Decentralized and Adaptive Internet of Vehicles: A Blockchain-Based Approach

JIAWEI SHI, Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications, Beijing, China
YEBO FENG, Nanyang Technological University, Nanyang Technological University, Singapore, Singapore
KONGLIN ZHU, Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications, Beijing, China and Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance, Beijing, China
TINGDA SHEN, Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications, Beijing, China
LIN ZHANG, Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications, Beijing, China

The Internet of Vehicles (IoV) enhances road safety and supports autonomous driving through real-time communication, but current methods face key challenges: rigid resource allocation due to static architectures, communication failures in low-signal areas from infrastructure reliance, and passive defense mechanisms struggle to counter coordinated attacks, while high-latency encryption algorithms further compromise framework real-time performance. To address this, we propose a blockchain-based dynamically adaptive restructuring framework. It enables real-time IoV cluster restructuring by splitting overloaded IoVs to reduce communication overhead, or merging nearby IoVs to optimize resource utilization. In infrastructure-sparse zones, vehicles establish temporary multi-hop communication links based on relative mobility to ensure continuous connectivity. A multi-layered security mechanism integrates physical validation, event verification, and majority voting, achieving over 95% resistance to data tampering. Compared to Raft, PoS, and PBFT, our framework improves consensus speed by 27.06%–38.56%, and reduces transaction latency by 7%–35%, 15%–54%, and 27%–66%, respectively. It also maintains high robustness under dense traffic, high mobility, and weak signals, offering a proactive, adaptive security paradigm for intelligent transportation frameworks.

This work was supported in part by the National Key Research and Development Program of China (2023YFB2704500), in part by the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, in part by the Science and Technology Program of Xizang Autonomous Region under Grant XZ202401ZY0027, in part by the Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance, and in part by the Ningbo Natural Science Foundation under Grant 2023J285.
Authors' Contact Information: Jiawei Shi, Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications, Beijing, Beijing, China; e-mail: shijiawei@bupt.edu.cn; Yebo Feng (corresponding author), Nanyang Technological University, Singapore, Singapore; e-mail: yebo.feng@ntu.edu.sg; Konglin Zhu, Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications, Beijing, Beijing, China and Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance, Beijing, China; e-mail: klzhu@bupt.edu.cn; Tingda Shen, Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications, Beijing, Beijing, China; e-mail: shentingda@bupt.edu.cn; Lin Zhang, Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications, Beijing, Beijing, China; e-mail: zhanglin@bupt.edu.cn.

ACM 1533-5399/2026/07-ART41
https://doi.org/10.1145/3820161

CCS Concepts: • **Distributed architectures** → **Peer-to-peer architectures**; • **Distributed algorithms** → **Self-organization**; • **Network mobility**; • **Information systems applications** → *Mobile information processing systems*; • **Network dynamics**;

Additional Key Words and Phrases: Blockchain, Internet of Vehicles (IoV), distributed networks, smart contracts

ACM Reference Format:
Jiawei Shi, Yebo Feng, Konglin Zhu, Tingda Shen, and Lin Zhang. 2026. Decentralized and Adaptive Internet of Vehicles: A Blockchain-Based Approach. *ACM Trans. Internet Technol.* 26, 3, Article 41 (July 2026), 27 pages. https://doi.org/10.1145/3820161

1 Introduction

The **Internet of Vehicles** (**IoVs**) enables real-time data exchange between vehicles, enhancing road safety and traffic efficiency. For instance, in hazardous situations, **vehicle-to-vehicle** (**V2V**) and **vehicle-to-infrastructure** (**V2I**) communication allows the IoV to deliver real-time traffic information [31], helping drivers anticipate hazards and reduce accidents [14]. IoV also supports intelligent traffic management to ease congestion [3], and provides a foundational infrastructure for autonomous driving by enabling vehicles to better perceive their environment and make safer decisions.

Current IoV research mainly focuses on data transmission performance and security. To improve efficiency, researchers optimize road infrastructure and reduce network congestion. RSU deployment strategies across various scenarios aim at maximizing coverage and performance while minimizing investment [15, 30, 45]. For congestion control, real-time traffic monitoring and big data analysis support dynamic vehicle routing [11]. Meanwhile, **Software-Defined Networking** (**SDN**) and **Network Function Virtualization** (**NFV**) technologies are used to flexibly allocate network resources [36, 44]. However, most studies focus on standardized scenarios, neglecting challenging environments like mountainous areas or underground parking lots. In areas lacking base station coverage, traditional infrastructure-dependent networks often fail to maintain stable V2V communication, leading to service interruptions and delayed transmission of safety information. The static structure of IoV also limits dynamic reorganization based on real-time traffic, leading to inefficient resource use. Centralized SDN controllers struggle with frequent mobility and unstable links, while NFV introduces high latency and requires stable connectivity and resource-rich conditions, which are unsuitable for resource-constrained vehicles.

IoV data security research focuses on developing advanced encryption algorithms and security authentication frameworks. **Identity-Based Encryption** (**IBE**) enables anonymous vehicle authentication while preserving data confidentiality and integrity [32, 48]. Multi-layered frameworks enhance security from the physical to application layer, protecting user privacy [26]. However, the high computational complexity of existing encryption algorithms can cause communication delays, compromising the real-time performance of the framework [28]. Moreover, passive cryptographic schemes (e.g., Identity-Based Encryption, IBE) lack proactive defenses against coordinated **false data injection** (**FDI**) attacks.

To address these challenges, we propose a blockchain-based IoV framework that supports dynamic networking, adapts to diverse environments, and enables decentralized, secure data sharing. Leveraging blockchain's consensus mechanism and tamper-proof ledger, the framework achieves reliable, infrastructure-independent communication while enhancing resilience against attacks and single points of failure. At its core is an adaptive restructuring mechanism that adjusts communication clusters and allocates resources based on real-time traffic and vehicle mobility, reducing congestion and improving efficiency. To maintain stable communication in high-mobility or infrastructure-poor

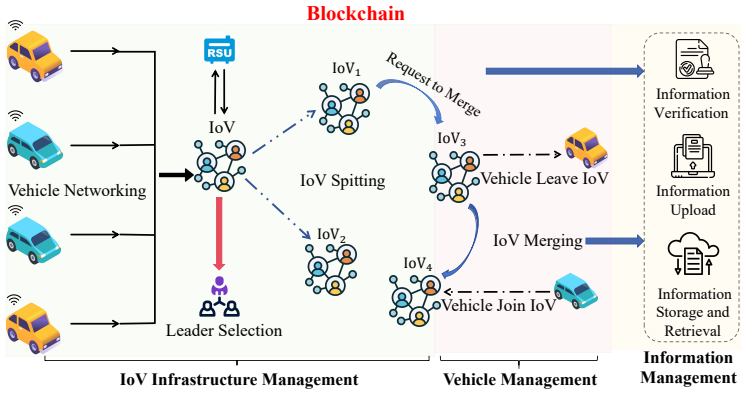


Fig. 1. The architecture of the proposed framework, which consists of IoV infrastructure management, vehicle management, and information management. All activities associated with these components are recorded on the blockchain to ensure transparency, traceability, and integrity of transactions.

areas (e.g., tunnels, mountains), it employs a mobility-aware multi-hop strategy based on vehicle positions. For security, the framework uses a multi-layered defense mechanism combining physical reachability validation, event authenticity verification, and majority-based decision-making. This reduces reliance on heavy cryptography while ensuring integrity and responsiveness in critical situations. Through these mechanisms, the proposed framework overcomes the limitations of static IoV frameworks, ensures stable communication in challenging environments, and defends against coordinated false data attacks. It consists of three main modules (see Figure 1).

(1) **IoV Infrastructure Management:** This module ensures flexibility and scalability in the IoV by handling dynamic changes. It includes leader election, IoV splitting, and IoV merging algorithms. The leader election algorithm selects a coordinator to enhance IoV management efficiency and IoV stability. The IoV splitting algorithm automatically divides overloaded IoVs to reduce congestion and optimize resource allocation, while the IoV merging algorithm integrates nearby IoVs to support information sharing.
(2) **Vehicle Management:** This module manages the dynamic joining and leaving of vehicles in the IoV to ensure efficient resource use. The vehicle joining algorithm integrates approaching vehicles to expand coverage, while the vehicle leaving algorithm removes departing vehicles to avoid resource waste.
(3) **Information Management:** This module ensures secure information management in the IoV by handling real-time data verification, upload, storage, and retrieval. It enhances communication reliability, prevents data tampering and loss, and supports reliable decision-making and effective collaboration among vehicles.

Our evaluation shows that the proposed blockchain-based IoV framework is efficient, secure, and adaptable. The proposed leader election-based consensus mechanism reduced transaction latency by 7%–35%, 15%–54%, and 27%–66% compared to Raft, **Proof-of-Stake** (**PoS**), and **Practical Byzantine Fault Tolerance** (**PBFT**), respectively, and demonstrated linear scalability as the IoV size increased. In urban and mountainous simulations, it achieved over 95% resistance to data tampering using a multi-layered defense mechanism. With up to 15% malicious nodes, FDI success remained near 0%, and consensus integrity was preserved under Byzantine attacks. In areas lacking infrastructure, dynamic vehicle networking keeps the data packet reception rate above 85%. Case studies further validated the framework's adaptability: IoV split/merge operations balanced traffic load and enabled seamless integration during emergencies.

Our contributions are summarized as follows:

— We proposed a blockchain-based IoV framework that supports dynamic vehicle networking, enables decentralized and secure information sharing, and adapts to environments with limited or no infrastructure support.
— To enhance consensus efficiency, we designed a lightweight mechanism incorporating leader election, in which a representative node is selected to coordinate consensus operations. This approach reduces communication overhead and lowers transaction latency.
— We designed a dynamic restructuring mechanism for the IoV that automatically splits during congestion or merges when IoVs are nearby, enabling adaptive scaling and optimizing performance and resource allocation.
— To ensure data security, we also introduced a multi-layered defense mechanism that performs physical reachability validation, event authenticity checks, and majority-based verification prior to uploading data to the blockchain.

The remainder of this article is organized as follows: Section 2 introduces the background and motivation of our framework. Section 3 describes the framework architecture and core design. Section 4 presents a performance evaluation. Section 5 reviews related work. Section 6 discusses the limitations and outlines directions for future research. Finally, Section 7 concludes the article.

2 Background

2.1 Motivation and Problem Statement

The IoV serves as a key enabler of intelligent transportation, enhancing road safety, real-time traffic management, and autonomous driving. Nevertheless, current frameworks face limitations in scalability, resilience, and reliability, particularly in dynamic or infrastructure-deficient environments.

First, connectivity remains unstable in infrastructure-deficient areas such as mountains and tunnels, where traditional infrastructure-dependent architectures fail to support reliable V2V and V2I communication, causing service interruptions and safety delays.

Second, static network architectures cannot adapt to dynamic mobility and traffic changes. Centralized schemes like SDN and NFV require stable, resource-rich environments, making them unsuitable for mobile, resource-limited IoV networks. Without adaptive reconfiguration, resources are used inefficiently, latency increases, and congestion may occur in dense scenarios.

Third, current security mechanisms struggle to balance protection and performance. While methods like IBE and multi-layer authentication ensure privacy, they incur high computational costs and offer limited defense against coordinated attacks. Their passive design also lacks the proactive verification needed for real-time vehicular environments.

These limitations underscore the necessity for a decentralized, adaptive, and security-enhanced IoV framework that can (i) operate effectively in environments with intermittent or absent infrastructure, (ii) dynamically restructure IoV clusters to optimize resource distribution and reduce communication delays, and (iii) incorporate lightweight, proactive defense mechanisms to ensure data verifiability and resistance against adversarial behaviors. Addressing these challenges is essential to realizing a robust, scalable, and trustworthy IoV ecosystem suitable for future intelligent transportation applications.

