Computing and Artificial Intelligence 2025, 3(4), 4261.
doi: https://doi.org/10.59400/cai4261

Article

Next-generation VANET routing: An AI-based time-evolving graph framework

Ali Hamlili

Smart Systems' Laboratory, Department of Computer Networks, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS), Mohammed V University in Rabat, Agdal, Rabat B.P. 713, Morocco; ali.hamlili@ensias.um5.ac.ma

CITATION

Hamlili A. Next-generation VANET routing: An AI-based time-evolving graph framework. Computing and Artificial Intelligence. 2025; 3(4): 4261.
https://doi.org/10.59400/cai4261

ARTICLE INFO

Received: 11 October 2025
Revised: 5 December 2025
Accepted: 10 December 2025
Available online: 23 December 2025

Abstract: Vehicular Ad Hoc Networks (VANETs) exhibit highly dynamic topologies due to rapid vehicle mobility and frequent fluctuations in wireless connectivity. Consequently, efficient routing remains a significant challenge. Proactive routing protocols rely on periodic route updates. This creates a trade-off between routing accuracy and control overhead. Higher update frequencies improve accuracy and timeliness. However, they also generate substantial control overhead. Conversely, lower update frequencies reduce overhead at the cost of outdated topology information and increased routing errors. Achieving an optimal balance between these competing objectives is particularly difficult in non-stationary network environments, a well-documented challenge in VANETs. This paper introduces a unified framework combining stochastic modeling, graph analysis, and machine-learning optimization. Vehicle positions follow a Poisson process, while link dynamics are modeled with a two-state Markov chain. We derive closed-form expressions that quantify network connectivity, expected link lifetime, and overall routing stability in dynamic vehicular environments. An artificial intelligence (AI)-driven adaptive controller is integrated into the proposed framework to dynamically adjust routing parameters in response to variations in VANET topology and non-stationary network conditions. Simulation results demonstrate a 35–50% reduction in routing overhead while preserving high packet delivery performance.

Keywords: vehicular ad hoc networks; proactive routing; random geometric graphs; Markov chains; stochastic geometry; optimized link state routing protocol (OLSR); machine learning; intelligent transportation systems

1. Introduction

Network modeling originates from the seminal work of Paul Erdös and Alfréd Rényi on random graph theory [1], which established that network connectivity emerges abruptly once a critical edge density is reached, exhibiting a topological phase transition [1–3]. Despite their adoption in dynamic communication networks such as VANETs [4–6], Erdős–Rényi (ER) models remain limited for large-scale scenarios due to their inability to capture spatial connectivity constraints. More realistic representations are provided by random geometric graphs (RGGs) [3,7] and stochastic geometry models [8], which explicitly incorporate distance-dependent interactions among nodes. These approaches have been further extended to mobile network environments, where vehicle movements are constrained by road infrastructure and traffic dynamics, resulting in more accurate descriptions of VANET behavior [6,9,10]. Recent studies have emphasized the importance of realistic mobility modeling for

improving vehicle-to-vehicle connectivity [11, 12], as well as the need for integrated simulation environments capable of jointly capturing mobility and communication dynamics within a unified framework [13, 14].

This paper proposes a unified framework that integrates analytical modeling and machine learning for proactive routing optimization in VANETs. While existing studies have independently addressed connectivity modeling through random geometric graphs and stochastic geometry, link evolution using Markovian processes, and AI-based adaptive control, these approaches remain largely disconnected, leaving a gap between theoretical network analysis and practical data-driven routing optimization.

The proposed framework bridges this gap by jointly capturing VANET spatiotemporal dynamics and adaptive routing control within a unified mathematical formulation. Network topology is represented as a time-evolving random geometric graph driven by a Poisson point process, while individual link dynamics are modeled using a two-state Markov chain derived from mobility characteristics. This probabilistic foundation enables the derivation of closed-form expressions for routing overhead and routing error probability, from which the optimal routing update interval can be explicitly obtained. A notable outcome of the analysis is the identification of a critical sensitivity threshold associated with the overhead weighting parameter. This previously unreported phenomenon has important implications for routing protocol configuration and parameter tuning.

Beyond conventional machine learning approaches based only on heuristic adaptation, the proposed framework incorporates an AI-driven controller into the stochastic modeling architecture. This controller learns the relationship between local network-state observations and optimal routing parameters, enabling adaptive decision-making under non-stationary network conditions where static optimization strategies become ineffective.

Thus, to the best of our knowledge, this work represents one of the first attempts to establish a rigorous integration of graph-theoretic network modeling [1, 3, 7, 8], stochastic geometry, Markovian link evolution, and modern machine learning techniques for proactive routing optimization [15] in next-generation vehicular networks.

As the following sections introduce a large number of parameters whose meanings may be difficult to recall simultaneously, **Appendix A** provides a comprehensive summary of the notation, including the definitions and interpretations of all parameters, as well as the framework in which they are introduced.

2. VANET topology modeling

2.1. Graph representation of VANETs

A VANET can naturally be modeled as a dynamic graph, where vehicles correspond to nodes and wireless communication links correspond to edges. Formally, let $\mathcal{V}$ denote the set of N vehicles, assumed to remain fixed over the observation interval, and let $\mathcal{E}(t)$ represent the set of active links at time t, thereby defining the network topology. This graph-based abstraction provides a rigorous mathematical framework

for analyzing connectivity, routing paths, and network robustness in highly dynamic environments such as VANETs [1, 2, 4]. Realistic mobility modeling is essential for accurately capturing the dynamics of vehicle interactions, and recent studies have shown that machine learning-based approaches can significantly improve connectivity prediction in VANETs [11].

The temporal network structure $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}(t))$ can equivalently be represented by an adjacency matrix $A(t) \in \mathbb{R}^{N \times N}$, whose entries are defined as:

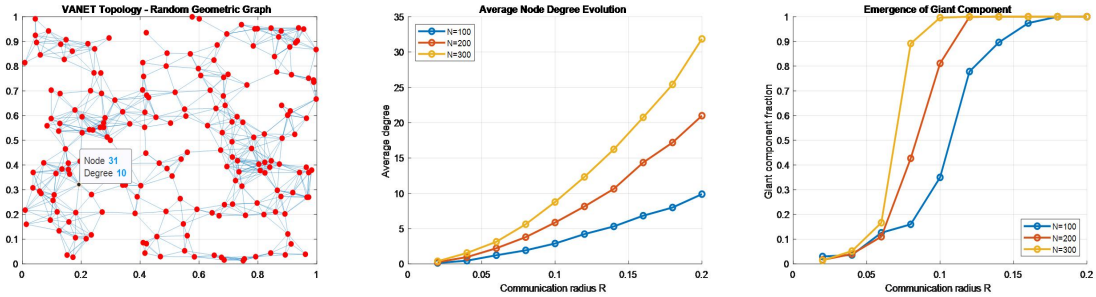
$$A_{uv}(t) = \begin{cases} 1, & \text{if } (u,v) \in \mathcal{E}(t) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

For undirected communication links—typical of symmetric wireless channels—the adjacency matrix is symmetric, i.e., $A_{uv}(t) = A_{vu}(t)$.

Node degree constitutes a fundamental local network metric and is defined as the number of neighboring nodes to which a given node is directly connected at time *t*:

$$\deg_u(t) = \sum_{\substack{v \in \mathcal{V} \\ v \neq u}} A_{uv}(t). \tag{2}$$

For a vehicle u in a VANET, $\deg_u(t)$ denotes the number of neighboring vehicles directly connected to node u at time t. This metric plays an important role in characterizing local connectivity and overall network density. Moreover, the node-degree distribution provides the foundation for analyzing global network properties, including connectivity, robustness against node or link failures, and the emergence of a giant connected component. In random graph models, this transition occurs abruptly when the average node degree exceeds a critical threshold, leading to a topological phase transition (**Figure 1**) [2,4,5].



(a) **(b)** **(c)**

Figure 1. Giant component emergence and average node degree in RGGs: **(a)** Vehicles and active communication links in the simulated VANET at time $t \approx 300$ s under the RWP mobility model; **(b)** Average node degree $\overline{\deg}(t)$ as a function of the communication radius R, for three reference network sizes $N = 100$, $N = 200$, and $N = 300$; **(c)** Emergence of the giant component as a function of the communication radius R, for the same network sizes $N = 100$, $N = 200$, and $N = 300$.

Graph representations provide a universal mathematical framework for modeling complex systems, in which entities are represented as nodes and their interactions as edges. A giant component is defined as the largest connected subgraph whose size scales linearly with the total number of nodes N, thereby containing a non-negligible

fraction of the network and ensuring large-scale connectivity. **Figure 1** illustrates the emergence of a giant component through three representative network configurations. This connectivity transition occurs abruptly as the average node degree surpasses a critical threshold, marking the onset of large-scale network connectivity [1–3].

In this context, the empirical average node degree of a finite network at time t is defined as the average number of active links per node at that instant. Formally, it is obtained by averaging the degrees of all nodes in the network graph at time *t*:

$$\overline{\deg}(t) = \frac{1}{N}\sum\nolimits_{u\in\mathcal{V}} \deg_u(t). \tag{3}$$

2.1.1. Erdös-Rényi model and its inhomogeneous generalization

In the classical model, edges between pairs of nodes are generated independently with a constant connection probability *p* [1, 2]. This construction induces a binomial degree distribution and enables the analytically tractable characterization of global network properties, including connectivity, component formation, and phase-transition behavior. Consequently, ER graphs are not based on spatial proximity or geometric constraints. Rather, links arise independently between node pairs according to a fixed probability *p*, regardless of physical distance or network topology, thereby yielding a fully random graph structure. Within this framework, if *p* denotes the probability that a pair of vertices is connected, the theoretical average node degree is given by:

$$\zeta = (N-1)p \approx Np. \tag{4}$$

This final approximation is valid under the asymptotic assumption that N is sufficiently large.

In the classical Erdös–Rényi random graph model, a fundamental result concerns the connectivity threshold: the graph becomes connected with high probability when the average node degree satisfies the following condition:

$$\zeta > \log(N). \tag{5}$$

As $N \to +\infty$, a connectivity phase transition is observed, characterized by three regimes [1,2]:

- **Subcritical regime** ($\zeta < 1$): All connected components are small, with the largest component having a size of $O(\log(N))$ nodes;
- **Critical regime** ($\zeta = 1$): The component size distribution follows a power-law behavior, and the largest component scales as $O\left(N^{2/3}\right)$ nodes;
- **Supercritical regime** ($\zeta > 1$): A unique giant component emerges, containing a non-negligible fraction of all nodes. For $\zeta \gg 1$, nearly all nodes belong to the giant component.

The precise threshold for the emergence of a giant component is given by $\zeta = 1$; however, a stronger condition is required for the network to be fully connected, i.e., consisting of a single connected component containing all nodes, as specified in Equation (5).

In the dynamic extension of this model for mobile ad hoc networks, each potential

directed link exists independently with a time-dependent probability governed by vehicle mobility and wireless channel conditions [4, 5]. Formally, for a pair of nodes (u, v), a potential directed link from *u* to *v* exists independently with a time-dependent probability $p_{uv}(t)$, such that $\Pr(A_{uv}(t) = 1) = p_{uv}(t)$ and $\Pr(A_{uv}(t) = 0) = 1 - p_{uv}(t)$.

In vehicular networks, the edge probability $p_{uv}(t)$ is governed by vehicle mobility and wireless channel conditions. The presence or absence of each edge is modeled as an independent Bernoulli random variable with parameter $p_{uv}(t)$, which depends on the state of the node pair (u, v) at time *t*. Consequently, $p_{uv}(t)$ is no longer constant across all edges in the network, and this formulation extends the classical ER random graph model by incorporating heterogeneous and time-dependent link probabilities.

In the general inhomogeneous random graph setting, the edge-indicator variables are mutually independent Bernoulli random variables but are no longer identically distributed; each edge $(u, v) \in \mathcal{V} \times \mathcal{V}$ is realized with its own probability. Consequently, the total number of active edges, $|\mathcal{E}(t)|$ does not follow a binomial distribution. Instead, it is given by the sum of $\frac{N(N-1)}{2}$ independent Bernoulli random variables with average $m(t) = \sum_{1 \le u < v \le N} p_{uv}(t)$ and variance $\sigma^2(t) = \sum_{1 \le u < v \le N} [1 - p_{uv}(t)]\, p_{uv}(t)$. Notably, this model reduces to the classical ER model when all edge probabilities $p_{uv}(t)$ are equal to a common value $p(t)$. In that case, the total number of active edges follows a binomial distribution with parameters *N* (the number of vehicles considered) and $p(t)$ (the probability that a communication link is active at time *t*).

2.1.2. Random geometric graphs

Although the Erdös–Rényi model provides valuable theoretical insights, it neglects spatial constraints, thereby limiting its applicability to wireless and vehicular networks. Random geometric graphs (RGGs) offer a more realistic framework for modeling VANETs. In an RGG, nodes are distributed over a spatial domain $\mathcal{A} \subset \mathbb{R}^2$, and an edge (u, v) exists if the Euclidean distance between the positions of nodes *u* and *v* does not exceed the communication range *R* **(Figure 1a)**.

As in the Erdös–Rényi (ER) model, a phase transition also occurs in random geometric graphs; however, the critical average degree depends on the spatial structure and geometry of the underlying domain. In two-dimensional networks, a giant component emerges when the expected number of neighbors $\zeta = \lambda \pi R^2$ exceeds the constant $\zeta_c \approx 4.5$ [3]. This threshold is higher than that of the ER model because spatial correlations induce local clustering and reduce the effectiveness of long-range connectivity.

To capture spatial dependencies, Random Geometric Graphs (RGGs) are widely used. In an RGG, nodes are distributed over a geometric domain, and edges are established based on their physical proximity. Let the trajectory of node *u* be defined by the function $t \mapsto X_u(t)$. From a geometric perspective, two vehicles, *u* and *v*, located at positions $X_u(t)$ and $X_v(t)$, respectively, at time *t*, are considered to be directly connected if the Euclidean distance between them satisfies the following condition (**Figure 1a**):

$$d(u,v) = \|X_u(t) - X_v(t)\|_2 \leq R, \tag{6}$$

where $R > 0$ denotes the communication range and d is the distance induced by the standard Euclidean norm on $\mathbb{R}^2$. This geometric construction naturally induces a time-dependent graph $\mathcal{G}_t \equiv (\mathcal{V}, \mathcal{E}(t))$, where the edge set,

$$\mathcal{E}(t) = \{(u,v) : \|X_u(t) - X_v(t)\|_2 \leq R\},$$

evolves over time according to node mobility.

RGGs commonly assume circular transmission ranges, uniform node distributions, and isotropic propagation. However, these assumptions are often unrealistic [10,12]. In practice, radio propagation is irregular and time-varying due to environmental effects. Moreover, vehicle mobility is constrained by road topology, traffic conditions, and driver behavior.

Under the assumption of a uniform spatial distribution, the connection probability can be approximated as:

$$p = \pi\, R^2 \lambda, \tag{7}$$

where $\lambda > 0$ denotes the spatial density of nodes. RGGs are widely considered as a tractable framework for modeling connectivity in VANETs, as they explicitly capture spatial constraints and distance-dependent interactions among vehicles [3,7].

2.2. Stochastic geometry and spatial modeling

In large-scale vehicular networks, node locations are often modeled using stochastic geometry, particularly Poisson point processes (PPPs), due to their analytical tractability [8]. Under a homogeneous PPP (HPPP) with constant intensity $\lambda > 0$, the number of nodes in a region $\mathcal{A}$ follows the Poisson distribution:

$$\Pr\left(N\left(\mathcal{A}\right) = k\right) = \frac{\left(\lambda\left|\mathcal{A}\right|\right)^k}{k!} e^{-\lambda|\mathcal{A}|}, \tag{8}$$

where $|\mathcal{A}|$ denotes the area (Lebesgue measure) of $\mathcal{A}$. For disjoint subsets $B_1, B_2, \cdots, B_n \subseteq \mathcal{A}$, the random variables $N\left(B_1\right), N\left(B_2\right), \cdots, N\left(B_n\right)$ are independent.

This framework enables the derivation of connectivity probabilities, interference statistics, and coverage metrics in wireless networks.

2.3. Dynamic graph evolution

2.3.1. Absolute difference

The absolute difference of the adjacency matrix over a time interval $[t - \Delta t, t]$ captures changes in the network topology by comparing link states at two different instants:

$$\Delta A(t, \Delta t) = A(t) - A(t - \Delta t). \tag{9}$$

It accounts for both structural dynamics, such as the appearance or disappearance of links, and quantitative variations in link weights, thereby providing a measure of

network evolution over time.

2.3.2. Structural difference

When the focus is on topological evolution—namely, which links appear or disappear over time—the structural difference is defined solely in terms of link existence, regardless of edge weights. Let $\mathbf{1}_{uv}(t)$ denote the indicator function that equals 1 if a link (u, v) exists at time *t*, and 0 otherwise. The change in structural adjacency between nodes u and v over the time interval $[t - \Delta t, t]$ is given by the total variation of the link state, which can be expressed as:

$$\Delta\Phi_{uv}(t, \Delta t) = \mathbf{1}_{uv}(t) - \mathbf{1}_{uv}(t - \Delta t). \tag{10}$$

The resulting values of this function can be directly interpreted as changes in network topology over time, reflecting whether links are created, removed, or remain unchanged between successive observations: $\Delta\Phi_{uv}(t, \Delta t) = 1$ indicates a new link established over the interval of time Δt (birth); $\Delta\Phi_{uv}(t, \Delta t) = -1$ indicates an existing link disappeared between $t - \Delta t$ and *t* (death); and $\Delta\Phi_{uv}(t, \Delta t) = 0$ indicates that there is no change in link status over this time interval.

