ISSN Number: 2454-2024(Online)

International Journal of Technical Research & Science

INTEGRATION OF VEHICAL-TO-VEHICAL (V2V) COMMUNICATION WITH ADVANCE DRIVER ASSISTANCE SYSTEM

Karan Patel, Madhusudan Barot, Jinesh Kamdar
E-Mail Id: vipulbhaikaran.22.am@iite.indusuni.ac.in
Indus Institute of Technology & Engineering, Ahmedabad, Gujarat, India

Abstract- The integration of Vehicle-to-Vehicle (V2V) communication with Advanced Driver Assistance Systems (ADAS) represents a significant advancement in the field of intelligent transportation. This research explores a prototype implementation of V2V communication embedded with ADAS capabilities, utilizing components such as LIDAR, camera modules, an Ackerman steering chassis, and an encoder-based DC motor system. The project aims to demonstrate real-time incident detection, risk broadcasting, and cooperative vehicle response in a simulated track environment. Key use cases include hazard recognition and alert transmission, coordinated platooning for improved efficiency and safety, and early warning systems for vehicles lacking ADAS sensors. The approach emphasizes proactive decision-making among connected vehicles, showcasing the potential of V2V networks to mitigate accidents, reduce traffic congestion, and enhance road safety (U.S. Department of Transportation, 2020). The experimental setup validates the feasibility of using direct communication via ESP modules to enable decentralized, real-time coordination between vehicles.

Keywords: ADAS, Autonomous Driving, Real-Time Communication, Driver Assistance Technology, Road Safety System, Intelligent Transportation, Vehicle Networks.

1. INTRODUCTION

As the automotive industry transitions toward intelligent mobility, autonomous and semi-autonomous vehicles are becoming increasingly viable and expected to dominate future road networks. A critical component of this shift is not just self-driving capabilities, but also how vehicles perceive and respond to dynamic road conditions-both within their environment and beyond their immediate sensor range. Traditional sensing systems like radar, LIDAR, and cameras, while powerful, are limited by their field of view and environmental constraints.

Consider a common scenario: a sudden obstacle appears, and the lead vehicle brakes abruptly. The trailing driver, depending solely on visual cues like brake lights, may react too late — especially in high-speed conditions. Further behind, vehicles are even more disadvantaged due to delays in both perception and driver reaction time. This cascade effect can result in multi-vehicle collisions, particularly when the hazard is beyond visual range Human drivers typically require between 0.7 to 1.5 seconds to respond to unexpected events (Yang et al.,2004v). In highway scenarios, where vehicles move at around 35 meters per second, even a slight delay in reaction or visibility can render collisions unavoidable (Yang et al.,2004). These limitations are exacerbated under poor weather conditions, low visibility, or road curvature.

To address these issues, Vehicle-to-Vehicle (V2V) communication offers a paradigm shift (U.S. Department of Transportation, 2020). V2V enables real-time exchange of safety-critical information between vehicles, independent of driver perception or sensor range. By broadcasting data such as sudden braking, road obstructions, or speed adjustments, vehicles can collaboratively respond to hazards more efficiently (Kenney, 2011)

When integrated with Advanced Driver Assistance Systems (ADAS), this communication enables autonomous vehicles to act predictively rather than reactively (Bosch Mobility Solutions, 2023). This research explores the integration of V2V communication with an ADAS-enabled prototype platform consisting of LIDAR, camera modules, an Ackerman steering chassis, and encoder-based DC motors. The aim is to demonstrate three primary use cases: early hazard detection and broadcasting, coordinated vehicle platooning, and assisting non-ADAS vehicles through shared situational awareness through practical implementation.

2. REVIEW OF LITERATURE

The advancement of Vehicle-to-Vehicle (V2V) communication systems has prompted extensive research focused on improving vehicular safety, coordination, and intelligent decision-making.

Table-2.1 The following literature review highlights

S. No.	Title of the Paper	Authors	Year of Publication	Key Findings	Remark
1	Cooperative Collision Warning Using V2V Protocols	Xue Yang et al.	[Not Provided]	Proposed a decentralized protocol for real-time vehicular state sharing; highlighted the need for hybrid systems integrating V2V with ADAS.	Did not address integration with onboard autonomous systems in obstructed line-of-sight scenarios.

2	Dynamic Vehicle Grid Infrastructure to Sense & Respond to Traffic	U.S. Patent No. 7,782,227 B2	2010	Introduced a distributed vehicular grid for real-time road condition updates and vehicle reaction coordination.	Did not consider ADAS perception systems like LiDAR and camera integration.
3	V2VNet: Vehicle-to-Vehicle Communication for Joint Perception	Wang et al.	2020	Developed a deep learning framework enabling V2V joint perception and future state prediction in complex urban settings.	System was simulation-heavy; real-time ADAS feedback integration was not analyzed.
4	Characterizing Urban Vehicle-to-Vehicle Communications	Feng Lyu et al.	2020	Studied urban V2V communication characteristics and their reliability for safety-critical applications.	Did not provide practical ADAS integration for perception or obstacle avoidance.
5	DSRC Standards in the United States	John B. Kenney	2011	Detailed the communication standards and protocols for Dedicated Short-Range Communications (DSRC) in vehicular environments.	ADAS system-level interaction and feedback loops were not explored.
6	Strategy on Cooperative Intelligent Transport Systems (C-ITS)	European Commission	2021	Defined European strategy for implementing cooperative systems including V2V and V2I, aiming to reduce accidents and improve traffic efficiency.	Implementation with embedded microcontroller ADAS systems and real testbed-based models not addressed.

