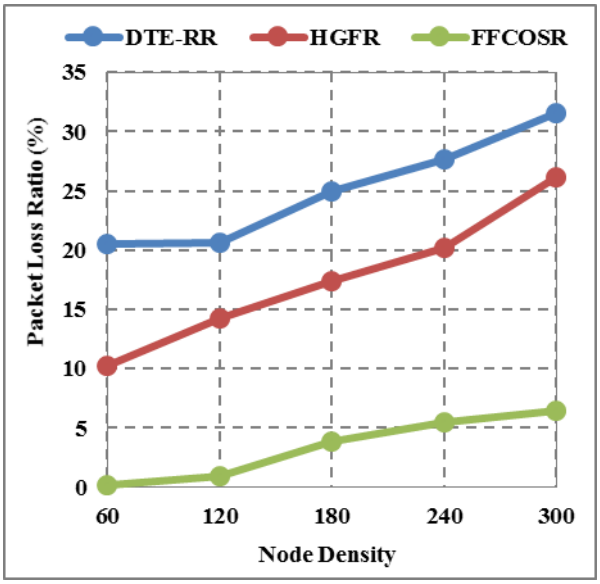

Figure 5 Packet Loss

In Figure 5, an intricate packet loss analysis unfolds across varying node densities, shedding light on the performance nuances of the routing protocols DTE-RR, HGFR, and FFCOSR within VANETs. DTE-RR, characterized by its dynamic topology evolution, showcases its adaptability and resilience to varying node densities. At lower node densities, precisely 60 nodes, DTE-RR exhibits a packet loss of 20.512%, reflecting its ability to efficiently optimize communication paths in sparse vehicular presence. This demonstrates its proficiency in managing data packet delivery in less congested environments. HGFR, leveraging a hybrid genetic-firefly approach, closely follows with a packet loss of 10.220% at 60 nodes. This indicates the cooperative and adaptive nature of HGFR, which employs genetic algorithms and firefly optimization to enhance routing decisions. The lower packet loss emphasizes the efficiency of HGFR in minimizing data loss, especially in scenarios with a lower node density. Remarkably, FFCOSR outperforms both protocols with a shallow packet loss of 0.208% at 60 nodes. FFCOSR's unique approach, inspired by FAst Furious Cheetah Optimization, showcases its efficacy in secure routing with minimal packet loss, even in low-vehicular

RESEARCH ARTICLE

density scenarios. This highlights the protocol's ability to swiftly and securely deliver data packets, setting it apart in less congested VANET environments. DTE-RR maintains its competitiveness as node density increases, demonstrating adaptability with a packet loss of 27.657% at 240 nodes. HGFR and FFCOSR also adapt, showcasing packet losses of 20.163% and 5.479%, respectively, at 240 nodes. FFCOSR consistently excels, emphasizing its secure routing approach inspired by cheetah optimization, ensuring minimal packet loss even in denser VANET scenarios.