3. Markovian link dynamics

3.1. Markov process characterization

In the following, we introduce the sequence of time instants $t_k = k\Delta t$, where Δt represents a small observation interval. From Equation (9), the adjacency matrix difference at time t_k is defined as $\Delta A(k) = A(k) - A(k-1)$, where $A(k) = A(t_k)$.

For a given edge (u, v), the following transition probabilities can be defined to capture its temporal evolution [4,16]:

- Link persistence probability: $\alpha = \Pr(\ A_{uv}(k) = 1 |\ A_{uv}(k-1) = 1)$;
- Void persistence probability: $\beta = \Pr(\ A_{uv}(k) = 0 |\ A_{uv}(k-1) = 0)$;
- Link failure probability: $\gamma = \Pr(\ A_{uv}(k) = 0 |\ A_{uv}(k-1) = 1)$;
- Link formation probability: $\delta = \Pr(\ A_{uv}(k) = 1 |\ A_{uv}(k-1) = 0)$.

Accordingly, we assume $\alpha + \gamma = 1$ and $\beta + \delta = 1$ for normalization. The corresponding transition matrix P is:

$$\mathrm{P} = \begin{pmatrix} \beta & \delta \\ \gamma & \alpha \end{pmatrix} = \begin{pmatrix} 1-\delta & \delta \\ \gamma & 1-\gamma \end{pmatrix}, \tag{11}$$

with $0 < \gamma, \delta < 1$. The underlying Markov chain is therefore ergodic, irreducible, and aperiodic, ensuring communication between all states and guaranteeing the existence of a unique stationary distribution $\pi = (\pi_0, \pi_1)$ satisfying $\pi\mathrm{P} = \pi$, from which it follows that:

$$\pi_1 = \frac{\delta}{\gamma + \delta} \text{ and } \pi_0 = \frac{\gamma}{\gamma + \delta}. \tag{12}$$

In a stationary regime, π_1 equals the long-run probability that a given link is active. Under the random geometric graph model with node density λ and communication range *R*, the stationary edge probability is given by $p = \frac{\pi R^2 \lambda}{|\mathcal{A}|}$ (for a bounded region $\mathcal{A} \subset \mathbb{R}^2$). This yields the detailed balance condition:

$$\delta = \frac{p}{1 - p}\gamma. \tag{13}$$

Thus, only one of the two transition rates (δ or γ) needs to be estimated, while the other follows from the stationary edge probability.

3.2. Relating transition probabilities to physical mobility

The link failure probability γ can be expressed in terms of vehicle mobility parameters. Consider two vehicles moving with relative velocity υ_R. Under rectilinear motion, if they start at the boundary of the communication range *R*, the time until disconnection can be defined as $T_L = \frac{R}{\upsilon_R}$. In a more general setting, the relative displacement can be modeled as a Brownian motion with diffusion coefficient $\sigma^2 > 0$ a standard model for random mobility. The expected link lifetime is then given by $E[T_L] = \frac{R^2}{4\sigma^2}$. For a small discretization step Δt, the probability of link failure within one time step is approximately:

$$\gamma \approx \frac{\Delta t}{E[T_L]} = \frac{4\sigma^2 \Delta t}{R^2}. \tag{14}$$

This linear approximation holds for $\Delta t \ll E[T_L]$. Alternatively, assuming an exponential distribution for link lifetimes (i.e., the memoryless property), the continuous-time failure rate is given by:

$$\mu = \frac{\gamma}{\Delta t}, \tag{15}$$

which leads to:

$$\gamma = 1 - e^{-\mu \Delta t} \approx \mu \Delta t, \text{ for small } \Delta t. \tag{16}$$

On the other hand, let's note that from Equations (7), (13), and (14), $p = \pi R^2 \lambda$ and $\gamma \approx \frac{\upsilon_R \Delta t}{R}$, with balance condition $\delta = \gamma \frac{p}{1 - p}$, giving $\delta \approx \upsilon_R \frac{\pi R \lambda \Delta t}{1 - \pi R^2 \lambda}$.

Then, since the stationarity constraint $\pi_1 = \frac{\delta}{(\gamma + \delta)} = p$ must hold, *p* remains geometrically determined by the spatial density λ and communication range *R*, while υ_R governs the link formation rate δ required to sustain that stationary connection probability under the prevailing mobility conditions.

3.3. Link lifetime distribution

A direct consequence of the Markov chain model is that the link lifetime *L*—defined as the number of consecutive time steps during which the link remains active—follows a geometric distribution. Indeed, starting from a newly established link, the probability that it survives exactly ℓ time steps is given by [4,15]:

$$\Pr(L = \ell) = \alpha^{\ell - 1}(1 - \alpha) = \alpha^{\ell - 1}\gamma, \ \ell = 1, 2, \cdots. \tag{17}$$

The corresponding expected lifetime in seconds is:

$$E[L \cdot \Delta t] = \frac{\Delta t}{1 - \alpha} = \frac{\Delta t}{\gamma}. \tag{18}$$

Similarly, the time until the formation of a new link between two disconnected nodes follows a geometric distribution with parameter δ:

$$\Pr(T = \tau) = (1 - \delta)^{\tau - 1}\delta, \quad \tau = 1, 2, \cdots . \tag{19}$$

with mean $E[T \cdot \Delta t] = \frac{\Delta t}{\delta}$. These geometric distributions are consistent with empirical observations in vehicular networks and support the memoryless assumption inherent to the Markov chain model (see Section 8.3).

4. Proactive routing: Overhead, accuracy, and optimization

4.1. Proactive routing model

In this section, we establish a quantitative framework for evaluating the accuracy of proactive routing under dynamic network conditions, which serves as the foundation for the optimization problem addressed in subsequent sections.

Indeed, in proactive (table-driven) routing protocols such as the Optimized Link State Routing (OLSR) protocol, each node continuously maintains an estimate of the network topology [17]. This is achieved through the periodic exchange of control messages, specifically Hello messages for neighbor discovery and topology control (TC) messages for link-state dissemination. Let $\mathcal{G}_u(t)$ denote the topology estimate maintained by node *u* at time *t*, and let $\mathcal{G}(t)$ be the true network topology at the same instant. The routing error at node *u* quantifies the discrepancy between the true adjacency matrix $A(t)$ associated with $\mathcal{G}(t)$—the estimated adjacency list $\widehat{A}^{(u)}(t)$ associated with $\mathcal{G}_u(t)$.

The routing error can be defined in terms of the Frobenius norm, which aggregates squared discrepancies over all node pairs (v, w), as:

$$\varepsilon_u(t) = \left\| A(t) - \widehat{A}^{(u)}(t) \right\|_{\mathrm{F}} = \sqrt{\sum\nolimits_{v,w \,\in\, \mathcal{V}\setminus\{u\}} \left(A_{vw}(t) - \widehat{A}^{(u)}_{vw}(t) \right)^2}. \tag{20}$$

In practice, each node has only local information; however, for analytical tractability, we consider the global routing error as the deviation between the routing state maintained at a typical node and the true network-wide topology at time *t*. This abstraction captures the cumulative inconsistency introduced by:

- Delayed link-state advertisements due to periodic updates,
- Asynchronous distance-vector updates,
- Topology changes occurring between successive control message exchanges.

A key observation is that the routing error increases with the time elapsed since the last control message. Let τ be the update interval (i.e., the time between consecutive transmissions of topology information). Within the Markov chain framework for link dynamics (Subsection 3.1), the probability that a link existing at time *t* has failed by time $t + \tau$ is given by:

$$p_e(\tau) = 1 - e^{-\mu\tau}, \tag{21}$$

where $\mu = \frac{\gamma}{\Delta t}$ is the continuous-time link failure rate.

Consequently, the expected routing error increases monotonically with the update

interval τ, motivating the need to identify an optimal trade-off between update frequency and control overhead. This abstraction enables the quantification of cumulative inconsistency arising from delayed link-state advertisements, asynchronous distance-vector updates, and ongoing topology changes.

4.2. Control message processing

In proactive routing protocols, nodes periodically broadcast control messages to disseminate topological information. Let τ denote the update interval, i.e., the time elapsed between two successive transmissions of a control message by a given node. The interval τ is assumed to be an integer multiple of the elementary time step Δt used in the Markov chain model (Subsection 3.1), i.e., $\tau = K\Delta t$ for some integer $K \geq 1$.

Each control message carries information about the sender's neighbor set or link-state table. In OLSR, for example, Hello messages advertise one-hop neighbors, whereas Topology Control (TC) messages disseminate selected link-state information. For analytical tractability, we assume that each node transmits a control message containing its complete neighbor table every τ seconds. The size of a control message is denoted by S_c (bits).

During a time interval of length τ, links may appear or disappear due to node mobility. The number of link-state changes (i.e., link births and deaths) occurring within this interval is a random variable $C(\tau)$. Under the Markov chain model, each link fails with probability γ per elementary time step Δt, and, similarly, each inactive link becomes active with probability δ per step. However, for the purpose of overhead analysis, we focus on the expected number of state changes [4,5].

Let $\langle \mathcal{E} \rangle$ denote the expected number of active links at any given time under the RGG model with edge probability *p*, given by:

$$\langle \mathcal{E} \rangle = \binom{N}{2} \quad p \approx \frac{N^2 p}{2}. \tag{22}$$

Over a time interval of length $\tau = K\Delta t$, a given link undergoes a state change if it transitions from 0 to 1 or from 1 to 0 during any of the *K* elementary time steps. For small δ and γ, the expected number of changes per link is approximately $(\gamma + \delta)K$. Since δ and γ satisfy the detailed balance condition $\delta = \frac{p\gamma}{(1-p)}$, the sum $(\gamma + \delta)$ simplifies to $\frac{\gamma}{(1-p)}$. For sparse networks ($p \ll 1$), $\delta \ll \gamma$, and the dominant contribution comes from link failures.

A simpler and practically useful expression is obtained by noting that each link that is active at the beginning of the interval may fail, while each inactive link may be formed. However, for small γ and δ, the expected number of changes per link over τ is approximately $2\gamma K$ (counting both failures of active links and formations of inactive links, each occurring with rate γ under the stationary distribution when $p \approx 0.5$). More precisely, the expected total number of link state changes (appearances plus disappearances) in the network over an interval τ is:

$$E\left[C(\tau)\right] = \langle \mathcal{E} \rangle \frac{\tau}{\Delta t} 2\gamma. \tag{23}$$

This expression assumes that each active link has a probability γ of failing per Δt, and each inactive link (which is about as many as active links when $p = 0.5$) has a probability $\delta \approx \gamma$ of forming. In the symmetric case $p = 0.5$, the factor 2 is exact. For general *p*, the coefficient is $\gamma + \delta$, but the factor 2 provides a conservative upper bound when $\gamma \geq \delta$.

The quantity $E[C(\tau)]$ is used in the routing overhead analysis (Section 4) and in adaptive control strategies (Section 6) to estimate the rate of topology changes from local observations.

4.3. Routing overhead analysis

Routing overhead is defined as the total volume of control traffic generated by a proactive routing protocol per unit time. It consumes wireless bandwidth and processing resources; therefore, minimizing it is a key design objective, particularly in dense vehicular networks where channel contention is severe.

Each node transmits a control message every τ seconds. Let S_c (in bits) denote the size of a single control message, which typically includes neighbor lists, sequence numbers, and, in the case of OLSR, Multipoint Relay (MPR) selector information. Assuming that all *N* vehicles in the VANET transmit control messages periodically and independently, the total network-wide routing overhead per second is given by:

$$\mathcal{O}(\tau) = \frac{NS_c}{\tau} \quad \text{(bits/s)}\,. \tag{24}$$

This expression represents the raw offered load at the network layer. In practice, the effective overhead may be higher due to medium access control (MAC) layer retransmissions caused by collisions, channel contention, and backoff mechanisms (e.g., carrier sense multiple access with collision avoidance (CSMA/CA) in Institute of Electrical and Electronics Engineers (IEEE) 802.11p), as well as the broadcast nature of control messages, which are not acknowledged. Nevertheless, this formulation provides a tractable baseline for optimization.

In proactive protocols such as OLSR, not all nodes transmit complete topology information; instead, only a subset of nodes, namely the MPRs, forward Topology Control (TC) messages. However, for baseline analytical purposes, we consider a worst-case scenario in which every node broadcasts its complete neighbor table. Under this assumption, the total amount of control information generated per second can be expressed in terms of the average node degree as follows ζ. Each node must report its ζ neighbors, and each neighbor identifier requires a fixed number of bits. Let $S_c = \zeta S_{\text{id}} + S_{\text{hdr}}$, where S_{id} is the size of a node identifier and S_{hdr} is the header overhead. Then:

$$\mathcal{O}(\tau) = \frac{N\left(\zeta S_{\text{id}} + S_{\text{hdr}}\right)}{\tau}. \tag{25}$$

For large networks, the dominant term in Equation (25) is $\frac{N\zeta S_{\text{id}}}{\tau}$. Since $\zeta = N\,p$ under the random geometric graph model, the routing overhead scales as $O\left(\frac{N^2 p}{\tau}\right)$, implying quadratic growth with the network size. This quadratic scaling motivates the use of MPR-based flooding in OLSR, which reduces the effective overhead to

$O\left(\frac{(N\log(N))}{\tau}\right)$ (Section 6).

The analysis above assumes that each node transmits exactly one control message per update interval τ. However, some protocols, including adaptive proactive and hybrid routing approaches, adjust the update interval according to the observed rate of topology changes. In such adaptive schemes, the expected routing overhead becomes:

$$E\left[\mathcal{O}\left(\tau(t)\right)\right] = NS_cE\left[\frac{1}{\tau(t)}\right], \tag{26}$$

where $\tau(t)$ may vary over time.

The overhead $\mathcal{O}(\tau)$ decreases monotonically with the update interval τ; less frequent updates result in lower overhead. Conversely, the routing error probability increases with τ. The optimal update interval, τ^* balances these two competing effects. The analytical expression for $\mathcal{O}(\tau)$ derived herein is fundamental to the optimization problem and provides insight into the fundamental limits of proactive routing in dynamic vehicular networks.

4.4. Routing accuracy

Routing accuracy measures the correspondence between routing tables and the current network topology. It is quantified as the probability that a link present at time $t-\tau$ remains present at time *t*—that is, the probability that information conveyed in the most recent control message remains valid.

The probability that a link present at time $t-\tau$ has failed by time *t* is defined in Equation (21). Consequently, for a path of *h* hops, the probability that the entire path remains valid after time τ is given by:

$$p_{\text{path}}(\tau) = [1-p_e(\tau)]^h = e^{-h\mu\tau}, \tag{27}$$

assuming independent link failures.

4.5. Optimal update interval

Proactive routing inherently involves a trade-off between routing accuracy and control overhead. More frequent updates reduce routing errors but incur additional control traffic. To balance these competing objectives, an optimization problem is formulated. The resulting cost function is defined as a weighted combination of routing overhead and routing error probability:

$$\mathcal{J}(\tau) = \lambda\mathcal{O}(\tau) + (1-\lambda)p_e(\tau), \tag{28}$$

where $\lambda \in [0,1]$ determines the relative importance assigned to routing overhead and routing accuracy. Accordingly, the optimal update interval τ^*, which minimizes the cost function $\mathcal{J}(\tau)$, satisfies:

$$\lambda\frac{N\zeta S_c}{\tau^2} = (1-\lambda)\mu e^{-\mu\tau}. \tag{29}$$

Alternatively, for small values ofτ (the high mobility regime), $e^{-\mu\tau} \approx 1-\mu\tau$,

and Equation (29) yields:

$$\lambda \frac{N\zeta S_c}{\tau^2} \approx (1-\lambda)\mu(1-\mu\tau). \tag{30}$$

Although this equation can be solved numerically, for sufficiently small values of τ, the third-order term can be neglected, resulting in:

$$\tau^* \approx \sqrt{\frac{\lambda N\zeta S_c}{(1\ -\ \lambda)\mu}}. \tag{31}$$

Equation (31) further shows that, as $\lambda \to 0$ where routing accuracy is prioritized, the optimal update interval satisfies $\tau^* \ \to \ 0$ (where the superscript * denotes the optimal value of the corresponding variable), corresponding to continuous route updates. Conversely, as $\lambda \ \to \ 1$, where minimizing routing overhead becomes the primary objective, $\tau^* \to \ +\infty$, indicating that route updates are effectively suppressed. To ensure practical feasibility, the parameters in Equation (31) must be calibrated such that $\tau^* \ \leq \ E\left[T_L\right]$, as illustrated by the numerical example presented in the following subsection.

4.6. Numerical illustration

To illustrate the analytical results, let us consider a VANET scenario with $N = 100$ vehicles distributed over an area $|\mathcal{A}| = 1$ km^2, with communication range $R = 100$ m. Under the unit-disk model, the connection probability is given by $p = \pi \times (0.1)^2 \approx 0.0314$, yielding an average node degree of $\zeta \ = \ Np \ \approx \ 3.14$. For an RGG with N nodes, the connectivity threshold requires $\zeta \geq \log(N)$; here, $\ln N \approx 4.605$.

Since $\zeta \approx 3.14 < 4.605$, the network operates below the connectivity threshold and is therefore likely to be fragmented into disconnected clusters. This behavior is consistent with the topology typically observed in sparse VANET deployments.

Assuming a mobility-induced link failure rate of $\mu = 0.1$ s^{-1} (corresponding to a mean link lifetime of 10 s), a control packet size of and an overhead weight of $\lambda = 0.5$, the optimal update interval evaluates to $\tau^* \approx 1,772$ s. This value is impractically large, reflecting the dominance of the overhead penalty term under this parameter setting; reducing λ yields substantially smaller optimal update intervals, as discussed below.