A broad range of work has demonstrated the feasibility of V2V integration in real-world traffic conditions and its synergy with advanced driver systems.

3. OBJECTIVES OF THE STUDY

- Combine perception-based ADAS (using LiDAR and camera) with V2V messaging to enable predictive and cooperative vehicle behaviour for enhanced road safety.
- Develop an event-triggered message architecture (using ESP32 and Raspberry Pi) capable of broadcasting critical safety events such as emergency braking or accidental vehicle
- Validate scenarios where a vehicle's ADAS response (e.g., braking or evasive maneuver) triggers immediate, automated mitigation actions in following vehicles via V2V communication.

4. RESEARCH METHODOLOGY

4.1 Problem Statement

Identified the limitations of standalone ADAS systems in detecting non-line-of-sight hazards and the need for inter-vehicle communication to improve road safety and response times.

4.2 System Design & Hardware Selection

Designed an embedded system integrating LiDAR, camera, encoder-based drive control, Raspberry Pi 5, and ESP32 for perception, decision-making, and V2V communication.

4.3 Software Development

Developed ROS-based modules for sensor fusion, obstacle detection, and automated braking, along with a custom V2V messaging protocol using Python.

4.4 Scenario Implementation & Testing

Simulated real-world use cases (emergency braking, lane changes, platooning) on a test track using two or more vehicles, evaluating ADAS-V2V interactions in controlled conditions.

4.5 Performance Analysis

Measured system metrics such as reaction time, message delivery latency, and collision avoidance success, and compared them with traditional ADAS and human driver baselines.

5. ANALYSIS AND DISCUSSION

5.1 System Component & Architecture

The integration of Vehicle-to-Vehicle (V2V) communication with Advanced Driver Assistance Systems (ADAS) requires a robust hardware-software co-design capable of real-time perception, decision-making, and wireless communication. The system architecture of this project is developed to simulate intelligent vehicle behavior under realistic conditions using modular components. The architecture can be divided into three major subsystems: Perception & Control, Communication, and Decision Coordination. number of pages. Use italics for emphasis; do not underline.

5.1.1 Hardware Components

5.1.1.1 Ackerman Steering Chassis with Encoder DC Motor

To simulate real-world vehicle dynamics, we employ a miniature Ackerman chassis platform equipped with encoder-based DC motors. The Ackerman geometry provides accurate steering simulation, essential for autonomous manoeuvring during lane changes or obstacle avoidance. Encoders offer precise feedback on wheel rotation, enabling closed-loop speed and position control.

5.1.1.2 Servo Motor for Steering

The steering mechanism is actuated using a high-torque servo motor to mimic realistic turning behavior. The servo is controlled through PWM signals generated via Raspberry Pi or motor control modules.

5.1.1.3 Camera (USB or Pi Cam)

The onboard camera serves as the primary visual sensor for lane detection, obstacle recognition, and path tracking. Using computer vision techniques implemented in OpenCV, the system can analyze road markings and dynamic objects in real time.

5.1.1.4 Raspberry Pi 5 (Central Control Unit)

The Raspberry Pi 5 acts as the main processing unit, handling sensor fusion, image processing, and communication with actuator systems. It runs Python scripts integrated with ROS (Robot Operating System) nodes for modular and distributed control.

5.1.1.5 ESP Modules (For V2V Communication)

ESP32 microcontrollers are used to implement the V2V communication layer. These modules operate over Wi-Fi (2.4GHz) and are responsible for broadcasting vehicle status, hazard alerts, and coordination commands to other vehicles within communication range.

5.2 Software Stack

5.2.1 Python & OpenCV

Python scripts form the control layer of the system. Using OpenCV, real-time image processing tasks such as lane detection, object recognition, and region-of-interest analysis are carried out. These scripts interact with the control logic to adjust speed and steering accordingly.

5.2.1 MATLAB/Simulink

MATLAB and Simulink are used for modelling vehicle dynamics, control algorithms, and simulating the response behavior under various V2V interaction scenarios. Simulink blocks allow real-time co-simulation with physical components for rapid prototyping.

5.2.2 ROS (Robot Operating System)

ROS provides the communication backbone between software nodes, ensuring modularity and real-time data handling. Different ROS nodes manage perception, motor control, and inter-vehicle message parsing , allowing for distributed computation and scalability.

