SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I7P112

Volume 13 Issue 7, 166-184, July 2026

Original Article

COD-STAR: The Energy Efficient Vehicular Communication Using Chinchilla-Based Adaptive Path Selection and Trust Evaluation

R. Jasmine Immaculate Shelly[1], M. Milton Joe[2], B. Ramakrishnan[3]

[1,3]Department of Computer Science and Research Centre, S.T. Hindu College
(Affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli), Nagercoil, Tamil Nadu, India.
[2]Department of Computer Science, St. Jerome's College
(Affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli), Nagercoil, Tamil Nadu, India.

[1]Corresponding Author : shellyrvk@gmail.com

Received: 09 February 2026 Revised: 30 May 2026 Accepted: 19 June 2026 Published: 29 July 2026

Abstract *- Vehicular Ad Hoc Networks require routing protocols which are able to handle high mobility, uncertain connectivity, and changing environmental conditions. Proposed COD-STAR presents a hybrid method involving Chinchilla Optimization and Dempster-Shafer Theory that can support intelligent routing that is trust-aware and context sensitive. The protocol starts with building anchor paths based on the structured urban topology, and then real-time contextual data of the mobile nodes and infrastructure is collected. Various routing metrics are assigned belief values and fused with the help of DST to determine how reliable the paths are when there is uncertainty. Adaptive exploration and exploitation of candidate routes is done by Chinchilla agents, which are informed by dynamic fitness models that combine fused beliefs and environmental volatility. The continuous updating of belief scores is facilitated by feedback mechanisms, which allow the recalibration of the path in real-time and refinement of decisions. The combination of optimization intelligence and belief-based reasoning allows COD-STAR to deliver valid and effective communication routes in demanding conditions of VANETs, which guarantees the routing stability and scalability in the ever-changing traffic situations.*

Keywords *- Adaptive routing, Belief-based decision making, Chinchilla Optimization, Dempster-Shafer Theory, Vehicular Ad Hoc Networks (VANETs)***.**

1. Introduction

The Vehicular Ad Hoc Networks (VANETs) have become an interesting communication paradigm that is capable of supporting the development of intelligent transportation systems using dynamic communication between vehicles (V2V) and between vehicle and infrastructure (V2I). VANETs enable several real-time applications such as road safety information services, traffic management, route planning, coordination of emergency response services, and infotainment services [1].

In contrast to the conventional wireless network setting, VANETs are deployed in highly dynamic network settings where the vehicles move with high speed, the topology changes often, wireless links are intermittent, and traffic densities are changing [2]. These features cause significant instability of communication and routing uncertainty, which makes reliable packet delivery hard in big environments like VANET. With the rapidly growing interconnection of transportation systems, the need for routing protocols that can ensure stable communication in dynamically changing road networks has increased greatly [3].

Routing is very important for VANET communication as it is responsible for determining the efficient dissemination of information between the source and destination vehicle [4]. In the traditional routing protocol, the frequent changing topology and high mobility of vehicles are the main reasons causing unstable path formation, slow packet forwarding, and frequent route breakage [5]. These challenges have been tackled by existing routing strategies such as topology-based, geographic-based, cluster-based, and hybrid routing methods with various path discovery and maintenance strategies. In particular, A-star-Routing, based on map knowledge and heuristic planning using road topology and traffic data, has gained significant interest in urban vehicles' environment [6]. A-Star is useful for providing better connectivity in a structured road environment, but its reliance on preloaded traffic assumptions reduces it is not being very flexible in real-time environments. Deterministic path selection mechanisms are often less effective due to sudden changes in mobility, temporary road congestion, and communication uncertainty [7].

Some of the recent works have adopted intelligent optimization approaches and trust-based decision models to enhance the routing performance [8]. Recently, some studies have been carried out that use intelligent optimization and trust-based decision models to improve the routing performance in VANETs [9, 10]. Bio-inspired optimization algorithms have demonstrated good performance, as they have been able to balance exploration/exploitation and adapt to different network environments. Similarly, uncertainty management methods like Dempster–Shafer Theory (DST) offer a good way of dealing with incomplete, inconsistent, and uncertain observations of routing through the evidence-based reasoning. Even though these developments have taken place, most of the current routing frameworks still have significant drawbacks [11]. Many protocols make strong assumptions about their routes and do not respond to changing vehicular conditions. Second, the uncertainty of routing metrics like node reliability, signal strength, vehicular density, and link quality is not sufficiently modeled, and hence, unreliable forwarding decisions. Thirdly, optimization-driven routing techniques often disregard shortest path or energy considerations and fail to incorporate contextual assessment of trust and adaptive adjustment of routes [12].

1.1. Problem Statement

VANETs work in extremely dynamic conditions with frequent changes in the topology, unpredictable mobility patterns, and dense node distributions. Current routing algorithms are not well adapted to handle such changing conditions because of a lack of flexibility, context-blindness, and inflexible route choice policies. The majority of traditional methods do not combine real-time monitoring of the environment and smart route planning, which leads to the loss of packets, high latency, low throughput, and inefficient use of energy. Most current protocols do not have a mechanism to effectively deal with uncertainty and conflict in observed network data, compromising the reliability of routing in large-scale deployments. As traffic congestion grows, the medium access level further reduces performance, as well as scalability and real-time responsiveness. The lack of a dynamic, evidence-based, and resource-conscious routing mechanism is an essential loophole in the provision of stable and effective communication in changing VANET infrastructures.

1.2. Motivation and Objective

The dense vehicular networks require a routing protocol that can handle the dynamic mobility, the changing traffic patterns, and the fluctuating link conditions. Traditional procedures are not always flexible and do not incorporate environmental intelligence and optimization strategies. Chinchilla-Optimized Dempster–Shafer Smart Traffic-Aware Routing (COD-STAR) presents a hybrid algorithm that is a union of Dempster-Shafer Theory of decision-making using belief and Chinchilla Optimization, which uses adaptive exploration of a path. Its main goal is to achieve better routing reliability, lower latency, and better energy efficiency through dynamically recalibrated routes with real-time feedback and trust. By providing context-sensitive, uncertainty-tolerant, and self-tuning routing behaviour across the communication structure, COD-STAR is designed to guarantee high delivery rates, low loss rates, and long network life, namely in congested, volatile VANET environments.

1.3. Novelty and Research Contributions

There are several routing researches in the VANET field which have been concentrated on optimizing, clustering, improving QoS, and secure communication. Most of the techniques in place, however, focus mainly on one aspect of the problem, e.g., shortest path selection, congestion control, clustering efficiency, or authentication. In highly dynamic vehicular environments, uncertainty of the routing metrics, such as signal strength, vehicular density, and link reliability, is not well handled by many protocols, resulting in less routing stability. Moreover, optimization-based methods generally do not take into account feedback on optimality on a continuous basis.

To overcome these drawbacks, a new integrated routing approach named COD-STAR (Chinchilla-Optimized Dempster–Shafer Smart Traffic-Aware Routing) has been introduced that integrates the Chinchilla Optimization (CO) with the Dempster–Shafer Theory (DST). Unlike previous works, COD-STAR conducts adaptive route optimization in conjunction with uncertainty-aware trust evaluation, taking into account multiple features' evidence fusion. The protocol also features a continual tuning algorithm based on feedback to dynamically adapt routing decisions based on traffic fluctuations. The main innovation of this work is the combination of the three techniques of bio-inspiration optimization, belief-based trust evaluation, and real-time route adaptation in a single VANET routing protocol. This design enhances the reliability of routing, packet delivery, delay, energy efficiency, and stability of communication in dense and highly dynamic vehicular environments.

2. Related Works

QRAVDR proposes a reinforcement learning framework in which deep Q-learning controls the routing of video information in vehicle settings. The model also adapts dynamically to learn the best data forwarding strategy depending on the current traffic density, network topology, and channel characteristics by incorporating Road Side Units (RSUs) as intelligent agents. The policy of rewards is involved in each decision to reduce the loss of data and provide steady video streaming [13]. Privacy-preserving authentication through efficiently and escrow-free batch authentication in dynamically configured VANETs without a trusted thirdparty escrow authority is made possible. The scheme proposal proposes identity concealment, dynamic issuance of keys, and batch verification, which guarantees the user anonymity and fast authentication of the message [14].

Fuzzy-AHP-ORD solves the RSU placement problem by means of using the road structure and traffic flow information in the fuzzy multi-criteria decision-making model. This is a combination of the Analytic Hierarchy Process (AHP) and fuzzy that is used to represent the uncertainty in vehicle movement and environmental conditions. The accessibility, vehicle density, potential connectivity, and reachability of the signal of each road segment are evaluated [15]. Fr-Aro fuses bio-inspired optimization with fuzzy interference management to come up with a secure clustering and routing strategy.

The fuzzy logic is used to cluster signals based on signal interference, node stability, and cost of communication. Hybrid optimization methods inform the choice of cluster heads and routes to take that provide a balance between the energy consumption and efficiency in delivery [16]. The contention-based strategy is used to optimize multi-hop dissemination by considering several network metrics to exclude redundant broadcasts. The model takes into consideration distance to destination, prediction of mobility, density of nodes, and signal quality in a broadcast scoring mechanism. Rebroadcasting is only done to high-priority nodes, which minimizes the bandwidth and prevented collisions. With local neighborhood conditions, suppression thresholds are optimally adjusted so that they are adaptive [17].

PCAS provides a detailed cryptanalysis of a pairing-free certificate-less signature scheme and introduces a more resilient one that admits conditional privacy. Weaknesses of the original protocol are overcome with the help of sophisticated key generation and aggregation processes. The updated mechanism provides authenticity of messages and traces them in specified legal circumstances [18]. By ranking nodes according to the stability of links, the signal strength, and the forwarding utility, adaptively prioritizing forwarding candidates can improve opportunistic routing. The model gives priority weights dynamically, adapting to the existing vehicular velocity, direction, and link expiration probability. The sets of candidates are continually updated, and the node with the highest rank will send the packet with the shortest delay [19]. Introduced in Leading the Way, a traffic reduction model in VANETs provides the use of cluster leader-based control. Nodes are grouped into temporary clusters, and some leaders are chosen to oversee intra and inter-cluster communications. The leaders receive and sift through redundant messages, blocking unneeded messages. The model recognizes the formation of clusters depending on the location, velocity, and their communication range. Heuristic rules are employed to elect the leaders, making it stable and low overhead [20]. A scalable traffic management system is a system that combines the concepts of Named Data Networking (NDN) and secure data handling optimized for VANETs. The architecture proposes the dynamic prefix advertisement, secure name-based forwarding, and real-time content caching. Vehicles are both consumers and distributors of traffic-related content, and every data transfer is bound by lightweight cryptographic binding [21].