2.2 Formalization

The IoV is a cyber-physical framework that integrates vehicles, infrastructure, pedestrians, and networks to enable intelligent transportation and cooperative services. We define the key entities as follows:

— Vehicles: $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, where each v_i has communication and computing capabilities.
— Infrastructure: $\mathcal{F} = \{f_1, f_2, \ldots, f_M\}$, including base stations, RSUs, traffic lights, and cameras.

Table 1. Notation Summary

Category	Symbol	Description
Entities	$\mathcal{V}$	Set of vehicles: $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$
	$\mathcal{F}$	Set of infrastructure nodes: $\mathcal{F} = \{f_1, f_2, \ldots, f_M\}$
	$\mathcal{P}$	Set of pedestrians: $\mathcal{P} = \{p_1, p_2, \ldots, p_L\}$
	$G = (N, E)$	Road network graph where N represents intersections or infrastructure points, and E represents connecting roads.
Vehicle Attributes	(x_i, y_i)	Position coordinates of vehicle v_i
	$\vec{v}_i$	Velocity vector of vehicle v_i
	θ_i	Direction of vehicle v_i
Communication Actions	V2V(v_i, v_j)	Direct data exchange between vehicles v_i and v_j
	V2I(v_i, f_k)	Communication between vehicle v_i and infrastructure node f_k
	V2P(v_i, p_l)	Communication between vehicle v_i and pedestrian p_l
Sensing and Computation	Sense(e)	Sensing operation performed by entity $e \in \mathcal{V} \cup \mathcal{F}$
	Compute(e, D)	Computation or decision-making process performed by entity e using dataset or input information D

— Pedestrians: $\mathcal{P} = \{p_1, p_2, \ldots, p_L\}$, who interact via mobile or wearable devices.
— Environment: A road network graph $G = (N, E)$, where N are intersections or infrastructure points, and E are connecting roads.

Each vehicle v_i has a position (x_i, y_i), velocity $\vec{v}_i$, and direction θ_i. IoV operations are characterized by a variety of actions, including but not limited to

— V2V Communication: Denoted as V2V(v_i, v_j), indicating a direct data exchange between vehicles v_i and v_j.
— V2I Communication: V2I(v_i, f_k), referring to communication between vehicle v_i and infrastructure f_k.
— Vehicle-to-Pedestrian Communication: V2P(v_i, p_l), used for safety notifications or route awareness.
— Sensing and Perception: Each vehicle and infrastructure node collects data from the physical environment, expressed as Sense(e), where $e \in \mathcal{V} \cup \mathcal{F}$.
— Computation and Decision Making: Vehicles and infrastructure units process data and make decisions, noted as Compute(e, D), where D is the dataset or input information.

These entities and operations form a dynamic, distributed IoV that supports real-time data sharing, cooperative driving, traffic optimization, and road safety. The key notations have been summarized in Table 1.

2.3 Threat Model

Our framework operates under a partially trusted environment, where the majority of nodes are assumed to behave honestly, and up to f out of N total nodes may act maliciously. We adopt the Byzantine adversary model, in which malicious nodes may conduct attacks, including but not limited to the following:

— Location Spoofing: Falsifying geographic positions to mislead other vehicles about the attacker's actual position.

— Message Tampering: Intentionally modifying the content of legitimate messages during transmission to distort traffic information, such as vehicle states or incident reports.
— Message Suppression: Disrupting communication by jamming channels or selectively dropping key messages, thereby hindering consensus processes, increasing packet loss, and reducing message credibility.
— Maliciously delaying the verification or acknowledgment of transmitted information to disrupt consensus, block communication processes, and degrade overall framework responsiveness.

We exclude Sybil attacks, as they are considered impractical in IoV due to strong physical presence and identity binding requirements. Within this context, we make the following security assumptions to define the adversarial model:

— Cryptographic Assumptions: All used cryptographic primitives (e.g., digital signatures) are assumed to be secure. Adversaries are computationally bounded and cannot break these primitives within polynomial time.
— Authenticated Channels: Communications between honest nodes are authenticated and protected against spoofing or tampering. Secure transport layers prevent Man-in-the-Middle attacks.
— No Physical Tampering: Adversaries have no access to or ability to tamper with the physical hardware of honest vehicles or sensors. Attacks are limited to protocol-level manipulation rather than hardware-level intrusion.

Under this threat model, the framework integrates layered defenses, including physical reachability validation, event authenticity verification, and majority voting mechanisms to mitigate the impact of compromised nodes and ensure data integrity and trustworthiness in dynamic IoV environments.

3 Methodology

This section presents the architecture and core design of our blockchain-based IoV framework, addressing adaptability, scalability, and security in dynamic IoV environments. Central to the framework is a decentralized restructuring mechanism that reorganizes vehicle groups in real time based on traffic density, communication quality, and geographic conditions.

The framework consists of three integrated modules: IoV infrastructure management for leader election, IoV splitting, and merging to ensure load balancing and flexibility; vehicle management for handling dynamic vehicle joining and leaving IoV; and information management for secure storage, retrieval, upload, and multi-layered verification to maintain data reliability and integrity. Figure 2 shows the structure and interactions of the modules. The following sections introduce the core mechanisms and design principles of each module.

3.1 IoV Infrastructure Management

To support adaptive and decentralized operation, the IoV infrastructure management module incorporates core algorithms, including leader election for coordination, IoV splitting for load balancing, and merging for efficient reintegration. These algorithms form the basis of a self-organizing IoV capable of responding in real time to traffic dynamics, vehicle mobility, and network conditions. The following subsections detail the design and functionality of each algorithm.

3.1.1 Consensus Establishment. In distributed IoV frameworks with blockchain or decentralized coordination, electing a leader is essential for managing communication, task scheduling, and consensus. Prior work, such as Tendermint and PBFT [6, 24], shows that leader-based consensus improves efficiency and reduces overhead, enhancing the scalability and robustness of decentralized frameworks. Building on these insights and the dynamic nature of IoV, where node composition

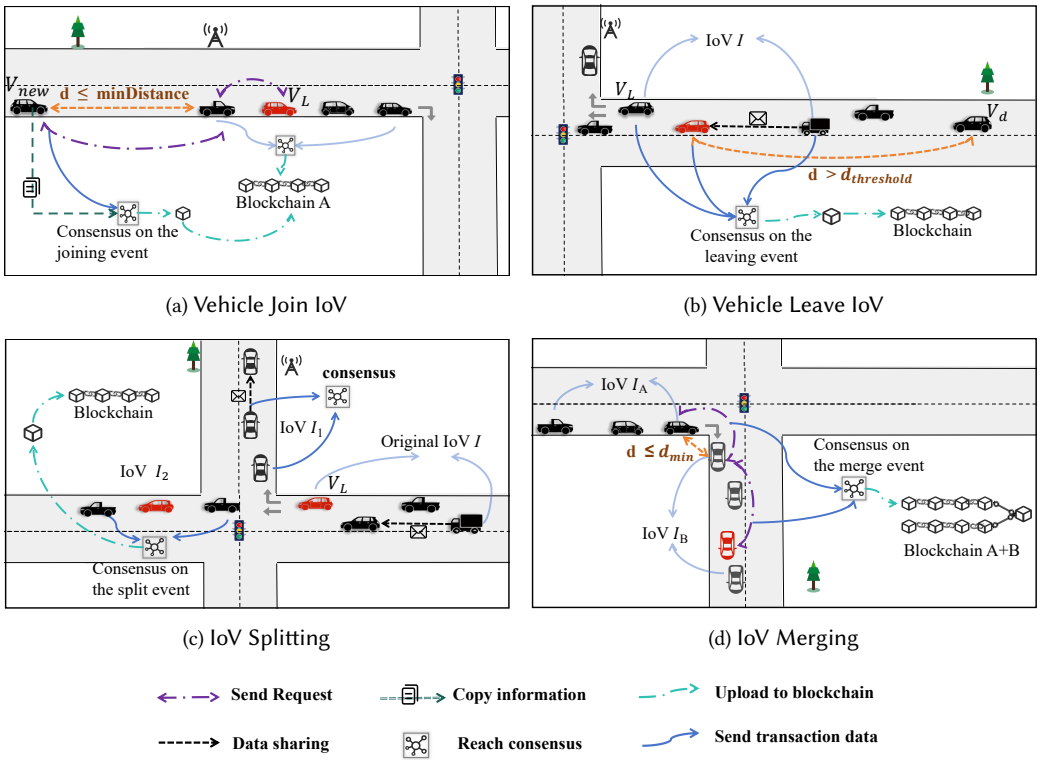


Fig. 2. Illustrations of vehicle operations and how the proposed IoV framework addresses them.

changes frequently, we propose a two-stage leader election algorithm. It combines Gossip-based configuration sharing with performance-based deterministic selection to choose the most capable and stable node. This design ensures efficient, resilient coordination without relying on centralized control.

Compared to random or rotating leader selection, our method offers multiple advantages: (1) Adaptability to dynamic environments: selects candidates based on geographic centrality and network connectivity; (2) Security and tamper-resistance: the leader is selected based on attributes such as computing capability and bandwidth, which are hardware-constrained and less prone to manipulation; (3) Scalability: the Gossip-based configuration exchange avoids broadcast storms and enables efficient local convergence of state; (4) Fault tolerance: the framework incorporates a heartbeat mechanism for failure detection and enables fast re-election, thereby maintaining operational continuity.

The election process (Algorithm 1) begins with each vehicle sharing configuration data (e.g., computing power, bandwidth) using the Gossip protocol. The node with the highest computing power initiates the election, selects the central $\frac{2}{3}$ of nodes based on geographic position as candidates, and elects the one with the highest bandwidth as the leader.

We specifically choose computing power and bandwidth as election metrics. These attributes are closely tied to physical hardware and secured by cryptographic verification and secure firmware, making them difficult to forge or falsify. To ensure stability, all nodes use a heartbeat mechanism to monitor the leader. If three consecutive heartbeats are missed, a re-election is automatically triggered.

ALGORITHM 1: Leader Election Algorithm.

Require: Configuration set $\{C_i = (\text{comp}_i, \text{bw}_i) \mid V_i \in \mathcal{V}\}$,
where comp_i and bw_i denote the computing power and bandwidth of node V_i.
Ensure: Elected leader node V_L.
Each node $V_i \in \mathcal{V}$ broadcasts C_i using a Gossip protocol
if All C_i have been received **then**
  $V_k \leftarrow \arg\max_{V_i \in \mathcal{V}} \text{comp}_i$
end if
if V_k receives $> \frac{2}{3}|\mathcal{V}|$ valid votes **then**
  Execution node V_k initiates leader election
  Let $\mathcal{C} \subseteq \mathcal{V}$, $|\mathcal{C}| = \left\lfloor \frac{2}{3}|\mathcal{V}| \right\rfloor$, such that $V_i \in \mathcal{C}$ have minimal geographic dispersion
  $V_L \leftarrow \arg\max_{V_i \in \mathcal{C}} \text{bw}_i$
  Broadcast V_L and synchronize ledgers across all $V_i \in \mathcal{V}$
else
  Restart election process
end if

ALGORITHM 2: IoV Splitting Algorithm.