5.3 Communication & Integration Architecture

The integration of all components follows a node-based distributed architecture:

- The Perception Node processes visual data and identifies road elements such as lanes, objects, and road signs.
- The Control Node receives inputs from the perception module and encoder feedback, computes motion commands, and actuates the motors.
- The Communication Node (ESP) broadcasts and receives V2V messages in JSON format over Wi-Fi, transmitting information such as braking status, hazard alerts, and positional updates.
- A Decision Node fuses local and received data to make high-level driving decisions, including emergency braking, lane switching, or platoon synchronization.

All nodes interact through ROS topics and Python-based data pipelines, ensuring real-time performance. The system is designed for plug-and-play expansion, allowing additional vehicles to join or leave the network without reconfiguring the architecture.

6. VEHICLE TO VEHICLE COMMUNICATION(V2V)

The dawn of connected and autonomous vehicles has ushered in a paradigm shift in transportation, moving from isolated, driver-centric systems toward networked ecosystems in which vehicles cooperate to enhance safety, efficiency, and user experience. Vehicle-to-Vehicle (V2V) communication stands at the forefront of this revolution, enabling direct, peer-to-peer exchange of critical data among vehicles without relying on centralized infrastructure.

6.1 Historical Context and Evolution

Early research into vehicular ad hoc networks (VANETs) emerged in the late 1990s and early 2000s, driven by advances in wireless networking and embedded systems (Kenney, 2011). Projects such as the U.S. Department of Transportation's Safety Pilot Model Deployment and Europe's C-Mobile initiative laid the groundwork by demonstrating basic safety applications—collision warnings, emergency electronic brake lights, and intersection movement assists—using Dedicated Short-Range Communications (DSRC) based on the IEEE 802.11p standard. Parallel efforts explored cellular technologies (e.g., 3G, LTE) for Vehicle-to-Infrastructure (V2I) and Vehicle-to-Network (V2N) use cases.

By the 2010s, automotive OEMs, tier-one suppliers, and regulatory bodies recognized that true "cooperative intelligent transport systems" (C-ITS) required vehicles to share more than just periodic beacons; they needed to transmit event-driven alerts and contextual data in real time. The emergence of Cellular V2X (C-V2X) under the 3GPP umbrella further expanded possibilities, offering both direct device-to-device communication (PC5) and network-assisted modes (Uu) under LTE and later 5G (5G Automotive Association, 2022). Today, V2V research has evolved into multi-disciplinary investigations spanning wireless protocols, control theory, cybersecurity, and human-machine interaction.

6.2 Motivations for V2V Integration

While Advanced Driver Assistance Systems (ADAS) have significantly reduced accident rates by leveraging onboard sensors—radar, LIDAR, cameras—they inherently suffer from limitations in field of view, range, and environmental robustness. For instance, radar can be blinded by heavy rain, cameras struggle in low light, and LIDAR may not penetrate fog. Moreover, these sensors cannot "see" around corners or beyond obstructing vehicles.

V2V addresses these gaps by enabling vehicles to share precise, time-stamped state information and event notifications with neighbours up to several hundred meters away. Key motivations include:

- Enhanced Safety: Real-time alerts of emergency braking, lane changes, or road hazards propagate instantly, reducing reaction delays. Studies indicate that up to 60% of rear-end collisions could be averted if drivers received warnings 0.5 s earlier (Yang et al., n.d.).
- Improved Traffic Flow: Cooperative manuever—platooning, adaptive speed harmonization—optimize road capacity and reduce congestion, yielding fuel savings and lower emissions.
- Resilience to Sensor Occlusion: By aggregating data from multiple vehicles, V2V networks provide redundancy; if one vehicle's sensors are blocked, others can relay relevant information.
- Foundation for Autonomous Driving: Fully autonomous vehicles require a 360° view of their environment. V2V extends perception to "beyond-line-of-sight," facilitating complex maneuvers like coordinated lane merges and intersection crossing without traffic lights.

6.3 Core Benefits and System Requirements

The overarching benefit of V2V is string stability—maintaining consistent inter-vehicle distances in dynamic traffic. Traditional Adaptive Cruise Control (ACC) relies solely on radar feedback, leading to amplification of speed oscillations (the "accordion effect"). In contrast, V2V-enabled Cooperative Adaptive Cruise Control (CACC) synchronizes acceleration and braking commands across the platoon, damping oscillations and improving safety margins.

Every vehicle periodically (e.g., 10 Hz) broadcasts its current motion state. Recipients infer hazards by analyzing relative kinematics (e.g., closing speed, inter-vehicle distance below safe threshold).

To realize these benefits, V2V systems must meet stringent requirements:

- Low Latency: Safety messages must traverse the network in under 100 ms end-to-end.
- High Reliability: Packet delivery ratios above 95% are essential, even in high-density or urban canyon environments.
- Scalability: Protocols must gracefully handle hundreds of vehicles within communication range without channel saturation.
- Security & Privacy: Robust authentication, encryption, and certificate management are needed to prevent spoofing and ensure data integrity.