A safe, lightweight authentication architecture is a combination of small cryptographic tasks and an interruption-conscious routing architecture of VANETs. The model allows verifying a message safely with low computational cost and thus is applicable to vehicles moving at high speeds. The interference of the neighbouring signals is measured to direct stable route choice. A detection system is inbuilt which tracks the pattern of the packets and the behaviour of the nodes in order to detect possible attacks [22]. C-QoSRP proposes the implementation of a quality-of-service routing protocol designed as a cluster-based protocol to operate in a high-speed highway environment. The network is split into dynamic clusters, with each cluster head being a stable cluster head that is chosen through vehicle speed, direction, and link strength. The routing system takes into consideration the QoS parameters like delay, bandwidth, and reliability of the packet in selecting the path [23].

The VANET cluster-based routing protocol that accommodates link breakage presents a hybrid optimization algorithm that incorporates clustering and proactive route stability. The protocol forms adaptive clusters in which the routing is controlled by the leaders and the neighbourhood status is maintained. Predictive algorithms based on signal strength and mobility vectors [24] are used to implement link breakage detection. Secured VANET introduces the optimized routing structure using the COOT algorithm, along with multi-layered authentication and the detection of fake messages. The COOT algorithm is improved to give priority to the secure energy-efficient routes in consideration of the density of vehicles, their mobility, and the quality of links. The various stages of authentication that are performed on each node are signature verification and identity validation, which help to eliminate spoofing. Anomaly detection models can detect fake messages based on the patterns of messages and timing deviations [5].

2.1. Design and Implementation of a Routing Protocol

The aim of the paper is to improve the Quality of Service (QoS) of VANETs through the addition of a customized routing architecture. The model considers the QoS parameters such as latency, provision of assurance in the delivery of packets, and throughput when building routes. Vehicles are grouped based on relative mobility and direction to form stable communication paths. Routing decisions consider real-time changes in node density and traffic conditions [25].

ADR, a novel cross-layer adaptive fuzzy-based routing protocol, enhances the traditional AODV model by integrating decision-making from multiple network layers. Fuzzy logic is applied to combine parameters of physical, MAC, and network layers to direct the route discovery and maintenance. The protocol will consider the signal strength, node energy, and mobility to decide on reliable paths on demand. Fuzzy rules are delineated to balance performance measures like delay, overhead, and link stability. The adjustment of dynamic routes is triggered in case major alterations are observed in the conditions of links [26].

The Intelligent Fuzzy Bald Eagle Optimization is used to help maximize the VANET performance when using cluster head selection to ensure stability and efficiency. This technique is a mix of fuzzy logic and Bald Eagle-inspired optimization in order to assess nodes on the basis of mobility, density, link strength, and energy. Candidate nodes are assessed using fuzzy scores, and the optimization algorithm explores, selects, and dives toward the best cluster head configurations. Each phase of the algorithm mimics the hunting strategy of a bald eagle, ensuring exploration, exploitation, and convergence in node selection [27].

3. Chinchilla Optimization with DST-Based Smart Traffic-Aware Routing

COD-STAR is an adaptive routing protocol that is a hybrid between Chinchilla Optimization and Dempster-Shafer Theory to make smart, belief-based, and traffic-aware path choices in dynamic Vehicular Ad Hoc Networks (VANETs). It provides dependable communication through a synthesis of bio-inspired exploration and probabilistic trust assessment of real-time routing choices.

3.1. Urban Anchor Path Initialization Using Topology Maps

Urban topology acts as the initial scaffold for route prediction in COD-STAR, serving as the first layer where Chinchilla Optimization starts encoding the structural behavior of candidate paths. The process begins with extracting road segments from standardized topology datasets, typically digital street maps. These segments are first converted into a road-graph structure $G(V, E)$, where V denotes the set of intersections (vertices), and E represents the road segments (edges) connecting them. This graph provides the static urban layout required for primary anchor path estimation.

Each edge $e_{ij} \in E$ is associated with spatial attributes such as segment length d_{ij}, average legal speed limit v_{ij}, and expected traffic density range ρ_{ij}. The initial cost function assigned to each edge integrates these static features to form a pre-evaluation matrix, which is represented in Equation (1).

$$C_{ij}^{(0)} = \alpha \cdot \frac{d_{ij}}{v_{ij}} + \beta \cdot \rho_{ij} \tag{1}$$

Important weights of delay and congestion impact are tunable parameters, α and β, respectively. This cost gives the foundation to assess the suitability of every road segment prior to engaging the dynamic belief or optimization measures.

After the initial graph is filled with fixed cost weights, the path-finding mechanisms will find the anchor paths with emphasis on route compactness and structure, as opposed to real-time reliability. These anchor paths can be expressed as a sequence of n connected edges, and are denoted by Equation (2):

$$P_k = \left\{e_{i_1 i_2}, e_{i_2 i_3}, \dots, e_{in-1^i n}\right\} \tag{2}$$

Each anchor path P_k becomes a member of the Chinchilla search space, acting as an encoded individual in the optimization population. During this stage, the Chinchilla mechanism forms a diverse set of paths traversing high-connectivity areas in the urban grid. These are candidates that have not yet been optimized, but which act as the first forage trails in the solution space.

The process of pruning a graph is used to remove unfeasible anchor sequences according to traffic constraints and structural disconnections to determine if it is feasible. It is a dynamic process that updates the edge set E by removing segments with a structural penalty more than some threshold provided in Equation (3):

$$E' = \left\{e_{ij} \in E \middle| C_{ij}^{(0)} \leq \theta\right\} \tag{3}$$

and θ is the pruning threshold, which is usually determined according to some established urban routing norms. It is replaced by Chinchilla Optimization, which is able to simulate the movement of agents along the encoded paths due to its adaptive trail-shaping behavior. Cost-efficiency of a path is measured by each agent as in a movement-inspired fitness model in Equation (4):

$$F_k = \sum_{e_{ij} \in P_k} C_{ij}^{(0)} + \lambda \cdot Var(v_k) \tag{4}$$

Term $Var(v_k)$ represents the inconsistency of speed over the path, and punishes paths with unpredictable travel situations. The smoothness parameter λ is used as a control parameter that is consistent with Chinchilla behaviour, minimizing sharp turns and favouring smooth transitions. The agents create new anchor sequences as they simulate their virtual traversal, converting towards structurally efficient paths that become stable over time.

3.2. Contextual Data Sensing from Neighbouring Vehicles and Roadside Units

COD-STAR relies on a reliable routing intelligence formation that starts with the extraction of real-time context of vehicular neighbors and stationary infrastructure nodes. Once

Chinchilla agents have learned the possible anchor routes during the initial topology encoding, adaptive sensing triggers the dynamic phase by taking fine-grained metrics on the environment that capture the live network state at each candidate path.

Each node within the VANET periodically broadcasts and receives beacon messages embedded with spatiotemporal information, including vehicle speed. v_t, inter-vehicular distance δ_t, node longevity τ_t, and signal strength η_t. These attributes are represented in a dynamic feature matrix. Φ_{ij} of a segment e_{ij} as is shown in Equation (5).

$$\Phi_{ij} = [v_t, \delta_t, \tau_t, \eta_t] \quad (5)$$

This matrix is a temporal evidence store to approximate the stability and viability of forward links in Chinchilla fitness assessment and Dempster-Shafer fusion in subsequent stages.

A local fluctuation index ξ_{ij} is calculated in the form of an Equation (6) in order to measure the volatility of segments. This index is a measure of the inconsistency of the environment across every segment under a weighted standard deviation model.

$$\xi_{ij} = \gamma_1 \cdot \sigma(v_t) + \gamma_2 \cdot \sigma(\delta_t) \quad (6)$$

The weighting coefficients γ_1 and γ_2 are adjusted based on sensitivity to motion variance and vehicular density fluctuation. A lower ξ_{ij} indicates higher environmental stability, which is rewarded during Chinchilla agent movement scoring.

Simultaneously, the confidence of link availability of any edge is computed based on instantaneous connectivity and past survivability. This measure, denoted as ψ_{ij}, that is based on empirical ratios of connectivity, is calculated as expressed in the form of Equation (7).

$$\psi_{ij} = \frac{Successful\ Links_{ij}}{Total\ Attempts_{ij}} + \omega \cdot \frac{\tau_t}{T} \quad (7)$$

where ω is a confidence weight, and T is the observation window. The larger the values of ψ_{ij}, The more likely the segment is to maintain stable communication during the forwarding of packets.

Each agent dynamically updates its internal map using contextual heatmaps formed from local sensing values to support Chinchilla Optimization. The optimization agents have a real-time adjustment cost of each edge, which is given as Equation (8)

$$\zeta_{ij} = \frac{1}{\eta_t} + \delta_t \cdot (1 - \psi_{ij}) \quad (8)$$

The cheaper this cost, the more appropriate a segment would be in the forward exploration process, which portrays the foraging economy that is shown in the Chinchilla search process. The agents will not follow paths with large. ζ_{ij}, pretending to avoid energy inefficient paths. This measure makes sure that the path of each agent is steered by geometrical closeness or building plan, and a refined understanding of the feasibility of the link at the current time. The interactions between beacon-level measurements and aggregate segment scoring affirm the exploratory reasoning by integrating the topological knowledge with behavioral knowledge.

3.3. Belief Assignment for Each Routing Metric Using Dempster–Shafer Theory

Dynamic vehicular routing demands that uncertainty in various parameters that vary be quantified. Once the contextual sensing matrix has been populated, COD-STAR passes into an inferential stage, where raw environmental measurements are turned into belief measures. DST offers the mathematical foundation to allocate trust across possible routing hypotheses without requiring complete probability distributions, perfectly aligning with the fluid nature of VANETs.

For each observed routing metric from the previous step, such as signal strength η_t, node longevity τ_t, vehicular density δ_t, and speed stability $\sigma(v_t)$ A discrete frame of discernment Θ is defined. This frame consists of three fundamental hypotheses and is defined as Equation (9).

$$\Theta = \{H_1: High\ Reliability, H_2: Moderate\ Reliability, H_3: Low\ Reliability\} \quad (9)$$

Each metric is independently analyzed, and a Basic Belief Assignment (BBA) function. $m_i(H_j)$ is constructed for each feature i and hypothesis j. It starts with mapping the observed value of each metric onto a predefined reliability scale with the help of monotonic transformation functions. $f_i(x)$, which are modeled to fit metric semantics. The BBA can be in the form of signal strength.

$$m_\eta(H_1) = f_\eta(\eta_t) = \frac{\eta_t - \eta_{min}}{\eta_{max} - \eta_{min}} \quad (10)$$

With η_{min} and η_{max} being empirically determined limits, the rest of the mass of the belief is distributed apportionately between H_2 and H_3 based either on inverse scale logic or predetermined split positions. To deal with ambiguity and non-deterministic situations, an uncommitted belief mass in the power set 2^Θ, is allocated to the power set $H_1 \cup H_2$, which captures partial confidence, is given as in Equation (11).

$$m_\eta(H_1 \cup H_2) = 1 - \left(m_\eta(H_1) + (H_3)\right) \quad (11)$$

This assignment scheme recognizes uncertainty in borderline measurements and does not force a categorization, but permits belief to have partial truth.