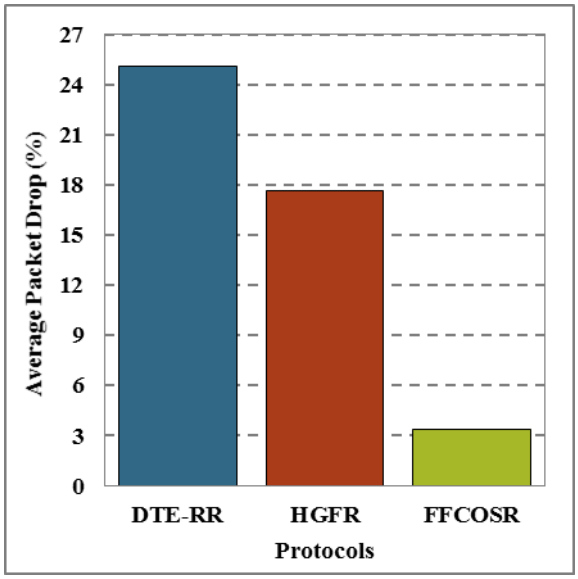

Figure 6 Average Packet Loss

Figure 6 unfolds a compelling narrative about the average packet loss dynamics across diverse node densities, providing insights into the overall performance of DTE-RR, HGFR, and FFCOSR within VANETs. DTE-RR, with an average packet loss of 25.075%, establishes a reliable and balanced performance metric. This underscores DTE-RR's prowess in adapting dynamically to changing network conditions, where its dynamic topology evolution optimizes communication paths efficiently. The protocol's consistent average packet loss across varying node densities showcases its reliability in real-world VANET scenarios. HGFR closely follows with an average loss of 17.657%, underscoring its efficacy in minimizing data loss. The hybrid genetic-firefly approach in HGFR contributes to efficient routing decisions, making it particularly suitable for scenarios with diverse node densities. HGFR's cooperative nature aligns well with the demands of VANETs, ensuring a commendable performance average. Notably, FFCOSR stands out with an impressively low average packet loss of 3.407%. This exceptional performance solidifies FFCOSR's position as a standout solution, especially in high-density VANET environments. FFCOSR's unique optimization approach inspired by Fast Furious Cheetah Optimization effectively ensures secure and swift routing decisions, resulting in minimal average packet loss. This feature makes FFCOSR appealing for environments demanding heightened security and efficiency. Figure 6 portrays a nuanced understanding of the average packet loss landscape in VANETs. DTE-RR showcases reliability, HGFR excels in efficiency, and FFCOSR stands out for its exceptional performance in minimizing average packet loss. Researchers and practitioners can leverage these distinctions to align their protocol selection with specific VANET deployment scenarios, tailoring their choices based on the unique strengths of each protocol in managing average packet loss across varied node densities.

4.5. Throughput

Throughput refers to the data transfer rate or the amount of data transmitted successfully over a network within a specific time frame. In the context of VANETs, throughput measures the network's capacity to efficiently deliver data packets, reflecting its overall performance in data transmission.

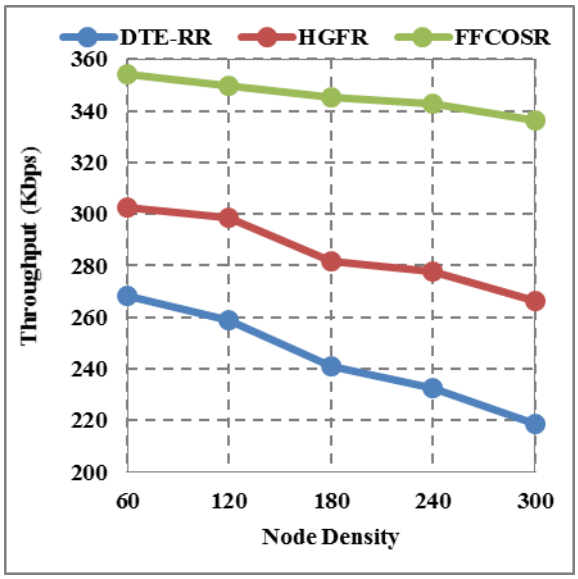

Figure 7 Throughput

Figure 7 scrutinizes the throughput dynamics across diverse node densities, unraveling the operational intricacies of three prominent routing protocols in VANETs: DTE-RR, HGFR, and FFCOSR. At the initial node density 60, FFCOSR emerges as a frontrunner, showcasing an impressive throughput of 354.265 Kbps. This underscores FFCOSR's ability to efficiently facilitate high-speed data transfer, leveraging its unique optimization inspired by Fast Furious Cheetah behavior. The protocol's adeptness in achieving superior throughput in scenarios with lower node densities positions it as a promising solution for applications demanding swift and reliable data transmission in less congested VANET environments. DTE-RR and HGFR closely follow with throughputs of 268.367 Kbps and 302.446

RESEARCH ARTICLE

Kbps, respectively, at 60 nodes. These results highlight the commendable performance of both protocols in facilitating efficient data transfer. With its dynamic topology evolution and HGFR, DTE-RR utilizes a hybrid genetic-firefly approach to showcase competitive throughput values, emphasizing their efficacy in supporting reliable communication in scenarios with moderate node densities.