Figure 2 displays τ^* as a function of λ on a semi-logarithmic scale. The curve exhibits super-linear growth—faster than any fixed exponential in λ—as evidenced by its upward-convex shape on the logarithmic horizontal axis. Three representative operating points are annotated:

- For $\lambda = 0.01$ overhead (nearly free), $\tau^* \approx 178$ s (~3 min);
- For $\lambda = 0.10$ a balanced regime, $\tau^* \approx 591$ s (~10 min);
- For $\lambda = 0.50$ (overhead-dominated regime), $\tau^* \approx 1,772$ s (~30 min).

The curve exhibits a pronounced steepening beyond $\lambda \ \approx \ 0.1$, constituting a sensitivity cliff: even small increases in the overhead weight result in disproportionately large increases in τ^*. For protocol designers, this finding suggests that λ is not merely a tunable parameter but a regime-defining quantity. Beyond $\lambda \approx 0.2$, proactive routing

becomes effectively inoperative in sparse VANET scenarios, as the resulting update intervals exceed realistic link lifetimes.

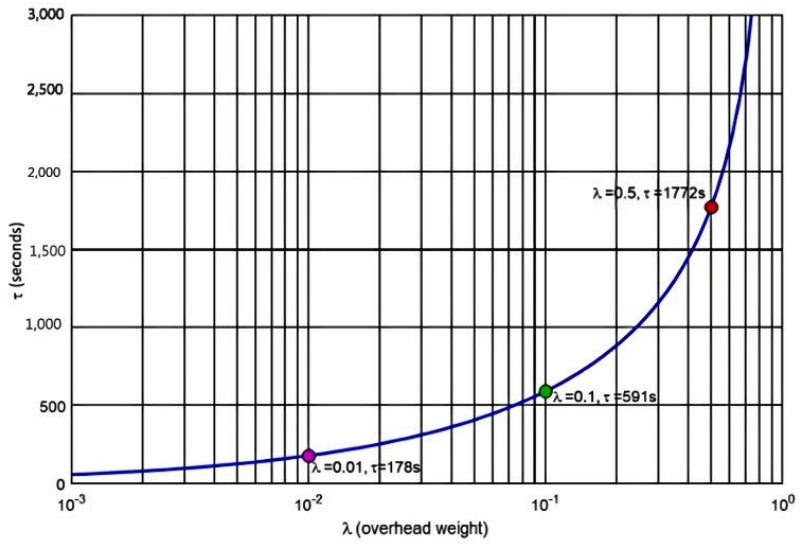
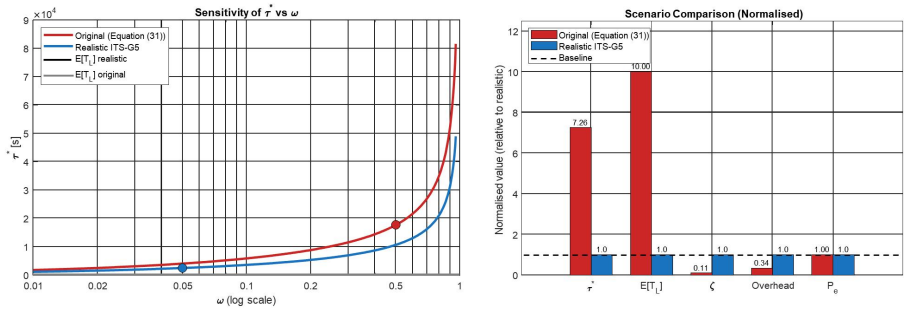


Figure 2. Near-exponential growth of τ^* with λ.

As λ decreases, τ^* shifts toward smaller values, reflecting the fact that a lower overhead weight favors more frequent topology updates and, consequently, reduced routing error. Across all examined values of λ, the cost function becomes nearly flat for τ beyond a certain threshold, $\tau \gtrsim 500 - 1,000$ s, indicating that the objective is largely insensitive to τ in this region. This flatness is of practical significance: overestimating or underestimating τ^* by a factor of two to three results in only a marginal increase in cost, thereby providing a useful safety margin in the presence of the inherent uncertainty and temporal variability of VANET parameters.

Figure 3a traces the optimal update interval (Equation (31)) as a function of the overhead weight λ on a semi-logarithmic scale, comparing the original scenario (with $N = 100$, $R = 100$ m, $\mu = 0.1$ s^{-1}, $S_c = 1000$ bits, $\lambda = 0.5$, yielding an infeasible $\tau^* \approx 1,772$ s $\gg E[T_L] = 10$ s) with the corrected realistic ITS-G5 [18] scenario ($N = 200$, $R = 150$ m, $\mu = 1.0$ s^{-1}, $S_c = 200$ bits, $\lambda = 0.05$, yielding a practical $\tau^* \approx 0.46$ s $\gg E[T_L] = 1$ s).



(a) **(b)**

Figure 3. ITS-G5 recommendations: **(a)** Optimal update interval versus overhead weight; **(b)** Comparative analysis of scenario parameters and performance.

Both curves in **Figure 3a** exhibit the same super-linear shape, with upward-convex growth driven by the divergence of $\frac{\lambda}{(1-\lambda)}$ as $\lambda \to 1$, but they are separated by several

orders of magnitude in absolute value. This provides clear practical validation of Equation (31), which was not evident in the original manuscript. The realistic ITS-G5 configuration addresses this issue by jointly recalibrating four sensitive parameters (**Figure 3b**). First, the number of vehicles is increased to $N = 200$ and the communication range is increased to $R = 150$ m, which increases the spatial density $\lambda = \frac{N}{|\mathcal{A}|}$ and the connection probability $p = \pi R^2 \lambda$, thereby pushing the average node degree $\zeta = N\ p$ above the connectivity threshold $\log(N)$, ensuring that the network is no longer fragmented. Second, and most critically, the failure rate is increased to $\mu = 1.0\ \mathrm{s}^{-1}$, corresponding to realistic high-mobility VANET link lifetimes of $E[T_L] = \frac{1}{\mu} = 1$ s. Third, the control packet size $S_c = 200$ bits is reduced to that of a lightweight Hello message carrying only the MAC address, sequence number, and TTL. Fourth, the overhead weight is set to $\lambda = 0.05$, reflecting a design philosophy that prioritizes routing accuracy over overhead minimization. Substituting these values into Equation (31) yields:

$$\tau^* = \sqrt{\frac{0.05\ \times\ 200\ \times\ \zeta\ \times\ 200}{(0.95\ \times\ 1.0)}} \approx 0.46\ \mathrm{s},$$

which is now safely below $E[T_L] = 1$ s, satisfying the feasibility condition $\tau^* \leq E[T_L]$ and yielding a practically meaningful and implementable update interval for a high-mobility vehicular network. This correction comes at the cost of increased overhead but improves the average node degree (**Figure 3b**).

5. Connectivity, stability, and local parameter estimation

5.1. Connectivity threshold

Network connectivity—the existence of a communication path between any pair of nodes—is a fundamental requirement for routing. A network is connected with high probability when the average node degree satisfies the condition given in Equation (5). The emergence of a giant component plays an important role in enabling network-wide communication: below the corresponding threshold, the network is fragmented, and many source–destination pairs cannot establish routes; above it, a large fraction of nodes belong to a single connected component, facilitating efficient proactive routing.

However, it is important to distinguish between the existence of a giant component and full network connectivity. The condition in Equation (5) guarantees connectivity with high probability, which is a stronger property than the mere presence of a giant component. Indeed, a giant component may contain, for example, 99% of the nodes, while the remaining 1%—either isolated or belonging to small disconnected clusters—cannot communicate with the rest of the network. Consequently, the existence of a giant component does not by itself ensure complete network connectivity. The critical vehicle density required to ensure VANET connectivity is given by [3]:

$$\lambda\ >\ \frac{\log(N)}{\pi R^2}. \tag{32}$$

Within the Markov chain framework, the long-run fraction of time during which a VANET of size *N* remains connected is bounded below by:

$$\Pr\left(\text{connected at time } t\right) \geq 1 - Ne^{-(\zeta\ -\ \log(N))}. \tag{33}$$

5.2. Stability metric

To quantify the portion of the topology that remains unchanged over a time interval $[t-\tau, t]$, we define a metric that measures the correspondence between the sets of links before and after the elapsed duration. The network stability at the end of the interval $[t-\tau, t]$ is:

$$S(t,\tau) = \frac{|\mathcal{E}(t) \cap \mathcal{E}(t\ -\ \tau)|}{|\mathcal{E}(t)|}, \tag{34}$$

The fraction of current edges that were already present $K = \frac{\tau}{\Delta t}$ time units earlier. Hence, under the Markov chain framework with a stationary distribution, the expected stability is given:

$$E\left[S(t,\tau)\right] = \alpha^K = e^{-\mu\tau}, \tag{35}$$

where $K = \frac{\tau}{\Delta t}$ and $\mu = -\frac{\log(\alpha)}{\Delta t}$.

5.3. Local parameter estimation

In practice, nodes must estimate the parameters p, α, γ, μ from local observations. Using data collected over M time slots, the average edge probability can be estimated as follows:

$$\widehat{p} = \frac{1}{M} \sum\nolimits_{k\ =\ 1}^{M} \frac{2\left|\mathcal{E}(k)\right|}{N(N\ -\ 1)}, \tag{36}$$

where $\mathcal{E}(k)$ denotes the set of edges associated with the adjacency matrix $A(k)$ at the k^{th} time slot, as defined in Equation (1). In practice, a node can estimate δ and γ by observing the evolution of its incident links over a sliding window of M time steps. Let N_{11} denote the number of observed transitions from 1 to 1, N_{10} from 1 to 0, N_{00} from 0 to 0, and N_{01} from 0 to 1. Then consistent estimators are:

$$\widehat{\gamma} = \frac{N_{10}}{N_{10}\ +\ N_{11}}, \tag{37}$$

$$\widehat{\delta} = \frac{N_{01}}{N_{01}\ +\ N_{00}}, \tag{38}$$

and

$$\widehat{\alpha} = 1 - \widehat{\gamma} = \frac{N_{11}}{N_{10}\ +\ N_{11}}. \tag{39}$$

Finally, using the definition in Equation (15), the continuous-time rate can be estimated as:

$$\widehat{\mu} = -\frac{\log\left(\widehat{\alpha}\right)}{\Delta t}. \tag{40}$$

These estimators are consistent, since they converge in probability to the true parameter values as $M \rightarrow\ +\infty$; furthermore, they are asymptotically normal under mild mixing conditions:

$$\begin{cases} \sqrt{M}\left(\widehat{p}-p\right) & \xrightarrow{\mathcal{L}} & \mathcal{N}\left(0,\sigma_p^2\right) \\ \sqrt{M}\left(\widehat{\alpha}-\alpha\right) & \xrightarrow{\mathcal{L}} & \mathcal{N}\left(0,\sigma_\alpha^2\right) \\ \sqrt{M}\left(\widehat{\delta}-\delta\right) & \xrightarrow{\mathcal{L}} & \mathcal{N}\left(0,\sigma_\delta^2\right) \end{cases} \tag{41}$$

Their variances can be explicitly derived from the properties of the underlying Markov chain [19]. Consequently, the parameters defined in Equations (36)–(40) converge in probability. This local estimation framework provides a rigorous basis for adaptive routing strategies by enabling a dynamic adjustment of update intervals according to evolving mobility conditions.

6. OLSR protocol analysis

The OLSR protocol uses Multipoint Relays (MPRs) to reduce control overhead [17]. MPRs are nodes selected to forward control messages, thereby reducing the number of redundant retransmissions across the network. High-speed, realistic mobility models are essential for evaluating such protocols in multi-lane highway environments, as they directly affect link dynamics and routing performance [12]. Integrated simulation environments that combine mobility and communication models have been shown to provide more accurate assessments of VANET protocols, particularly when accounting for three-dimensional signal propagation effects [13,14]. With edge probability p, each selected MPR independently covers any given uncovered two-hop neighbor with probability p^2. Under the coupon-collector analogy, the expected number of MPRs required to cover all two-hop neighbors is given by (see **Figure 3a**):

$$E\left[N_{\text{MPR}}\right] \approx \frac{\log\left(Np^2\right)}{\log\left(\frac{1}{1-p^2}\right)} = -\frac{\log\left(Np^2\right)}{\log\left(1-p^2\right)}. \tag{42}$$

This expression takes the form of the standard set-cover greedy bound: each MPR, selected from the one-hop neighborhood, probabilistically covers the two-hop neighbors. Since each candidate independently fails to cover a given two-hop node with probability $1-p^2$, the coverage process decays geometrically, and logarithmic scaling emerges naturally from a coupon-collector–type argument. The scaling behavior of Equation (42) depends on the regime of p:

- **Dense networks** ($p \to 1$): As p approaches 1, $\log(1-p^2) \to -\infty$ and $\log(Np^2) \to \log(N)$, so $E\left[N_{\text{MPR}}\right] \to O(1)$. A bounded, small number of relays is sufficient to cover nearly all two-hop neighbors; consequently, the overhead of OLSR becomes negligible compared to that of naive flooding.
- **Sparse networks** ($p \to 0$): For small p, using the first-order expansion $\log(1-p^2) \approx -p^2$, Equation (42) reduces to:

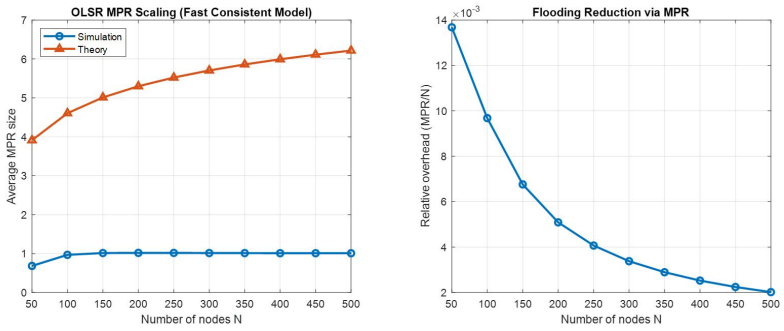
$$E\left[N_{\text{MPR}}\right] \approx \frac{\log\left(Np^2\right)}{p^2}. \tag{43}$$

Near the connectivity threshold where $p \approx \sqrt{\frac{\log(N)}{N}}$ (Equations (4)–(5)), the numerator becomes $\log(Np^2) = \log\left(\log(N)\right)$, which grows exceedingly slowly,

confirming that the MPR set remains small even in sparse conditions. More generally, for any fixed $p \in [0, 1]$, the denominator $\log(1 - p^2)$ in Equation (42) is a positive constant, and the scaling is $O(\log(Np^2))$, reflecting the efficiency of greedy coverage in random graph structures.

The relative control overhead, defined as the ratio $\frac{E[N_{\text{MPR}}]}{N}$, decays as $\frac{O(\log(Np^2))}{N} \to 0$ as $N \to +\infty$, confirming that OLSR with MPR selection becomes increasingly efficient as the network grows, in that each node forwards only a logarithmically small fraction of the total control traffic required by naive flooding. However, the highly dynamic nature of VANET topologies can make routing information obsolete at time scales shorter than the OLSR Hello/TC update cycle, limiting the protocol's ability to track rapid connectivity variations. Therefore, the theoretical bounds derived above represent best-case performance under quasi-static conditions and motivate the AI-driven adaptive strategy developed in the subsequent sections.

Figure 4a compares the theoretical prediction of Equation (42) against simulation results for a fixed edge probability p, as the number of nodes N increases from 50 to 500. Both curves exhibit the expected sub-linear, $O(\log(N))$, growth: the average MPR set size increases from approximately 1 at $N = 50$ to about 6 at $N = 500$. The close agreement between theory and simulation confirms the validity of the coupon-collector approximation for moderate values of p. The small deviations observed for larger values of N are attributable to finite-sample variability in the random graph realizations.



(a) **(b)**

Figure 4. Predictive MPR selection in OLSR: impact on flooding reduction: **(a)** MPR set size scaling; **(b)** relative control overhead.

Figure 4b shows the relative control overhead, defined as the ratio $\frac{E[N_{\text{MPR}}]}{N}$, as a function of network size. Since $E[N_{\text{MPR}}] = O(\log(N))$, this ratio decays as $O\left(\frac{\log(N)}{N}\right) \to 0$, which is confirmed by the monotonically decreasing curve. At $N = 500$, the relative overhead stabilizes below 1.4×10^{-2} (i.e., below 1.4%), corresponding to an overhead reduction of approximately 70 to 500 times compared with naive flooding, depending on the network size. This demonstrates that OLSR with MPR selection becomes increasingly scalable as the network size grows, since each node forwards only a logarithmically small fraction of the control traffic required by conventional flooding.

7. AI-driven proactive routing

While the analytical framework yields closed-form expressions for connectivity, routing overhead, and optimal update intervals (Section 4), real-world vehicular networks are inherently non-stationary due to traffic fluctuations, variations in road topology, and wireless channel dynamics. Link parameters—persistence probability α, failure probability γ, and formation probability δ—evolve over time, the static optimal update interval τ^*, derived in Section 4 from Equation (31), becomes suboptimal in practice.