Each metric results in a belief vector that is independent and adds to the decision matrix of the current segment of the route. These vectors are kept apart to maintain the integrity of evidence prior to fusion. Chinchilla agents make use of such belief distributions in the course of the exploratory evaluation of alternative courses of action. The agent's trial memory is further enhanced with layered confidence over reliability hypotheses in lieu of relying on numeric fitness only.

To enhance consistency of beliefs across metrics, a conflict coefficient K is calculated between sources of beliefs prior to fusion in the form of Equation (12).

$$K = \sum_{A \cap B = \theta} m_i(A) \cdot m_j(B) \quad (12)$$

Large values of routing criteria discord values K signal, which is typically caused by a sudden change in the environment or data anomalies.

This conflict is used to inform the Chinchilla trail avoidance model, which agents use to avoid locations where there is conflicting information about reliability. Next, the metrics are not integrated but instead, they are independent and fulfill two important roles in COD-STAR. The next stage is first, they feed DST fusion.

They are then filled in the belief-sensitive fitness layer of Chinchilla Optimization, in which the agent scores the candidate segments with weighted belief confidence instead of deterministic cost.

The Chinchilla agents have a composite belief potential. B_k, corresponding to the path of each candidateP_k, in the form of the Equation as given in Equation (13).

$$B_k = \sum_{e_{ij} \in P_k} \omega_{ij} \cdot m_{ij}(H_1) \quad (13)$$

ω_{ij} denotes the adaptive weighting function shaped by path smoothness and local traffic entropy. The outcome is a map of probabilistic certainties distributed along structural paths, which enables agents to use inferred confidence and intuition about the environment.

The combination of Dempster-Shafer belief logic in path evaluation at the agent level makes the optimization geometry-agnostic; it is cognition-conscious. COD-STAR enables Chinchilla agents to move over belief-weighted maps instead of hard-fitness landscapes, resulting in more context-sensitive as well as uncertainty-robust routing choices.

3.4. Multi-Feature Evidence Fusion via Dempster's Combination Rule

Vehicular networks can require a combined view of heterogeneous and variable inputs to make routing decisions. Once the initial belief values of each routing metric have been assigned, COD-STAR then proceeds to a sequence of evidence consolidation steps, based on Dempster's Combination Rule.

This step balances the discontinuous belief measurements into one and composite measure of trust in each path segment. Chinchilla agents use this amalgamated belief in order to rank and follow candidate paths, and select candidates with consistent reliability cues on all the observed parameters.

Every segment e_{ij} contains several Basic Belief Assignments (BBAs) of the last stage, denoted as Equation (14)

$$m_\eta = m_\tau, m_\delta, m_\sigma \quad (14)$$

In relation to signal strength, the life of the node, inter-vehicular distance, and speed consistency, the BBA of each of the sources is mapped to a common frame of discernment $\Theta = \{H_1, H_2, H_3\}$ and the resulting maps are aligned.

In order to combine these BBAs, the rule of combination introduced by Dempster is used repeatedly to combine two sets of beliefs at a time until a final fused assignment. m^* is issued on each segment. The generalized fusion equation takes the form of Equation (15).

$$m^*(H) = \frac{1}{1-K} \sum_{A \cap B = H} m_1(A) \cdot m_2(B) \quad (15)$$

where $H \subseteq \Theta$ represents a possible hypothesis, and K is the conflict mass, defined in Equation (16):

$$K = \sum_{A \cap B = \theta} m_1(A) \cdot m_2(B) \quad (16)$$

This fusion operation preserves consistency by penalizing contradictory evidence. In scenarios where K approaches 1, the source beliefs are marked unreliable, prompting Chinchilla agents to reduce exploration pressure on those paths.

After the fusion is completed, the assignment of the hypothesis H_1 will be the reliability belief of the segment, which can be expressed using Equation (17) as a resulting assignment.

$$m^*(H_1) = Fused\ trust\ in\ high\ routing\ reliability \quad (17)$$

This value is one of the key values in the optimization trail memory that has a direct influence on the Chinchilla agent movement. Similar segments where $m^*(H_1)$ are more likely to experience convergence and serve as safe paths in the routing of packets.

Mass K that has been in conflict with the mass is converted to a segment risk index. ρ_{ij} as Equation (18):

$$\rho_{ij} = \frac{K}{1+K} \tag{18}$$

The index allows Chinchilla agents to strike a balance between trust and caution. Reliable routes where there is a high level of conflict are identified as ambivalent and searched carefully, whereas low-conflict, high-belief routes are given preference.

This fusion of beliefs is advantageous to the adaptive movement strategy developed by Chinchilla. The updated fitness function incorporates fused belief and risk awareness, as expressed in Equation (19).

$$F_{ij} = \lambda_1 \cdot \left(1 - m^*(H_1)\right) + \lambda_2 \cdot \rho_{ij} \tag{19}$$

Parameters λ_1 and λ_2 reflect the trade-off between trust preference and ambiguity avoidance. As a result, the agent trajectory becomes smoother and more informed, navigating with a dual lens of probabilistic trust and conflict mitigation.

Every Chinchilla agent has a belief-weighted history of trial, affecting the swarm behavior. The memory re-prioritizes paths in later iterations, and dynamically varies exploration pressure according to the local fusion feedback. Fusion, in this configuration, is not a passive calculation but an optimization-conscious decision filter. COD-STAR enables the Chinchilla-based routing logic to operate as a risk-aware, finely-tuned evaluation framework by merging a variety of sources of context using DST. The result is a stratified belief system that governs the membership of routes and agent actions in path search, to support the accuracy of high-mobility vehicular networks.

3.5. Dynamic Path Reliability Scoring Using Fused Beliefs

Quantification of segment-wise trust in COD-STAR is based on fused beliefs of a combination of routing indicators. Once evidence has been combined by applying Dempster's combination logic, the adaptive reliability score is given to each candidate segment in the Chinchilla population.

This reliability assessment is no longer a fixed or heuristically-based assessment but is a probabilistically weighted result based on real-time traffic dynamics and sensor-based confidence.

The final belief of high reliability is achieved through the fusion of a given edge. e_{ij}, is denoted as $m^*(H_1)$. The weight of reliability R_{ij} used to score is directly affected by this belief, as in the case of the definition of R_{ij} in Equation (20).

$$R_{ij} = \mu \cdot m^*(H_1) \tag{20}$$

The μ is a scaling coefficient to standardize the influence of different path lengths, which are not homogeneous. The weighting of reliability attributed to each of the segments gives the downstream path selection a probabilistic merit to prefer structurally trusted routes.

In cases where segments hold comparable R_{ij}, Further discrimination is introduced using a composite volatility penalty V_{ij}, derived from recent contextual inconsistency, as shown in Equation (21).

$$V_{ij} = \sigma(v_t) + \delta_{var} + \frac{1}{\eta_t} \tag{21}$$

With standard deviation of speed $\sigma(v_t)$, density change in δ_{var}, and signal strength at any moment η_t This penalty degrades segments that have varying or noisy conditions, even though there is moderate belief in reliability.

A weighted average of the reliability and volatility penalty is the dynamic reliability score. S_{ij}, which is calculated by the following Equation (22).

$$S_{ij} = R_{ij} - \lambda \cdot V_{ij} \tag{22}$$

The parameter λ modulates the impact of instability on the final score. Higher S_{ij} values indicate segments that are not only structurally trustworthy but also behaviourally consistent. These scores form the basis for Chinchilla path ranking in subsequent decision cycles.

The agents of the Chinchilla system, on considering a complete candidate path P_k, calculate an aggregate route reliability R_k as given in Equation (23).

$$R_k = \sum_{e_{ij} \in P_k} S_{ij} \tag{23}$$

It is a cumulative measure that represents the expected quality of the entire path, with beliefs at the segment level, smoothness of the environment, and continuity. When exploring swarms, agents spend more attention on those paths that have high. R_k, and focus movement on more promising solution space corridors.

The reliability score also has an impact on the local update rule of Chinchilla memory, as given in Equation (24) below:

$$M_k^{(t+1)} = \theta \cdot R_k + (1 - \theta) \cdot M_k^{(t)} \tag{24}$$

$M_k^{(t)}$ is the state of memories at iteration t and the forgetting factor, θ is denoted as 0. This updated rule guarantees behavioral change of the agent is gradual in maintaining the stability of learning, but responsive to the changes in route conditions.

With repeated crosses and updates of the agents, depending on R_k, High-reliability areas are enforced naturally. The swarm self-organizes into these areas without having to have explicit control measures. This reliability-oriented trend substitutes fixed, cost-based path-finding with self-optimizing reasoning that is attuned to reliability and flow regularity.

The reliability scoring system is a real-time behavioral compass of COD-STAR that drives the optimization process towards more informed, belief-grounded directions. This feedback layer transforms the DST fusion results into actionable navigation priorities, which closely fuse the use of evidence logic and Chinchilla agent mobility intelligence. The optimization does not focus on the shortest route but on the confident, fluid, and intelligent movement through the uncertain vehicular environments.

3.6. Chinchilla Population Encoding for Path Candidates

The optimization step of COD-STAR starts with creating a Chinchilla population, with each agent being a representative of a candidate path coded on the urban graph. This encoding codes the reliability-informed path structure to a solution vector that drives exploration and convergence. The Chinchilla encoding is also deep in terms of behavior, in contrast to the generic population-based models in which spatial memory, reliability scoring, and movement adaptability are incorporated into a single structure. All candidate paths P_k are represented as a series of interconnected segments, which are written as in Equation (25).

$$P_k = \{e_{i_1 i_2}, e_{i_2 i_3}, \dots\dots e_{i_{n-1} i_n}\} \tag{25}$$

This trail is stored as a population matrix χ, so that the rows represent the urchin. χ_k of the matrix are the sorted lists of nodes visited by the k^{th} Chinchilla. The integrity of the structure is that every transition e_{ij} is feasible on the anchor graph.

To make the interpretation more interpretable and dynamic in the search process, the reliability score S_{ij} In the previous step, is annotated on each segment of χ_k as indicated in Equation (26). This yields an extended encoding matrix.

$$\chi_k = [(v_1, S_{12}),\ (v_2, S_{23}), \dots\dots (v_n, S_{n-1,n})] \tag{26}$$

Scores, including dynamic reliability, allow agents to rank their paths internally when the reputation of their paths is being assessed and to use this embedded trust value to guide future exploration.