Require: $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$: confirmation messages from nodes;
N_c: current size of IoV, computed as $|\mathcal{N}(V_L)| + 1$;
T_{split}: threshold size for triggering split;
$\text{Delay}_{\text{last}}$: average delay in the last minute;
$\alpha \in (0, 1]$: required confirmation ratio.
Ensure: Two disjoint IoVs $\mathcal{I}_1, \mathcal{I}_2$ with ledgers updated and leader elected in $\mathcal{I}_2$.
Initialize split $\leftarrow$ False
while ¬split **do**
  $N_c \leftarrow |\mathcal{N}(V_L)| + 1$ ▷ Include leader V_L itself
  if $N_c > T_{\text{split}} \wedge \text{Delay}_{\text{last}} > 500$ ms **then**
    Broadcast signed split request to all $V_i \in \mathcal{V}$
    $\mathcal{F} \leftarrow \{F_i \mid V_i \in \mathcal{V},\ \text{response}(V_i) = F_i,\ \text{within } \Delta t\}$ ▷ Collect responses within a fixed time window
    if $\sum_{i=1}^{|\mathcal{F}|} \mathbb{I}(F_i = \text{confirm}) \geq \alpha \cdot |\mathcal{V}|$ **then**
      Partition $\mathcal{V}$ into $(\mathcal{I}_1, \mathcal{I}_2)$ such that $\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{V}$ and $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$
      $V_L^{\text{new}} \leftarrow \text{LeaderElection}(\mathcal{I}_2)$
      split $\leftarrow$ True and synchronize ledgers
    end if
  end if
end while

3.1.2 IoV Splitting. As the IoV grows, the rising number of connected vehicles creates scalability challenges. Large consensus groups lead to higher latency, leader overload, and reduced throughput, especially in dense traffic or emergencies that require real-time response. Static IoV structures cannot adapt to changing vehicle locations or communication needs. Thus, a self-restructuring mechanism is essential to maintain operational efficiency under dynamic conditions.

To address scalability issues, we propose an IoV splitting algorithm (Algorithm 2) that partitions overloaded IoVs into smaller sub-IoVs under high-load conditions. The leader monitors two conditions: (1) the number of non-leader nodes exceeds threshold T_{split}, and (2) the average transaction delay over 60 seconds surpasses 500 ms. Both must be met to trigger a split, ensuring it's due to overload rather than mere density. In addition, we have introduced a splitting cooldown period

ALGORITHM 3: Node Allocation Algorithm.

Require: $\text{Distance}(V_i, V_j) \in \mathbb{R}^+$;
L_{original}: The leader node of the IoV before splitting;
L_{new}: The new leader generated after the IoV splits;
$\delta_{\min}$: The minimum distance from L_{original}.
Ensure: Two independent IoVs with allocated nodes.
Step 1: Geo-Constraint Prefiltering
$\mathcal{V}_{\text{eligible}} \leftarrow \emptyset$
for each node $V_i \in \mathcal{V}$ **do**
  if $\text{Distance}(V_i, L_{\text{original}}) \geq \delta_{\min}$ **then**
    $\mathcal{V}_{\text{eligible}} \leftarrow \mathcal{V}_{\text{eligible}} \cup \{V_i\}$ ▷ Select nodes far enough from original leader
  end if
end for
Step 2: Score Computation and Leader Proposal
for each $V_i \in \mathcal{V}_{\text{eligible}}$ **do**
  Compute $\text{Score}_i = \frac{3}{5} \cdot \frac{\text{Computation}_i}{\max(\text{Computation})} + \frac{2}{5} \cdot \frac{\min(\text{Proximity})}{\text{Proximity}_i}$
end for
$L_{\text{new}} \leftarrow \arg\max_{V_i \in \mathcal{V}_{\text{eligible}}} \text{Score}_i$ ▷ Elect node with highest score as new leader
Step 3: Assign Nodes to IoVs Based on Distance
for each node $V_i \in \mathcal{V}$ **do**
  Compute $d_{\text{original}} = \text{Distance}(V_i, L_{\text{original}})$
  Compute $d_{\text{new}} = \text{Distance}(V_i, L_{\text{new}})$
  if $d_{\text{new}} < d_{\text{original}}$ **then**
    V_i signs vote: $(\text{assign}, V_i, \mathcal{J}_{\text{new}}, \text{sig}_{V_i})$ ▷ Assign to new IoV if closer
  else
    V_i signs vote: $(\text{assign}, V_i, \mathcal{J}_{\text{original}}, \text{sig}_{V_i})$ ▷ Otherwise stay in original IoV
  end if
  Append all votes to block buffer ▷ Record assignment decisions for consensus
end for

T_{cooldown}, which enforces a minimum waiting interval after each completed split before a new splitting operation can be triggered. This mechanism prevents excessively frequent splits and maintains the stability of the framework. Before splitting, the leader broadcasts a proposal, and peers validate it using local metrics. Upon consensus, the IoV is divided into two sub-IoVs, each with its own leader, ledger, and consensus process.

After an IoV splits into two independent IoVs, nodes must be reallocated to ensure both balanced computational resources and geographical proximity, which are critical for maintaining performance and scalability. This process involves a scoring mechanism to evaluate node suitability and a distance-based assignment to optimize efficiency. The leader node in the original IoV performs the following steps in Algorithm 3:

(1) *Geo-Constraint Prefiltering:* The original leader node within the IoV is retained, and an additional leader is elected to manage the new sub-IoV. Each non-leader vehicle computes its distance to the original leader, and vehicles located within a predefined threshold are filtered out. This prevents geographical overlap between the resulting sub-IoVs, ensuring sufficient spatial separation.
(2) *Score Calculation for Leader Selection:* The leader scores each node based on computational capability (*Computation*) and proximity to others (*Proximity*). The node with the highest score becomes the provisional leader of the new sub-IoV, ensuring strong processing capacity and

ALGORITHM 4: IoV Merging Algorithm.

Require: V_{adj}: adjacent vehicle node from another IoV;
$d_{\min}$: distance threshold for merge trigger;
V_e: current edge node;
V_{L_A}, V_{L_B}: leader nodes of respective IoVs;
N_{c1}, N_{c2}: sizes of the two IoVs;
Ensure: Merge status: `Success` or `Failure`.

Edge Node Behavior:

while ¬merged **do**
  $d \leftarrow \text{Distance}(V_e, V_{\text{adj}})$ ▷ Monitor distance between edge node and adjacent node
  if $d \leq d_{\min}$ **then**
    Send merge_req to V_{L_A}, V_{L_B}
    Wait for authorizations: $\text{Auth}(V_{L_A}), \text{Auth}(V_{L_B})$
    if $\text{Auth}(V_{L_A}) = \texttt{True} \wedge \text{Auth}(V_{L_B}) = \texttt{True}$ **then**
      Create block B_{new} with B_{new}.isMerged ← 1
      Append B_{new} to both $\mathcal{B}_A$ and $\mathcal{B}_B$ ▷ Commit merge block to both ledgers
      merged ← `True`;
      return `Success` ▷ Update merge status and exit loop
    end if
  end if
end while
return `Failure`

Leader Node Behavior (upon merge_req):

if $N_{c1} + N_{c2} \leq T_{\text{split}}$ **then**
  return $\text{Auth}(L) \leftarrow \texttt{True}$ ▷ Approve merge if combined size is acceptable
else
  return $\text{Auth}(L) \leftarrow \texttt{False}$
end if

geographic centrality. Higher computation indicates better processing power, while lower proximity reflects a more central geographic position. The weight ratio between computational capability (3/5) and geographical centrality (2/5) is set based on the role analysis of a leader. A higher weight (3/5) is assigned to computational capability due to the computationally intensive nature of the leader's critical tasks, where the cumulative processing delay is paramount for avoiding IoV splits. Conversely, geographical centrality is assigned a lower weight (2/5) since its impact on reducing communication latency is secondary and can be mitigated via multi-hop mechanisms. Furthermore, the rationality of this specific weight ratio is supported by empirical evidence from our experimental studies, detailed in Appendix C.

(3) *Node Assignment Based on Distance:* Each remaining node calculates its distance to both the original and new leader nodes. Nodes are assigned to the IoV of the closer leader, minimizing communication delays and enhancing IoV efficiency.
(4) *State Update and Blockchain Finalization:* After node assignment, the leader updates the IoV state, finalizing the division. Each blockchain updates its membership, ensuring all nodes are aware of their respective IoV and designated leader.

3.1.3 Blockchain-Based IoV Merging. IoV merging addresses fragmentation and resource underutilization that arise in low-density or post-split scenarios, serving as a complement to IoV splitting. Isolated sub-IoVs may waste resources and hinder data exchange. To address this, the IoV merging algorithm (Algorithm 4) merges adjacent sub-IoVs $\mathcal{I}_A$ and $\mathcal{I}_B$ when: (1) their distance is below $d_{\min}$; (2) the combined node count is below T_{split}; and (3) inter-IoV data exchange is needed.

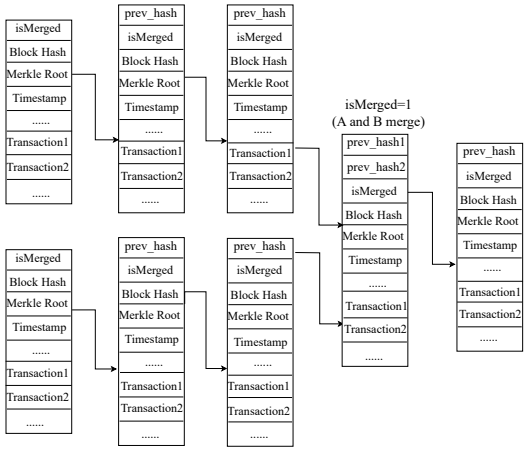


Fig. 3. The data structure used to store information following the merger of two blockchain-based IoVs.

Edge nodes monitor proximity, initiate merge requests, and finalize the process upon dual-leader approval by linking the ledgers.

To support IoV merging, we introduce a block-level data structure, *isMerged*, to indicate the merging status of a block:

- *isMerged* = 0: The block points to a single blockchain, maintaining the traditional structure.
- *isMerged* = 1: The block serves as a "bridge" block, linking two separate blockchains. It contains two previous hash pointers, *prev_hash1* and *prev_hash2*, to ensure continuity in both chains.

A bridge block is a specially designed data structure that facilitates the secure integration of two independent blockchains during IoV cluster merging. It is identified by a designated flag isMerged = 1 and contains cryptographic references to the latest blocks of both predecessor chains. By linking these chains at a convergence point, the bridge block ensures continuity of transaction history, maintains data consistency, and preserves the immutability of both ledgers throughout the merging process. The IoV merging process follows the steps in Figure 3:

(1) *Initialization Phase:* Two separate blockchain-based IoVs (IoV A and IoV B) are initialized.
(2) *Merge Operation:* A merge point is selected, and a block from chain A is combined with one from chain B. The merged block's `isMerged` flag is set to 1, and its `prev_hash` is updated to reference the previous blocks of chains A and B.
(3) *Merge State Processing:* Check the consistency and immutability of the blockchain to ensure that the merged block remains tamper-proof.
(4) *Conflict Handling:* If no conflicts are detected, the merge is finalized. Otherwise, the framework rolls back to the pre-merge state and reattempts the merge.

To ensure that the dynamic reorganization process does not introduce significant delays or inconsistencies, the framework incorporates the following design principles: The splitting mechanism employs dual thresholds and a cooldown period to prevent unnecessary operations triggered by transient load fluctuations. And all splitting and merging actions are contingent upon consensus agreement among peers, ensuring synchronous state transitions and preventing configuration conflicts. For consistency during merges, the novel bridge block mechanism acts as an atomic commit point, guaranteeing that ledger integration is an all-or-nothing operation. Furthermore, post-split node reallocation is optimized for geographical proximity and load balance to swiftly restore

ALGORITHM 5: Vehicle Joining Algorithm.