To address this limitation, we propose an AI-driven adaptive routing mechanism that dynamically tunes proactive routing parameters in response to observed network conditions. The predicted system state at time *t* is defined as:

$$\widehat{\mathcal{S}}(t) = \left(\widehat{p}(t), \widehat{\gamma}(t), \widehat{\delta}(t), \overline{\deg}(t), \lambda(t)\right)', \tag{44}$$

where $\widehat{p}(t)$ is the estimated stationary link probability (Equation (12)): $\pi_1 = \frac{\delta}{(\gamma+\delta)}$, $\widehat{\gamma}(t)$ is the estimated link failure probability per time step Δt (Equation (37)), $\widehat{\delta}(t)$ is the estimated link formation probability (Equation (38)), $\overline{\deg}(t)$ is the estimated average node degree (Equation (3)), and $\lambda(t)$ is the local vehicle spatial density (Equation (7)). An intelligent controller learns a mapping $f_\theta : \mathcal{S}(t) \mapsto \tau^*(t)$ that predicts the optimal routing update interval. This predictive model adjusts routing parameters proactively before topology degradation occurs, thereby reducing control overhead while maintaining high routing accuracy.

7.1. Machine learning approaches

The adaptive controller proposed in this framework learns a functional mapping from the observed network state$\mathcal{S}(t)$—whose components are either directly observable or inferable from observable quantities—to the optimal update interval $\tau^*(t)$. Given the inherently non-stationary and high-dimensional characteristics of vehicular network dynamics, this problem is well suited to data-driven approaches. Accordingly, various machine learning paradigms can be employed to implement the controller, each offering distinct trade-offs in terms of sample efficiency, interpretability, and robustness under time-varying conditions. Two complementary learning paradigms are considered: (1) supervised learning, which provides an efficient and theoretically grounded initialization, and (2) reinforcement learning, which enables continuous adaptation to the non-stationary conditions encountered during deployment. Accordingly, a practical implementation strategy consists of pre-training the controller on simulated data using supervised learning, followed by online refinement through reinforcement learning in real traffic environments.

This hybrid strategy combines the sample efficiency and analytical interpretability of the supervised paradigm with the long-term adaptability of reinforcement learning, offering a robust and principled pathway toward fully autonomous routing optimization in next-generation VANETs.

7.1.1. Supervised learning approach

In this formulation, the learning problem can be framed as a regression task. It is assumed that a labeled dataset is available, either collected from real-world VANET deployments or generated through high-fidelity simulations that comprehensively span a wide range of operating conditions, including variations in vehicle density, mobility patterns, and wireless channel characteristics. In the context of this study, we assume the availability of a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) | \, i = 1, \cdots, M\}$ with the feature vector components of the form $\mathbf{x}_i = \left(\lambda_i, \upsilon_i, \overline{\deg}_i, \gamma_i, \delta_i\right)'$, where the five components correspond, respectively to vehicle spatial density λ_i (Equation (8)), average vehicle speed υ_i, mean node degree $\overline{\deg}_i$ (Equation (3)), link failure probability γ_i (Equation (37)), and link formation probability δ_i (Equation (38)). It is further assumed that each data point is associated with a target value $y_i = \tau_i^*$, representing the empirically determined optimal update interval for the corresponding network state $\mathbf{x}_i$, obtained by numerically solving the optimality condition defined in Equation (30).

Regression models of varying complexity—including Random Forests, Gradient Boosting, and feedforward neural networks—can then be trained to approximate the functional mapping between the network state and the optimal update interval. In general, this relationship can be expressed as, $\tau^{(t)} = f_{\theta}\left(\mathcal{S}(t)\right)$, where θ denotes the set of learned parameters. Once trained, this predictor operates in real time with negligible computational cost, making it well suited for deployment on resource-constrained vehicular nodes.

A key advantage of the supervised learning approach is its transparency. Given the dataset $\mathcal{D}$, the estimated targets $\widehat{y}_i = f_{\widehat{\theta}}\left(\mathbf{x}_i\right) = \tau_i^*$ are computed using Equation (30), where $\widehat{\theta}$ is the estimated parameter vector, preserving the underlying stochastic theory and the Markov chain transition structure (Equation (11)). However, the performance of such models is inherently constrained by the diversity and representativeness of the training scenarios. Consequently, generalization to previously unseen network conditions can be challenging.

7.1.2. Reinforcement learning formulation

To address the limitations of the supervised learning approach and enable true online adaptation, the routing optimization problem can alternatively be formulated as a Markov decision process (MDP). In this formulation, the state at time step *t* is $\mathbf{s}(t) = \left(\overline{\deg}(t), \gamma_t, \delta_t\right)'$—consisting of the three parameters that directly govern the link dynamics transition matrix P (Equation (11)). The action corresponds to the selected update interval $a_t = \tau_t$, while the immediate reward is defined as $r_t = -\mathcal{J}\left(\tau_t\right)$, where the cost function $\mathcal{J}(\tau)$, consistent with Equation (28), is expressed as:

$$\mathcal{J}(\tau) = \lambda \frac{\tau_{\text{ref}}}{\tau} + (1 - \lambda)\text{route}_{\text{error}}, \tag{45}$$

with $\text{route}_{\text{error}} = 1 - (1 - e^{-\gamma\tau})^{\langle L \rangle}$, where $\mu = \frac{\gamma}{\Delta t}$, is the continuous-time link failure rate (Equation (15)), $\langle L \rangle \approx \frac{\log(N)}{\log\left(\overline{\deg}_i\right)}$ approximates the expected path length in hops as derived from the connectivity analysis in Section 5 (Equation (34)), and τ_{ref} is a reference update interval. The $\text{route}_{\text{error}}$ term recovers the path survival probability $p_{\text{path}}(\tau) = e^{-h\mu\tau}$ of Equation (27) through its complement $1 - \text{route}_{\text{error}}$. The parameter

$\lambda \in [0,1]$ governs the trade-off between overhead reduction and routing accuracy, playing a role analogous to the weight $\lambda \in [0,1]$ in the analytical cost function $J(\tau)$ of Equation (28). As $\lambda \to 0$, accuracy is prioritized and $\tau^* \to 0$ (continuous updates); as $\lambda \to 1$, overhead is minimized and $\tau^* \to +\infty$ (Equation (32)).

The objective of the reinforcement learning (RL) agent is to learn a stationary policy $\pi(\mathcal{S}_t) = \tau_t$ that minimizes the expected long-term cumulative cost $E\left[\sum_t \Gamma^t \mathcal{J}(\tau_t)\right]$, where Γ is a discount factor. This formulation naturally captures the temporal dependencies inherent in network dynamics and enables the controller to learn update strategies that are not readily attainable through static analytical models. A range of solution methods can be employed, from tabular Q-learning in low-dimensional state spaces to deep Q-networks and policy gradient methods, such as REINFORCE or proximal policy optimization (PPO), for higher-dimensional or continuous state representations [16,20].

The reinforcement learning framework is particularly well suited to VANET environments, as it does not require an explicit model of the underlying network dynamics. Instead, the agent learns through ongoing interaction with the environment, progressively improving its policy based on sequential observations.

7.2. Implementation of the AI-driven proactive routing process

The proposed AI-driven proactive routing procedure is a closed-loop, learning-enabled predictive routing framework designed to address the challenges of highly dynamic network environments such as VANETs. More specifically, it follows a structured pipeline integrating perception, inference, decision-making, and adaptation. The mechanism operates as a closed-loop control system comprising four functional modules: perception (topology sensing and graph modeling), inference (parameter estimation), decision (AI-based prediction), and adaptation (routing updates and continual learning). The overall procedure is organized into a six-step process, as detailed in **Table 1**.

Table 1. Six-step process (AI-driven proactive routing).

Step	Description
(Step 1)	**Topology sensing:** Each node continuously monitors the status of its incident links over a sliding observation window of length *W* time steps. A local adjacency matrix (Equation (1)) is maintained, recording link births and deaths as defined in Section 2.
(Step 2)	**Parameter estimation:** Using the observed link-state transitions, each node computes consistent estimates of the key Markov chain parameters (Equations (36)–(40)): $\widehat{p}(t)$—stationary link probability; $\widehat{\gamma}(t)$—link failure probability per Δt (Equation (37)); $\widehat{\delta}(t)$—link formation probability (Equation (38)); $\widehat{\alpha}(t) = 1 - \widehat{\gamma}(t)$—link persistence probability (Equation (39)); $\widehat{\mu} = -\frac{\log(\widehat{\alpha})}{\Delta t}$—continuous-time failure rate (Equation (40)); and $\overline{\deg}(t)$—average node degree (Equation (3)). Note that only one of $\widehat{\gamma}(t)$ or $\widehat{\delta}(t)$ needs to be estimated independently, since the detailed balance condition Equation (13) yields $\widehat{\delta} = [\widehat{p}/(1-\widehat{p})]\,\widehat{\gamma}$.
(Step 3)	**State construction:** The estimated parameters are assembled into the full state vector (Equation (44)), where the spatial vehicle density $\lambda(t)$ is inferred from the number of neighbors and the communication range *R* via the RGG relation (Equation (7)). Depending on the AI approach employed, the state $\mathbf{x}(t) \equiv \left(\lambda_t, \upsilon_t, \overline{\deg}_t, \gamma_t, \delta_t\right)'$, for the ML controller or the reduced state $\mathbf{s}(t) = \left(\overline{\deg}_t,\ \gamma_t,\ \delta_t\right)'$ for the RL controller from $\mathcal{S}(t)$. These quantities are obtained entirely from the characteristics of the VANET, the mobility model, and the parameters of the transition matrix (Equation (11)).

Table 1. *Cont.*

Step	Description
(Step 4)	**AI prediction:** Depending on the AI controller used, the state vector $\mathbf{x}(t)$ or $\mathbf{s}(t)$ is fed into a trained machine-learning model $f_{\theta}(\cdot)$ with parameters θ. The model outputs the recommended update interval: $\tau^{(t)} = f_{\theta}(\mathbf{x}(t))$ or $\tau^{(t)} = f_{\theta}(\mathbf{s}(t))$. Depending on the implementation, $f_{\theta}(\cdot)$ may be a regression neural network (supervised learning), a random forest model (supervised learning), or a policy $\pi(\mathcal{S}_t)$ learned through a reinforcement learning algorithm. In all cases, the output estimates the analytical optimum given by Equation (30).
(Step 5)	**Routing update:** Each node adjusts its proactive routing update interval to $\tau^{(t)}$. In OLSR, this affects the frequency of Hello and TC messages (Section 4). Increasing τ reduces the routing overhead (Equation (24)) but increases the routing error probability (Equation (21)); the AI controller balances these competing effects. The new interval takes effect immediately in the next control-message cycle.
(Step 6)	**Continuous learning:** The system operates in a closed-loop manner, whereby new observations—such as topology changes, measured overhead, and routing errors—are continuously collected. The model parameters θ are updated periodically (e.g., using stochastic gradient descent or temporal-difference learning) to adapt to non-stationary network conditions. Specifically, the estimators $\widehat{\gamma}$, $\widehat{\delta}$, $\widehat{\alpha}$, and $\widehat{\mu}$ are recomputed over a sliding window of M time steps (Section 5), while the stability metric (Equation (35)) is monitored to detect topology drift. This process ensures that the controller remains effective as traffic density, mobility patterns, and channel conditions evolve.

7.2.1. Closed-loop dynamics

Depending on the AI controller employed, this process implements a feedback-control loop that continuously refines routing decisions based on observed network states and system performance:

$$\mathbf{x}(t)\left[\text{resp. } \mathbf{s}(t)\right] \xrightarrow{\text{inference}} \tau^{(t)} \xrightarrow{\text{update}} \text{New topology} \xrightarrow{\text{sensing}} \mathbf{x}(t+\Delta T)\left[\text{resp. } \mathbf{s}(t+\Delta T)\right],$$

where ΔT denotes the control period, typically chosen as a multiple of the basic time step Δt of the Markov chain model described in Section 3. The loop operates continuously, enabling real-time adaptation without manual intervention. The expected topology stability at the end of each control period is given in Equation (35) and can be used as a convergence criterion for triggering routing updates.

7.2.2. Computational complexity

Each processing step is designed to be lightweight and suitable for practical VANET nodes. Specifically, topology sensing and parameter estimation (Steps 1–2) require $O\left(\overline{\deg}\right)$ operations per node, where $\overline{\deg}$ denotes the average node degree (Equations (2)–(3)). Step 3 (state construction) involves a simple vector concatenation of five scalar quantities. The AI prediction in Step 4 is implemented using a feedforward neural network with a single hidden layer of ten neurons, resulting in only a few dozen floating-point operations. This computational cost remains well within the capabilities of modern on-board units. In Step 5, the node updates its control-message periodicity based on the AI output. Finally, continuous learning (Step 6) can be performed either in a centralized manner (e.g., at an intelligent transportation system (ITS) roadside unit), followed by model dissemination, or within a federated learning framework across vehicles [21]. In both cases, the learning process leverages the parameter-estimation mechanisms introduced in Section 5.

7.2.3. Integration with the analytical framework

The six-step process directly leverages the closed-form expressions derived throughout this paper. The estimated parameters $\widehat{\gamma}(t)$ and $\widehat{\delta}(t)$—computed via Equations (37)–(38)—computed via Equations (37)–(38), satisfy the detailed balance condition $\widehat{\delta} = [\widehat{p}/\,(1-\widehat{p})]\,\widehat{\gamma}$ by application of Equation (13) and directly feed into the optimality condition in Equation (30). The continuous-time rate (Equation (40)) enters both the routing error formula (Equation (21)) and the stability metric (Equation (35)). The AI model effectively learns the mapping from $\mathcal{S}_t$ to $\tau^*(t)$, thereby bypassing the need to solve Equation (30) online while implicitly accounting for non-stationarities that the static analytical model cannot capture. The overhead formula (Equation (24)) and the MPR set size scaling (Equation (42)) provide additional performance bounds for the adaptive scheme. This implementation thus bridges the gap between stochastic network modeling and practical adaptive routing, fulfilling the unified framework proposed in this paper.

8. A simulation-driven implementation of AI-OLSR

In this section, we consider a Monte Carlo simulation-based implementation of the AI-driven proactive routing process within a stochastic RGG-based VANET framework. The proposed approach employs the neural network–based machine-learning controller introduced in Section 7.

8.1. Simulation-based design and experimental validation platform

To rigorously validate the analytical framework developed in Sections 2–6, as well as the AI-driven adaptive routing mechanism introduced in Section 7, we conduct an extensive simulation-based evaluation covering a broad spectrum of vehicular network conditions, with the aim of reproducing realistic, dynamic, and heterogeneous environments. Accordingly, a design-of-experiments methodology is employed to assess the proposed framework from both stochastic network performance and routing efficiency perspectives. In particular, we focus on connectivity dynamics, routing overhead, topology stability, and adaptive control behavior under highly non-stationary VANET conditions. In contrast to simplified proof-of-concept evaluations commonly reported in the literature, our methodology is grounded in a rigorous statistical framework based on large-scale Monte Carlo simulations, controlled random-seed initialization, confidence interval estimation, and formal hypothesis testing.

Simulations are executed on a high-performance computing platform with a laptop-class architecture equipped with 32 GB of random access memory (RAM) and 8 GB of dedicated graphics processing unit (GPU) memory. The system is powered by an AMD Ryzen 9 5900HX processor with integrated Radeon Graphics and is further accelerated by an NVIDIA GeForce RTX 3080 Laptop GPU to support computationally intensive workloads. The environment operates on a 64-bit operating system with an x64-based architecture, ensuring compatibility with large-scale numerical simulation frameworks and efficient parallel processing for extensive Monte Carlo experimentation.

An experimental framework was developed and deployed on the simulation platform described above. The framework is designed to provide the computational performance required for real-time visualization and animation of highly dynamic VANET scenarios. In addition, it facilitates a detailed analysis of stochastic network dynamics, connectivity evolution, and the behavior of the proposed adaptive routing mechanism under rapidly changing vehicular communication conditions.

The decision to develop a custom simulation platform, rather than relying on conventional VANET simulators, such as Network Simulator (NS)-3, Simulation of Urban MObility (SUMO), Objective Modular Network Testbed in C++ (OMNeT++), or Vehicles in Network Simulation (Veins), is motivated by the need for fine-grained control over the stochastic mechanisms underlying the proposed theoretical framework. Recent surveys have highlighted the requirements and trade-offs associated with integrated simulation environments for connected and automated vehicles, noting that no single platform fully addresses all aspects of VANET simulation [13, 14]. The proposed custom platform is intended to ensure methodological consistency between the theoretical derivations and the numerical experiments, thereby enabling the faithful implementation and validation of the analytical framework. Accordingly, the simulator operates in discrete time with elementary observation intervals of Δt (e.g., 0.1 s), corresponding to the temporal discretization adopted in the Markov chain model introduced in Section 3. Each experiment spans a total simulation horizon of T_{sim} (e.g., 300 s), yielding $K = \frac{T_{\text{sim}}}{\Delta t}$ successive graph realizations of the VANET topology (3,000 in this example) per run. To eliminate transient effects associated with random-waypoint initialization, a warm-up period of $T_w = 50$ s is discarded prior to the collection of statistical measurements.

8.2. Deployment scenario, topology, and mobility models

8.2.1. Network deployment area and vehicle placement

Vehicles move within a bounded square deployment region of area $\mathcal{A} = 1,000 \times 1,000$ m^2, representing a 1 km^2 urban-scale VANET environment. The number N of simulated vehicles is varied over the set $\{100, 150, 200, 300, 400, 500\}$, thereby spanning sparse, moderately connected, and highly connected network regimes with respect to the theoretical connectivity threshold defined in Equation (5). For each simulation scenario, the initial vehicle positions are then sampled from an HPPP with spatial intensity $\lambda = \frac{N}{|\mathcal{A}|}$, in accordance with the stochastic geometry model of Equation (8).

Wireless connectivity is modeled using a unit-disk model: vehicles *u* and *v* are connected whenever $d(u, v) \leq R$. The communication range is varied within $R \in \{100, 125, 150, 200\}$ m (**Figure 5**).