The first path generation uses two modes of structural shortest paths and reliability-based detours. An anchor-based route is associated with a stochastic bias function. π_{ij}, which is defined by the following Equation (27).

$$\pi_{ij} = \frac{s_{ij}}{\sum_{e_{mn} \in alt(v_i)} s_{mn}} \tag{27}$$

This value is used to select probabilistically the segments to be used at the initial stage to favor the higher quality of the edges, without fully excluding the less-explored ones. Consequently, the population that is coded is diverse and remains reliability-conscious.

At the same time, every agent has a search intensity coefficient. k_k which depends on the cumulative confidence of its path, in terms of segment, as shown in Equation (28).

$$k_k = \frac{1}{n} \sum_{e_{ij} \in P_k} S_{ij} \tag{28}$$

The coefficient controls the exploration range in the movement phase, which resembles the cautious energy-conscious movement style exhibited by chinchillas. The larger the k_k, The more exploratory boundaries are tighter, promoting deep exploration in high-confidence areas.

The path set has a uniqueness constraint to maintain the diversity of populations and minimize redundancy. A collision index χ_{km} is defined between two encoded paths P_k and P_m, as shown in Equation (29).

$$\chi_{km} = \frac{|Common\ segments|}{min(|P_k|, |P_m|)} \tag{29}$$

If χ_{km} exceeds a predefined threshold, reinitialization is triggered for one of the agents. This mechanism ensures the swarm does not prematurely converge around a narrow subset of paths.

The decisions made by the movement and the reliability of the paths are used to update the encoded paths in each iteration, depending on their historical fitness. F_k and reliability. The dynamic update of the solution vector of a specific agent is given by the formula in Equation (30).

$$\chi_k^{(t+1)} = \chi_k^{(t)} + \rho_k \cdot \Delta_k \tag{30}$$

$\Delta_k$is a mutation path based on the neighbors or historical bests, and ρ_k Does the environment volatility form an inertia term. The encoding is a mutation with graph feasibility and continuity of paths. The Chinchilla swarm turns into a context-sensitive optimization layer through the process of population encoding and honors not only environmental truth but also strategic diversity.

COD-STAR, path encoding encodes the condition with structure, but does not separate the two, but instantiates spatial geometry and dynamic trust, priming each agent to move intelligently through the reliability-optimized routing space.

3.7. Exploration of Optimal Forwarding Nodes via Foraging-Inspired CO Movement

Chinchilla Optimization is an effort to introduce a biologically inspired behavioral model by means of which agents model intelligent foraging to find optimal routing decisions. This mechanism becomes central to navigating the fused-belief landscape within COD-STAR, guiding each path candidate to evolve toward optimal next-hop selections. The foraging analogy is embedded directly into the agent's mobility strategy, where segment-level evaluations are no longer driven solely by cost or geometry but are governed by learned trust, proximity confidence, and environment-induced diversity.

The candidate path χ_k Chinchilla of each of the agents delimits its location within the search space, with the vertices constituting the path, $\{v_1, v_2, \ldots, v_n\}$. The foraging behavior commences with the identification of the viable next-hop choices at the present vertex. v_i with the constraint of the neighbors of $N(v_i)$. These candidates are evaluated on a probabilistic desirability function as in Equation (31).

$$P_{ij} = \frac{S_{ij}^{\alpha}}{\sum_{e_{im} \in N(v_i)} S_{im}^{\alpha}} \quad (31)$$

where S_{ij} is the dynamic reliability score, and α is a sensitivity constant influencing attraction sharpness. This function creates a belief-weighted movement map, directing the agent's trail toward segments with higher perceived stability.

Besides segment trust, a local density avoidance D_{ij} is added to minimize the risk of congestion, as indicated in Equation (32).

$$D_{ij} = 1 - \frac{\delta_{ij}}{\delta_{max}} \quad (32)$$

In which δ_{ij} is the local vehicle density and δ_{max} is the maximum tolerated. Adding this to desirability gives the adjusted transition probability, in the form of Equation (33).

$$\tilde{P}_{ij} = P_{ij} \cdot D_{ij} \quad (33)$$

The avoidance instinct is captured by this modulation, which resembles the Chinchilla's foraging strategies of avoiding dense or high-risk places.

The direction of foraging is not taken deterministically. Rather, the transition is sampled via roulette-wheel logic, which makes it possible to explore non-greedily and allows the swarm to cover more broadly. After choosing a next-hop vertex, the agent adds it to its path vector, updates the local history, and propagates the move, ensuring that the resulting partial path is feasible and increases cumulative confidence. An adaptive step-size modulation $\Delta_k^{(t)}$, which is influenced by the entropy in the environment and variations in the segments, is further implemented in the movement model according to the expression of Equation (34).

$$\Delta_k^{(t)} = \beta \cdot \left(1 - \frac{\sigma\left(S_k^{(t)}\right)}{\sigma_{max}}\right) \quad (4)$$

This term modulates the aggressiveness of an agent that tries to find some new nodes or optimize the existing path. The regions with low entropy cause bigger steps, and the volatile zones require more cautious with minor updates. Consequently, the swarm would inherently adjust itself to the road dynamics without the need to have central control.

When a complete path has been reconstituted with these local transitions, then the agent determines its fitness by adding the rewards based on belief and also punishing segmental inconsistency as depicted in Equation (35).

$$F_k = \sum_{e_{ij} \in \chi_k} \left(S_{ij} - \lambda \cdot V_{ij}\right) \quad (35)$$

Where,V_{ij} - Volatility score and λ the penalty rate. Increased fitness trajectories become powerful in future transitions and become elite solutions that shape it in their behavior.

Elite agents share with others their most popular paths by a limited-scale memory-sharing protocol. Shared trails are not directly copied but to bias neighbor agents through a directional reinforcement vector defined in Equation (36).

$$R_k = \theta \cdot (\chi_{elite} - \chi_k) \quad (36)$$

This bias is used to predict the next movement of an agent without compromising search individuality. Chinchilla agents will provide the desired diversity of routes with exploration that is consistent with reliability and confidence through these biologically inspired, adaptively modulated trails. The process converts next-hop selection to a group-think-based, belief-directed navigation behavior, so that COD-STAR is not just following fixed shortest routes but smartly finding the most credible paths as the vehicular conditions evolve dynamically.

3.8. Exploitation Phase Using DST-Guided Fitness Evaluation

The COD-STAR exploitation phase converts the previous exploratory learning into the process of intensifying the paths. Once the initial foraging by the Chinchilla population on the reliability landscape is over, it is time to shift attention to high-quality candidate refinement. This is done by using the combined results of belief of DST and driving them into an adaptive fitness assessment mechanism so that agents only reinforce routes that are structurally viable as well as probabilistically reliable.

The belief-guided score of each candidate path P_k, a sequence of edges, $\{e_{i_1 i_2}, e_{i_2 i_3}, \ldots\ldots e_{i_{n-1} i_n}\}$ is the sum of the segment-level reliability beliefs $m^*(H_1)$ of that path, as in Equation (37).

$$B_k = \sum_{e_{ij} \in P_k} m^*_{ij}(H_1) \tag{37}$$

This overall faith score is the sum of the evidence of the candidate pathway, combining evidence of contextual sensing, assigning belief, and multi-source fusion.

Environmental fluidity cannot be totally explained by belief alone. A route volatility penalty v_k as in Equation (38), is added in order to capture the temporal volatility, based on dynamic measures like speed inconsistency, density spikes, and connectivity disruption.

$$v_k = \sum_{e_{ij} \in P_k} \left(\sigma(v_t) + \delta_{var} + \frac{1}{\eta_t} \right) \tag{38}$$

Disruption at the link level is contained in each term. The cumulative penalty discourages unstable paths by preventing agents from using structurally sound but behaviourally unreliable paths.

The volatility-aware penalty is coupled with the fitness function defined as DST-based belief in the form of a fitness functional, as shown in Equation (39).

$$F_k = B_k - \lambda \cdot v_k \tag{39}$$

The instability is multiplied by a parameter λ, so that high-belief but high-volatility paths have their fitness weighted down. The Equation is used to direct the exploitation plan, which enhances the agent convergence along routes that strike the balance between confidence and consistency.

Every agent has a path buffer of χ_{elite} that contains paths that have been shown to perform highly in the past, so that they can be exploited locally. The current path vector χ_k is affected by the best-so-far solution chi elite to an adjustment operator directed, illustrated in Equation (40).

$$\chi_k^{(t+1)} = \chi_k^{(t)} + \phi \cdot \left(\chi_{elite} - \chi_k^{(t)} \right) \tag{40}$$

ϕ is the exploitation gain, modulated by the relative fitness difference between the elite and current path. This update guides the agent to some areas in the routing graph that are promising without swamping swarm diversity.

In exploitation, a fitness gradient ∇_{ij} of each edge is computed using a DST-based fitness gradient, which can be represented as follows, Equation (41):

$$\nabla_{ij} = \frac{\partial F_k}{\partial m^*_{ij}(H_1)} = 1 \tag{41}$$

This gradient confirms that a linear change in belief leads to a linear change in fitness, as the contribution of belief to overall fitness is linear. Agents will focus on improving the segments whose belief scores can be strengthened through re-evaluation or re-wiring their structures.

In order to achieve type 1 convergence, a mechanism of belief thresholding is imposed in Equation (42). The segments that have belief scores less than a soft threshold τ_b are not reinforced.

$$m^*_{ij}(H_1) < \tau_b \Rightarrow prune \;\; e_{ij} \tag{42}$$

This trimming operation narrows down the choice pool of paths, speeds up convergence, and concentrates the movement of agents in the areas of stable beliefs. It saves computing time since unlikely candidates are filtered out prior to the final selection step. Exploitation in COD-STAR enhances the search around high-fitness regions and retains adaptability by retaining sensitivity to DST-guided confidence. The optimization swarm acts with a discriminatory approach and tapers selectively, converging towards forwarding nodes that are long-term reliable, in dynamically sensed conditions. The combination of Chinchilla movement logic and DST certainty quantification makes exploitation an evidence-believed refinement step instead of a local search, which guarantees routing flexibility in the presence of dynamic real-world cars.

3.9. Elite Route Selection Based on Combined CO Fitness and DST Belief Score

The choice of final route in COD-STAR is the result of a stratified assessment plan which combines probabilistic confidence in belief with optimization-informed fitness. Elite validation of each path candidate in a dual-scoring system is performed after swarm-wide exploitation. This system combines the CO fitness score with a DST-based belief value, and it guarantees the representation of elite routes by the learned efficiency and situational reliability.

Every candidate path P_k is linked to the CO fitness score F_k previously computed, which is the balance of accumulated reliability and observed volatility as illustrated in Equation (43).

$$F_k = B_k - \lambda \cdot v_k \tag{43}$$

B_k represents the sum of all high-reliability beliefs and v_k represents volatility characteristics including speed variance, density spikes, and connectivity noise. The scalar λ is used to modulate the fitness penalty on path instability.