6.4 Overview and Functional Blocks

At its core, V2V enables each vehicle to broadcast and receive time-sensitive messages containing its kinematic state (position, speed, acceleration, heading) and event flags (e.g., emergency braking, obstacle detection). By sharing these data, vehicles gain awareness of hazards beyond line-of-sight and react cooperatively, reducing collision risk and improving traffic flow.

Figure 1 illustrates a three-vehicle "dynamic grid." Each vehicle's transmission range overlaps its neighbour's', allowing multi-hop propagation of warnings. When Car 1 detects a critical event, it broadcasts an Emergency Warning Message (EWM) that Car 2 and Car 3 can receive directly or via relays.

ISSN Number: 2454-2024(Online)

International Journal of Technical Research & Science

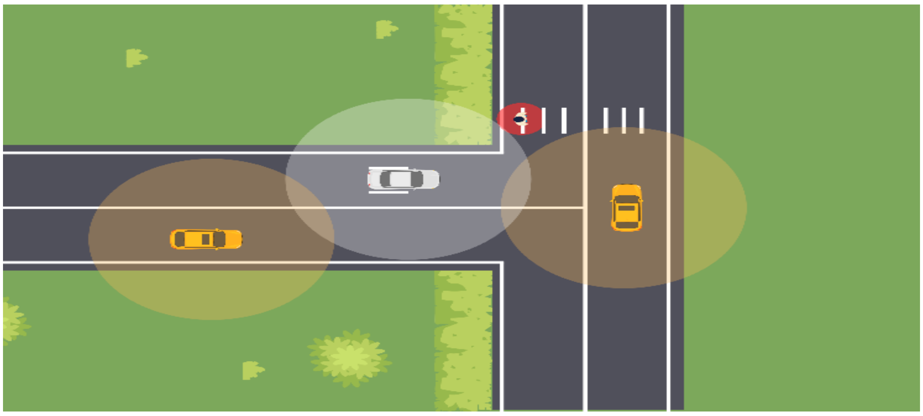


Fig. 6.1 Three-car V2V grid showing overlapping Transmit Ranges

Here we can see a potential accident likely to happen as the car turns left the pedestrian is in danger and most likely hit by the Vehicle B.

7. MESSAGE ARCHITECTURE & PROTOCOL

A robust V2V system relies on well-defined message structures and protocol mechanisms to ensure interoperability, security, and real-time performance. This section outlines the design of safety-critical messages, security frameworks, and protocol lifecycles essential for reliable V2V communication.

7.1 Basic Safety Message (BSM) Structure

The Basic Safety Message (BSM), standardized under SAE J2735 for DSRC and mirrored in C-V2X specifications, serves as the fundamental payload for cooperative safety applications (SAE International, 2021). It comprises two main parts:

Part I – Mandatory Data Elements (Fixed Payload):

- Vehicle ID: A temporary, pseudonymized identifier refreshed periodically to protect privacy.
- Timestamp: GPS-synchronized time for latency measurement and message ordering.
- Position: Latitude, longitude, elevation (±0.1 m accuracy).
- Motion State: Speed (±0.02 m/s), heading (±0.1°), steering angle.
- Acceleration: Longitudinal and lateral acceleration (±0.02 m/s²).

Part II – Optional Data Elements (Variable Payload):

- Event Flags: Emergency brake, hard acceleration, traction control activation, ABS activation.
- Vehicle Size & Type: Dimensions and classification (passenger, truck, motorcycle).
- Path History & Prediction: Sequence of past positions and predicted trajectory points.
- Additional Sensor Data: LIDAR point cloud summary, camera-based object count.

A typical BSM packet size ranges from 100 bytes (minimal) to 300 bytes (with full Part II). To conserve bandwidth, applications should tailor the inclusion of optional elements based on use-case priority and channel conditions.

7.2 Emergency Warning Message (EWM)

For the active approach, we define a streamlined Emergency Warning Message (EWM) variant optimized for minimal size and maximal urgency:

```
+----------------+----------------+----------------+
| Field          | Size (bytes)   | Description    |
+----------------+----------------+----------------+
| MsgType        | 1              | EWM identifier|
| VehicleID      | 8              | Pseudonym ID   |
| Timestamp      | 8              | Unix ms        |
| Latitude       | 8              | Degrees        |
| Longitude      | 8              | Degrees        |
| Speed          | 2              | km/h           |
| Acceleration   | 2              | m/s²           |
| EventCode      | 1              | E.g., 0x01=Emergency Brake |
| HopCount       | 1              | Relay counter  |
+----------------+----------------+----------------+
```

Fig. 7.1 Code 1: EWM

At 12 bytes minimum (excluding overhead), EWMs ensure sub-50 ms transmission times even under congested channel loads (5G Automotive Association, 2022). The HopCount prevents infinite relays; vehicles discard messages exceeding a preset maximum (e.g., 3 hops)

7.3 Message Types & Priority Classes

To ensure channel efficiency and meet real-time requirements, V2V messages are classified into three priority levels:

Class 1 – Emergency Warning Messages (EWMs):

- Trigger: Generated only upon detection of abnormal events (e.g., deceleration > threshold, sudden swerving, crash sensor activation).
- Contents: Vehicle ID, GPS coordinates, speed, acceleration, heading, event type, hop count.
- Latency: Must be delivered within tens of milliseconds.