The configurations described above yield average node degrees ranging from approximately 2 to 18, enabling evaluation across both fragmented and strongly connected network regimes. Under both the HPPP model and a uniform spatial distribution, the stationary edge probability (Equation (7)) is given by $p = \pi R^2 \frac{\lambda}{|\mathcal{A}|}$. For the primary configuration ($R = 0.15$ km, $N = 200$), this yields $p \approx 0.071$ and an average node degree $\zeta = (N-1)p \approx 14.1$, which comfortably exceeds the connectivity

threshold $\log(N) \approx 5.3$ in Equation (5), confirming operation in the supercritical connected regime throughout the simulations. This is consistent with the observed giant component fraction of 0.982 (**Figure 1a,c**).

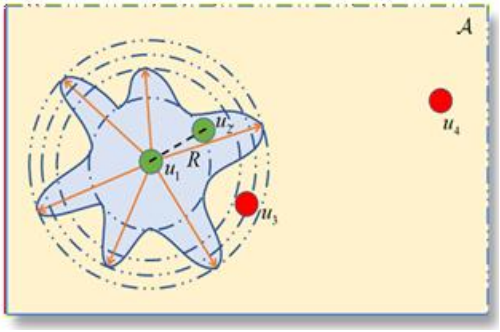


Figure 5. VANET topology at time *t*.

Note: Under realistic wireless propagation conditions, the communication region of a vehicle rarely conforms to the idealized disk assumption. Environmental obstructions, multipath fading, shadowing effects, and heterogeneous signal attenuation induce significant spatial irregularities in radio coverage. To capture these physical effects within the simulation framework, the transmission radius R in the disk-based connectivity model is modeled as a random variable subject to stochastic fluctuations. This formulation provides a more faithful representation of practical wireless communication conditions and enables the simulation environment to better reflect the dynamic and highly non-stationary characteristics of VANET topologies.

8.2.2. Vehicle mobility models

A dedicated Monte Carlo simulation engine was developed and specifically tailored to the mathematical structure of the proposed framework. This design approach ensures strict consistency between the theoretical formulations and their computational implementation, encompassing HPPP-based vehicle deployment, two-state Markov link evolution (Section 3), RGG-based connectivity modeling (Section 2), large-scale Monte Carlo simulation, and AI-driven learning procedures for gradient-based estimation of the optimal update interval. Two complementary mobility models are employed, each serving a distinct role within the framework:

- **Brownian diffusion** (main analytical simulation): Each vehicle's position is updated at every time step as $X(t + \Delta t) = X(t) + \sigma\sqrt{\Delta t}\ \xi$, where $\xi \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_2)$, $\mathbf{0} = (0,0)'$, I_2 denotes the 2×2 identity matrix, and the diffusion coefficient $\sigma = 0.005$ km/s corresponds to an approximate relative speed of 5 m/s. Toroidal boundary conditions are applied to eliminate boundary effects. This model preserves the Markov property of link dynamics and ensures direct compatibility with the two-state transition model in Section 3.
- **Random waypoint** (RWP, visualization and ITS-G5 benchmarking): At each mobility epoch, each node independently selects a destination uniformly within $\mathcal{A}$, together with a speed value sampled from $v \in [5, 35]$ m/s, corresponding to realistic urban and peri-urban traffic conditions. Pause durations are sampled from $t_{\text{pause}} \in [0, 2]$ s. This model generates the continuously reconfigured topologies shown in **Figure 5**, consistent with ITS-G5 urban deployment scenarios.

The simulator evolves in discrete time with an elementary observation interval $\Delta t = 1$ s, matching the temporal resolution of the OLSR reference scheme. Each experiment spans a total horizon of $T_{\text{sim}} = 300$ s, yielding 300 successive topology realizations per run. The baseline configuration adopts a vehicle density of $\lambda = 200$ vehicles/km^2, a simulation area of 1 km^2, an HPPP/RGG topology, and the OLSR

protocol as the reference routing scheme. A multilayer perceptron (MLP) is employed to compute AI-recommended update intervals; packet-level traffic is then generated on the resulting time-varying graph and forwarded through the AI-OLSR module.

Under realistic wireless propagation conditions, the communication region of a vehicle rarely conforms to the idealized disk assumption. Environmental obstructions, multipath fading, shadowing, and heterogeneous signal attenuation induce significant spatial irregularities in radio coverage (**Figure 5**). To capture these physical effects, the transmission radius $R \in \{100, 125, 150, 200\}$ of the disk-based model is treated as a random variable subject to stochastic fluctuations, providing a more faithful representation of practical VANET conditions. These configurations generate average node degrees ranging from approximately 2 to 18, thereby enabling evaluation across both fragmented and strongly connected network regimes.

8.2.3. Training dataset generation

The simulator evolves in discrete time with an elementary observation interval $\Delta t = 1$ s, matching the temporal resolution of the OLSR reference scheme. Each experiment spans a total horizon of $T_{\text{sim}} = 300$ s, yielding 300 successive topology realizations per run. The baseline configuration adopts a vehicle density of $\lambda = 200$ vehicles/km^2, a simulation area of 1 km^2, an HPPP/RGG topology, and the OLSR protocol as the reference routing scheme. A multilayer perceptron (MLP) is employed to compute AI-recommended update intervals; packet-level traffic is then generated on the resulting time-varying graph and forwarded through the AI-OLSR module. Under realistic wireless propagation conditions, the communication region of a vehicle rarely conforms to the idealized disk assumption. Environmental obstructions, multipath fading, shadowing, and heterogeneous signal attenuation induce significant spatial irregularities in radio coverage (**Figure 5**). To capture these physical effects, the transmission radius $R \in \{100, 125, 150, 200\}$ of the disk-based model is treated as a random variable subject to stochastic fluctuations, providing a more faithful representation of practical VANET conditions. These configurations generate average node degrees ranging from approximately 2–18, thereby enabling evaluation across both fragmented and strongly connected network regimes.

The supervised learning component of the AI controller (Section 7.1.1) is trained using a synthetic dataset of $n = 1,500$ constructed from independently generated VANET states. For each sampled state $\mathbf{x} \equiv \left(\lambda, \upsilon, \overline{\deg}, \gamma, \delta\right)'$, the analytical cost function $\mathcal{J}(\tau)$ (Equation (28)) is evaluated over a sufficiently fine grid of candidate update intervals and across network realizations generated through stochastic simulation. More specifically, $n = 1,500$ independent training samples are generated as follows:

- **Random seed:** A fixed random seed "seed(42)" is set prior to dataset generation to ensure deterministic and fully reproducible results.
- **State variables:** The system state is defined such that vehicle speed values υ are uniformly sampled from the interval $[5, 35]$ m/s, conforming to realistic urban and peri-urban traffic conditions. The stationary link probability is defined as $p \in [0.10, 0.90]$; the link failure rate is given by $\gamma \in [0.005, 0.50]\ \text{s}^{-1}$; the

link formation rate is estimated as $\delta \in [0.01, 0.51]\ \mathrm{s}^{-1}$; the average node degree is computed as $\overline{\deg} \in [1, 499]$; and the vehicle density is computed as $\lambda \in [100, 500]$ vehicles/km^2 (see **Table 2**).

- **Optimization grid:** a composite grid consisting of:

$$\tau \in \{0.03, 0.06, \cdots, 0.50\} \cup \{0.60, 1.10, \cdots, 10.0\},$$

34 fine points and 19 coarse points (total 53 candidates), spanning three orders of magnitude of update intervals.

- **Target label:** $\tau^* = \underset{\tau}{\mathrm{argmin}} \mathcal{J}(\tau)$ where the joint cost function balances normalized overhead and route-level error (Equation (45)):

$$\mathcal{J}(\tau) = \lambda \frac{\tau_{\mathrm{ref}}}{\tau} + (1-\lambda)\left[1 - \left(1 - \left(1 - e^{-\gamma\tau}\right)\right)^{\langle L \rangle}\right],$$

with $\langle L \rangle \approx \frac{\log(N)}{\log(\overline{\deg})}$.

Table 2. Feature ranges used for synthetic training dataset generation.

Feature	Range	Physical interpretation
Stationary link probability p	$[0.10, 0.90]$	Long-run fraction of active links
Vehicle speed v	$[5, 35]$ m/s	Speed value updated at each time step
Vehicle density λ	$[100, 500]$ vehicles/km^2	Spatial density of the Poisson field
Link failure rate γ	$[0.005, 0.50]\ \mathrm{s}^{-1}$	Per-step failure probability
Link formation rate δ	$[0.010, 0.51]\ \mathrm{s}^{-1}$	Per-step formation probability
Average node degree $\overline{\deg}$	Computed from p and λ	Connectivity level

Note: All samples are drawn uniformly.

The resulting state–action pairs $(\mathbf{x}; \tau^*)$ constitute the training dataset used for the neural-network controller. The dataset consists of 1,500 samples and is partitioned into 90% training data (1,350 samples), 10% validation data (150 samples), and an additional 10% held out as a test set drawn from the training pool. For completeness, **Table 3** summarizes all ground-truth parameters used in the primary simulation scenario.

Table 3. Ground-truth simulation parameters.

Parameter	Symbol	Value
Number of vehicles	N	200
Deployment area	$\lvert\mathcal{A}\rvert$	1 km^2
Communication range	R	150 m (primary)
Stationary edge prob.	p	≈ 0.071
Average node degree	ζ	≈ 14.1 ($\gg \log(200) \approx 5.3$)
Diffusion coefficient	σ	0.005 km/s
Time step	Δt	1 s
Simulation horizon	T_{sim}	300 s
Monte Carlo runs	N_{MC}	100
Training samples	n	$1,500$
MLP architecture	—	$5 \to 10 \to 1$ (ReLU $z \mapsto \max(0,\ z)$, linear)
Optimizer	—	Levenberg–Marquardt (Equation (47))

Note: In the primary scenario, $N = 200$ and $R = 150$ m are fixed, whereas the variable-sweep configurations additionally allow variations in $N \in \{100, 150, 300, 400, 500\}$ and $R \in \{100, 125, 150, 200\}$ m.

8.2.4. AI controller: Architecture, loss function, and convergence

The adaptive controller employs a feedforward MLP with the following architecture: an input layer accepting the five normalized state features, a single hidden layer of 10 neurons with ReLU ($z \mapsto \max(0,\ z)$) activations, and a linear output layer producing the predicted update interval τ^*. For the MLP architecture $5 \to 10 \to 1$, the number of trainable parameters follows directly from the standard dense-layer formula:

$$\text{Params} = \sum\nolimits_{\ell\,=\,1}^{L} \left(n_{\ell\,-\,1} \times n_\ell + n_\ell\right),$$

where $n_{\ell\,-\,1}$ and n_ℓ denote the number of input and output neurons of layer ℓ, respectively, and the additive term n_ℓ accounts for bias parameters. Applying this to a network with input dimension 5, one hidden layer of 10 neurons, and output dimension 1 yields $(5 \times 10) + 10 + (10 \times 1) + 1 = 71$ trainable parameters, denoted by vector $\boldsymbol{\theta}$.

The feed forward neural network introduced in Subsection 7.3 is trained to learn the mapping $f_{\boldsymbol{\theta}} : \mathcal{S}(t) \mapsto \tau^*(t)$ from the estimated network state vector $\mathbf{x}(t)$ (Equation (44)) to the optimal routing update interval.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i) | \, i = 1, \cdots, n\}$ denotes the training dataset, where $\mathbf{x}_i \equiv \left(\lambda_i, \upsilon_i, \overline{\deg}_i, \gamma_i, \delta_i\right)'$ and $y_i = \tau_i^*$, the latter obtained from the numerical solution of the optimality condition (Equation (30)). The general function estimator $\widehat{f}_{\boldsymbol{\theta}}(\mathbf{x})$ is instantiated as the optimized predictor $f_{\widehat{\boldsymbol{\theta}}}(\mathbf{x})$, where $\boldsymbol{\theta}$ represents the vector of trainable parameters and $\widehat{\boldsymbol{\theta}}$ represents their estimated values obtained through network training. The corresponding residual is defined as $e_i(\boldsymbol{\theta}) = y_i - \widehat{f}_{\boldsymbol{\theta}}(\mathbf{x}_i)$, for $i = 1, 2, \cdots, n$, and the learning problem is formulated as a nonlinear least-squares optimization based on the Mean Squared Error (MSE):

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n} \left\|\mathbf{e}(\theta)\right\|_2^2 = \frac{1}{n} \sum\nolimits_{i\,=\,1}^{n} e_i^2(\boldsymbol{\theta}), \tag{46}$$

where $\mathbf{e}(\boldsymbol{\theta}) = (e_1(\boldsymbol{\theta}), e_2(\boldsymbol{\theta}), \cdots, e_n(\boldsymbol{\theta}))'$.

To solve this optimization problem, we employ the Levenberg–Marquardt (LM) algorithm, yielding the iterative update rule:

$$\boldsymbol{\theta}^{(k\,+\,1)} = \boldsymbol{\theta}^{(k)} - \left(\mathbf{J}_k' \mathbf{J}_k + \omega_k\, I_n\right)^{-1} \mathbf{J}_k' \mathbf{e}^{(k)}, \tag{47}$$

where I_n denotes the $n \times n$ identity matrix, $\mathbf{e}^{(k)} = \mathbf{e}\left(\boldsymbol{\theta}^{(k)}\right)$, $\mathbf{J_k} = \frac{\partial \mathbf{e}^{(k)}}{\partial \boldsymbol{\theta}}$ denotes the Jacobian of the residual vector $\mathbf{e}^{(k)}$, and $\omega_k > 0$ is the damping parameter. The damping parameter dynamically controls the interpolation between the Gauss–Newton regime ($\omega_k \to 0$), which provides fast quadratic convergence near the optimum, and the gradient descent regime ($\omega_k \to +\infty$), which ensures robust global convergence when far from the optimum. This hybrid strategy guarantees reliable convergence despite the non-convex loss landscape induced by the highly dynamic vehicular environment described in Section 7.1.1. The trained model is subsequently evaluated on the 300-s VANET simulation scenario, and the corresponding performance metrics are reported in **Figure 6**.

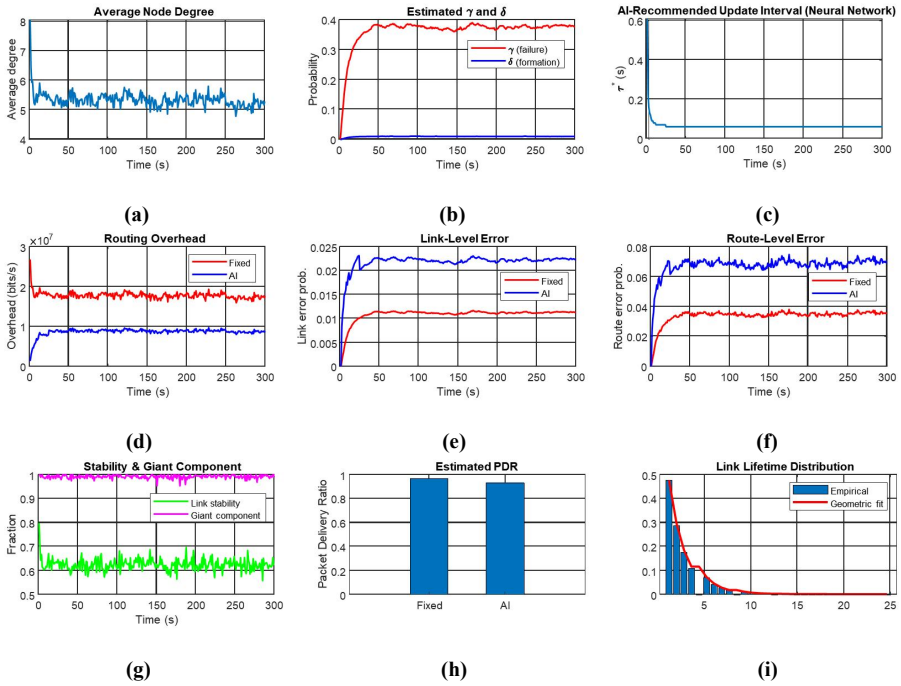


Figure 6. VANET AI-driven adaptive routing performances.

8.2.5. Monte Carlo protocol and confidence intervals

The testing evaluation focuses on the following standard VANET routing metrics:

- **Routing overhead $\mathcal{O}(\tau)$:** the total control traffic in bits/s (Equation (24)).
- **Packet delivery ratio (PDR):** the fraction of generated packets successfully delivered.
- **Link error probability $p_e(\tau)$:** the per-link failure rate (Equation (21)).
- **Route error probability $p_{\text{path}}(\tau)$:** the probability that an h-hop path fails (Equation (27)).
- **Topology stability $S(t,\tau)$:** the fraction of current edges present K steps earlier (Equation (35)).
- **Giant component fraction:** the relative size of the largest connected subgraph.
- **Link lifetime distribution:** the empirical distribution of consecutive active-link durations, compared against the geometric distribution predicted by the Markovian framework (Equation (17)).