Alongside fitness, each path is assigned a global belief confidence. Ω_k, as shown in Equation (44), derived from the fusion-normalised average of segment beliefs.

$$\Omega_k = \frac{1}{n}\sum_{e_{ij}\in P_k} m_{ij}^*(H_1) \quad (44)$$

This metric provides a stability-agnostic estimation of DST-confirmed segment quality unaffected by short-term environmental disruptions. Routes with high Ω_k maintain strong DST endorsement and are likely to sustain performance over time.

To formalize elite selection, COD-STAR defines a combined selection score. ε_k for each candidate's path, as expressed in Equation (45).

$$\varepsilon_k = \alpha \cdot F_k + \beta \cdot \Omega_k \quad (45)$$

The weights α and β regulate the trade-off between environmental adaptability and long-term route certainty. Higher values of α favour exploitation-tuned paths, while higher β promotes belief consistency. This allows flexible prioritization depending on route lifespan requirements or application criticality.

All candidate routes are ranked by ε_k and the best subset made of the candidates is promoted to the elite. Diversity filtering is used to prevent a single path collapse to swarm collapse by applying a dissimilarity matrix in pairs of paths, Δ_{km}, in order to prevent swarm collapse to a single path, as in Equation (46).

$$\Delta_{km} = 1 - \frac{|shared\ segments\ between\ P_k\ and\ P_m|}{min(|P_k|,|P_m|)} \quad (46)$$

Only those paths that have a dissimilarity score greater than a predetermined threshold are kept in the elite pool.

This prevents overlapping solutions from dominating subsequent routing cycles and ensures a robust response to future topological changes.

The elite set is maintained in a global memory buffer. M_{elite}, which influences future initialization and movement decisions. During the next iteration, newly spawned Chinchilla agents use elite paths as partial templates, guiding exploratory routes toward historically trusted segments. Each segment e_{ij} appearing in more than δ_{elite} paths receive a reliability boost coefficient, as shown in Equation (47).

$$\zeta_{ij} = 1 + \gamma \cdot \frac{f_{ij}}{N_{elite}} \quad (47)$$

Where f_{ij} is the frequency of appearance, and γ is the influence factor. This coefficient scales up segment-level belief or fitness in early-stage exploration, reinforcing trust in high-frequency, high-certainty transitions.

In this reinforcement, a path rejection gate will be triggered among low-ranked candidates. A path whose ε_k falls under a soft boundary τ_e is removed off the swarm pool as given in Equation (48).

$$\varepsilon_k < \tau_e \Rightarrow remove\ P_k \quad (48)$$

This adaptive boundary changes with the perceived fitness distribution, where the selective pressure is maintained, and premature convergence is prevented. The selection of elites in COD-STAR arises due to the static ranking as well as an optimization-belief synthesis. The swarm codifies the consensus among the crowds on how to route intelligence in this way, capitalizing on previous exploitation but open to re-assessment. The last paths chosen have stable DST trust, Chinchilla-informed environmental fitness, and structural differentiation, which provides high readiness to VANET deployment.

3.10. Adaptive Path Update with Chinchilla-Converged Route

COD-STAR Path adaptation finishes with the incorporation of the Chinchilla-converged path into the active routing layer. After swarm convergence has refined a subset of elite paths based on combined belief scores and fitness values, the highest-ranking path undergoes activation for forwarding decisions. This final path reflects the optimized outcome of repeated exploratory refinement and probabilistic evaluation, maintaining a dynamically viable and contextually endorsed structure.

The consensus is reached by each agent by a convergence check with the score vector. ε_k of the elite set. The optimal scoring direction $P^*$is chosen, which is given in Equation (49).

$$P^* = arg\ arg\ \varepsilon_k \quad (49)$$

This route is the most reliable and valid course of transitions with existing mobility and reliability factors. Its structure incorporates a belief-evaluated segment. e_{ij} that have passed through multiple layers of evidence validation, environmental scoring, and agent reinforcement.

To enable the seamless adoption of P^* A time-weighted reliability reinforcement is applied to its segments by the vehicular nodes. Each segment e_{ij} in P^* is assigned an adjusted confidence score, as expressed in Equation (50).

$$\tilde{S}_{ij} = S_{ij} + \delta_t \cdot (1 - \rho_{ij}) \quad (50)$$

δ_t represents the temporal freshness of the data and ρ_{ij} is the conflicting mass from earlier DST fusion. This modified reliability will tell the longevity of the segment in the existing environment, which would help define the intervals of path re-evaluation.

The new route is stored in a route cache, and the adaptive decay model takes care of prioritization of segments. In the long run, when the route is not challenged by other, more efficient routes, the segment belief decays at a rate determined by environmental entropy. ε_t as in Equation (51).

$$\tilde{S}_{ij}(t+1) = \tilde{S}_{ij}(t) \cdot e^{-\varepsilon_t} \quad (51)$$

Higher values of ε_t accelerate the decay in unstable conditions, encouraging the swarm to revisit path construction more frequently.

As vehicles begin routing packets along P^*, new context data is captured via onboard sensors and roadside infrastructure. This data reactivates the belief assignment phase, prompting the swarm to engage in lightweight re-exploration when environmental conditions begin to deviate. Segment-wise belief is updated through a micro-adjustment rule, as expressed in Equation (52).

$$m_{ij}^*(H_1)^{(t+1)} = (1-\eta) \cdot m_{ij}^*(H_1)^{(t)} + \eta \cdot New\ Evidence_{ij} \quad (52)$$

The response rate to environmental changes is determined by the smoothing factor η. The high value results in quickness of adaptation of trust, whereas the low one remains with the remembrance of the previous trust.

A triggered re-convergence protocol is added to avoid stagnation on a long-term basis. In case performance e of P^*, steps in terms of packet delivery rate or delay index decrease to a threshold ψ_{min}, Swarm reinitialization will follow, as indicated in Equation (53).

$$\psi_{P*} < \psi_{min} \Rightarrow trigger\ CO\ re-run \quad (53)$$

This guarantees that the optimal path is maintained in changing VANET conditions, and only when performance is statistically significant does the swarm re-engage. The Chinchilla-intersecting path is the foundation of routing stability amid continual mobility. Its dynamic model of adaptive usage driven by decay, evidence refreshing, and triggered re-evaluation makes the route fluid, environment sensitive, and most likely responsive in an optimal manner. COD-STAR has a great deal of flexibility to network turbulence through permitting convergence without inflexibility, facilitated by the perpetual cycle of sense, belief propagation, and adaptation through optimization.

3.11. Real-Time Routing Recalibration with Continuous Feedback and Learning

COD-STAR maintains routing intelligence by recalibration in real time, so that the path choice can be responsive to changing network conditions of the vehicular network. After initial path deployment, environmental irregularities, fluctuating signal strengths, and node mobility can diminish the relevance of previously converged routes. To overcome such instabilities, an evolving learning system is implemented where routing measures are recalibrated with feedback streams obtained when packets are traversed.

Each active segment e_{ij} in the deployed route has a feedback registry F_{ij}, which keeps important delivery information, including: packet success rate ψ_{ij}, hop delay Δt_{ij}, and retransmission count ξ_{ij}. These measurements will be converted into a real-time reinforcement score, which is defined in Equation (54).

$$\chi_{ij} = \omega_1 \cdot \psi_{ij} - \omega_2 \cdot \Delta t_{ij} - \omega_3 \cdot \xi_{ij} \quad (54)$$

Where $\omega_1, \omega_2, \omega_3$ represent important weights and are dynamically updated based on traffic class or application sensitivity. The score ψ_{ij} serves as a continuous health indicator for the path segment, signaling whether further reinforcement or de-emphasis is necessary.

To combine this real-time signal and routing logic, the assigned belief $m_{ij}^*(H_1)$ at an assigned point in the previous step (Equation) (54) is corrected with a feedback-adjusted correction rule, as illustrated in Equation (55).

$$m_{ij}^*(H_1)^{(t+1)} = m_{ij}^*(H_1)^{(t)} + \eta_t \cdot \left(\chi_{ij} - m_{ij}^*(H_1)^{(t)}\right) \quad (55)$$

η_t is a time-dependent learning rate which is adjusted in consideration of the current network volatility. This gradual change allows the system to reconcile historical confidence with the current observations to make sure that confidence is not dismissed prematurely or stuck.

Chinchilla agents restart the exploration by updating their exploration parameters according to aggregated feedback. Agents recompute their local coefficient of reward adjustment. k_k, as indicated in Equation (56).

$$k_k = \frac{1}{|P_k|} \sum_{e_{ij} \in P_k} \chi_{ij} \quad (56)$$

This coefficient determines the motion of the agent in the next stage of exploration as the swarm moves towards the direction of amplifying the route that exhibits a good performance in the runtime. High k_k paths naturally self-reinforce in the optimization memory and lead to adaptive exploitation, without reinitiating swarm reinitiations.

To avoid overfitting to short-term data anomalies, a decay-controlled reinforcement buffer is used. The exponential moving average of the score χ_{ij} is used in order to smooth the score χ_{ij}, as given in Equation (57).

$$\chi_{ij}^{(t+1)} = \gamma \cdot \chi_{ij}^{(t)} + (1-\gamma) \cdot \chi_{new} \tag{17}$$

This mechanism is designed to give more importance to long-term performance continuity rather than short-lived bursts or failures, particularly in highly volatile VANET corridors.

A hybrid score that is a combination of DST belief and operational feedback is then incorporated in each routing decision thereafter, as shown by Equation (58), below:

$$\zeta_{ij} = \alpha \cdot m_{ij}^{*}(H_1) + \beta \cdot \chi_{ij} \tag{58}$$

Segments with higher ζ_{ij} are considered more reliable in real-time, influencing both next-hop selection and full-path re-evaluation. This fused measure becomes a live compass for Chinchilla-driven path reformation.

The model is a historical reinforcement model in which regularly changed segments across vehicles are stored in a distributed memory layer to accomplish global learning. Every segment has a cumulative trust trajectory, which is updated, as in Equation (59).

$$\Theta_{ij}(t+1) = \Theta_{ij}(t) + \delta \cdot \zeta_{ij} \tag{59}$$

The memory supports pattern recognition of long-term network behavior, enabling COD-STAR to anticipate path degradation and pre-emptively adjust its search strategy. Recalibration in COD-STAR is not just a correction process but a loop of learning, adaptation, and decision enhancement.

The constant feedback layer is used to make sure that the Chinchilla population is relevant in motion-sensitive networks and that it changes beliefs and exploration rules without compromising convergence to optimization. The routing logic can evolve, along with the landscape it is applied to.

Pseudocode 1 provides the overall functionals of COD-STAR as pseudocode.