Class 2 – Forwarded EWMs:

- Purpose: Extend the warning's reach by relaying EWMs
- Hop Count: Incremented by each relay node to prevent infinite loops.

Class 3 – Non-Time-Sensitive Messages:

- Examples: Periodic beacons of position and speed for passive hazard inference.
- Latency Tolerance: Hundreds of milliseconds or more.

By restricting continuous background traffic to Class 3 only when bandwidth permits, the network reserves capacity for critical EWMs (Class 1/2) under high-density conditions.

8. PASSIVE vs ACTIVE V2V PARADIGMS

V2V communication systems generally adopt one of two paradigms for disseminating vehicular state information: the passive paradigm, which relies on periodic broadcasting of vehicle kinematics, and the active paradigm, which transmits only when critical events occur. Each approach presents distinct advantages and trade-offs in terms of channel utilization, latency, scalability, and detection accuracy.

8.1 Passive Paradigms

- Continuous Situational Awareness: Recipients always have up-to-date neighbour states, enabling early inference of abnormal behavior even before explicit warnings.
- Simplicity: No need for event-detection thresholds; hazard inference is purely algorithmic based on relative metrics.
- Redundancy: Multiple BSMs compensate for occasional packet loss; smoothing algorithms can filter out spurious data.

8.2 Challenges

- High Channel Load: At 10 Hz and ~200 bytes per BSM, a platoon of 50 vehicles can generate ~80 kbps per vehicle, saturating the channel in dense scenarios (Lyu et al., 2020).
- Scalability Limits: As vehicle density increases, contention and collisions on the shared medium lead to elevated packet loss and latency spikes.
- Inference Delay: Hazard detection relies on computing metrics like time-to-collision, which may require multiple BSMs to confirm, introducing processing delays.

8.3 Analytical Model

Let N be the number of vehicles within range, each sending BSMs of size S bytes at rate f Hz. The aggregate offered load L is:

$$L=N\times f\times S\times 8(\text{bits/s})$$

For a 10 MHz DSRC channel with effective throughput T≈6T Mbps, the channel becomes saturated when L≥T. Thus, maximum sustainable N is:

$$Nmax = \frac{T}{(8fST)}$$

With f =10Hz and S=200bytes, Nmax≈375N vehicles—ignoring MAC overhead. In practice, contention reduces this number substantially, limiting effective platoon sizes to tens of vehicles.

8.4 Active Paradigms

Mechanism: Vehicles remain silent during normal operation Upon detecting a threshold-exceeding event (e.g., deceleration > 6 m/s², sudden steering angle change, crash sensor activation) (Yang et al., 2004) , a vehicle immediately broadcasts an Emergency Warning Message (EWM). Recipients act on the warning and may relay the EWM for up to H hops, extending coverage without periodic background traffic.

ISSN Number: 2454-2024(Online)

International Journal of Technical Research & Science

Advantages

- Minimal Channel Overhead: EWMs occur only during critical events, drastically reducing average channel load.
- Focused Urgency: By isolating safety-critical alerts, the network ensures that high-priority messages face minimal contention.
- Scalable in Dense Traffic: With infrequent EWMs, the system remains robust even as NNN grows large.

Challenges

- Detection Reliability: Effective event detection relies on accurate, low-latency sensing. False negatives (missed events) can lead to unalerted hazards.
- Limited Background Awareness: Without periodic beacons, vehicles lack continuous neighbor state information, potentially delaying hazard inference for events not directly sensed.
- Relay Complexity: Multi-hop relays must manage hop-count limits and loop prevention, requiring additional protocol logic.

8.5 Analytical Model

Assume events occur with average rate λ events/s per vehicle. The average offered load Lactive is:

$$\text{Lactive}=N\times\lambda\times \text{SEWM}\times 8\times(1+\bar{H})$$

where S_{EWM} is the EWM size (e.g., 50 bytes) and $\bar{H}$ is the average relay hops per event. For N=100N $\lambda=0.01\ s^{-1}$ (one event per 100 s), S_{EWM}=50 bytes, and $\bar{H}$=2, we get:

$$\text{Lactive}\approx 100\times 0.01\times 50\times 8\times 3=1200 \text{ bits/s}$$

This load is negligible compared to periodic BSM traffic, demonstrating the active paradigm's channel efficiency.

8.6 Hybrid Approaches

To leverage both paradigms' strengths, hybrid schemes combine low-rate periodic beacons (e.g., 1 Hz) with event-driven EWMs. This ensures minimal background awareness while preserving channel headroom for emergencies. Congestion control algorithms adjust beacon rates dynamically based on channel load and vehicle density.