Require: V_{new}: a new vehicle requesting to join;
minDist: minimum distance threshold for discovery;
isDirSame(V_a, V_b): check direction consistency;
V_L: leader of the IoV.
Ensure: Updated $\mathcal{J} \leftarrow \mathcal{J} \cup \{V_{\text{new}}\}$ if joining successful.
Behavior of V_{new}:
V_{new} senses nearby $V_e \in \mathcal{V}$ with $\Delta L(V_{\text{new}}, V_e) \leq \text{minDist}$
Create message $M = (\texttt{join\_req}, V_{\text{new}}, \text{timestamp})$
V_{new} signs M with private key $\text{sk}_{\text{new}} \rightarrow \text{sig}_{\text{new}}$ ▷ Ensure authenticity of join request
Send $\langle M, \text{sig}_{\text{new}} \rangle$ to V_e, $V_e \rightarrow V_L$ ▷ Forward request to the leader
if Auth(V_L, V_{new}) = `True` **then**
  $\mathcal{J} \leftarrow \mathcal{J} \cup \{V_{\text{new}}\}$ ▷ Join successful; add to IoV
  break
end if
Behavior of V_L:
Upon receiving `join_req` for V_{new}
$N_c \leftarrow |\mathcal{N}(V_L)| + 1$ ▷ Count current members including applicant
if $N_c \leq T_{\text{split}} \wedge$ isDirSame(V_L, V_{new}) **then**
  V_L broadcast `agree` to all $V_i \in \mathcal{V}$ and collects all votes
  if valid `agree` votes $\geq 2/3 \cdot |\mathcal{V}|$ **then**
    return Auth(V_L, V_{new}) $\leftarrow$ `True` ▷ Majority agreed, approve join
  else
    return Auth(V_L, V_{new}) $\leftarrow$ `False` ▷ Not enough support, reject join
  end if
end if

performance in new sub-IoVs. These mechanisms ensure that reorganization is a controlled process with minimal disruption, as evidenced by the scalable operation latencies shown in Section 4.2.2, Figure 5.

3.2 Vehicle Management

This module manages the vehicle participation lifecycle within the IoV, including joining, communication, and leaving. The vehicle joining and leaving algorithms handle access and disconnection, respectively.

3.2.1 Vehicle Join IoV. In dynamic IoV environments, flexible vehicle joining is essential for maintaining coverage, scalability, and data integrity. In its absence, vehicles outside the IoV may be unable to contribute valuable information or benefit from IoV services. To support this, we design a vehicle joining algorithm (Algorithm 5) with three stages: proximity detection, capacity evaluation, and blockchain-based integration. The algorithm ensures that only vehicles within communication range can join, prevents IoV overload, and records join events on-chain for traceability. The process is detailed as follows:

— *Distance Monitoring:* A vehicle not in any IoV monitors its distance to nearby vehicles. If it is within a threshold (*minDistance*) of an IoV edge node, it sends a join request, which the edge node forwards to the IoV leader for further processing.
— *Capacity Evaluation:* The leader node evaluates whether the IoV can accommodate additional members without exceeding the capacity threshold, while ensuring that the vehicles requesting to join are in the same direction as the leader's travel, preventing frequent joining and

leaving of vehicles, and maintaining the relative stability of the dynamic network. If so, the join request is approved.

— *Integration and Verification:* The vehicle can formally join the IoV only if it receives more than 2/3 of the nodes' votes. Its identity and confirmation record are immutably recorded on-chain, ensuring the access event is verifiable. Both leader election and vehicle joining algorithms adopt a 2/3 majority threshold, which follows the classic Byzantine Fault Tolerance principle. This theory shows that in a system with N nodes, to tolerate at most f malicious nodes (Byzantine nodes), two conditions must be satisfied [7], [13], [23], [29]: The total number of nodes satisfies $N \geq 3f + 1$ (meaning malicious nodes cannot exceed 1/3 of the total), and the voting threshold requires the agreement of at least $2f + 1$ nodes, which is approximately 2/3 of the total nodes given $N \geq 3f + 1$.

3.2.2 Vehicle Leave IoV. In decentralized IoVs, detecting vehicles that leave the communication range is critical to avoid IoV state inconsistencies and resource waste. We propose a vehicle leave algorithm that accurately identifies departures by combining distance checks with heartbeat failure detection. This approach balances responsiveness and fault tolerance, reducing false positives. All departures are recorded on-chain to ensure traceability and ledger consistency. The full procedural logic is provided in Appendix A for completeness.

— *Monitoring Phase:* The leader node V_L continuously monitors all participating vehicles through distance measurements and periodic heartbeat signals.

— *Disconnection Criteria:* If V_L detects that a vehicle's distance exceeds a predefined threshold $d_{\text{threshold}}$, or if heartbeat messages are not received for three consecutive cycles, the vehicle is identified as disconnected from the IoV.

— *State Update and Notification:* Upon confirmation, V_L marks the vehicle status as "disconnected," broadcasts this status update to all remaining nodes, and appends a departure event to the blockchain ledger.

3.3 Information Management

The information management module ensures reliable processing of vehicular data through four key processes: information verification, uploading, storage, and retrieval. These steps collectively maintain data integrity, security, and privacy.

3.3.1 Information Verification. This section outlines the framework's security design [18]. Vehicular nodes are defined as either honest—strictly following the protocol—or malicious, which may collude to tamper with or forge messages. To mitigate risks posed by such adversarial behavior, the framework incorporates the following security mechanisms.

— *Physical Reachability Validation:* This mechanism ensures the consistency of real-time vehicle data and prevents malicious data injection.

— *Event Authenticity Verification:* This mechanism verifies critical reports such as accident warnings and congestion alerts to prevent the dissemination of false information.

— *Majority Voting Mechanism:* This mechanism aggregates judgments from surrounding vehicles to identify false data and enhance resilience against coordinated attacks.

The framework first applies physical reachability validation to filter unrealistic data before uploading to the blockchain. Node A checks whether the distance between two reported positions from node B exceeds the maximum possible distance $\Delta d > (v_{\max} + \epsilon)\Delta t$, where $v_{\max}$ is the maximum speed of conventional vehicles (e.g., 30 m/s), ϵ is a safety margin, and Δt is the time difference between two consecutive messages. Messages violating this constraint are rejected as physically implausible to mitigate forgery attempts.

Next, the framework verifies event authenticity by checking if a node is close enough to detect the reported event location. Upon receiving an event packet, the node extracts the event coordinates $(x_{\text{event}}, y_{\text{event}})$ and timestamp t_{event}, then computes the distance to the event:

$$d_{\text{event}} = \sqrt{(x_{\text{event}} - x_{\text{self}})^2 + (y_{\text{event}} - y_{\text{self}})^2}. \tag{1}$$

If $d_{\text{event}} \leq R_{\text{comm}}$, with R_{comm} denoting the communication range for event detection, the vehicle proceeds to verify the event. It performs a local scan using onboard sensors to detect evidence and requests confirmation from nearby nodes.

Subsequently, the node checks the time difference $\Delta t = |t_{\text{current}} - t_{\text{event}}|$ to ensure it is within the propagation delay threshold T_{prop} to prevent replay attacks. If all checks are passed, the node signs and forwards the event; otherwise, it flags the event as potentially malicious. For out-of-range cases ($d_{\text{event}} > R_{\text{comm}}$), the data is stored for later verification via collaborative majority voting.

To further defend against coordinated attacks, the framework aggregates reports on the same event and accepts a consensus result only if a predefined threshold (e.g., 2/3) agrees. Nodes that submit contradictory data are flagged as suspicious and must follow the consensus. For example, if the majority of reports deny an accident while a few affirm it, the minority is treated as potentially forged. This mechanism ensures that only consistent and reliable event data is accepted.

3.3.2 Information Upload. Each vehicle periodically uploads data (e.g., position, speed, events) to the IoV. The data is validated through mechanisms such as physical reachability and event authenticity. Validated data is then sent to the leader, who packages it into transaction proposals for blockchain submission. Once consensus is reached, the data is recorded on-chain, ensuring integrity, transparency, and tamper resistance.

3.3.3 Information Storage. Once data is approved through consensus, it is permanently recorded on the blockchain. Each block contains a transaction list, timestamp, Merkle root, and the hash of the previous block, ensuring tamper resistance and traceability. In special cases such as IoV merging, the framework employs a specialized block structure, termed a "bridge block," which contains dual hash links to preserve ledger consistency. These designs maintain consistent and auditable records during IoV merges. The decentralized consensus mechanism guarantees that all IoV participants can verify data without relying on centralized authorities, thereby reinforcing trust in the recorded information. Furthermore, we propose an information retrieval mechanism, with detailed implementation provided in Appendix B.

3.4 Security Analysis

In this section, we analyze the security of our proposed framework against specific attacks under clearly stated threat models.

3.4.1 Resistance to Location Spoofing Attacks. Physical reachability validation leverages the inherent physical movement characteristics of vehicles to filter data at the source, effectively blocking a large volume of simple forged data from entering the framework. By comparing the displacement of a vehicle between two consecutive message intervals with the maximum speed constraint, the framework can accurately detect anomalies such as abrupt location jumps. Since the trajectory of a normal vehicle must comply with physical laws, any data that exceeds the reasonable range is highly likely to be forged or erroneous. This detection mechanism effectively prevents malicious nodes from fabricating false positions or other misleading data to disrupt the framework. It can be mathematically demonstrated using inductive reasoning that within a continuous message stream, any malicious tampering that violates these constraints will be detected within a bounded time window, thereby ensuring the global physical consistency of location data. This verification

process can be completed rapidly upon message reception by a node, without requiring complex computation or excessive resource consumption. As a result, suspicious data can be filtered in a timely manner without compromising framework efficiency, thereby reducing the burden on downstream processing and ensuring the framework's overall real-time responsiveness.

3.4.2 Integrity Protection Against Message Tampering. We model the success probability of a message tampering attack: $P_{\text{bypass}} \leq \alpha \cdot (1 - P_p) \cdot (1 - P_e) \cdot P_{\text{vote}}$, where: α represents the fraction of Byzantine nodes; P_p denotes the detection probability by Physical Validation; P_e represents the detection probability by Event Authenticity verification; P_{vote} denotes the probability of bypassing the BFT-based Consensus. This probability is constrained by the following factors:

Firstly, physical reachability validation provides a low-cost, high-precision filter ($P_p \approx 1$). This layer transforms physical laws into a first-line defense, ensuring any message violating kinematic constraints is discarded. This can be formally proven: Let the real position of a vehicle at times t_1 and t_2 be P_1 and P_2, with $\Delta t = t_2 - t_1$. The maximum allowable displacement is $d_{\max} = (v_{\max} + \epsilon)\Delta t$. If an attacker tampers with the position to P_2', creating a displacement $d_{\text{tamper}} = \|P_2' - P_1\|$, then: If $d_{\text{tamper}} > d_{\max}, \quad P_p = 1$ (Certain Detection). Only tampering with $d_{\text{tamper}} \leq d_{\max}$ proceeds, making $(1 - P_p)$ negligible for any significant attack.

Secondly, for event-related messages that pass the physical check, sensor semantics require corroboration. An attacker must now fabricate an event consistent with multi-modal sensor observations (e.g., camera, radar) from nearby vehicles. This makes semantic tampering practically infeasible without compromising the physical sensing environment, leading to a high probability P_e of detection for fabricated events. Thus, the term $(1 - P_e)$ is also driven to a very low value.