Pseudocode 1: COD-STAR

```
InitializeNetwork()

Nodes ⟵ Deploy mobile + sink nodes
Parameters ⟵ Set node energy, mobility, MAC protocol
AnchorPaths ⟵ ExtractAnchorPaths(TopologyMap)
InitializePopulation(PopulationSize)

For each Agent ∈ Population do
Agent.Path ⟵ RandomSelect(AnchorPaths)
Agent.Memory ⟵ ∅
Agent.Fitness ⟵ 0
Agent.BeliefScore ⟵ 0

For Iteration = 1 to MaxIterations do

For each Agent ∈ Population do
ContextData ⟵ SenseData(Agent.Position)
BeliefSet ⟵ AssignDSTBeliefs(ContextData)
FusedBelief ⟵ FuseEvidence(BeliefSet)
Reliability ⟵ ScorePathBelief(FusedBelief)
Volatility
⟵ MeasureSegmentVolatility(ContextData)
Agent.Fitness
⟵ EvaluateFitness(Reliability, Volatility)
Agent.Memory
⟵ UpdateTrail(Agent.Path, FusedBelief)

EliteSet ⟵ SelectTopK(Population, k
        = ceil(0.1 * PopulationSize))
Population
⟵ ChinchillaMovement(EliteSet, Population)
If ConvergenceStable(EliteSet, Thresholds.
        ConvergenceLimit) then
        break
OptimalRoute ⟵ SelectBestRoute(EliteSet)
RouteCache.Store(OptimalRoute)
While NetworkActive() do
Feedback ⟵ CollectMetrics(OptimalRoute)
UpdateBeliefs(OptimalRoute, Feedback)
If PerformanceDrop(Feedback, Thresholds) then
return COD_STAR_Routing()
```

4. Simulation and Results

Simulation refers to the digital recreation of how a real-world system or process behaves over time. It is based on a computer-generated model created to capture the key aspects of the real system in a simplified form. Through testing this model in various conditions, researchers can monitor trends, test system reactions, and make valuable conclusions. Simulation is essential in various fields such as engineering, economics, healthcare, and social sciences, where it is used to comprehend complex systems, experiment with potential results, and assist in decision-making.

Network Simulator 3, also referred to as NS3, is an open-source, extensively used platform to simulate communication networks. It works based on discrete events, which enables the correct timing and scheduling of network events.

NS3 can be used both in academia and as a research tool, and it provides the ability to construct elaborate network environments. In-depth simulation experiments allow the users to design network structures, implement or modify communication protocols, and measure network performance metrics.

The diversity of the network types supported by NS3, including the conventional wired connection and dynamic wireless and mobile systems, makes it a useful environment to investigate the dynamics of network behavior and performance given different technical environments. The setting of the simulations is presented in Table 1.

Table 1. Simulation settings

Setting / Metric	Value / Description
Total Nodes in Network	500 nodes
Active Network Size	70 mobile nodes
Simulation Duration	100 seconds
Extended Execution Time	300 seconds
Mobility Behavior	Random Walk 2D Mobility Model
Application Behavior	Continuous data generation and transmission
Mobility Tracing	Enabled for all moving nodes
Network Coverage Area	850 meters × 850 meters
Maximum Layer Spread	≤ 150 meters per layer
Data Transmission Speed	21 kilobits per second (kbps)
Communication Bandwidth	97 Hz
Individual Packet Size	78 bytes
Initial Node Energy	1 Joule per node
Power in Idle Mode	164 milliwatts
Node Voltage Level	3.0 Volts
Number of Sink Nodes	Minimum of 4 sink nodes
Medium Access Protocol	IEEE 802.11p standard

The Packet Delivery Ratio (PDR) performance across varying node densities is tabulated in Table 2, and visually illustrated in Figure 1. COD-STAR consistently outperforms both ADR and CLR protocols across all node counts. At 50 nodes, COD-STAR achieves a peak PDR of 79.18%, while ADR, CLR, and T-ASTAR (Traditional-ASTAR) reach only 58.34%, 65.06%, and 54.77%, respectively. COD-STAR has a higher delivery rate of 64.31 with an increase in node count to 500 than ADR with 34.09%, CLR with 47.41, and T-ASTAR with 32.01%.

Table 2. Packet Delivery Ratio

No. of Nodes	ADR (%)	CLR (%)	T-ASTAR (%)	COD-STAR (%)
50	58.34	65.06	54.77	79.18
100	56.18	62.74	52.74	76.61
150	53.71	59.78	50.42	75.46
200	52.98	59.11	49.74	73.54
250	51.18	58.17	48.05	71.37
300	45.60	56.20	42.81	70.03
350	40.91	54.10	38.41	68.85
400	39.28	51.29	36.88	66.97
450	37.12	49.31	34.85	66.17
500	34.09	47.41	32.01	64.31
Average	46.939	56.317	44.066	71.249

The slow decrement in PDR of all protocols is explained by the raised network congestion and collisions. COD-STAR has a higher resilience, as it recalibrates the adaptive routes and chooses the path to follow based on beliefs. This dominance is seen in the average PDR: COD-STAR achieves a 71.249%, which is much higher than ADR with 46.939%, CLR with 56.317%, and T-ASTAR with 44.066%. This strength in dense environments underscores the increased efficiency of delivery of COD-STAR, which makes it a more reliable protocol in large-scale and dense vehicular networks.

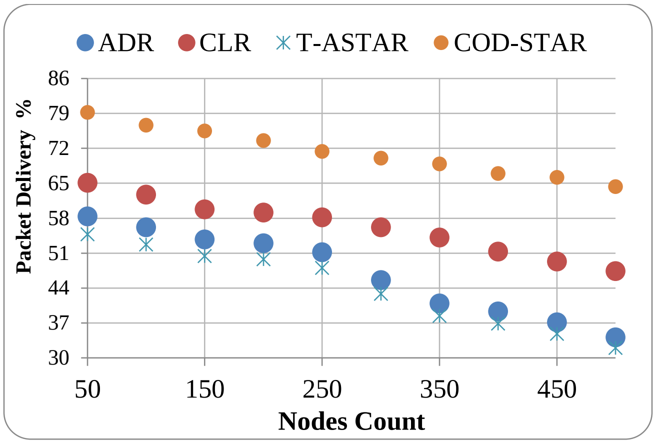

Fig. 1 Packet Delivery Ratio

Table 3 shows the Percentage Distribution of Packet Drops (PDrR) with varying node density, and Figure 2 illustrates it graphically. In all of the configurations, COD-STAR has a much lower drop rate compared to ADR, CLR, and T-ASTAR. At 50 nodes, COD-STAR records a drop of only 20.82%, while CLR and T-ASTAR report higher values of 41.66%, 34.94%, and 45.23%, respectively. As the node count increases to 500, COD-STAR maintains control with a drop ratio of 35.69%, whereas CLR and T-ASTAR escalate to 65.91%, 52.59%, and 68.00%. The rise in drop ratios as the number of nodes increases can be attributed to congestion and other effects, such as increased contention at the MAC layer.

Table 3. Packet Drop Ratio

No. of Nodes	ADR (%)	CLR (%)	T-ASTAR (%)	COD-STAR (%)
50	41.66	34.94	45.23	20.82
100	43.82	37.26	47.26	23.39
150	46.29	40.22	49.58	24.54
200	47.02	40.89	50.26	26.46
250	48.82	41.83	51.95	28.63

300	54.40	43.80	57.19	29.97
350	59.09	45.90	61.59	31.15
400	60.72	48.71	63.12	33.03
450	62.88	50.69	65.15	33.83
500	65.91	52.59	68.00	35.69
Average	53.061	43.683	55.934	28.751

The routing of COD-STAR is more stable due to its belief-oriented path correction and routing optimization at Chinchilla, minimizing the loss of packets.

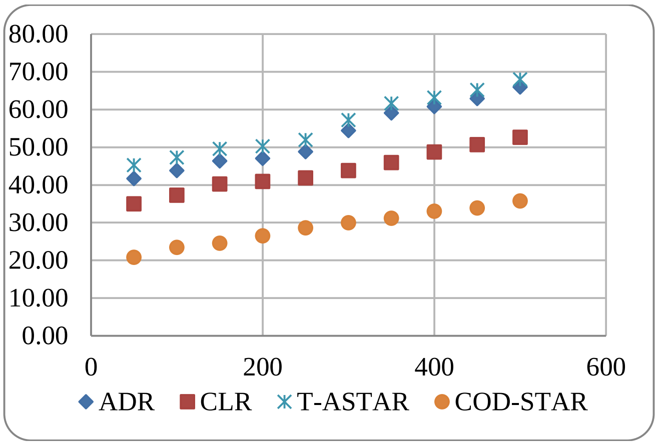

Fig. 2 Packet Drop Ratio

This advantage is further supported by the average packet drop ratio: COD-STAR has an average rate of 28.751%, which is significantly lower than the one of ADR (53.061%), CLR (43.683), and T-ASTAR (55.934**),** which justifies its efficiency in reducing losses even in high-density conditions.

Throughput analysis across varying node densities is detailed in Table 4, and visualized in Figure 3. COD-STAR consistently achieves higher throughput than ADR, CLR, and T-ASTAR, highlighting its superior data handling capability. Under the condition of 50 nodes, COD-STAR provides a throughput of 59.48 kbps, which is quite high compared to ADR at 34.41 kbps, CLR at 48.68 kbps, and T-ASTAR at 32.71 kbps. When the number of nodes grows to 500, COD-STAR increases again to 76.79 kbps, whereas CLR and T-ASTAR only attain 44.52 kbps, 57.51 kbps, and 42.32 kbps, respectively.

Table 4. Throughput analysis

Nodes Count	ADR (Kbps)	CLR (Kbps)	T-ASTAR (Kbps)	COD-STAR (Kbps)
50	34.41	48.68	32.71	59.48
100	34.88	49.26	33.16	61.82
150	35.67	50.92	33.91	64.39
200	36.87	51.22	35.05	64.70
250	37.59	52.78	35.73	69.49
300	38.34	53.65	36.45	69.84
350	42.58	55.55	40.48	70.00
400	43.22	55.66	41.08	71.13
450	43.87	56.76	41.70	73.89
500	44.52	57.51	42.32	76.79
Average	**39.195**	**53.199**	**37.259**	**68.152**

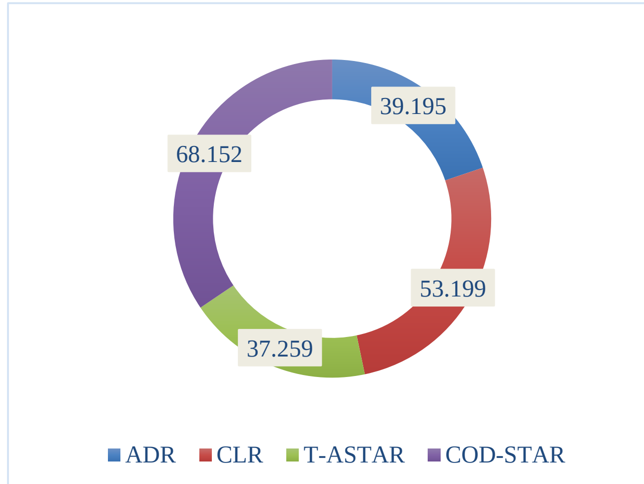

Fig. 3 Throughput

Such a constant increase in throughput with growing load proves the scalability and robustness of the protocol. The adaptive routing and dynamic recalibration mechanisms in COD-STAR, driven by Chinchilla, lead to efficient utilisation of the channel and improved data forwarding speed. On average, COD-STAR records 68.152 kbps, clearly surpassing CLR's 53.199 kbps, ADR's 39.195 kbps, and T-ASTAR's 37.259 kbps. The results confirm that COD-STAR maintains high data flow rates even in dense environments, validating its effectiveness for high-throughput demands in vehicular networks.