As node density escalates to 300, DTE-RR maintains competitiveness with a throughput of 218.402 Kbps, showcasing its adaptability to denser vehicular environments. HGFR and FFCOSR also adapt, with throughputs of 266.350 Kbps and 336.446 Kbps, respectively, at 300 nodes. FFCOSR consistently excels, emphasizing its secure routing approach inspired by cheetah optimization, ensuring superior throughput even in denser VANET scenarios. Figure 7 provides a nuanced understanding of how each protocol responds to increasing node densities, reflecting their adaptability and efficiency in managing data transfer demands. These insights are crucial for network planners and researchers aiming to deploy VANETs with routing protocols that align with specific density conditions, ensuring optimal throughput and reliable communication in dynamically changing vehicular environments.

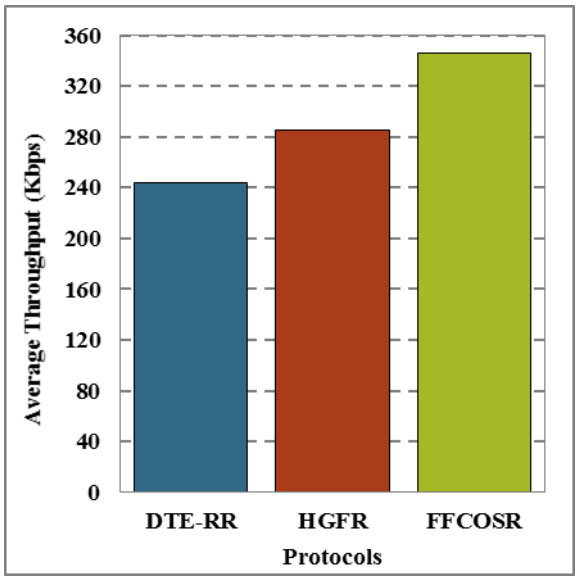

Figure 8 Average Throughput

Figure 8 offers a comprehensive overview of the average throughput landscape, presenting a holistic perspective on the efficiency of three prominent routing protocols in VANETs: DTE-RR, HGFR, and FFCOSR. DTE-RR maintains an average throughput of 243.836 Kbps, reflecting its balanced and competitive performance across diverse node densities. The protocol's dynamic adaptability and optimization mechanisms contribute to a consistent and reliable throughput, positioning it as a dependable choice for VANET applications requiring efficient and steady data transmission. DTE-RR's ability to maintain competitiveness in average throughput underscores its versatility and adaptability in various vehicular scenarios. HGFR closely follows with an average throughput of 285.406 Kbps, showcasing the effectiveness of its hybrid genetic-firefly approach in optimizing routing decisions. The protocol's adaptability and cooperative nature contribute to competitive average throughput values, making it a promising solution for VANET scenarios with varying node densities. HGFR's ability to consistently achieve average throughput demonstrates its suitability for applications demanding reliable and efficient data transfer in dynamic vehicular environments.

FFCOSR stands out with an impressive average throughput of 345.688 Kbps, establishing itself as exceptionally efficient across diverse node densities. FFCOSR's success can be attributed to its unique optimization approach inspired by Fast Furious Cheetah behavior, ensuring high-speed and reliable data transfer. The protocol's ability to consistently achieve superior average throughput values signifies its suitability for data-intensive applications in VANETs, emphasizing its role as a secure and efficient routing solution. Figure 8 provides valuable insights into the average throughput performance of DTE-RR, HGFR, and FFCOSR, allowing network planners and researchers to make informed decisions based on specific deployment requirements and the need for high-speed data transfer in diverse vehicular environments. The distinctions in average throughput underscore the unique strengths of each protocol, enabling stakeholders to choose the most suitable routing solution for their VANET applications.