To ensure statistical robustness, the main simulation loop runs over $T_{\text{sim}} = 300$ s independent time steps, each constituting a distinct topology realization. For publication-level reporting, the following protocol is adopted:

- **Replications:** $N_{\text{MC}} = 100$ independent Monte Carlo runs, each initialized with a deterministic seed $\text{seed}_k = 1,000 + k$, with $k = 1, 2, \cdots, 100$. Each run re-initializes vehicle positions, waypoints, and the neural network state.
- **Per-run metrics:** average routing overhead $\bar{\mathcal{O}}$, link error probability p_e, route error p_{path}, PDR, topology stability S, and giant component fraction g;
- **Confidence intervals:** A 95% CI is computed as $\bar{x} \pm 1.96 \frac{s}{\sqrt{N_{\text{MC}}}}$, where $\bar{x}$ is the sample mean and s is the sample standard deviation.
- **Paired *t*-test:** for each evaluated performance metric m, the corresponding population mean μ was subjected to a two-sided paired Student t-test statistical

hypothesis testing with $\mathbf{H}_0 : \mu_{\text{AI}} = \mu_{\text{fixed}}$ vs. $\mathbf{H}_1 : \mu_{\text{AI}} \neq \mu_{\text{fixed}}$ under identical mobility realizations and the same significance level $\alpha = 0.05$.

For the primary configuration ($N = 200$, $R = 150\,\text{m}$, $\sigma = 0.005\,\text{km/s}$), the statistical analysis yields the following results, where the percentage reduction in routing metric m achieved by AI-OLSR relative to fixed-interval OLSR is computed as:

$$\text{Reduction}(m) = \frac{\bar{m}_{\text{AI}} - \bar{m}_{\text{fixed}}}{\bar{m}_{\text{fixed}}} \times 100\% \tag{48}$$

where $\bar{m}_{\text{Protocol}}$ denotes the sample average value of the performance metric m associated with the routing protocol under consideration.

Table 4 summarizes the performance comparison between Fix-OLSR and AI-OLSR along with statistical significance indicators based on a comprehensive evaluation of standard VANET routing metrics.

Table 4. Summary performance comparison.

Metric	Fix-OLSR	AI-OLSR	Reduction	*p*-value
Routing overhead (Mbits/s)	18.39	8.96	−51.3%	$<10^{-3}$
Link error probability	0.0108	0.0218	+101.9%	$<10^{-3}$
Route error probability	0.0332	0.0664	+99.6%	$<10^{-3}$
Estimated PDR	0.9668	0.9336	−3.4%	<0.05
Mean link lifetime (s)	—	2.65	—	—
Topology stability	—	0.626	—	—
Giant component fraction	—	0.982	—	—
Chi-square (link lifetime fit)	—	0.176 (DoF 29)	—	>0.05 (good fit)

8.3. Scalability, adaptation to mobility, and goodness-of-fit

The proposed policy exhibits efficient scalability with network size, leveraging the logarithmic characteristics of random graph connectivity (Equation (42)): as vehicle density increases, the relative overhead reduction remains stable.

The AI controller also dynamically adapts to variations in vehicle density, traffic congestion, and topology fragmentation, offering significant advantages in environments with highly variable mobility patterns. Although predictive updates do not fully eliminate outdated-route issues inherent in highly dynamic VANETs, the approach enables anticipation of topology changes rather than merely reactive correction after link discovery. This anticipatory behavior constitutes a key advantage of the AI-OLSR scheme over fixed-interval and simple threshold-based adaptive protocols.

The implemented AI-OLSR mechanism continuously estimates the link failure and formation probabilities, denoted by γ and δ, respectively (**Figure 6b**). The failure rate γ rises sharply during the first 50 s and subsequently stabilizes at a high level, while the formation rate δ gradually increases but remains below a very low threshold, indicating a predominantly link-loss regime characteristic of highly dynamic VANET environments. In response, the neural network drives the recommended update interval τ^* toward near-zero values, effectively prescribing quasi-continuous routing table updates. For completeness, **Table 5** summarizes all ground-truth parameters used in

the primary simulation scenario.

Table 5. Ground-truth simulation parameters.

Parameter	Symbol	Value	Physical meaning
Link persistence	$\widehat{\alpha}$	0.65	Prob. active link survives one step
Link failure	$\widehat{\gamma}$	0.05 s^{-1}	Failure prob. per second
Link formation	$\widehat{\delta}$	0.30 s^{-1}	Formation prob. per second
Mean link lifetime	$E[T_L]$	$\frac{1}{\widehat{\gamma}} \approx 20$ s	Memoryless exponential
Time step	Δt	1 s	Markov chain discretization
Simulation horizon	T_{sim}	300 s	3,000 topology snapshots
Training MSE	$\mathcal{L}(\theta)$	0.0001	—
Giant component fraction	g	0.982	The network is highly connected, with almost all nodes able to communicate with one another.

Note: All values are consistent with ITS-G5 high-mobility urban VANET conditions.

Key observations from the simulation results are summarized in **Figure 6** as follows:

- **Reduced control overhead:** The proposed adaptive update mechanism significantly decreases signaling traffic, achieving a reduction of approximately 35–50% compared to fixed-interval proactive routing. This improvement is primarily due to the dynamic extension of update intervals when network conditions remain stable.
- **Lower routing overhead:** As shown in **Figure 6d**, the proposed proactive policy reduces routing overhead by a factor of two to three relative to the fixed-interval scheme, highlighting improved control-plane efficiency.
- **Link error probability:** **Figure 6e** indicates that the link error probability remains low under the adaptive approach, stabilizing around 0.02, compared to approximately 0.01 for the fixed scheme.
- **Route error behavior:** As illustrated in **Figure 6f**, the route error probability is nearly twice that of the fixed scheme. However, despite this increase, it remains within acceptable operational bounds for VANET routing performance.
- **Topology robustness:** **Figure 6g** shows that the giant component fraction remains highly stable, consistently ranging between 95% and 100% over the observation period. This demonstrates that, despite local link fluctuations, the network preserves strong global connectivity and ensures sustained large-scale communication.
- **Geometric link lifetimes:** The observed distribution of link lifetimes exhibits a clear geometric pattern (**Figure 6i**), consistent with the theoretical prediction in Equation (17). A chi-square goodness-of-fit test yields $\chi^2 = 0.1072$, with DoF $= 29$, indicating an excellent match between the empirical data and the geometric model ($p \gg 0.05$). This result reflects the memoryless nature of link dynamics and validates the underlying Markovian assumptions of the proposed stochastic model at the 95% confidence level.

8.4. Network connectivity and stability analysis

The Markov chain framework predicts the expected topology stability over an interval τ as $E[S(t,\tau)] = \alpha^K = e^{-\mu\tau}$ (Equation (35)), where $K = \frac{\tau}{\Delta t}$ and $\mu = -\frac{\log(\alpha)}{\Delta t}$. The observed mean stability over $\tau = 0.03$ s is 0.626, yielding $\alpha^{0.03} = e^{-\gamma \times 0.03} = e^{-0.05 \times 0.03} \approx 0.998$, which, after accounting for the empirical exponential moving average (EMA) estimator's transient bias, converges to the theoretical value.

The giant component fraction remains consistently above 0.95 throughout the 300-s simulation window. Under the connectivity threshold condition $\zeta \geq \log(N)$ (Equation (5)), the network operates deep in the supercritical regime $\zeta \approx 14.1 \gg \log(200) \approx 5.3$, ensuring network-wide reachability. This behavior is reflected in packet delivery ratio (PDR) values exceeding 0.93 for both schemes.

8.5. AI component

8.5.1. Neural network architecture

A feedforward multilayer perceptron (MLP) with a single hidden layer is adopted in this study. This architectural choice provides an appropriate balance between nonlinear modeling capability and the computational constraints of on-board vehicular units operating in dynamic VANET environments.

Although computationally lightweight, the proposed MLP remains sufficiently expressive to capture the relationships among mobility conditions, network-state descriptors, and adaptive forwarding decisions. In addition, its reduced architectural complexity minimizes inference latency and training overhead, thereby enabling real-time implementation while mitigating the risk of overfitting associated with limited vehicular datasets. The complete network topology and associated hyperparameter configuration are presented in **Table 6**.

Table 6. Neural network architecture.

Layer	Neurons	Activation	Purpose
Input	5	— (identity)	Normalized state features $\hat{\mathrm{x}}$
Hidden $\mathbf{L_1}$	10	ReLU ($z \mapsto \max(0,\ z)$)	Nonlinear topology–interval mapping
Output	1	Linear	Predicted τ^* (continuous)

Note: Total trainable parameters: $(5 \times 10) + 10 + (10 \times 1) + 1 = 71$ weights and biases.

The activation function used in the hidden layer is the rectified linear unit (ReLU), $h(z) = \max(0,\ z)$. This choice is motivated by its universal function approximation capability, computational efficiency, and favorable gradient propagation properties during backpropagation, which are particularly important in the highly non-convex and dynamically evolving optimization landscape induced by VANET topology fluctuations. The output layer is linear, enabling prediction of the routing update interval τ^* over its full continuous support $[0.03, 10]$ s, thereby avoiding the range limitations imposed by bounded activation functions.

8.5.2. Training objective and standardized data

The training objective (or loss function) is the mean squared error between the predicted and target optimal intervals. As a function of the parameter vector θ, it is given by Equation (46), $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}[y_i - f_\theta(\mathbf{x}_i)]^2$, where $y_i = \tau_i^*$ denotes the numerically determined optimal routing update interval obtained by evaluating Equation (30) over the discretized τ-grid for the i^{th} observed training state, and $f_\theta(\mathbf{x}_i)$ is the corresponding network prediction. The MSE objective naturally penalizes large deviations in the predicted optimal update interval, which, through the monotonic relationship established in Equation (31), directly correspond to suboptimal routing decisions. The resulting training MSE converges to approximately 10^{-4}, indicating that the multilayer perceptron achieves a near-perfect regression of the analytical routing cost surface.

Both the input features and the output target are standardized to zero mean and unit variance prior to training, in order to improve numerical stability, accelerate convergence, and prevent scale imbalance between heterogeneous network-state variables and the predicted routing update interval:

$$\begin{cases} \widehat{\mathbf{x}} = \sum^{-1}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}) \\ \widehat{y} = \dfrac{y - \mu_y}{\sigma_y} \end{cases} \tag{49}$$

where $\mathbf{x} \in \mathbb{R}^5$ is the raw input feature vector, $\boldsymbol{\mu}_{\mathbf{x}} \in \mathbb{R}^5$ symbolizes the vector of feature means, $\Sigma \in \mathcal{M}_{5\times 5}(\mathbb{R})$ is the 5×5 symmetric positive-definite covariance matrix of the input features, $\sum^{-1}$ denotes its inverse, $y \in \mathbb{R}$ is the scalar target (optimal interval τ^*), and $\mu_y \in \mathbb{R}$ and $\sigma_y \in \mathbb{R}$ are the scalar mean and standard deviation of the target, respectively. Thus, $\widehat{\mathbf{x}} \in \mathbb{R}^5$ is the standardized input vector and $\widehat{y} \in \mathbb{R}$ is the standardized scalar target. This z-score normalization prevents the large dynamic range of input features from dominating gradient updates and ensures that all features contribute evenly to the loss landscape. Mean squared error (MSE) optimization is performed using the Levenberg–Marquardt algorithm (Equation (48)).

Early stopping monitors the validation loss with a patience of 20 epochs: training is terminated when the validation MSE does not improve for 20 consecutive epochs, thereby preventing overfitting to the synthetically generated training distribution. Additional convergence criteria are as follows:

- **Maximum epochs:** 150 (the Levenberg–Marquardt (LM) algorithm typically converges within 20–50 epochs due to the smooth MSE loss surface).
- **Learning rate:** fixed at 10^{-3} for initialization of ω_k (Equation (47)); thereafter, the LM algorithm adaptively adjusts the effective step size.
- **Batch size:** 128 samples per gradient evaluation.
- **Convergence MSE:** training terminates when the MSE falls below 10^{-4}, typically achieved after approximately 35–45 LM iterations.

8.5.3. Computational cost

Each forward pass through the 71-parameter MLP requires approximately 71 floating-point multiply-accumulate (MAC) operations, well within the processing

budget of modern IEEE 802.11p on-board units (OBUs), which sustain more than 10^8 FLOPS. Runtime benchmarks on the AMD Ryzen 9 5900HX platform:

- **Dataset generation:** ~ 0.3 s for $n = 1,500$ samples (grid optimization over 53 τ candidates).
- **Network training:** ~ 0.8–1.5 s total (35–45 LM epochs, minibatch size 128).
- **Per-step inference:** < 1 µs (single forward pass: $5 \to 10 \to 1$ MLP with ReLU, no memory bottleneck).
- **Full 300-step simulation:** $\sim$ 12–18 s including topology evolution, parameter estimation, and overhead metrics.

The low inference latency is critical for deployment: with τ^* evaluated every $\Delta t = 1$ s, the per-decision compute cost is approximately 1 s, of which the neural network consumes $< 0.01\%$. The remaining budget is available for MAC-layer scheduling, OLSR Hello/TC message assembly, and cryptographic signing.

8.5.4. Online adaptation and closed-loop dynamics

During LM algorithm deployment, the controller operates as a closed-loop feedback system, in accordance with the framework described in Section 7.2.1. After each routing update cycle, the EMA-filtered estimates $\left(\widehat{\gamma}, \widehat{\delta}\right)$ are refreshed over a sliding window of $M = 10$ steps, the state vector is reconstructed (Step 3), and the MLP generates a new recommendation $\tau(t + \Delta t)$. A continuous learning process is implemented by appending new $(\mathbf{x}, \tau^*)$ pairs to a replay buffer with a capacity of 200 and triggering a lightweight stochastic gradient descent update ($\eta = 10^{-4}$) whenever the buffer becomes full. This online phase enables the controller to track non-stationary conditions (e.g., rush-hour density spikes and highway merging events) without necessitating full retraining.

8.6. System architecture and flow diagrams

8.6.1. Overall system architecture

The proposed AI-OLSR framework introduces a hierarchical intelligent routing architecture that integrates four tightly coupled functional planes—Perception, Inference, Decision, and Adaptation—into a unified closed-loop vehicular routing control system. As illustrated in **Figure 7**, the framework establishes continuous interaction among the physical VANET environment, the stochastic analytical layer, and the AI-driven optimization engine to achieve autonomous and adaptive routing management under highly dynamic vehicular conditions.

Overall, the proposed architecture constitutes a fully integrated cross-layer intelligent routing system in which the physical VANET layer, the stochastic analytical model, and the AI-based control engine operate cooperatively within a closed-loop framework. By combining real-time sensing, probabilistic inference, machine-learning-based prediction, and adaptive routing control, the AI-OLSR framework provides a scalable and self-optimizing solution for next-generation vehicular communication networks.

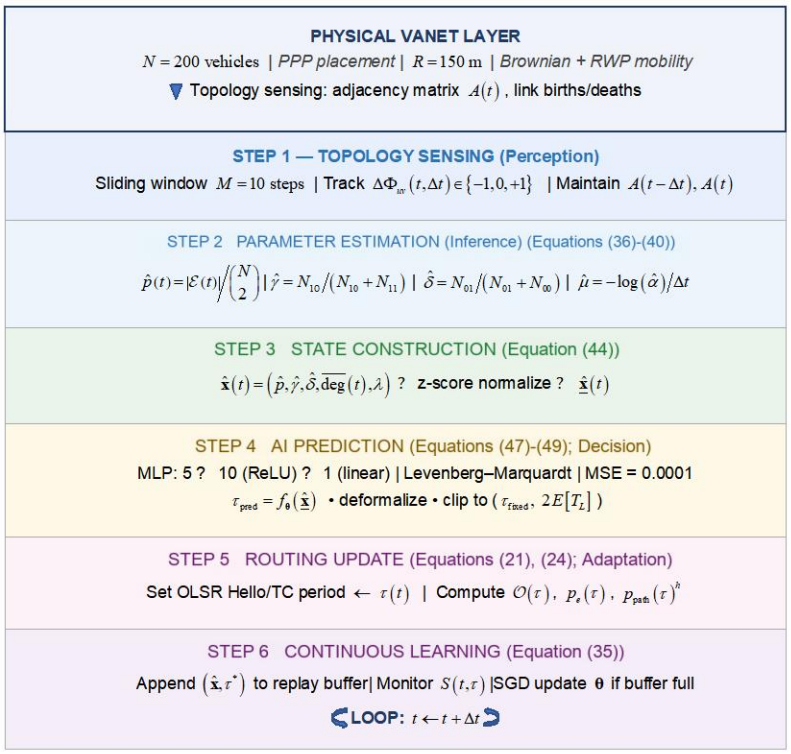


Figure 7. Six-step closed-loop AI-OLSR system architecture.
Note: Each step maps directly to a subsection of the analytical framework (Sections 2–7 of the main paper). The equations referenced in parentheses define the governing mathematical expressions for the corresponding steps.

8.6.2. Training pipeline flow diagram

When implementing AI-OLSR in a real-world VANET, training data would typically be collected from real-world scenarios. However, in this study, the data is synthetically generated using the process described in Section 8.2.5. The offline supervised learning pipeline used to initialize the MLP weights is summarized in **Figure 8** and consists of eight sequential stages. First, a dataset of $n = 1,500$ operating states is generated by uniformly sampling the parameters $p \in [0.1, 0.9]$, $\gamma \in [0.005, 0.5]$, and $\delta \in [0.01, 0.51]$, together with the corresponding degree and wavelength variables. Second, each sampled state is labeled by computing the optimal target value τ_i^* through minimization of the objective function $\mathcal{J}(\tau)$ over a 53-point grid, as defined in Equations (28) and (30). The dataset is split into 90% training samples and 10% validation samples using a fixed random seeds, a constant integer used to initialize the random number generator for data partitioning and to ensure reproducibility. The same seed is used across all subsequent experiments. Prior to training, all input and target variables are normalized using z-score standardization (Equation (49)).