Finally, if a message is unverifiable by individual nodes, it undergoes majority voting. The consensus mechanism provides statistical guarantees, as a tampered message is discarded unless the attacker controls a quorum. This is rooted in Byzantine Fault Tolerance theory: Let N be the total nodes, f be the number of malicious nodes. The consensus threshold is set at $\tau > \frac{2}{3}N$. The probability P_{vote} that a tampered message achieves consensus is: $P_{\text{vote}} = 0 \quad \text{when} \quad f < N/3$. Even as f increases beyond $N/3$, P_{vote} remains negligible unless a large fraction of honest nodes coincidentally err, which is highly improbable given their sensor reliability $p_h \approx 1$.

Since P_p and P_e are both close to 1, the product $(1 - P_p)(1 - P_e)$ becomes negligible. Combined with a very small P_{vote}, the overall bypass probability P_{bypass} is reduced to a negligible level, providing rigorous integrity protection. In addition, combining with Figure 6(a), when the number of malicious nodes is $\leq 15\%$, the FDI success rate is close to 0, which can verify the rationality of the above inference.

3.4.3 Resilience Against Message Suppression and Packet Loss Attacks. Our framework incorporates temporal redundancy and distributed consensus mechanisms that inherently tolerate partial communication failure. Specifically, majority voting is performed within a configurable time window T_{vote}, during which event-related messages are collected in spatial–temporal buckets. Even if a portion of messages are lost due to adversarial suppression, the aggregation window allows sufficient time for honest nodes to submit their information. Let the total number of nodes be N, and assume a packet loss probability p_{loss} uniformly applied to all honest messages. The probability P_{survive} that at least τN honest votes are received can be modeled as a binomial survival probability:

$$P_{\text{survive}} = \sum_{k=\lceil \tau N \rceil}^{N-f} \binom{N-f}{k} (1 - p_{\text{loss}})^k \, p_{\text{loss}}^{(N-f-k)}. \tag{2}$$

This formulation quantifies the framework's resilience to packet loss: as long as P_{survive} remains sufficiently close to 1, the framework ensures that consensus can be correctly reached despite

adversarial suppression. Notably, by tuning parameters such as the aggregation window T_{vote} and the majority threshold τ, the framework can tolerate a wide range of packet loss rates without compromising safety. In particular, under reasonable network conditions where p_{loss} is bounded, the probability of failure becomes exponentially small in N, ensuring that honest observations are successfully aggregated and adversarial attempts to disrupt consensus through message loss are effectively mitigated.

3.4.4 Resilience Against Replay Attacks. A replay attack occurs when an adversary intercepts and retransmits a valid old message to disrupt the current system state. To mitigate such attacks, our framework incorporates a dual-mechanism defense strategy. First, each message is embedded with a fresh timestamp t; upon receipt, a node verifies that the difference between the current time t_{current} and t remains within a predefined propagation tolerance window T_{prop}, where messages with outdated timestamps are discarded as stipulated in Section 3.3.1. Second, every vehicle maintains a sequentially incremented number for its messages, which is validated by the consensus protocol to guarantee freshness. Together, these measures ensure that any intercepted message cannot be replayed beyond its validity period, thereby effectively neutralizing replay attacks.

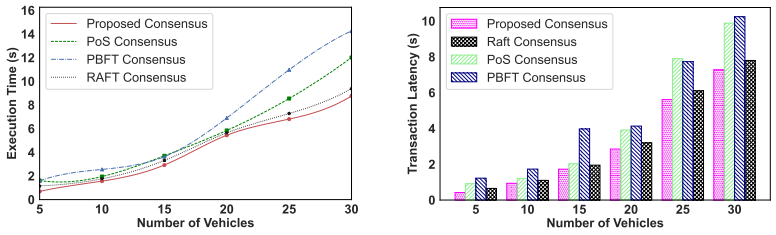
3.4.5 Sybil Attack Resistance. In Sybil attacks, a single malicious entity creates multiple fake identities to gain disproportionate influence. In our framework, the core prerequisite for a valid identity (i.e., a voting node) is that it must be bound to a physical hardware device with verifiable computational capability (C_{min}) and bandwidth (B_{min}). Let $Cost_{\text{legitimate}}$ represent the cost of purchasing and deploying a legitimate hardware node that meets the minimum requirements. For an attacker to achieve a high proportion in voting by creating K Sybil identities to manipulate the system, the minimum economic cost $Cost_{\text{sybil}}$ can be modeled as: $Cost_{\text{sybil}} \geq K \times Cost_{\text{legitimate}}$. Considering a medium-scale IoV cluster (N=30 nodes), if an attacker wants to control one-third of the total votes, the cost immediately becomes $10 \times Cost_{\text{legitimate}}$. This stands in stark contrast to traditional software-only identity registration systems where creating Sybil nodes incurs near-zero cost. Furthermore, our leader election algorithm, which prioritizes high-performance hardware, raises this barrier further. Ensuring Sybil nodes are elected as leaders requires investing in superior hardware, making $Cost_{\text{sybil}}$ prohibitively expensive and the attack impractical.

4 Experimental Evaluation

In this section, we conduct a comprehensive evaluation of the proposed framework from the following perspectives: (1) efficiency evaluation of the consensus algorithms and runtime distribution for other algorithms; (2) reliability and security evaluation; (3) performance evaluation of the four frameworks in two different cases.

4.1 Experimental Setup

We conducted experiments to evaluate the proposed algorithms with respect to efficiency, scalability, and robustness under varying network conditions. All tests were run on a Windows 11 machine (2.40 GHz Intel Core i5-9300H, 16 GB RAM). Vehicular dynamics were simulated using SUMO in urban road scenarios involving 10 to 60 vehicles. An Apache Kafka-based producer-consumer architecture enabled real-time message distribution through independent topics, with transmission controlled via authentication. The implementation used Python for simulation and off-chain processing, with smart contracts written in Solidity and deployed using Truffle. Blockchain experiments ran on a local Ethereum Ganache network with a 1-second manual mining interval.



(a) Comparison of execution time for different consensus algorithms across the number of vehicles.

(b) Comparison of transaction latency for different consensus algorithms across varying numbers of vehicles.

Fig. 4. Efficiency comparisons of different consensus algorithms.

4.2 Algorithmic Efficiency

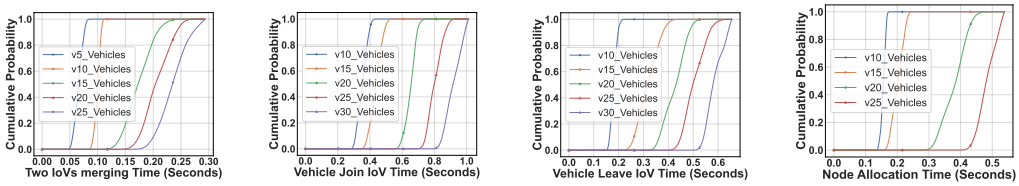
We assessed our algorithms based on three metrics: off-chain transaction execution time, on-chain transaction execution time, and transaction confirmation latency. On-chain execution time refers to the duration from the submission of the first transaction in a batch to the confirmation of the last. Transaction confirmation latency is the delay from a transaction's initiation by the user to its successful inclusion in a block [37].

In the following sections, we compare the on-chain runtime performance of our consensus algorithm with baseline algorithms and analyze their execution characteristics. We also present the **cumulative distribution function** (**CDF**) of off-chain and on-chain execution times for the various algorithms to illustrate performance across different vehicle counts.

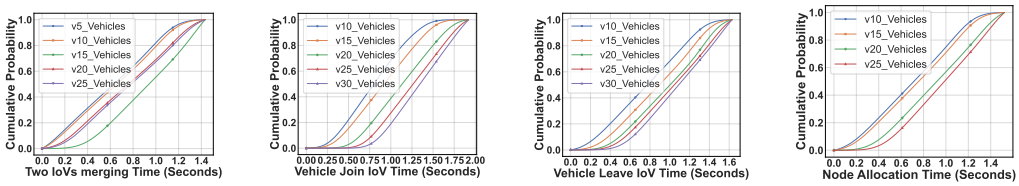
4.2.1 Efficiency of the Proposed Consensus Algorithm. Figure 4(a) shows the execution times of four consensus algorithms, including our proposed method, PoS, PBFT, and Raft, under varying numbers of vehicular nodes. As the number of nodes increases, the execution time of all four algorithms exhibits non-linear growth: when the number of nodes is small, the execution of all algorithms is similar. As it increases, the differences become significant: our method scales more effectively with slower growth in execution time, while PBFT and PoS increase sharply, and Raft grows moderately. Specifically, our proposed consensus algorithm achieves an efficiency improvement of 27.06%–38.56% compared to other baseline algorithms. These results demonstrate the scalability of our approach for large-scale IoV scenarios requiring low latency.

We conducted a scalability evaluation of transaction confirmation latency for four consensus algorithms (5 to 30 vehicles, Figure 4(b)). Our proposed algorithm significantly outperformed the baselines, with latency increasing linearly from 0.42 seconds (5 vehicles) to 7.28 seconds (30 vehicles). In contrast, the baseline algorithms exhibited steeper latency growth curves, particularly PBFT, which reached 10.25 seconds at 30 vehicles due to its $O(n^2)$ message complexity. Quantitative analysis of performance improvements reveals that our algorithm achieves latency reductions of 7%–35%, 15%–54%, and 27%–66% compared to Raft, PoS, and PBFT, respectively. These results highlight the effectiveness of our design in reducing communication overhead and optimizing consensus processes, making it better suited for large-scale IoV frameworks with stringent real-time requirements.

4.2.2 Execution Time Distribution for Vehicle Joining, Leaving, IoV Merging, and Node Allocation. Figure 5 shows the time distribution of vehicle join, leave, IoV merge, and node allocation operations in off-chain and on-chain modes. In the off-chain case (Figure 5(a)–(d), the legend denotes the number of vehicles concurrently performing each specific operation), where multiple vehicles



(a) IoV merging (off-chain) (b) Vehicle joining (off-chain) (c) Vehicle leaving (off-chain) (d) Node allocation (off-chain)



(e) IoV merging (on-chain) (f) Vehicle joining (on-chain) (g) Vehicle leaving (on-chain) (h) Node allocation (on-chain)

Fig. 5. Execution time distributions for vehicle joining IoV, leaving IoV, IoV merging, and node allocation. Here, on-chain refers to operations that include both execution and committing results to the blockchain, while off-chain refers to operations that are executed without committing the results to the blockchain.

perform operations concurrently, the framework achieves fast response with lightweight execution. The CDF curves converge rapidly, and execution time increases slowly with scale, indicating good scalability.

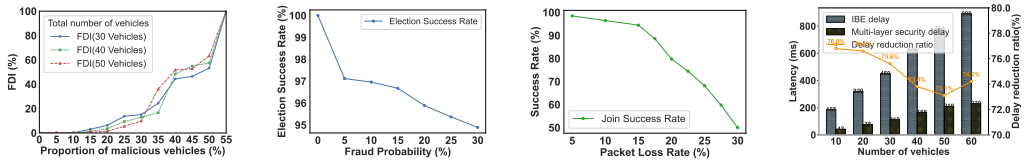
In the on-chain case (Figure 5(e)–(h)), blockchain ensures auditability through consensus, with execution time covering transaction packaging, validation, and ledger updates. Although execution time spans a wider range, it scales predictably with the number of nodes, demonstrating the blockchain's adaptability to large-scale IoV environments.

The off-chain and on-chain execution modes serve complementary roles: the former supports real-time vehicle coordination, while the latter ensures traceability and immutability for critical operations. Results demonstrate the framework's flexibility in adapting execution strategies to different scenario requirements.

4.3 Reliability and Security Evaluation

This section evaluates the reliability and security of the proposed IoV framework under different conditions. Reliability refers to maintaining operation and consensus under disturbances like packet loss, while security concerns resistance to attacks such as data forgery, event manipulation, and consensus disruption. We simulate threat scenarios to assess the effectiveness of the layered verification mechanisms, including physical reachability validation, event authenticity checks, and majority voting, across three key aspects.