Table 5. Delay

Nodes Count	ADR (ms)	CLR (ms)	T-ASTAR (ms)	COD-STAR (ms)
50	12532	9958	12864	5985
100	12566	10018	12899	6022
150	12588	10325	12922	6070
200	12624	10380	12959	6093
250	12836	10526	13176	6156
300	13045	10584	13391	6194
350	13150	10702	13498	6271
400	13263	11092	13614	6300
450	13309	11352	13662	6397
500	13588	12359	13948	6628
Average	**12950**	**10730**	**13293**	**6212**

The average delay across the simulation confirms this advantage: COD-STAR is 6212 ms, and CLR, ADR, and T-ASTAR are 10,730 ms, 12,950 ms, and 13,293 ms, respectively. These results validate COD-STAR's efficiency in ensuring quicker data delivery, even under high node density and network strain.

Delay is mostly in milliseconds, which is summarised in Table 5 and shown in the Figure 4. COD-STAR shows the lowest delay with all node configurations than ADR and CLR. The delay recorded by COD-STAR at node density 50 is 5985 ms, which is significantly lower than ADR, CLR, and T-ASTAR, which are 12,532 ms, 9958 ms, and 12864 ms, respectively. COD-STAR at a node density of 500 has a moderate delay of 6628 ms, still lower than ADR, CLR, and T-ASTAR 13,588 ms, 12,359 ms, and 13948 ms. The upward trend in all protocols indicates the increasing congestion and processing time as a result of increasing communication overheads. Nevertheless, the timely recalibration of the route by COD-STAR, low retransmission, and efficient control of congestion allow a low latency.

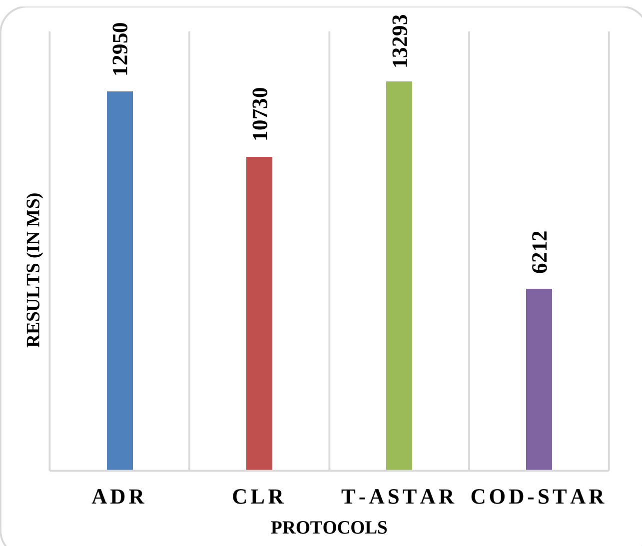

Fig. 4 Delay

Table 6 shows the energy consumption by the number of nodes; Figure 5 shows the visual representation of this table. In all of the simulation scenarios, COD-STAR shows a reduced energy consumption in comparison to ADR, CLR, and T-STAR. At 50 nodes, COD-STAR consumes 48.274 units, significantly less than ADR's 75.867, CLR's 59.787, and T_ASTAR's 74.114. When the density of nodes increases to 500, COD-STAR utilizes controlled energy consumption of 61.964 with ADR, CLR, and T-STAR surging to 91.774, 79.899, and 89.654, respectively. This trend demonstrates that COD-STAR minimizes energy consumption by optimal route finding and efficient packet routing.

The Chinchilla-based path convergence and belief-based routing minimize redundancy transmission and idle listening. COD-STAR registers 54.878 units on average energy consumption as compared to 69.306 at CLR, 82.282 at T-ASTAR, and 84.228 at ADR. The findings attest to the fact that COD-STAR can enable energy-aware communication that offers a long life of operation and sustainability in vehicle-dense networks where energy efficiency is an essential factor.

Table 6. Energy consumption

No.of nodes	ADR (%)	CLR (%)	T-ASTAR (%)	COD-STAR (%)
50	75.867	59.787	74.114	48.274
100	76.961	61.366	75.183	49.287
150	79.216	63.826	77.386	49.394
200	82.484	64.458	80.579	50.866
250	83.530	65.000	81.600	52.372
300	85.697	72.518	83.717	57.940
350	87.833	73.133	85.804	58.586
400	88.989	75.402	86.933	59.511
450	89.926	77.666	87.849	60.584
500	91.774	79.899	89.654	61.964
Average	84.228	69.306	82.282	54.878

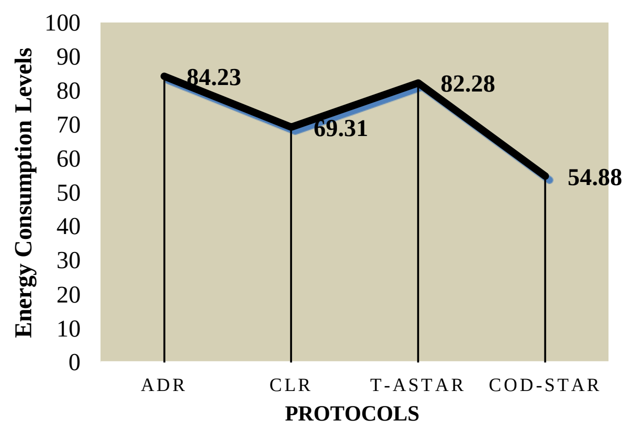

Fig. 5 Energy consumption

Table 7, and in Figure 6, the network lifetime (the time it takes for the first node to run out of energy) is given. COD-STAR shows better performance and longevity over all the node densities than ADR, CLR, and T-ASTAR. COD-STAR has a lifetime of 71.93 units at 50 nodes, which is much greater than the lifetime of CLR, ADR, and T-ASTAR: 56.58, 28.77, and 28.16, respectively. With 500 nodes, COD-STAR sticks with 54.57 units, but CLR falls to 30.24, ADR declines to 10.99, and T-ASTAR declines to only 10.76. This declining trend in ADR, CLR, and T-ASTAR is due to inefficient energy usage and increased packet loss, leading to retransmissions. COD-STAR's Chinchilla-based convergence mechanism optimizes routing decisions and conserves energy across all nodes. The average network lifetime further confirms COD-STAR's resilience, recording 64.867 units markedly higher than CLR's 41.395, ADR's 19.301, and T-ASTAR's 18.892. These results confirm the ability of COD-STAR to increase the operational time in energy-sensitive vehicular networks through load balancing in traffic and minimizing the communication overhead.

Table 7. Network lifetime analysis

No.of Nodes	ADR	CLR	T-ASTAR	COD-STAR
50	28.77	56.58	28.16	71.93
100	27.72	55.60	27.13	71.39
150	24.20	51.61	23.69	69.70
200	21.19	49.67	20.74	68.77
250	19.85	41.49	19.43	67.76
300	18.73	34.13	18.33	67.32
350	14.63	32.49	14.32	60.33
400	13.81	31.81	13.52	59.07
450	13.12	30.33	12.84	57.84
500	10.99	30.24	10.76	54.57
Average	**19.301**	**41.395**	**18.892**	**64.867**

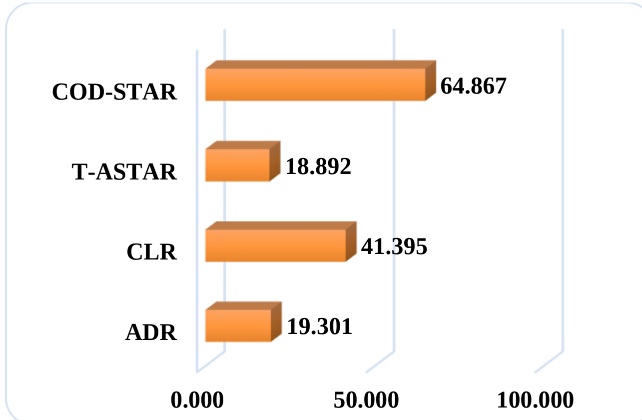

Fig. 6 Network lifetime

The control overhead of the various node numbers is tabulated and illustrated in Table 8 and graphically presented in Figure 7.

Table 8. Control overhead analysis

No.of Nodes	ADR	CLR	T-ASTAR	COD-STAR
50	0.296	0.281	0.309	0.262
100	0.300	0.285	0.314	0.266
150	0.310	0.295	0.325	0.275
200	0.318	0.302	0.333	0.282
250	0.326	0.310	0.341	0.289
300	0.330	0.313	0.345	0.292
350	0.335	0.318	0.351	0.297
400	0.339	0.321	0.354	0.3
450	0.343	0.326	0.359	0.304
500	0.347	0.329	0.362	0.307
Average	**0.324**	**0.308**	**0.341**	**0.287**