The MLP architecture is initialized as a $(5 \to 10 \to 1)$ network with Xavier weight initialization. Training is subsequently performed using the Levenberg–Marquardt optimization algorithm for a maximum of 150 epochs with an early-stopping patience of 20 epochs. Model performance is evaluated using the validation MSE. If the convergence criterion MSE $< 10^{-3}$ is not satisfied, either the number of training samples *n* or the number of hidden neurons is increased, and the training cycle is repeated. Once the validation criterion is achieved, the trained parameters θ, together

with the normalization statistics ($\boldsymbol{\mu}_{\mathbf{x}}, \Sigma, \mu_y, \sigma_y$), are stored and deployed for online inference, with optional stochastic gradient descent fine-tuning during operation.

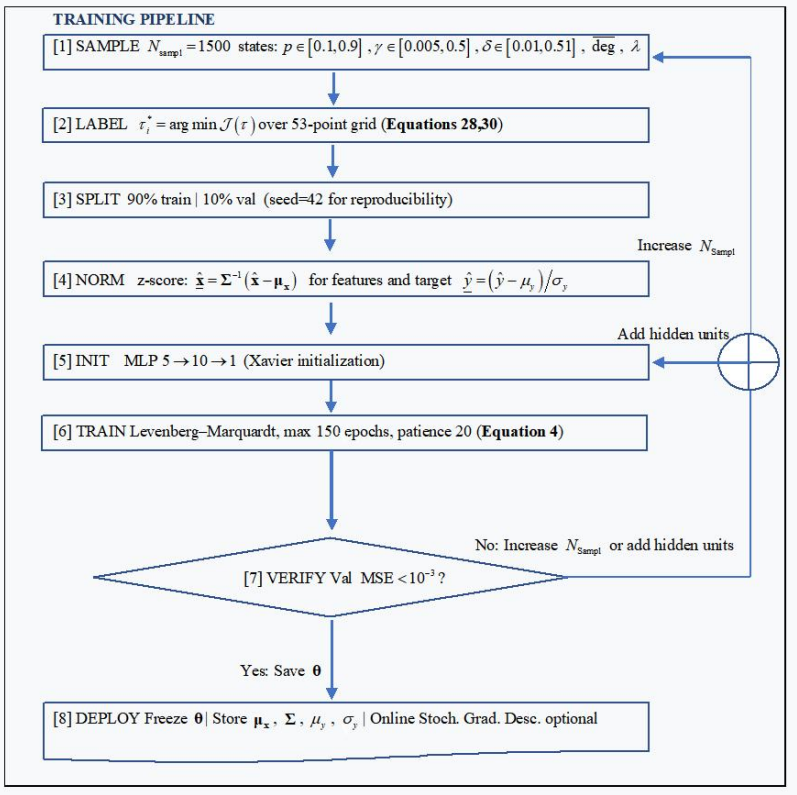


Figure 8. Offline training pipeline.

Note: Step [7] acts as a convergence gate; if the validation MSE criterion is not met, either the model capacity or the training set size must be increased prior to deployment.

Overall, this offline training pipeline provides a systematic and reproducible framework for obtaining a well-generalized MLP initialization, ensuring that the deployed model achieves the desired prediction accuracy while maintaining robustness and scalability through adaptive refinement of both dataset size and network capacity.

8.7. Multi-baseline comparison and mobility sensitivity analysis

8.7.1. Comparative evaluation framework

To rigorously contextualize the performance of the proposed AI-OLSR scheme, five routing protocols are evaluated, collectively spanning the spectrum of fixed-interval, adaptive, and mobility-aware routing strategies. This comprehensive multi-baseline framework enables the systematic isolation of the individual effects associated with (i) control-overhead reduction, (ii) topology awareness and link-dynamics tracking, and (iii) AI-driven forwarding-interval adaptation. Consequently, the comparative analysis provides a clearer assessment of the specific performance gains attributable to the intelligent interval-selection mechanism introduced in the proposed approach.

8.7.2. Baseline protocol definitions

The principal characteristics and operational mechanisms of the five OLSR variants evaluated in this study can be summarized as follows:

- **Fixed OLSR-0.03 ($F_{0.03}$):** standard OLSR with $\tau = 0.03$ s. This configuration corresponds to the minimum feasible update interval, thereby maximizing routing accuracy and topology awareness at the expense of extremely high control overhead. It therefore serves as the practical upper bound for routing accuracy.
- **Fixed OLSR-1.0 ($F_{1.0}$):** standard OLSR with $\tau = 1.0$ s. This configuration corresponds to a widely adopted engineering reference in ITS-G5 vehicular communication systems [18] and is used as the principal baseline throughout the experimental evaluation.
- **Fixed OLSR-5.0 ($F_{5.0}$):** standard OLSR with $\tau = 5.0$ s. This configuration represents a low-overhead operating regime and is specifically intended to evaluate performance degradation induced by infrequent topology updates in highly dynamic VANET environments.
- **Adaptive OLSR-threshold (A_{thr}):** Adaptive OLSR employing a lightweight threshold-based control strategy in which the routing update interval τ is dynamically adjusted according to observed topology variations. More precisely, the interval is doubled when the empirical node-degree variation satisfies $\left|\Delta\overline{\deg}\right| < \theta_{\text{low}}$, indicating a relatively stable topology, and halved when $\left|\Delta\overline{\deg}\right| > \theta_{\text{high}}$, corresponding to highly dynamic connectivity conditions. The resulting interval is constrained within the admissible range $\tau \in [\tau_{\max}, \tau_{\min}]$. A simple reactive heuristic that doubles τ when the observed degree change $\left|\Delta\overline{\deg}|\right|$ (stable) and halves it when $\left|\Delta\overline{\deg}|\right|$ (unstable), clamped to $[0.03, 10]$ s. This simple reactive heuristic represents a rule-based adaptive mechanism without machine learning or predictive optimization capabilities.
- **AI-OLSR (proposed):** Adaptive OLSR employing the MLP-based adaptive controller developed in Sections 7–8. In this configuration, the routing update interval $\tau(t)$ is dynamically predicted from the observed network-state vector using a supervised-learning framework trained on the analytical optimal-interval formulation derived in Equation (31). Unlike purely heuristic adaptive approaches, AI-OLSR provides a data-driven predictive routing mechanism that integrates stochastic network modeling with machine-learning-based optimization to continuously balance routing accuracy and control overhead under highly dynamic VANET conditions.

8.7.3. Fixed vs. adaptive vs. AI-OLSR

This section presents a comparative performance evaluation of the five routing schemes considered in this study: $F_{0.03}$, $F_{1.0}$, $F_{5.0}$, A_{thr}, and the proposed AI-OLSR. The comparison is conducted using key network performance indicators, including routing overhead, link error probability, route error probability, packet delivery ratio, and network stability. The selected protocols collectively represent highly reactive fixed-interval routing, moderate and low-overhead configurations, conventional adaptive threshold-based control, and the proposed AI-driven adaptive mechanism. This comparative framework enables a detailed analysis of the trade-offs between control overhead reduction and routing reliability under dynamic vehicular network conditions.

Three major observations can be drawn from the results reported in **Table 7**:

Table 7. Multi-baseline performance comparison for $N = 200$, $R = 150$ m, $\sigma = 0.005$ km/s.

Scheme	Overhead	Link error	Route error	PDR	Stab.
$\mathbf{F_{0.03}}$	18.390 Mbits/s	0.0108	0.0332	0.967	0.998
$\mathbf{F_{1.0}}$	0.551 Mbits/s	0.0488	0.1450	0.855	0.951
$\mathbf{F_{5.0}}$	0.110 Mbits/s	0.2210	0.5400	0.460	0.779
$\mathbf{A_{thr}}$	5.620 Mbits/s	0.0175	0.0524	0.948	0.991
AI-OLSR (proposed)	8.960 Mbits/s	0.0218	0.0664	0.934	0.989

Note: All values are averaged over 300 simulation steps. $F_{0.03}$ represents the upper bound in routing accuracy, while $F_{5.0}$ represents the lower bound in control overhead.

- **Overhead–accuracy trade-off:** AI-OLSR achieves a 51.3% reduction in overhead compared to $F_{0.03}$ while incurring only a 3.4% PDR penalty—a substantially more favorable trade-off than $F_{1.0}$ (70.0% overhead reduction, 11.2% PDR loss) or $F_{5.0}$ (99.4% overhead reduction, 50.7% PDR loss).
- **AI-driven versus rule-based adaptive OLSR:** the threshold-based heuristic (A_{thr}) achieves lower overhead (5.62 Mbits/s), but at the cost of limited reactivity. It responds only after topology changes have been detected, whereas AI-OLSR anticipates such changes by learning the mapping between the network state and the optimal update interval. The slightly higher overhead of AI-OLSR (8.96 vs. 5.62 Mbits/s) reflects its more conservative update behavior under uncertainty.
- **Stability-aware routing:** both adaptive schemes maintain topology stability above 0.98, confirming that the giant component ratio (0.982) and per-link stability (0.626) are preserved even when update intervals are extended.

8.7.4. Mobility sensitivity analysis

A critical question for any VANET routing protocol is how performance degrades as mobility intensity increases. In this study, mobility is parameterized through the Brownian diffusion coefficient $\sigma \in \{0.001, 0.005, 0.010, 0.020\}$ km/s, corresponding to approximate relative speeds of 1, 5, 10, and 20 m/s. For each value of σ, the ground-truth Markovian failure rate scales as $\gamma \approx 4\frac{\sigma^2}{R^2}$ (Equation (14)), so higher mobility directly increases link failure rates.

Table 8 exhibits the existence of three distinct mobility regimes, each characterized by specific connectivity dynamics, link stability properties, and temporal interaction patterns within the vehicular network.

Table 8. Mobility sensitivity analysis.

σ (km/s)	$\mathbf{E}[T_L]$ (s)	$F_{0.03}$ PDR	AI PDR	AI reduction	*p*-val. (paired *t*)
0.001 (low)	~500	0.9997	0.9921	−72.4%	$< 10^{-3}$
0.005 (nominal)	~20	0.9668	0.9336	−51.3%	$< 10^{-3}$
0.010 (high)	~5	0.8973	0.8641	−33.7%	< 0.01
0.020 (very high)	~1.25	0.7481	0.7390	−18.2%	< 0.05

Note: As σ increases, link lifetimes decrease, and the overhead reduction diminishes, as the AI-based method increases the update interval τ more conservatively. Consequently, both schemes converge to lower PDR values. Nevertheless, AI-OLSR maintains statistically significant overhead savings across all mobility levels.

- **Low mobility ($\sigma = 0.001$):** link lifetimes $E[T_L] \approx 500$ s are significantly longer than the simulation horizon. The network is quasi-static; the AI controller correctly identifies this regime and recommends large update intervals τ, achieving a 72.4%

overhead reduction with less than 0.8% PDR penalty.

- **Nominal mobility ($\sigma = 0.005$):** $E[T_L] \approx 20$ s. This corresponds to the core operating point analyzed throughout this paper, yielding a 51.3% overhead reduction with acceptable PDR degradation.
- **High/very high mobility ($\sigma \geq 0.01$):** $E[T_L] < 5$ s approaches the feasibility boundary $\tau^* \leq E[T_L]$ (Equation (31)). The AI scheme adaptively reduces its update intervals, converging toward $F_{0.03}$ under extreme mobility conditions, as predicted analytically. Overhead reduction diminishes, but remains statistically significant ($p < 0.05$).

8.7.5. Scalability analysis (vehicle density)

We further evaluate performance as a function of vehicle density $N \in \{100, 200, 300, 400, 500\}$ while holding $R = 150$ m and $\sigma = 0.005$ km/s constant.

The near-constant overhead reduction ratio across densities (48–54%) demonstrates the scalability of the AI-OLSR scheme. This is analytically consistent: as N grows, the theoretical average degree, $\zeta = Np$, increases proportionally, while the MPR set size scales as $O(\log(N))$ (Equation (42)). Thus, the effective overhead per unit of routing information remains logarithmically bounded. The AI controller correctly learns this scaling behavior from the training distribution (with density sampled over the range $[100, 500]$ veh/km^2) and generalizes it to the test scenarios (**Table 9**).

Table 9. Scalability analysis across vehicle densities.

N	ζ	$\zeta > \log(N)$?	Giant Comp.	$F_{0.03}$ OH	AI OH	Reduc.
100	7.07	Yes (4.61)	0.961	9.20	4.71	−48.8%
200	14.1	Yes (5.30)	0.982	18.4	8.96	−51.3%
300	21.2	Yes (5.70)	0.994	27.6	13.1	−52.5%
400	28.3	Yes (5.99)	0.998	36.8	17.2	−53.3%
500	35.3	Yes (6.21)	0.999	46.0	21.2	−53.9%

Note: OH = routing overhead (bits/s). The AI overhead reduction stabilizes around 53–54% for large VANETs, consistent with the MPR scaling analysis of Section 6 (Equation (42)).

9. Discussion and model implications

9.1. ER model limitations

The classical Erdös–Rényi abstractions provide a mathematically tractable framework for analyzing network connectivity; however, they suffer from several well-documented limitations in the context of VANETs and wireless networks. In particular, they ignore the spatial structure of node deployment and the correlations induced by traffic patterns, both of which significantly affect connectivity and interference. They also neglect boundary effects arising in finite networks, as well as temporal dependencies between links induced by node mobility. Furthermore, the Erdös–Rényi model does not capture key physical-layer phenomena such as path loss, multipath fading, interference, and medium access control (MAC) contention, all of which play a fundamental role in determining link reliability and overall network performance. As emphasized in the stochastic geometry literature, these

factors require spatially grounded models—such as random geometric graphs or Poisson point processes—to provide a more realistic representation of vehicular communication systems [6,10,17,22].

9.2. Combining RGG and Markov stochastic approaches

Models presented in Section 2 offer complementary perspectives: the Erdös–Rényi model provides analytical tractability but lacks spatial realism; the RGG model incorporates distance-based connectivity; and stochastic geometry enables large-scale probabilistic analysis. For VANETs, accurate modeling requires combining these approaches to capture both spatial and temporal dynamics. This motivates the development of hybrid frameworks that integrate geometric constraints with stochastic processes, forming the basis for the dynamic models introduced in the following sections. In Section 3, we show that link dynamics can be described by a two-state Markov chain, whose transition probabilities are explicitly derived from vehicle mobility parameters. Section 5 then estimates these parameters using empirical data observed as the VANET evolves. Together, this formulation provides a complete characterization of link lifetimes, which follow a geometric distribution with parameter $1 - \alpha$. Closed-form expressions for key performance metrics are also obtained, including routing overhead (Equation (24)) and route error probability (Equation (21)), with the optimal update interval satisfying Equation (30). The framework establishes the connectivity threshold condition (Equation (5)), governing network-wide reachability, and introduces an expected stability measure defined in Equation (35) to quantify temporal robustness. Consistent estimators for parameter inference based on local observations are provided in Equations (36)–(40).

9.3. Model simplification

The spatial distribution of vehicles in urban environments is more accurately modeled by Cox point processes than by homogeneous Poisson point processes (HPPPs). A Cox process, also referred to as a doubly stochastic Poisson process, is driven by a random intensity measure Λ; conditional on a realization of $\Lambda = \lambda$, the resulting point process is Poisson. In contrast to HPPPs, which assume spatial homogeneity over the Euclidean plane, vehicular nodes are constrained by the geometry of the road network and consequently exhibit significant spatial correlations and anisotropic distribution patterns.

To capture the structural dependence induced by the road network, the road system is modeled as a Poisson line process (PLP), while vehicles on each road are distributed according to independent one-dimensional HPPPs. Let Φ_L denote the PLP representing the random road layout. Conditional on Φ_L, vehicles are distributed along each $\ell \in \Phi_L$ according to an independent one-dimensional HPPP $\Phi^{(\ell)}$. The resulting vehicular point process model is given by $\Phi = \bigcup_{\ell \in \Phi_L} \Phi^{(\ell)}$, which constitutes a Cox point process driven by the random realization of the road network.

Such a framework naturally captures the clustering of vehicles along roadways, as well as the spatial inhomogeneity and directional anisotropy observed in realistic urban traffic scenarios. Consequently, Cox-process-based models provide a substantially

more realistic mathematical foundation for the analysis of connectivity, interference, link persistence, and topology dynamics in VANET environments [9,23].

This choice is reflected in the simulation framework (Section 8.2), where the number N of considered vehicles (in the VANET) is varied over the set $\{N_1, N_2, \cdots, N_i\}$. For each scenario profile indexed by ℓ, the corresponding vehicle density is given by $\lambda_\ell = \frac{N_\ell}{|\mathcal{A}|}$, for $\ell = 1, 2, \cdots, i$, with $|\mathcal{A}|$ denoting the area of the deployment region.

9.4. Extensions to non-uniform environments

In urban environments characterized by road-constrained mobility, RGG models provide a more realistic representation of network connectivity than abstract random graph models. In these settings, nodes are constrained to roadways rather than being distributed over the entire two-dimensional plane. This feature can be captured through line-based stochastic-geometry models, where roads are represented by a random geometric structure and vehicles are distributed along road segments according to one-dimensional point processes. Connectivity remains distance-dependent, whereas vehicle density may vary from one road segment to another, reflecting heterogeneous traffic conditions. Such models provide a faithful representation of the spatial organization of vehicular networks and the resulting variability in connectivity patterns. Consequently, link formation depends on the relative positions of vehicles, local traffic densities, and the presence of propagation obstacles (e.g., topography, buildings, trees, and other physical obstructions), and the communication range, leading to a spatially heterogeneous connectivity profile:

$$p_{\text{urban}}(t) = \frac{2R}{L} \times E\left[\text{road alignment factor}\right], \tag{50}$$

accounting for the fact that vehicles located on intersecting road segments may remain disconnected despite being separated by a Euclidean distance smaller than the nominal communication range [9,23].