— *FDI Defense:* We measure the FDI attack success rate under varying proportions of malicious vehicles injecting forged data, reflecting the framework's ability to reject tampered or implausible messages.
— *Consensus Robustness:* We evaluate the leader election success rate under varying malicious message forging probabilities, reflecting the protocol's resilience to adversarial manipulation.
— *Vehicle Join IoV Success Rate under Packet Loss:* We assess the probability of a vehicle successfully joining the IoV under varying packet loss levels, reflecting environmental degradation or message suppression attacks.



(a) Impact of the proportion of malicious vehicles on FDI success rate.

(b) Impact of the fraud probability on the election success rate.

(c) Impact of packet loss rate on the success rate of vehicles joining the IoV.

(d) Comparison of verification latency of the multi-layer security mechanism and the IBE.

Fig. 6. Reliability and security assessment of the framework.

Key parameters include a tolerance margin $\epsilon = 5$ m/s for physical plausibility checks and a voting interval $T_{\text{vote}} = 2$ s. $\epsilon = 5$ m/s is grounded in industry standards for speed monitoring tolerance (typically ±5%–10%) and is designed to be configurable, allowing adaptation to different road scenarios [9], [10]. The FDI attack success rate is formally defined as

$$\text{FDI} = \frac{\text{Number of Fake Messages Accepted}}{\text{Number of Fake Messages Sent}}. \tag{3}$$

The following results show that the proposed framework maintains low attack success rates and high reliability under adversarial interference and packet loss.

4.3.1 FDI Attack Resilience. Figure 6(a) shows that with fewer than 10% malicious vehicles, FDI attacks are mostly ineffective, confirming the framework's effectiveness against a small number of attacks. When malicious nodes exceed 30%, the success rate rises in small IoVs but more slowly in larger ones, reflecting Byzantine fault tolerance. Even with 50% malicious nodes, the framework blocks 50%–60% of false messages, indicating resilience against coordinated FDI attacks.

4.3.2 Consensus Robustness. Figure 6(b) shows that our consensus maintains nearly 100% leader election success when malicious forging is rare. Leader election success refers to the protocol's ability to select an honest leader for consensus. As the forging ratio increases, the success rate gradually declines but remains high under moderate attack levels. When forging approaches 30%, the success rate decreases more noticeably, yet still remains above 95%. These results demonstrate that the proposed design is highly resilient to typical Byzantine attacks.

4.3.3 Vehicle Join IoV Robustness. Figure 6(c) evaluates the framework's ability to admit new vehicles into the IoV under adverse communication conditions, by measuring the join success probability at various packet loss rates. The simulation reveals that the IoV is highly robust to packet loss: at low to moderate packet loss rates (e.g., up to 20%), nearly all vehicles requesting to join can join the IoV successfully, indicating that a few dropped messages do not prevent legitimate nodes from entering the IoV. As the packet loss rate increases further, the join success probability gradually declines, which is a natural outcome as communication reliability degrades. In addition, even at relatively high packet loss levels (for instance, 40%–50%), a substantial majority of vehicles are still able to successfully join the IoV. This resilience can be attributed to reattempts and cross-validation by multiple neighbors. In practice, this ensures that legitimate vehicles are not easily excluded from the IoV, even in the presence of unreliable channels or adversarial interference.

4.3.4 Verification Latency Analysis of Security Mechanism. To quantitatively validate the lightweight nature of our security mechanism, we compare its single-message verification latency against a traditional IBE scheme. As shown in Figure 6(d), our multi-layered mechanism maintains a low and stable latency (under 230 ms even with 60 vehicles), reducing latency by 74%–77% compared

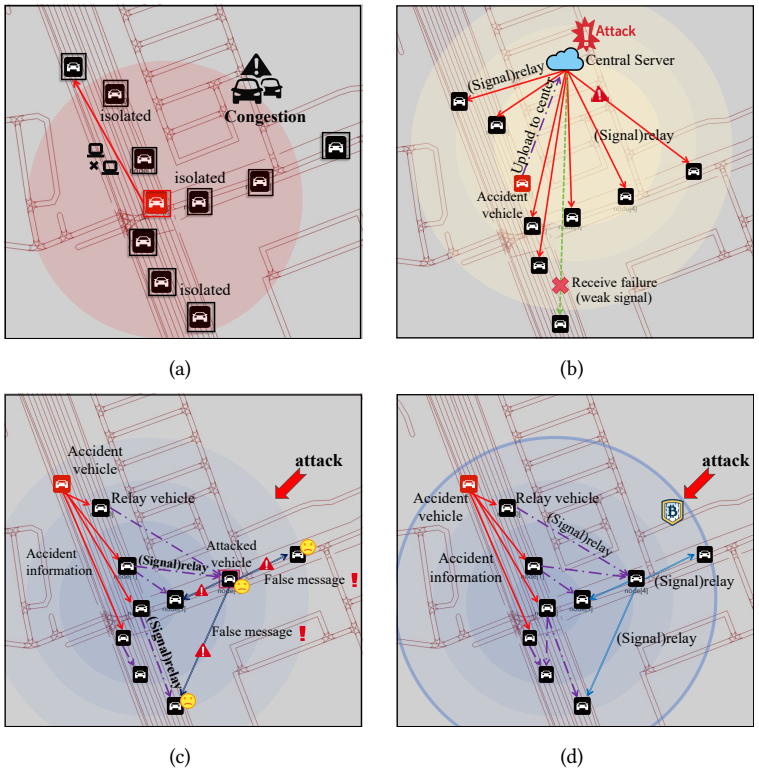


Fig. 7. Illustrations of the response differences among four frameworks under the same accident scenario.

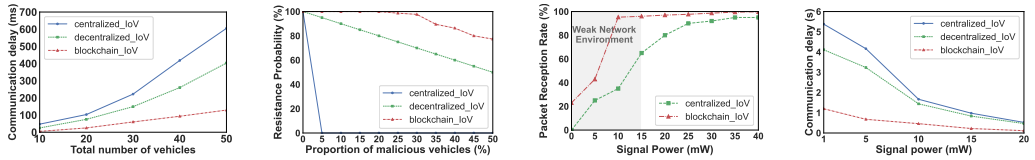
to IBE. This improvement stems from replacing IBE's computationally intensive operations (e.g., $O(n^3)$ bilinear pairings) with efficient, localized checks ($O(1)$ physical and sensor validation) and lightweight ($O(\log N)$) gossip-based voting.

4.4 Case Study: Performance of Vehicular Networks under Different Frameworks

The following case studies were evaluated via co-simulation using OMNeT++, VEINS, INET, and SUMO to assess four frameworks: (1) a non-IoV, where vehicles operate independently without communication; (2) a centralized IoV, where data is disseminated via a cloud server or RSU through single-hop transmission; (3) a decentralized IoV, where vehicles communicate through V2V multi-hop links; and (4) our blockchain-based IoV, which provides decentralized data management with integrity verification. To simulate resource heterogeneity among vehicles from different manufacturers, each vehicle was assigned distinct processing delays and receiver sensitivity values. This models differences in communication and computation capabilities among heterogeneous nodes. This approach enhances the practical applicability of simulation results to real-world deployment.

4.4.1 Case Study 1: In an Urban Road Environment with Good Signal Coverage. We evaluated four frameworks under stable network conditions in an urban road scenario, where a simulated vehicle breakdown led to congestion, particularly during rush hours. The evaluation focused on efficiency, security, and resilience.

Efficiency. The non-IoV framework (Figure 7(a)) relies on static routing, leading to poor incident response and persistent congestion. The centralized IoV framework (Figure 7(b)) depends on a central



(a) Impact of total vehicle count on communication delay in different frameworks.

(b) Impact of malicious vehicle proportion on the attack resistance of frameworks.

(c) Impact of signal power on packet reception probability in different frameworks.

(d) Impact of signal power on communication delay in different frameworks.

Fig. 8. A comparative analysis of the efficiency and security of different frameworks under two distinct scenarios.

server or RSU, but suffers from a single point of failure, poor edge reception, and delays during peak traffic. The decentralized IoV framework (Figure 7(c)) employs V2V multi-hop communication and achieves high data reception but lacks an adaptive mechanism, incurring delays under heavy load. In contrast, our blockchain-based framework (Figure 7(d)) maintains high packet reception while significantly reducing latency through dynamic IoV splitting and the flexible leaving of vehicles, effectively alleviating congestion. As evidenced in Figure 8(a), our Blockchain-IoV framework reduces average communication delay by approximately 30%–40% compared to the Centralized-IoV and about 15%–25% compared to the Decentralized-IoV approach across varying network scales. This consistent advantage demonstrates the efficacy of our adaptive mechanisms in sustaining low-latency communication under congested conditions.

Security. The non-IoV framework prevents external data manipulation but remains vulnerable to local data corruption and traffic disruptions. The centralized IoV framework is at risk of total compromise if the central server is attacked. Although the decentralized IoV avoids single points of failure through V2V broadcasting, it lacks data verification and is thus susceptible to FDI. Our framework integrates smart contracts to authenticate data and mitigates over 80% of tampering attempts, offering improved resilience. As shown in Figure 8(b), our blockchain-IoV framework sustains over 95% resistance with 30% malicious vehicles, outperforming decentralized and centralized approaches by 40% and 60%, respectively, a 2- to 3-fold resilience improvement against coordinated attacks.

4.4.2 Case Study 2: In Mountainous Areas with Weak Signal Coverage. This case study evaluates four frameworks using a 20-vehicle network in a low-signal mountainous environment.

Efficiency and Information Integration Ability. Non-IoV frameworks lack communication capabilities, making them ineffective in emergencies. Centralized frameworks suffer from rapid performance degradation in weak network environments, with packet reception dropping below 40% at low signal power. Although decentralized IoV improves packet reception through V2V communication, it still exhibits 20%–30% higher delays compared to our solution under the same conditions. As evidenced by Figure 8(c) and (d), our blockchain-IoV framework maintains over 85% packet reception even at 10 mW signal power through dynamic IoV merging and flexible vehicle joining, more than doubling the performance of centralized systems. Simultaneously, it reduces communication delays by approximately 35% compared to decentralized IoV, demonstrating robust connectivity and timely data dissemination in challenging network environments.

5 Related Work

Currently, IoV research primarily focuses on communication performance optimization and security privacy protection. However, existing solutions often overlook the balance between these two aspects, as well as key challenges such as high vehicle dynamics and infrastructure dependency.

5.1 Limitations of Existing Research in Communication Performance Optimization

Existing research mainly improves IoV communication efficiency through optimizing **roadside unit** (**RSU**) deployment [15], [30], [45], [33], [35], utilizing **mobile edge computing** (**MEC**) [43], [46], **wireless sensor networks** (**WSNs**) [1], or applying SDN [11], [36], [42]. While these methods are effective in scenarios with well-established infrastructure, they commonly share a significant limitation: heavy dependence on pre-established, stable infrastructure. For example, Huang et al.'s UAV-assisted model requires drones as intermediate nodes [20]; research on 5G-based IoV communication efficiency improvement relies on continuous base station coverage [27]. Additionally, some studies improve efficiency through enhanced communication protocols, such as Bintoro et al. [5] who introduced a **learning automata-driven ad-hoc on-demand distance vector** (**LA-AODV**) routing protocol to optimize V2V communication. Sugianto et al. evaluated the performance of three routing protocols in improving traffic efficiency [39]. However, these methods experience significant performance degradation in scenarios with sparse infrastructure or unstable signals (such as mountainous areas and tunnels). Although some blockchain-based studies [8], [19], [22], [47], [17] utilize decentralized mechanisms to achieve balanced resource allocation and system efficiency improvement, their architectures are typically static, assuming relatively stable node connections and steady resource supply, failing to adequately consider link interruptions or latency fluctuations caused by IoV's high dynamics. In contrast, our framework achieves infrastructure-independent communication between vehicles through a blockchain-based dynamic adaptive network. The core innovation lies in introducing dynamic splitting and merging mechanisms, enabling IoV to adaptively adjust according to real-time traffic conditions and connection quality, thereby maintaining high communication efficiency even in infrastructure-free areas (as demonstrated in the case study in Section 4.4.2).