8.7 Implementation in Our Prototype

Our testbed uses the active paradigm exclusively, with the following parameters:

- EWM Rate: Event-driven only; no periodic beacons.
- Hop Limit H: Configured to 3 to balance coverage and overhead.
- Deceleration > 5 m/s² triggers EWM.
- Steering angle change > 15° within 100 ms.
- Relay Logic: Each ESP32 increments HopCount and discards messages with HopCount > 3.

This configuration yielded an average EWM end-to-end latency of 35 ms over two hops in indoor tests and maintained a 98% delivery ratio under moderate Wi-Fi interference—validating the active approach's efficacy for our use cases.

9. MULIT HOP RELAY ALGORITHM

Efficient propagation of Emergency Warning Messages (EWMs) beyond direct communication range relies on multi-hop relay strategies. This section describes the graph-based relay model, hop-count management, loop prevention, and prioritization to ensure timely, reliable dissemination in dynamic vehicular networks.

9.1 Graph Model of Vehicular Network

We model the set of vehicles as a dynamic undirected graph G=(V,E) where:

- V represents vehicles within the network.
- An edge (u,v)∈E exists if vehicles u and v are within direct communication range (e.g., ≤ 200 m).

Upon generation of an EWM by vehicle s, the goal is to deliver the message to all nodes within k hops, where k is the maximum relay limit.

9.2 Hop-Count Management

Each EWM carries a HopCount field initialized to zero. Upon receiving an EWM:

- If HopCount < MaxHops, the relay node increments HopCount by one.
- The node rebroadcasts the updated EWM after a short, randomized delay $\delta\in[0,\Delta]$ ms to reduce simultaneous transmissions.
- If HopCount >= MaxHops, the node processes but does not relay the message further. This mechanism bounds the flood and prevents excessive channel.usage.

9.3 Loop Prevention

To avoid rebroadcast storms and loops, each node maintains a Recently Seen Table (RST) of message identifiers

ISSN Number: 2454-2024(Online)

International Journal of Technical Research & Science

(e.g., hash of VehicleID + Timestamp + EventCode). On receiving an EWM:

- If the identifier exists in RST, the message is discarded.
- Otherwise, the identifier is added to RST with a timeout equal to the message lifetime (e.g., 500 ms).

This simple cache prevents duplicate relays and ensures loop-free propagation.

9.4. Code Snippet: Relay Logic (Python)

```python
def on_ewm_received(ewm):
    msg_id = hash((ewm.vehicle_id, ewm.timestamp, ewm.event_code))
    if msg_id in recently_seen:
        return
    recently_seen.add(msg_id)
    process_ewm(ewm)
    if ewm.hop_count < MAX_HOPS:
        ewm.hop_count += 1
        delay = compute_backoff(ewm.source_distance)
        threading.Timer(delay / 1000.0, broadcast_ewm, args=[ewm]).start()
```

Fig. 9.1 Code 2: Loop Prevention

By implementing these multi-hop relay strategies and integrating them into our modular ROS-ESP32 platform, the prototype effectively demonstrates scalable, low-latency V2V communication aligned with both DSRC and C-V2X principles.

10. INTRODUCTION TO ADAS

Advanced Driver Assistance Systems (ADAS) are intelligent technologies integrated into modern vehicles to improve road safety, enhance driving comfort, and support semi-automated driving. These systems continuously perceive the driving environment and make real-time decisions based on the fusion of multiple sensor inputs.
ADAS typically assists in critical scenarios such as emergency braking, collision avoidance, lane departure warning, and adaptive cruise control. The key to their functionality lies in accurate perception, fast decision-making, and precise actuation.

10.1 Practical ADAS Reactions to V2V Inputs

V2V Event Type	ADAS Response
Emergency Brake (Car A)	Immediate braking (if ADAS active) or flashing dashboard warning
Road Hazard Ahead	Speed reduction, switch to cautious mode
Collision Warning	Lane change preparation, request driver override
Slippery Surface Alert	Adjust traction control, reduce speed

These responses are tiered by urgency. High-priority events bypass the driver alert step and immediately engage autonomous control, whereas moderate events may suggest driver intervention.

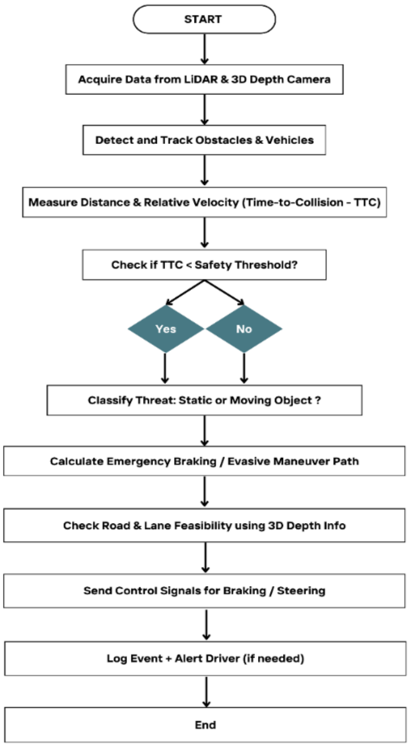
10.2 ADAS Algorithm for Emergency Collision Avoidance