5. CONCLUSION

FFCOSR is an innovative framework that addresses routing and security challenges inherent in Vehicular Ad Hoc Networks (VANETs). FFCOSR integrates Route Life Time Enhanced AODV (RLE-AODV), an optimized version of the Ad Hoc On-Demand Distance Vector (AODV) protocol, with Fast Furious Cheetah Optimization (FFCO) and Elliptic Curve Cryptography (ECC). RLE-AODV enhances routing efficiency by prolonging route stability through FFCO's dynamic adjustment of routing parameters, thus mitigating common VANET issues like route flapping and packet loss. FFCOSR fortifies data security using ECC, ensuring the integrity and confidentiality of transmitted data amidst potential security threats. Empirical validation is necessary to substantiate FFCOSR's effectiveness in real-world VANET environments, with future enhancements focusing on further refining its mechanisms to optimize routing efficiency and security features. FFCOSR holds significant promise for advancing vehicular communication systems' reliability, efficiency, and security, offering valuable contributions to the evolving landscape of connected and autonomous vehicles.

RESEARCH ARTICLE

REFERENCES

[1] M. Azizi and S. Shokrollahi, "RTRV: An RSU-assisted trust-based routing protocol for VANETs," Ad Hoc Networks, vol. 154, p. 103387, 2024, doi: 10.1016/j.adhoc.2023.103387.

[2] S. Hosmani and B. Mathapati, "Efficient Vehicular Ad Hoc Network routing protocol using weighted clustering technique," Int. J. Inf. Technol., vol. 13, no. 2, pp. 469–473, 2021, doi: 10.1007/s41870-020-00537-2.

[3] V. K. Quy, V. H. Nam, D. M. Linh, N. T. Ban, and N. D. Han, "Communication Solutions for Vehicle Ad-hoc Network in Smart Cities Environment: A Comprehensive Survey," Wirel. Pers. Commun., vol. 122, no. 3, pp. 2791–2815, 2022, doi: 10.1007/s11277-021-09030-w.

[4] K. Chandramohan, A. Manikandan, S. Ramalingam, and R. Dhanapal, "Performance Evaluation of VANET using Directional Location Aided Routing (D-LAR) Protocol with Sleep Scheduling Algorithm," Ain Shams Eng. J., vol. 15, no. 3, p. 102458, 2024, doi: 10.1016/j.asej.2023.102458.

[5] J. K. Shahrouz and M. Analoui, "An anonymous authentication scheme with conditional privacy-preserving for Vehicular Ad hoc Networks based on zero-knowledge proof and Blockchain," Ad Hoc Networks, vol. 154, p. 103349, 2024, doi: 10.1016/j.adhoc.2023.103349.

[6] L. R. Gallego-Tercero, R. Menchaca-Mendez, M. E. Rivero-Angeles, and R. Menchaca-Mendez, "Efficient time-stable geocast routing in delay-tolerant vehicular ad-hoc networks," IEEE Access, vol. 8, pp. 171034–171048, 2020, doi: 10.1109/ACCESS.2020.3024541.

[7] Z. Zhou, A. Gaurav, B. B. Gupta, M. D. Lytras, and I. Razzak, "A Fine-Grained Access Control and Security Approach for Intelligent Vehicular Transport in 6G Communication System," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 7, pp. 9726–9735, 2022, doi: 10.1109/TITS.2021.3106825.

[8] K. Ahed, M. Benamar, A. A. Lahcen, and R. El Ouazzani, "Forwarding strategies in vehicular named data networks: A survey," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 5, pp. 1819–1835, 2022, doi: 10.1016/j.jksuci.2020.06.014.

[9] N. H. Hussein, C. T. Yaw, S. P. Koh, S. K. Tiong, and K. H. Chong, "A Comprehensive Survey on Vehicular Networking: Communications, Applications, Challenges, and Upcoming Research Directions," IEEE Access, vol. 10, pp. 86127–86180, 2022, doi: 10.1109/ACCESS.2022.3198656.

[10] R. Agrawal et al., "Classification and comparison of ad hoc networks: A review," Egypt. Informatics J., vol. 24, no. 1, pp. 1–25, 2023, doi: 10.1016/j.eij.2022.10.004.