5.2 Deficiencies of Existing Research in Security and Privacy Protection

In terms of security, existing solutions mainly focus on complex cryptographic methods or communication protocols, such as Sutrala et al.'s **elliptic curve cryptography** (**ECC**)-based batch authentication mechanism [40], Tang et al.'s hybrid encryption scheme [41], Li et al. utilized homomorphic encryption and zero-knowledge proof schemes to provide privacy protection for the IoV [25], Guo et al.'s SSL/TLS protocol for communication link encryption [16], Thuvva et al. proposed a hybrid intelligent detection and encryption scheme [2], or Bansal's team's **physical unclonable functions** (**PUFs**) [4]. Although these methods can enhance IoV communication security, their significant computational overhead may fail to meet IoV's real-time requirements, and their passive defense mode struggles against coordinated FDI attacks. On the other hand, some blockchain-based research, such as the work by Javaid's team [21], Shi et al. [34], Gopal et al. [38] and Dwivedi's team [12], have all achieved decentralized IoV trust management, privacy protection, and secure data sharing. However, their architectures still focus on static resource management or data storage, assuming relatively stable node connections, failing to integrate blockchain consensus mechanisms with IoV's highly dynamic topology, thus lacking adaptive security capabilities at the network level. Different from the above work, we propose a lightweight, multi-layered proactive defense mechanism that combines physical reachability validation, event authenticity verification, and majority voting, filtering data before uploading to the blockchain. This significantly reduces dependence on complex encryption, achieves proactive defense against coordinated attacks, and ensures low latency.

6 Discussion

This section discusses the limitations of the proposed blockchain-based adaptive IoV framework and suggests avenues for future research.

6.1 Limitations

6.1.1 Storage Overhead and Ledger Scalability. While the framework guarantees data traceability through blockchain, the frequent execution of on-chain operations under high vehicular mobility may accelerate ledger expansion.

6.1.2 Simplified Environmental Modeling. We used simulation tools (e.g., SUMO, OMNeT++) to evaluate framework performance under different network structures and threat scenarios. However, these platforms may not fully reflect real-world factors like signal interference, terrain variations, and **non-line-of-sight** (**NLOS**) conditions. Therefore, further validation and tuning are needed for deployment in real-world environments.

6.1.3 Architectural Tradeoffs Between Stability and Reconfiguration Responsiveness. This framework integrates adaptive mechanisms such as dynamic IoV splitting/merging and multi-layered security verification. While these mechanisms enhance framework flexibility, they also introduce an inherent tension between stability and responsiveness.

First, regarding the IoV splitting and merging strategy, the dual-threshold mechanism based on node count and latency helps prevent excessive fragmentation, but it may reduce responsiveness in rapidly changing traffic conditions. The multi-layered security verification strengthens attack resistance but adds consensus latency, requiring a balance between security robustness and real-time performance. Second, the performance-based leader election mechanism favors nodes with high computational power, which benefits consensus stability, but in highly dynamic scenarios, communication efficiency may be affected due to geographical shifts. The bridge-block mechanism ensures ledger consistency during merging but also introduces synchronization overhead and merging delays.

In summary, these tradeoffs highlight that the adaptive capabilities of the framework are not without cost. In practical deployments, relevant parameters should be dynamically fine-tuned based on specific scenarios, such as urban congestion or highway platooning. Future research may explore adaptive strategies based on reinforcement learning to achieve a dynamic balance between stability and responsiveness.

6.2 Future Research

For improvement and to address the limitations listed above, our framework may be further improved as follows:

6.2.1 A Hybrid On-Chain/Off-Chain Storage Architecture. To address the storage scalability challenge of high-frequency vehicular data, our framework will employ a hybrid on-chain/off-chain storage model. Only critical data summaries (e.g., Merkle roots of batched sensor readings) are immutably anchored to the blockchain, while the voluminous raw data is stored off-chain (e.g., in local vehicle storage). This approach leverages the blockchain for tamper-evident auditing without incurring the burden of storing all data on-chain, making it feasible for resource-constrained onboard devices.

6.2.2 Real-World Pilot Deployment. Collaborating with transportation authorities to conduct small-scale real-vehicle trials in urban and rural scenarios to evaluate the robustness and performance of the framework in practical conditions.

7 Conclusion

This article proposes a blockchain-based dynamic adaptive restructuring framework for the IoV. In this framework, vehicles autonomously form communication clusters through a proximity-driven self-organization mechanism, while a consensus algorithm ensures data consistency among

distributed nodes. To accommodate network dynamics, the framework supports structural adaptation through proactive IoV splitting under high-load conditions to alleviate communication overhead, and intelligent IoV merging when adjacent IoV sub-networks converge, thereby improving resource utilization. In addition, a self-governing node mechanism enables vehicles to dynamically join or leave IoVs based on real-time communication quality. To ensure security and robustness, the framework incorporates a multi-layered defense mechanism that effectively mitigates threats such as data tampering and node misbehavior. Experimental results demonstrate that the proposed framework significantly enhances both communication efficiency and data security, offering a scalable solution for structural reconfiguration and resource optimization in intelligent transportation systems.

References

[1] 2025. The application of wireless sensor network in intelligent transportation. Retrieved February 20, 2025 from https://www.techphant.cn/blog/90558.html

[2] Thuvva Anjali, Rajeev Goyal, and G. N. Balaji. 2025. Enhanced gated sway network and hybrid henon encryption for secured vanet communication. *International Journal of Safety and Security Engineering* 15, 3 (2025), 543. DOI: https://doi.org/10.18280/ijsse.150313

[3] Ferheen Ayaz, Zhengguo Sheng, Daxin Tian, and Victor C. M. Leung. 2021. *Blockchain-Enabled Security and Privacy for Internet-of-Vehicles*. Springer International Publishing, Cham, 123–148. DOI: https://doi.org/10.1007/978-3-030-46335-9_9

[4] Gaurang Bansal, Naren Naren, Vinay Chamola, Biplab Sikdar, Neeraj Kumar, and Mohsen Guizani. 2020. Lightweight mutual authentication protocol for v2g using physical unclonable function. *IEEE Transactions on Vehicular Technology* 69, 7 (2020), 7234–7246. DOI: https://doi.org/10.1109/TVT.2020.2976960

[5] Ketut Bayu Yogha Bintoro, Tri Kuntoro Priyambodo, and Yadie Prasetyo Sardjono. 2024. Smart aodv routing protocol strategies based on learning automata to improve v2v communication quality of services in vanet. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control* 9, 3 (2024), 255–266.

[6] Ethan Buchman. 2016. *Tendermint: Byzantine Fault Tolerance in the Age of Blockchains*. Ph. D. Dissertation. University of Guelph.

[7] Ethan Buchman. 2017. Tendermint: Byzantine fault tolerance in the age of blockchains, 2016. Retrieved from http://hdl.handle.net/10214/9769. (2017).

[8] Haoye Chai, Supeng Leng, Ke Zhang, and Sun Mao. 2019. Proof-of-reputation based-consortium blockchain for trust resource sharing in internet of vehicles. *IEEE Access* 7 (2019), 175744–175757.

[9] China Road Traffic Safety Association. 2023. Specifications for the investigation and handling of road traffic accidents of intelligent connected vehicles. Retrieved September 25, 2025 from https://s0.crsa.net/1732246471071_64.pdf

[10] Committee of Transport Officials (COTO). 2018. TMH 3: Specifications for the Provision of Traffic and Weigh-in-Motion Monitoring Services. Retrieved September 25, 2025 from https://www.transport.gov.za/wp-content/uploads/2023/02/TMH-3-Traffic-Monitoring-Services-CD-2.pdf

[11] Santosh A Darade, M. Akkalakshmi, and Dr Nitin Pagar. 2022. SDN based load balancing technique in internet of vehicle using integrated whale optimization method. In *AIP Conference Proceedings* 2469, 1 (2022), 020006.

[12] Sanjeev Kumar Dwivedi, Ruhul Amin, Satyanarayana Vollala, and Rashmi Chaudhry. 2020. Blockchain-based secured event-information sharing protocol in internet of vehicles for smart cities. *Computers and Electrical Engineering* 86 (2020), 106719.

[13] Ruo Fei. 2025. Why can the BFT system tolerate at most one-third of faulty nodes. Retrieved September 25, 2024 from https://www.cnblogs.com/zhanchenjin/p/19033200

[14] Bimal Ghimire and Danda B. Rawat. 2022. Secure, privacy preserving, and verifiable federating learning using blockchain for internet of vehicles. *IEEE Consumer Electronics Magazine* 11, 6 (2022), 67–74. DOI: https://doi.org/10.1109/MCE.2021.3097705

[15] Abderrahim Guerna, Salim Bitam, and Carlos T. Calafate. 2022. Roadside unit deployment in internet of vehicles systems: A survey. *Sensors* 22, 9 (2022), 3190.

[16] Jiale Guo, Yuhan Tian, Pengren Ding, and Yang Gao. 2024. Design of an SSL/TLS encryption-based information security detection system for vehicular networks. In *Proceedings of the 2024 3rd International Conference on Artificial Intelligence and Computer Information Technology*. 1–4. DOI: https://doi.org/10.1109/AICIT62434.2024.10730670

[17] Muhammad Hassnain Ali Haider, Muhammad Fayaz, Yue Zhang, Haseena Noureen, Zeeshan Ali Haider, Fida Muhammad Khan, Inam Ullah Khan, and Md Moklesur. 2025. Enhancing authentication security in internet of vehicles: A blockchain-driven approach for trustworthy communication. *IECE Trans. Adv. Comput. Syst.* 1, 1 (2025), 48–62.

[18] Zicong Hong, Song Guo, Peng Li, and Wuhui Chen. 2021. Pyramid: A layered sharding blockchain system. In *Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.

[19] Zhuo Hu, Bozhi Liu, Ao Shen, and Jie Luo. 2024. Blockchain-based resource allocation mechanism for the internet of vehicles: Balancing efficiency and security. *IEEE Transactions on Network and Service Management* 21, 4 (2024), 3971–3987.

[20] Jiwei Huang, Man Zhang, Jiangyuan Wan, Ying Chen, and Ning Zhang. 2024. Joint data caching and computation offloading in UAV-assisted Internet of Vehicles via federated deep reinforcement learning. *IEEE Transactions on Vehicular Technology* 73, 11 (2024), 17644–17656.

[21] Uzair Javaid, Muhammad Naveed Aman, and Biplab Sikdar. 2019. DrivMan: Driving trust management and data sharing in VANETS with blockchain and smart contracts. In *Proceedings of the 2019 IEEE 89th Vehicular Technology Conference*. IEEE, 1–5.

[22] Jiawen Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Maharjan, Shengli Xie, and Yan Zhang. 2018. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet of Things Journal* 6, 3 (2018), 4660–4670.