The ADAS framework is equipped with a specialized algorithm for emergency collision avoidance, which functions as a real-time decision engine. Upon detection of a potential obstacle in the vehicle's forward path, the system initiates a threat evaluation process based on object distance, relative velocity, and projected trajectory. A key parameter in this evaluation is the Time-To-Collision (TTC), which is calculated by dividing the distance between the vehicle and the obstacle by their relative speed. When TTC falls below a predefined safety threshold, the system classifies the situation as critical.
The decision-making engine then distinguishes between stationary and dynamic objects to determine an appropriate manoeuvre. If braking is deemed the most effective response, a deceleration profile is generated, and commands are issued to the braking actuator. Alternatively, if a steering manoeuvre is safer and feasible based on the spatial clearance identified by LiDAR and camera data, an avoidance trajectory is computed using

ISSN Number: 2454-2024(Online)

International Journal of Technical Research & Science

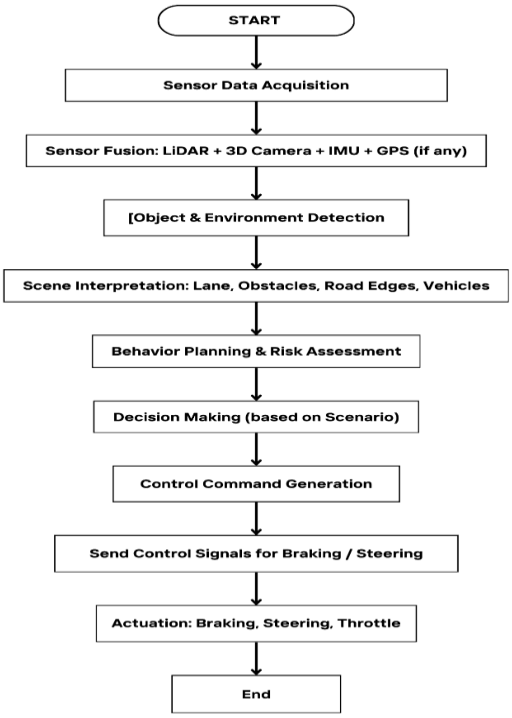
polynomial trajectory generation techniques. This entire process occurs within milliseconds, ensuring minimal delay in actuation. Additionally, driver alert systems may be activated to issue visual and acoustic warnings during the intervention



Flow Chart 10.1 Emergency Avoidance System

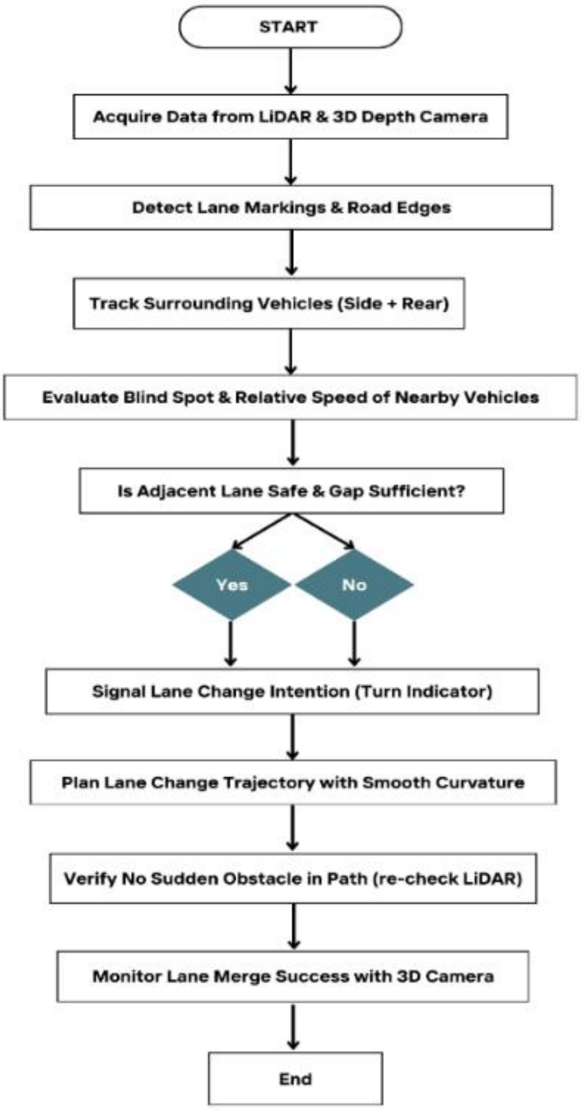
10.3 ADAS Algorithm for Lane Change Assist

Lane change scenarios are handled using a separate but integrated algorithm that relies heavily on the detection of lane geometry and surrounding traffic. Initially, the 3D depth camera identifies lane boundaries using deep learning-based lane segmentation models. Concurrently, the LiDAR scans for adjacent vehicles in both lateral and rear zones, particularly focusing on blind spot regions that are not visible to the human driver.