[11] E. Khoza, C. Tu, and P. A. Owolawi, "Decreasing traffic congestion in vanets using an improved hybrid ant colony optimization algorithm," J. Commun., vol. 15, no. 9, pp. 676–686, 2020, doi: 10.12720/jcm.15.9.676-686.

[12] J. Ramkumar, A. Senthilkumar, M. Lingaraj, R. Karthikeyan, and L. Santhi, "Optimal Approach for Minimizing Delays in Iot-Based Quantum Wireless Sensor Networks Using Nm-Leach Routing Protocol," J. Theor. Appl. Inf. Technol., vol. 102, no. 3, pp. 1099–1111, 2024.

[13] J. Ramkumar, R. Vadivel, B. Narasimhan, S. Boopalan, and B. Surendren, "Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) for Quality of Service Enhancement in Internet of Things-Based Public Cloud Networking," J. M. R. S. Tavares, J. J. P. C. Rodrigues, D. Misra, and D. Bhattacherjee, Eds., Singapore: Springer Nature Singapore, 2024, pp. 425–438. doi: 10.1007/978-981-99-5435-3_30.

[14] J. Ramkumar and R. Vadivel, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," Wirel. Pers. Commun., vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[15] R. Jaganathan, V. Ramasamy, L. Mani, and N. Balakrishnan, "Diligence Eagle Optimization Protocol for Secure Routing (DEOPSR) in Cloud-Based Wireless Sensor Network," Res. Sq., 2022, doi: 10.21203/rs.3.rs-1759040/v1.

[16] D. Xue, Y. Guo, N. Li, X. Song, and M. He, "Cross-domain cooperative route planning for edge computing-enabled multi-connected vehicles," Comput. Electr. Eng., vol. 108, p. 108668, 2023, doi: 10.1016/j.compeleceng.2023.108668.

[17] T. Li, F. Guo, R. Krishnan, and A. Sivakumar, "An analysis of the value of optimal routing and signal timing control strategy with connected autonomous vehicles," J. Intell. Transp. Syst. Technol. Planning, Oper., vol. 28, no. 2, pp. 252–266, 2022, doi: 10.1080/15472450.2022.2129021.

[18] K. Haseeb, A. Rehman, T. Saba, S. A. Bahaj, H. Wang, and H. Song, "Efficient and trusted autonomous vehicle routing protocol for 6G networks with computational intelligence," ISA Trans., vol. 132, pp. 61–68, 2023, doi: 10.1016/j.isatra.2022.09.035.

[19] R. Tirumalasetti and S. K. Singh, "Automatic Dynamic User Allocation with opportunistic routing over vehicles network for Intelligent Transport System," Sustain. Energy Technol. Assessments, vol. 57, p. 103195, 2023, doi: 10.1016/j.seta.2023.103195.

[20] Saifullah, Z. Ren, K. Hussain, and M. Faheem, "K-means online-learning routing protocol (K-MORP) for unmanned aerial vehicles (UAV) adhoc networks," Ad Hoc Networks, vol. 154, p. 103354, 2024, doi: 10.1016/j.adhoc.2023.103354.

[21] Parveen, S. Kumar, R. P. Singh, A. Kumar, R. Yaduwanshi, and D. P. Dora, "TS-CAGR:Traffic sensitive connectivity-aware geocast routing protocol in internet of vehicles," Ad Hoc Networks, vol. 147, p. 103210, 2023, doi: 10.1016/j.adhoc.2023.103210.

[22] M. V. Kadam, H. B. Mahajan, N. J. Uke, and P. R. Futane, "Cybersecurity threats mitigation in Internet of Vehicles communication system using reliable clustering and routing," Microprocess. Microsyst., vol. 102, p. 104926, 2023, doi: 10.1016/j.micpro.2023.104926.