[23] Seung Woo Kim. 2018. Why N = 3f+1 in the Byzantine Fault Tolerance system. Retrieved September 24, 2025 from https://medium.com/codechain/why-n-3f-1-in-the-byzantine-fault-tolerance-system-c3ca6bab8fe9

[24] Wenyu Li, Chenglin Feng, Lei Zhang, Hao Xu, Bin Cao, and Muhammad Ali Imran. 2020. A scalable multi-layer PBFT consensus for blockchain. *IEEE Transactions on Parallel and Distributed Systems* 32, 5 (2020), 1146–1160.

[25] Yijing Li, Ran Bi, Nan Jiang, Fengqiu Li, Mingsi Wang, and Xiangping Jing. 2024. Methods and challenges of cryptography-based privacy-protection algorithms for vehicular networks. *Electronics (2079-9292)* 13, 12 (2024).

[26] Hua Yi Lin and Meng-Yen Hsieh. 2022. A dynamic key management and secure data transfer based on m-tree structure with multi-level security framework for Internet of vehicles. *Connection Science* 34, 1 (2022), 1089–1118.

[27] Mahmoud M. Elsayed, Khalid M. Hosny, Mostafa M. Fouda, and Marwa M. Khashaba. 2023. Vehicles communications handover in 5G: A survey. *ICT Express* 9, 3 (2023), 366–378.

[28] Na Lv. 2022. Key Points for Governing Data Security Risks in the Internet of Vehicles. Retrieved February 19, 2025 from https://finance.sina.com.cn/tech/2022-05-01/doc-imcwiwst5050078.shtml

[29] Dahlia Malkhi, Kartik Nayak, and Ling Ren. 2019. Flexible byzantine fault tolerance. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1041–1053.

[30] Ming Mao, Peng Yi, and Tao Hu. 2023. Roadside infrastructure deployment scheme based on internet of vehicles information service demand. *Transactions on Emerging Telecommunications Technologies* 34, 1 (2023), e4671.

[31] Muhammad Baqer Mollah, Jun Zhao, Dusit Niyato, Yong Liang Guan, Chau Yuen, Sumei Sun, Kwok-Yan Lam, and Leong Hai Koh. 2021. Blockchain for the internet of vehicles towards intelligent transportation systems: A survey. *IEEE Internet of Things Journal* 8, 6 (2021), 4157–4185. DOI: https://doi.org/10.1109/JIOT.2020.3028368

[32] Manjari Singh Rathore, M. Poongodi, Praneet Saurabh, Umesh Kumar Lilhore, Sami Bourouis, Wajdi Alhakami, Jude Osamor, and Mounir Hamdi. 2022. A novel trust-based security and privacy model for internet of vehicles using encryption and steganography. *Computers and Electrical Engineering* 102 (2022), 108205.

[33] Ahmed Thair Shakir, Md Shabiul Islam, Jit Singh Mandeep, Mohammad Tariqul Islam, Nor Fadzilah Abdullah, Younus Hasan Taher, Omer T. Abdullahi, and Mohamed S. Soliman. 2024. Systematic review of data exchange for road side unit in a vehicular ad hoc network: Coherent taxonomy, prominent features, datasets, metrics, performance measures, motivation, opportunities, challenges and methodological aspects. *Discover Applied Sciences* 6, 9 (2024), 487.

[34] Kexin Shi, Liehuang Zhu, Can Zhang, Lei Xu, and Feng Gao. 2020. Blockchain-based multimedia sharing in vehicular social networks with privacy protection. *Multimedia Tools and Applications* 79, 11–12 (2020), 8085–8105.

[35] Shuyao Shi, Neiwen Ling, Zhehao Jiang, Xuan Huang, Yuze He, Xiaoguang Zhao, Bufang Yang, Chen Bian, Jingfei Xia, Zhenyu Yan, et al. 2024. Soar: Design and deployment of a smart roadside infrastructure system for autonomous driving. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 139–154.

[36] Rhodney Simões, Kelvin Dias, and Ricardo Martins. 2021. Dynamic allocation of SDN controllers in nfv-based mec for the internet of vehicles. *Future Internet* 13, 11 (2021), 270.

[37] Pranav Kumar Singh, Roshan Singh, Sunit Kumar Nandi, Kayhan Zrar Ghafoor, Danda B. Rawat, and Sukumar Nandi. 2020. Blockchain-based adaptive trust management in internet of vehicles using smart contract. *IEEE Transactions on Intelligent Transportation Systems* 22, 6 (2020), 3616–3630.

[38] Gopal Singh Rawat, Karan Singh, Mohd Shariq, Ashok Kumar Das, Shehzad Ashraf Chaudhry, and Pascal Lorenz. 2025. BTC2PA: A blockchain-assisted trust computation with conditional privacy-preserving authentication for connected vehicles. *IEEE Transactions on Intelligent Transportation Systems* 26, 1 (2025), 1134–1148. DOI: https://doi.org/10.1109/TITS.2024.3488184

[39] Rulliansyah Sugianto, Ketut Bayu Yogha Bintoro, and Ade Syahputra. 2025. V2V communication QoS comparison of reactive and proactive protocols: A simulation-based evaluation in urban and highway scenarios. *Jurnal PROCESSOR* 20, 1 (2025), 99–109. DOI: https://doi.org/10.33998/processor.2025.20.1.2206

[40] Anil Kumar Sutrala, Palak Bagga, Ashok Kumar Das, Neeraj Kumar, Joel J. P. C. Rodrigues, and Pascal Lorenz. 2020. On the design of conditional privacy preserving batch verification-based authentication scheme for internet of vehicles deployment. *IEEE Transactions on Vehicular Technology* 69, 5 (2020), 5535–5548. DOI: https://doi.org/10.1109/TVT.2020.2981934

[41] Yujian Tang, Ya Chen, and HoeKyung Jung. 2024. Hazardous chemicals logistics internet of vehicles based on encryption algorithm. *Scalable Computing: Practice and Experience* 25, 3 (2024), 1287–1300. DOI: https://doi.org/10.12694/scpe.v25i3.2064

[42] Syed Aizaz Ul Haq, Muqaddas Imran, Nadir Shah, and Gabriel-Miro Muntean. 2025. SDN-based edge computing in vehicular communication networks: A survey of existing approaches. *IEEE Access* 13 (2025), 74252–74287.

[43] Junyi Yang, Yuchuan Fu, Changle Li, and Xiaoming Yuan. 2023. Joint optimization scheme for user association and resource allocation in internet of vehicles. In *Proceedings of the 2023 IEEE 98th Vehicular Technology Conference*. IEEE, 1–5.

[44] Yuanqi Yang. 2021. A SDN-based traffic estimation approach in the internet of vehicles. *Wireless Networks* 32, 2 (2021), 1–12.

[45] Qiqi Mao Yuming Ge. 2023. Research on cross-domain cooperation deployment of V2X infrastructure. *Telecommunications Science* 39, 3 (2023), 24–31. DOI: https://doi.org/10.11959/j.issn.1000-0801.2023036

[46] Yahong Zhai, Yong Lv, and Longyan Xu. 2024. Optimized QoS routing in software-defined in-vehicle networks. *International Journal of Computers Communications and Control* 19, 1 (2024), 5962.

[47] Guobin Zhang, Xuqiang Chen, Guojun Han, Chang Liu, and Jiawen Kang. 2025. Blockchain-based distributed resource management for delay minimization in uav-assisted vehicular networks. *IEEE Transactions on Vehicular Technology* 74, 10 (2025), 1–16. DOI: https://doi.org/10.1109/TVT.2025.3568714

[48] Yuanjian Zhou, Zhenfu Cao, and Xiaolei Dong. 2023. Dynamic identity-based broadcast proxy re-encryption for data sharing in autonomous vehicles. *Transactions on Emerging Telecommunications Technologies* 34, 11 (2023), e4801.

Appendix

A Full Algorithm Descriptions

To provide a more comprehensive understanding of our framework's operational details, we present the complete procedure of the Vehicle Leave IoV Algorithm in this appendix. While Section 3(b) of the main text outlines the key principles and decision criteria, the full algorithmic workflow—including heartbeat monitoring, distance-based disconnection detection, and ledger update operations—is detailed here to facilitate implementation clarity.

ALGORITHM A1: Vehicle Leaving Algorithm.

Require: $d_{\text{threshold}}$: maximum allowed distance between vehicles within the IoV.
Ensure: Updated IoV membership set $\mathcal{J}'$ after removals.

Leader Node Behavior:
while $\mathcal{J}$ is active **do**
  for each $V_j \in \mathcal{N}(V_L)$ **do**
    if heartbeat_lost(V_j) **then**
      miss_count[V_j] $\leftarrow$ miss_count[V_j] + 1
    else
      miss_count[V_j] $\leftarrow$ 0
    end if
    if miss_count[V_j] $\geq 3 \vee \Delta L(V_j, V_L) > d_{\text{threshold}}$ **then**
      V_j.status $\leftarrow$ `Disconnected`
      Broadcast `leave_notice` to all $V_i \in \mathcal{J}$
      $\mathcal{J} \leftarrow \mathcal{J} \setminus \{V_j\}$ and update the ledger ▷ Remove vehicle from IoV
    end if
  end for
  Synchronization: $\forall V_i \in \mathcal{J}, \mathcal{J}_{V_i} \leftarrow \mathcal{J}$
end while

B Information Retrieval

IoV nodes, including vehicles, edge nodes, and infrastructure units, can initiate information retrieval operations through predefined smart contract interfaces. The system provides a query API that supports parameterized access to on-chain data, allowing users to specify query types (e.g., traffic incidents, reputation scores, or local connectivity status), temporal ranges, and geographic constraints.

To support efficient and structured data access, smart contracts expose read-only functions such as getRecentIncidents(areaID, timeWindow) or getReputation(vehicleID), which return verified, timestamped results directly from on-chain records. These queries are handled through mapping structures and event logs recorded on the blockchain, ensuring data integrity and traceability. For performance optimization, commonly accessed data (e.g., regional traffic summaries or leader election history) are periodically aggregated and cached at edge nodes. These edge-side caches synchronize with the blockchain in near real-time to balance efficiency and consistency. Furthermore, all retrieval requests are subject to access control policies enforced by the smart contract layer. Sensitive data, such as vehicle identifiers or behavioral history, is anonymized using pseudonym mechanisms or selectively encrypted to preserve privacy, especially during inter-cluster queries. This module ensures that decision-making processes such as route planning, emergency response, and trust assessment are supported by timely, verifiable, and context-aware information.

C Empirical Verification: Performance of Leaders under Different Weights

Based on the experimental settings of Section 4.1 and Section 4.4, a test platform is built: 10 high-configuration vehicles and 20 ordinary vehicles are used to simulate urban congestion scenarios, so as to test the average computing delay and communication delay of leaders under different weights shown in Table 2.

Table 2. Performance of Leaders under Different Weights

Weight Ratio (Computing: Geography)	High-configuration Node Election Rate	Average Computing Delay (ms)	Average Communication Delay (ms)	Total Delay (ms)
4:1	92%	280	180	460
3:2	75%	310	120	430
2:3	48%	520	80	600

With a 4:1 ratio: While computing delay remained low, the frequent election of edge leaders increased multi-hop forwarding, raising communication delay and resulting in a total delay of 460 ms. With the chosen 3:2 ratio: This balance resulted in a 75% election rate for high-configuration nodes. It achieved a computing delay of 310 ms (safely below the 500 ms split threshold) and a manageable communication delay of 120 ms, yielding an optimal total delay of 430 ms. With a 2:3 ratio: Ordinary vehicles were elected too frequently, causing the computing delay to rise to 520 ms (exceeding the split threshold). This led to frequent IoV splits and significantly reduced framework stability.

These experimental results confirm that the 3:2 ratio effectively balances computational and geographical factors, minimizing total delay while preventing framework instability.

Received 16 June 2025; revised 19 January 2026; accepted 14 February 2026