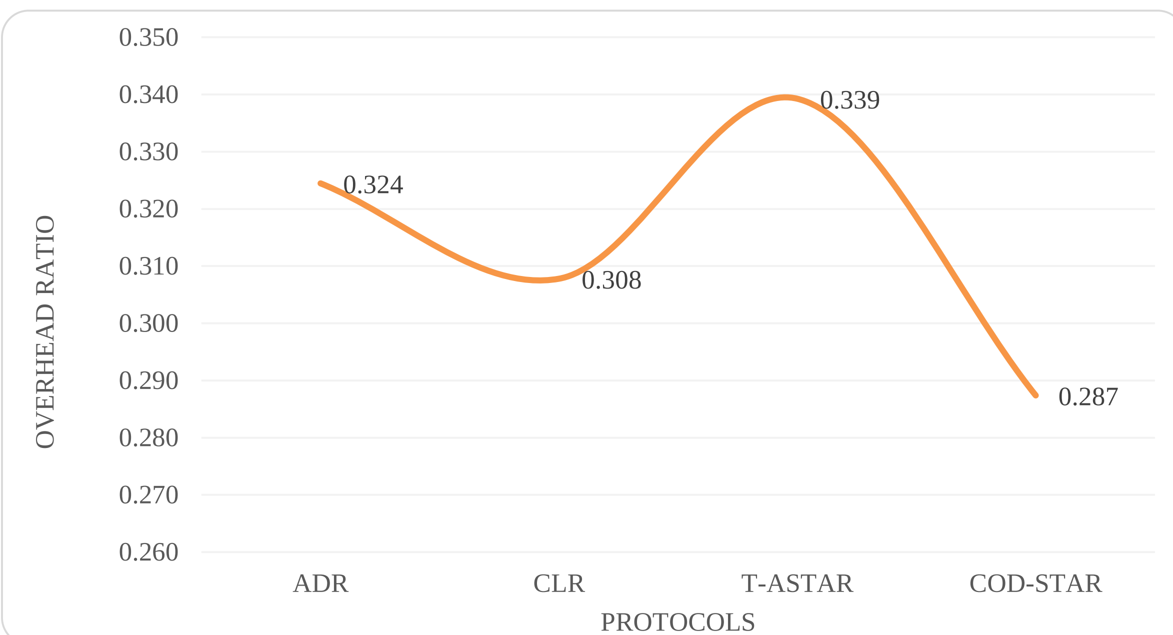

Fig. 7 Control overhead

COD-STAR has minimal control overhead in any situation, which means it has effectiveness in controlling routing-related message exchanges. COD-STAR has an overhead of 0.262 at 50 nodes, compared to higher values of 0.296, 0.281, and 0.309 in ADR, CLR, and T-ASTAR, respectively. At the node density of 500, COD-STAR remains at 0.307, but ADR, CLR, and T-ASTAR rise to 0.347, 0.329, and 0.362. This can be attributed to COD-STAR's intelligent recalibration of routes and belief-based control of broadcasts, which ensures that there is minimal propagation of control packets. The adaptive decision-making embedded in COD-STAR reduces unnecessary re-routing efforts. The average values across simulations confirm its consistent advantage, with COD-STAR at 0.287, outperforming CLR at 0.308, ADR at 0.324, and T-ASTAR at 0.341. Reduced control overhead also guarantees improved bandwidth and data flow, which supports the use of COD-STAR in high-density vehicular networks. The performance of COD-STAR outperforms the other methods like ADR and CLR as it is the combination of Chinchilla Optimization (CO) and Dempster–Shafer Theory (DST). Congestion and unstable routing paths are avoided by Chinchilla Optimization via efficient exploration and exploitation of the forwarding nodes to achieve adaptive forwarding-node selection. At the same time, DST-based belief fusion enhances the reliability of routing while addressing uncertainty in the signal strength, node stability, and vehicular density. In contrast to conventional routing techniques, which make static or limited routing decisions, COD-STAR continuously recalibrates the route based on the feedback received from the environment, thereby minimizing packet loss, delay, energy consumption, control overhead, and optimizing throughput, the packet delivery ratio, and the network lifetime. Under dynamic vehicular network conditions, COD-STAR outperforms state-of-the-art techniques thanks to this integrated routing intelligence.

5. Conclusion

COD-STAR, the proposed protocol, presents a technically robust solution to the challenges in dynamic vehicular ad hoc networks by integrating Chinchilla Optimization and Dempster–Shafer Theory. COD-STAR achieves intelligent routing through belief-based evaluation and bio-inspired exploration. The simulation outcomes confirm the superior performance of COD-STAR in delivering packets with a high delivery rate, low packet drop rate, enhanced throughput, minimized delays, limited energy expenditure, and reduced control overhead in different node density scenarios. The optimization algorithm effectively manages exploration and exploitation, whereas DST fusion guarantees trust-based decisions when network conditions are unknown. Feedback incorporation and learning processes increase the flexibility of the protocol. The cooperative migration of Chinchillas maximizes the efficiency of the protocol due to the reliability of the environment and the fusion of belief scores. The protocol outperforms competing protocols in congestion avoidance, energy saving, and delay reduction. The proposed COD-STAR protocol solves important problems associated with conventional routing methods by providing an efficient protocol for the vehicular ad-hoc networks' complex conditions. This comprehensive design contributes a significant advancement to the field of intelligent network routing under dynamic, large-scale conditions.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Funding Statement

The authors declare that no funding was received for the conduct of this research.

Author Contributions

All the authors contributed to the conceptualization and methodology. JS did data Collection and wrote the original draft of the manuscript. MJ reviewed, supervised, and did a formal analysis. RK reviewed and supervised the manuscript. All authors read and approved the final manuscript.

Data Availability Statement

The datasets generated and analyzed during the current study are not publicly available.

Research Involving Human and/or Animals

Not applicable.

AI Usage

The authors used an AI-based language tool to improve grammar and readability. The scientific content and conclusions were reviewed and validated by the authors.

References

[1] D. Antony Joseph Rajan, and E R Naganathan, "Trust Based Anonymous Intrusion Detection for Cloud Assisted WSN-IOT," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 104-108, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Pawan Singh et al., "W-GeoR: Weighted Geographical Routing for VANET's Health Monitoring Applications in Urban Traffic Networks," *IEEE Access*, vol. 10, pp. 38850-38869, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Mishri Saleh AlMarshoud, Ali H. Al-Bayatti, and Mehmet Sabir Kiraz, "Location Privacy in VANETs: Provably Secure Anonymous Key Exchange Protocol based on Self-Blindable Signatures," *Vehicular Communications*, vol. 36, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Jianhong Zhang, and Qijia Zhang, "Comment on "Secure and Lightweight Conditional Privacy-Preserving Authentication for Securing Traffic Emergency Messages in VANETs"," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1037-1038, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Mayur Jagdish Patil, and Krishnakant P. Adhiya, "Secured VANET: An Improved COOT-Algorithm-Based Optimal Routing Protocol with Multiple Authentication and Fake Message Detection for Secure Data Transmission," *Wireless Networks*, vol. 31, pp. 3315-3342, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Jianhang Liu et al., "Trajectory Prediction Training Scheme in Vehicular Ad-Hoc Networks based on Federated Learning," *Ad Hoc Networks*, vol. 178, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Xin-yu Hong, Na Lv, and Zhi-yuan Ren, "Oblivious Neighbor Discovery Algorithms in Airborne Networks with Directional Multi-Antenna," *Ad Hoc Networks*, vol. 141, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Phillip Garrad, and Saritha Unnikrishnan, "Reinforcement Learning in VANET Penetration Testing," *Results in Engineering*, vol. 17, pp. 1-7, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Sowmya Kudva et al., "Towards Secure and Practical Consensus for Blockchain based VANET," *Information Sciences*, vol. 545, pp. 170-187, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Yanwei Zhou et al., "An Efficient and Provably Secure Identity Authentication Scheme for VANET," *IEEE Internet of Things Journal*, vol. 10, no. 19, pp. 17170-17183, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] J. Ramkumar, R. Karthikeyan, and V. Valarmathi, "Alpine Swift Routing Protocol (ASRP) for Strategic Adaptive Connectivity Enhancement and Boosted Quality of Service in Drone Ad Hoc Network (DANET)," *International Journal of Computer Networks and Applications*, vol. 11, no. 5, pp. 726-748, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Tingjun Lei et al., "Graph-based Robot Optimal Path Planning with Bio-Inspired Algorithms," *Biomimetic Intelligence and Robotics*, vol. 3, no. 3, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Huahong Ma et al., "QRAVDR: A Deep Q-Learning-Based RSU-Assisted Video Data Routing Algorithm for VANETs," *Ad Hoc Networks*, vol. 171, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] Girraj Kumar Verma et al., "Escrow-Free and Efficient Dynamic Anonymous Privacy-Preserving Batch Verifiable Authentication Scheme for VANETs," *Ad Hoc Networks*, vol. 166, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[15] Samkit Jain, Vinod Kumar Jain, and Subodh Mishra, "Fuzzy-AHP based Optimal RSU Deployment (Fuzzy-AHP-ORD) Approach using Road and Traffic Analysis in VANETc," *Ad Hoc Networks*, vol. 161, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Gurjot Kaur, and Deepti Kakkar, "Fr-Aro: Secure Interference Aware Fuzzy based Clustering and Hybrid Optimization Driven Data Routing in VANETs," *Ad Hoc Networks*, vol. 151, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yonas Abate Debalki et al., "Multi-Hop Data Dissemination using a Multi-Metric Contention-Based Broadcast Suppression Strategy in VANETs," *Ad Hoc Networks*, vol. 140, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ziyan Gong, Tianhan Gao, and Nan Guo, "PCAS: Cryptanalysis and Improvement of Pairing-Free Certificateless Aggregate Signature Scheme with Conditional Privacy-Preserving for VANETs," *Ad Hoc Networks*, vol. 144, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Mohammad Naderi, and Mohammad Ghanbari, "Adaptively Prioritizing Candidate Forwarding Set in Opportunistic Routing in VANETs," *Ad Hoc Networks*, vol. 140, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] J.V.G. Ferreira et al., "Leading the Way: Reducing Network Traffic in Vehicular Ad Hoc Networks through Cluster Leader Algorithms," *Ad Hoc Networks*, vol. 178, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Guilherme Araujo, and Leobino Sampaio, "A Scalable, Dynamic, and Secure Traffic Management System for Vehicular Named Data Networking Applications," *Ad Hoc Networks*, vol. 158, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Gurjot Kaur, and Deepti Kakkar, "A Secure Lightweight Authentication Model with Interference Aware Routing and Attack Detection Approach in VANET," *Cluster Computing*, vol. 28, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Fatima Belamri, Samra Boulfekhar, and Djamil Aissani, "C-QoSRP: Cluster Based QOS-Routing Protocol for VANET in Highway Environment," *Automatic Control and Computer Sciences*, vol. 58, pp. 313-325, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] R. K. Mahesh, and Shivkumar S. Jawaligi, "VANET Cluster-based Routing Protocol with Link Breakage Handling: Introduction to Hybrid Optimization Algorithm," *Signal Image Video Process*, vol. 18, pp. 7527-7536, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Tumpa Pal, Ramesh Saha, and Suparna Biswas, "Design and Implementation of a Routing Protocol for VANET to Improve the QoS of the Network," *Journal of Network and Systems Management*, vol. 32, pp. 1-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Fatemeh Safari et al., "A Novel Cross-Layer Adaptive Fuzzy-Based Ad Hoc On-Demand Distance Vector Routing Protocol for MANETs," *IEEE Access*, vol. 11, pp. 50805-50822, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[27] A. Maria Christina Blessy, and S. Brindha, "Maximizing VANET Performance in Cluster Head Selection using Intelligent Fuzzy Bald Eagle Optimization," *Vehicular Communications*, vol. 45, 2024. [CrossRef] [Google Scholar] [Publisher Link]