[23] K. Matrouk, Y. Trabelsi, V. Gomathy, U. Arun Kumar, C. R. Rathish, and P. Parthasarathy, "Energy efficient data transmission in intelligent transportation system (ITS): Millimeter (mm wave) based routing algorithm for connected vehicles," Optik (Stuttg)., vol. 273, p. 170374, 2023, doi: 10.1016/j.ijleo.2022.170374.

[24] Y. A. Shah et al., "An Evolutionary Algorithm-Based Vehicular Clustering Technique for VANETs," IEEE Access, vol. 10, pp. 14368–14385, 2022, doi: 10.1109/ACCESS.2022.3145905.

[25] Y. Feng, Y. Huang, B. Li, H. Peng, J. Wang, and W. Zhou, "Connectivity Enhancement of E-VANET Based on QL-mRSU Self-Learning Energy-Saving Algorithm," IEEE Access, vol. 11, pp. 3810–3825, 2023, doi: 10.1109/ACCESS.2023.3235397.

[26] A. Salim, A. M. Khedr, B. Alwasel, W. Osamy, and A. Aziz, "SOMACA: A New Swarm Optimization-Based and Mobility-Aware Clustering Approach for the Internet of Vehicles," IEEE Access, vol. 11, pp. 46487–46503, 2023, doi: 10.1109/ACCESS.2023.3275446.

[27] J. Ramkumar and R. Vadivel, CSIP—cuckoo search inspired protocol for routing in cognitive radio ad hoc networks, vol. 556. 2017. doi: 10.1007/978-981-10-3874-7_14.

[28] L. Mani, S. Arumugam, and R. Jaganathan, "Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol," ACM Int. Conf. Proceeding Ser., pp. 1–5, Dec. 2022, doi: 10.1145/3590837.3590907.

[29] Z. Han, C. Xu, S. Ma, Y. Hu, G. Zhao, and S. Yu, "DTE-RR: Dynamic Topology Evolution-Based Reliable Routing in VANET," IEEE Wirel. Commun. Lett., vol. 12, no. 6, pp. 1061–1065, 2023, doi: 10.1109/LWC.2023.3260142.

[30] A. Sheela Rini, C. Meena, "Analysis of Machine Learning Classifiers to Detect Malicious Node in Vehicular Cloud Computing", International Journal of Computer Networks and Applications (IJCNA), 9(2), PP: 202-213, 2022, DOI: 10.22247/ijcna/2022/212336.

[31] G. D. Singh, M. Prateek, S. Kumar, M. Verma, D. Singh, and H. N. Lee, "Hybrid Genetic Firefly Algorithm-Based Routing Protocol for VANETs," IEEE Access, vol. 10, pp. 9142–9151, 2022, doi: 10.1109/ACCESS.2022.3142811.

RESEARCH ARTICLE

Authors

A. Sheela Rini is a dedicated educator and researcher with over a decade of experience in computer science. Holding a Master's degree in Computer Science and an M.Phil. with commendation, she qualified UGC-NET for Assistant Professor in Computer Science in 2018. She is currently working as an Assistant Professor at PSGR Krishnammal College for Women and pursuing her Ph.D. at Avinashilingam Institute for Home Science and Higher Education for Women. She has presented papers at national and international conferences, published in esteemed journals, and participated in workshops for professional development. She strives for academic excellence, emphasizing practical problem-solving skills for students, and is committed to quality education and the application of computer science principles.

Dr. C. Meena presently designated as Incharge of Computer Center in Avanishalingam Institute for Home Science and Higher Education for Women, Coimbatore. She has 27 years of rich & extensive experience in teaching, System Analysis, Technical Information and Programming Skills. She is expertise in Project Personals and Budget Estimation. She has presented papers in many countries like Malaysia, USA etc. She has published more than 45 papers in National and International Journals. She has undertaken many UGC sponsored projects.

How to cite this article:

A. Sheela Rini, C. Meena, "Dynamic Integration of Fast Furious Cheetah Optimization for Efficient and Secure Routing in Vehicular Ad Hoc Networks", International Journal of Computer Networks and Applications (IJCNA), 11(2), PP: 248-273, 2024, DOI: 10.22247/ijcna/2024/224449.