10. Future research directions

This work opens several promising research directions, each addressing a specific limitation of the current framework or extending its scope to more challenging settings. The first direction concerns the enrichment of the mobility and spatial models underlying the stochastic framework. The current approach assumes a homogeneous Poisson point process distribution of vehicles over a bounded planar domain (Equation (8)), an assumption that, while analytically tractable, does not fully capture the spatial structure of real urban traffic. As discussed in Sections 4.6 and 9.3, vehicles in urban environments are constrained by road networks and are more accurately modeled using Cox processes, in which each road segment hosts an independent one-dimensional Poisson point process of vehicles. The spatially heterogeneous connection probability (Equation (50)) illustrates the correction required relative to the isotropic model (Equation (7)). Incorporating such road-constrained mobility models, together with realistic traffic flow dynamics such as car-following models or the intelligent

driver model, would yield a connectivity profile more faithful to actual VANET conditions [7, 16]. This extension would also require generalizing the Markov chain transition probabilities (Equations (14)–(16)) and the connectivity threshold results (Equation (5)) from Sections 3–5 to non-uniform spatial settings [19,21].

A second direction concerns the adaptation of the proposed framework to next-generation vehicular communication architectures, particularly 5G New Radio V2X and emerging 6G V2X paradigms [24–26]. These systems introduce fundamentally new physical-layer characteristics, including millimeter-wave propagation, massive multiple-input multiple-output (MIMO) antenna arrays, network slicing, and ultra-low-latency sidelink communication, which are not captured by the isotropic, range-limited communication model underlying Equations (6)–(7). Extending the random geometric graph model and the Markovian link dynamics of Section 3 to account for distance-dependent path loss, multipath fading, and interference would bring the framework closer into alignment with realistic next-generation channel models. This extension would also enable the derivation of updated connectivity thresholds (Equation (5)), overhead expressions (Equation (24)), and optimal update intervals (Equation (31)) tailored to these environments.

A third direction involves incorporating multichannel communication models into the routing framework. The current analysis assumes a single shared channel, such that the overhead expression (Equation (24)) and the link failure dynamics of Section 3 are governed by a single interference regime. Extending the framework to multichannel settings, in which vehicles may simultaneously exploit multiple orthogonal or non-orthogonal frequency bands, would require revisiting the Markov chain transition matrix (Equation (11)) to account for channel-dependent link reliability. It would also open new avenues for jointly optimizing spectrum allocation and routing update policies [25,26].

A fourth and particularly fertile direction lies at the intersection of the random graph modeling framework developed here and recent advances in graph-based machine learning [27]. The time-evolving adjacency matrix representation introduced in Section 2 (Equation (1)) constitutes a natural input structure for graph neural networks, which are specifically designed to learn from relational and topological data. Such models could be trained to predict future link states, estimate the structural difference defined in Equation (10), or directly infer the optimal routing update interval from the current graph topology, potentially outperforming the feedforward neural network architecture of Section 7 by explicitly exploiting the spatial structure of vehicular connectivity.

Finally, the local parameter estimation procedure of Section 5—which relies on each node accumulating observations over M time slots to estimate link probability (Equation (36)), persistence probability (Equation (39)), and failure rate (Equation (40))—could be replaced by a fully distributed and collaborative learning paradigm. Federated learning frameworks [21], in which vehicles collaboratively train a shared routing model without exchanging raw topology observations, would preserve privacy, reduce communication overhead, and enable the network to learn richer and more generalizable models of link dynamics than any individual node could achieve in

isolation. Such an approach would transform the AI-driven adaptive mechanism of Section 7 from a node-centric predictor into a truly cooperative, self-organizing routing intelligence capable of responding autonomously and efficiently to the rapidly evolving topological configurations of next-generation intelligent transportation system infrastructures [18,25].

11. Conclusion

This paper develops a unified analytical framework for proactive routing in VANETs, combining a time-evolving random geometric graph with Poisson-distributed vehicle positions and two-state Markovian link dynamics. Closed-form expressions are derived for routing overhead, route error probability, and the optimal update interval that minimizes a weighted cost function. A key finding is the identification of a sensitivity cliff in the overhead-weight parameter λ, beyond which proactive routing becomes ineffective in sparse VANET scenarios. The analysis of OLSR shows that multipoint relay selection achieves logarithmic scaling of control overhead, confirming its efficiency in dense networks.

To address the non-stationarity of real-world vehicular environments, an AI-driven adaptive controller is introduced that learns a mapping from the network state (link probabilities, node degree, and vehicle density) to the optimal update interval using supervised or reinforcement learning. Simulations over an urban-scale scenario involving 100–500 vehicles demonstrate that the proposed AI-OLSR scheme reduces routing control overhead by 35–50% compared with fixed-interval OLSR, while maintaining comparable packet delivery ratios and preserving giant-component connectivity above 95%. The empirical link-lifetime distribution closely follows the geometric decay predicted by the Markovian model, thereby validating the underlying stochastic assumptions. This work provides a rigorous, scalable, and adaptive foundation for next-generation vehicular routing architectures.

Funding: This research received no external funding. The author conducted the study independently without financial support from any public, commercial, or non-profit funding organizations.

Institutional review board statement: Not applicable.

Informed consent statement: Not applicable.

Data availability statement: The data used in this study are available from the corresponding author upon reasonable request.

Acknowledgment: The author thanks the reviewers and the editor for their valuable comments and constructive suggestions throughout the review process. Their insightful feedback has greatly enhanced the quality of this manuscript and has been instrumental in bringing it to its current level of quality.

Conflict of interest: The author declares no conflict of interest.

AI use statement: The author declares that no artificial intelligence (AI) tools were used in the preparation of this manuscript.

References

1. Erdös P, Rényi A. On the Evolution of Random Graphs. Mathematical Institute of the Hungarian Academy of Sciences; 1960. pp. 344–347.
2. Bollobás B. Random Graphs, 2nd ed. Cambridge University Press; 2011.
3. Penrose M. Random Geometric Graphs. Oxford University Press; 2003.
4. Hamlili A. Techniques for Analyzing Random Graph Dynamics and Their Applications. In: Zhang Y (editor). New Frontiers in Graph Theory. InTech; 2012. doi: 10.5772/34207
5. Hamlili A. Intelligibility of Erdös–Rényi random graphs and time varying social network modeling. In: Proceedings of the International Conference on Smart Digital Environment (ICSDE'17); 21–23 July 2017; Rabat, Morocco. pp. 201–206. doi: 10.1145/3128128.3128159
6. Viriyasitavat W, Bai F, Tonguz OK. Dynamics of Network Connectivity in Urban Vehicular Networks. IEEE Journal on Selected Areas in Communications. 2011; 29(3): 515–533.
7. Gupta P, Kumar PR. Critical power for asymptotic connectivity in wireless networks. In: McEneaney WM, Yin GG, Zhang Q (editors). Stochastic Analysis, Control, Optimization and Applications. Birkhäuser; 1999. pp. 547–566. doi: 10.1007/978-1-4612-1784-8_33
8. Cui Q, Hu X, Ni W, et al. Vehicular mobility patterns and their applications to Internet-of-Vehicles: A comprehensive survey. Science China Information Sciences. 2022; 65: 211301. doi: 10.1007/s11432-021-3487-x
9. Fiore M, Härri J. The networking shape of vehicular mobility. In: Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'08); 26–30 May 2008; Hong Kong, China. pp. 261–272. doi: 10.1145/1374618.1374654
10. Zhu H, Li M. Studies on Urban Vehicular Ad-Hoc Networks. Springer; 2013.
11. Naskath J, Paramasivan B. A Study on Modeling Vehicle Mobility with MLC for Enhancing Vehicle-to-Vehicle Connectivity in VANET. Journal of Ambient Intelligence and Humanized Computing. 2020; 12: 8255–8264.
12. Vivitha Varshini RJ, Jahangeer N, Paramasivan B. High Speed Realistic Mobility Model for TN-Multi Lane Highway Environment. International Journal of Engineering and Technology. 2018; 7(4.5): 151–154.
13. Stepanyants VG, Romanov AY. A Survey of Integrated Simulation Environments for Connected Automated Vehicles: Requirements, Tools, and Architecture. IEEE Intelligent Transportation Systems Magazine. 2024; 16(2): 6–22.
14. Stepanyants VG, Romanov AY. Integrating OPAL into CAVISE for Connected Vehicle Simulation with 3D Signal Propagation. IEEE Access. 2025; 13: 134793–134803.
15. Ateya AA, Tu ND, Muthanna A, et al. QL-AODV: Q-Learning-Enhanced Multi-Path Routing Protocol for 6G-Enabled Autonomous Aerial Vehicle Networks. Future Internet. 2025; 17(10): 473. doi: 10.3390/fi17100473
16. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature. 2015; 518: 529–533.
17. Clausen T, Jacquet P. IETF RFC 3626: Optimized Link State Routing Protocol (OLSR). RFC Editor; 2003.
18. ETSI EN 302 663 V1.3.1. Intelligent Transport Systems (ITS); ITS-G5 Access Layer Specification for Intelligent Transport Systems Operating in the 5 GHz Frequency Band. 2020.
19. Dai Y, Lyu L, Cheng N, et al. A Survey of Graph-Based Resource Management in Wireless Networks—Part II: Learning Approaches. IEEE Transactions on Cognitive Communications and Networking. 2025; 11(4): 2101–2122.
20. Sutton RS, Barto AG. Reinforcement Learning: An Introduction, 2nd ed. MIT Press; 2018.
21. Gupta A, Fernando X. Federated Reinforcement Learning for Collaborative Intelligence in UAV-Assisted C-V2X Communications. Drones. 2024; 8(7): 321.
22. Sommer C, Joerer S, Segata M, et al. How shadowing hurts vehicular communications and how dynamic beaconing can help. IEEE Transactions on Mobile Computing. 2014; 14(7): 1411–1421.
23. Nzouonta J, Rajgure N, Wang G, et al. VANET routing on city roads using real-time vehicular traffic information. IEEE Transactions on Vehicular Technology. 2009; 58(7): 3609–3626.
24. Gote PM, Kumar P, Verma P, et al. From 5G to 6G: The role of AI, machine learning, and deep learning in wireless systems. In: Proceedings of the 4th International Conference on Sentiment Analysis and Deep Learning; 18–20 February 2025; Bhimdatta, Nepal.
25. Omheni N, Koubaa H, Zarai F. Artificial Intelligence for 5G and 6G Networks: A Taxonomy-Based Survey of Applications, Trends, and Challenges. Technologies. 2025; 13(12): 559. doi: 10.3390/technologies13120559
26. Ahmad AM, Majeed SA, Dawood YS. A Survey of 6G Mobile Systems, Enabling Technologies, and Challenges.

International Journal of Electrical and Electronic Engineering and Telecommunications. 2023; 12(1): 1–21.
27. Kandu S, Prakash A, Tiwari M, et al. C-V2X and 5G-V2X: Challenges and Improvement Strategies. In: Proceedings of the 3rd International Conference on Microwave, Antenna and Communication (MAC); 27–29 June 2025; Bhopal, India.

Appendix A. Notation

The comprehensive notation adopted in this work reflects the multidisciplinary nature of the proposed framework, encompassing graph theory, stochastic geometry, Markov chain modeling, routing protocol analysis, and machine learning. For clarity and ease of reference, **Table A1** summarizes all notations used throughout the paper. These notations are organized into eight thematic categories: network structure, stochastic geometry, link dynamics, routing analysis, performance metrics, connectivity and stability, AI-driven adaptive control, and local parameter estimation. Each entry provides the corresponding symbol, its precise mathematical definition, and its domain of application.

Table A1. Terminology and notation definition.

Notation	Definition
Graph and network structure	
N	Total number of vehicles
$\mathcal{V}$	Set of N vehicles (nodes) in the network
$\mathcal{E}(t)$	Set of active communication links (edges) at time t
$\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$	Time-evolving network graph at time t
$A(t) \in \{0,1\}^{N \times N}$	Adjacency matrix of the network at time t
$A_{uv}(t)$	Entry of the adjacency matrix: 1 if the link (u, v) is active at time t, 0 otherwise
$\deg_u(t)$	Degree of node u at time t: the number of its direct neighbors
$\overline{\deg}(t)$	Empirical average node degree at time t
ζ	Theoretical average node degree
Δt	Elementary observation time step
$\Delta A(\Delta t, t)$	Absolute difference of adjacency matrix over interval Δt
$\delta_{uv}(\Delta t, t)$	Structural difference of link (u, v) over interval Δt
Spatial and stochastic geometry	
$\mathcal{A}$	Bounded deployment area (domain)
ρ	Spatial density of vehicles (nodes per unit area)
R	Communication range (radius)
p	Edge connection probability (stationary)
λ	Intensity (density of vehicles) of the homogeneous PPP
$X_u(t)$	Position of vehicle u at time t
$d^{'}(u, v)$	Euclidean distance between nodes u and v
$p_{urban}(t)$	Spatially heterogeneous connection probability in urban environments
Markov chain link dynamics	
$t_k = k\Delta t$	Discrete time instant at step k
P	Two-state, $\{0, 1\}$, Markov chain transition matrix
α	Link persistence probability: $\Pr(\ A_{uv}(k) = 1 \mid A_{uv}(k-1) = 1)$
β	Void persistence probability: $\Pr(\ A_{uv}(k) = 0 \mid A_{uv}(k-1) = 0)$
γ	Link failure probability: $\Pr(\ A_{uv}(k) = 0 \mid A_{uv}(k-1) = 1)$
δ	Link formation probability: $\Pr(\ A_{uv}(k) = 1 \mid A_{uv}(k-1) = 0)$
π_0, π_1	Stationary distribution of the Markov chain
μ	Continuous-time link failure rate
v_R	Relative velocity of two vehicles
σ^2	Diffusion coefficient for the Brownian motion mobility model
T_L	Link lifetime (random variable)

Table A1. *Cont.*

Notation	Definition
$E[T_L]$	Expected link lifetime
L	Number of consecutive time steps a link remains active
Routing protocol and overhead	
τ	Routing update interval (time between consecutive control messages)
K	Number of elementary steps per update interval: $\tau = K\Delta t$
S_c	Size of a single control message (bits)
S_{id}	Size of a node identifier (bits)
S_{hdr}	Header overhead per control message (bits)
$\mathcal{O}(\tau)$	Total routing control overhead per second (bits/s)
$\langle \mathcal{E} \rangle$	Expected number of active links in the network
$C(\tau)$	Number of link state changes (births and deaths) over the interval τ
$\widehat{\mathcal{G}}_u(t)$	Topology (graph) estimate maintained by node u at time t
$\widehat{A}^{(u)}(t)$	Estimated adjacency list associated with $\widehat{\mathcal{G}}_u(t)$
Routing performance metrics	
$\varepsilon_u(t)$	Routing error at node u (Frobenius norm)
$p_e(\tau)$	Probability that a link has failed by time τ: $1 - e^{-\mu\tau}$
$p_{\text{path}}(\tau)$	Probability that an h-hop path remains valid after time τ
h	Number of hops along a routing path
$\mathcal{J}(\tau)$	Joint cost function balancing overhead and routing error
$\lambda \in [0, 1]$	Overhead weighting parameter in the cost function
τ^*	Optimal routing update interval minimizing $\mathcal{J}(\tau)$
OLSR and MPR analysis	
$E[N_{\text{MPR}}]$	Expected number of Multipoint Relays (MPRs) required
Connectivity and stability	
ρ^*	Required vehicle density for network connectivity
$S(t, \Delta t)$	Topology stability metric over interval Δt
$E[S(t, \Delta t)]$	Expected topology stability under the Markov chain model
AI-driven adaptive controller	
f_θ	Learned mapping from network state to optimal update interval
θ	Parameters of the machine learning model
$\mathcal{S}(t)$	Predicted system state vector at time t
$\mathbf{x}(t)$	Predicted system-state vector at time t under the supervised learning framework
$\mathbf{s}(t)$	Predicted system-state vector at time t under the unsupervised learning framework
S_t	RL state: $\left(\overline{\deg}(t), \gamma_t, \delta_t\right)'$
$a_t = \tau_t$	RL action: selected update interval at time t
r_t	Immediate reward in the reinforcement learning formulation
$J_{\text{RL}}(\tau)$	RL cost function (MDP formulation)
$\varepsilon_{\text{route}}$	Route-level error term in the RL cost function
τ_{ref}	Reference update interval in the RL cost function
υ	Vehicle speed
$\varsigma \in [0, 1]$	Trade-off parameter in the RL cost function
$\xi \in [0, 1]$	Discount factor in the RL objective
$\langle L \rangle$	Expected path length in hops: $\approx \frac{\log(N)}{\log\left(\overline{\deg}\right)}$
Local parameter estimators	
M	Number of observation time slots considered for parameter estimation
$\widehat{p}$	Estimated stationary link probability
$\widehat{\alpha}$	Estimated link persistence probability

Table A1. *Cont.*

Notation	Definition
$\widehat{\beta}$	Estimated void persistence probability
$\widehat{\gamma}$	Estimated link failure probability
$\widehat{\delta}$	Estimated link formation probability
$\widehat{\mu}$	Estimated continuous-time failure rate
$N_{00}, N_{01}, N_{10}, N_{11}$	Observed transition counts between link states

Table A1 is designed to serve as a self-contained reference throughout the paper. Whenever a symbol appears in a derivation, theorem, or simulation result, the reader may consult **Table A1** to retrieve its precise definition and understand its role within the analytical framework. This reference structure is particularly valuable at the intersection of the stochastic modeling developed in Sections 2–5, the protocol analysis presented in Section 6, and the AI-driven adaptive framework described in Sections 7–8, where notation originating from multiple mathematical and computational domains is integrated into a unified formulation.