Flow Chart 10.2 Lane Change Assist

To evaluate the safety of a lane change, the system performs a gap analysis, determining the spatial and temporal distance to the closest vehicles in the target lane. If the system calculates that the vehicle can change lanes without compromising safety, it initiates a decision-making protocol. This involves planning a lateral trajectory that ensures a smooth transition between lanes, maintaining vehicular stability and passenger comfort. Before executing the maneuver, the system verifies one final time—using live LiDAR and camera data—that the intended lane is still free of obstacles. Once confirmed, control commands are transmitted to the steering module to carry out the maneuver seamlessly.



Flow Chart 10.3 Decision-Making Protocol

10.4 General ADAS Decision-Making Framework

The overarching decision-making process in ADAS systems involves three primary stages: perception, planning, and control. The perception stage fuses sensory input into a coherent environmental model. The planning stage uses this model to forecast the behavior of surrounding agents and generate optimal responses. Finally, the control stage translates these responses into physical actions by actuating the vehicle's brake, throttle, and steering systems.

The decision-making logic is often implemented using a combination of rule-based systems and reinforcement learning techniques. This hybrid approach enables the ADAS to balance deterministic safety rules (e.g., safe following distance, legal speed limits) with adaptive learning from real-world driving scenarios. Predictive models further enhance safety by forecasting the future positions of dynamic obstacles, thereby allowing the vehicle to respond proactively rather than reactively.

11. ADAS SENSOR OPERATION WITH V2V INTEGRATION

This section describes, at a high level, how the ADAS sensors (LIDAR and camera) work together with V2V messaging in critical scenarios.

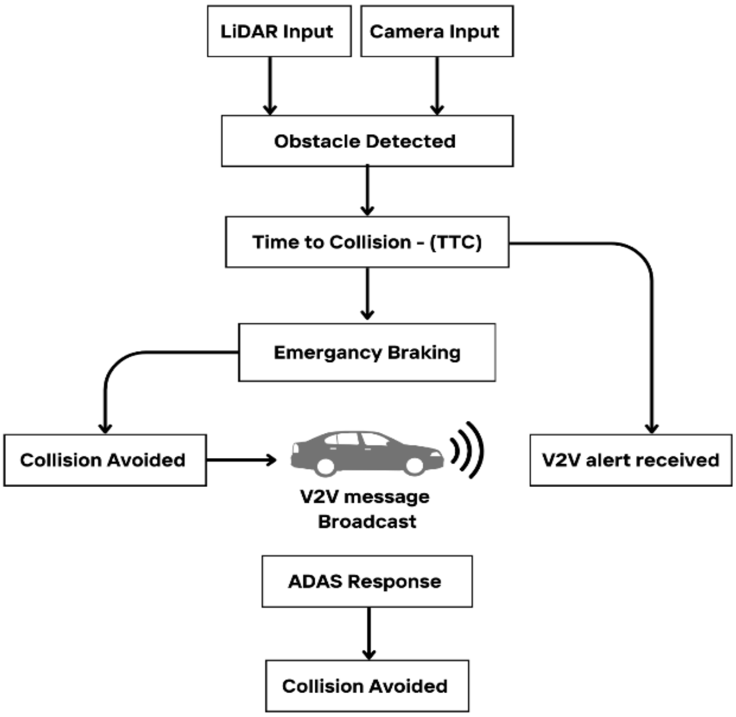


Fig. 11.1 ADAS Sensor Operation with V2V Integration

11.1 In-Depth: ADAS Mitigation + V2V Mitigation Sequence

This subsection unpacks **Flow Chart 1**, detailing how Vehicle A's ADAS pipeline detects a critical obstacle, executes mitigation, and issues a V2V warning that Vehicle B's ADAS then uses to autonomously react. We break the process into sensing, perception, decision, actuation, and communication phases.

11.2 Sensor Configuration & Synchronization

LiDAR: Mounted at the front bumper, the 16-beam LiDAR scans a 120° horizontal field at 10 Hz, with a range of up to 50 m. Each sweep yields ~10,000 points; onboard driver timestamps scans using a hardware PPS (pulse-per-second) from GPS to align with camera frames.

- Camera: A global-shutter RGB camera captures 720p frames at 30 Hz. Intrinsic parameters (focal length, distortion coefficients) are pre-calibrated and stored on the Raspberry Pi.

A shared system clock (via ROS Time) fuses timestamps so that each LIDAR point cloud and camera image share a common temporal reference.

11.3 Real-Time Data Acquisition

- LIDAR Scan Capture: At t_0, the LiDAR completes a 10 ms sweep. The driver node publishes a /lidar_points ROS message containing the point cloud.
- Camera Frame Grab: Within 5 ms of t_0, the camera captures an image, published as /camera_image.

Both streams are queued in ROS with header timestamps, ensuring synchronized processing.

11.4 Preprocessing & Filtering

11.4.1 Point Cloud Processing

- Ground Removal: A RANSAC-based plane fitter removes ground points (road surface).
- Downsampling: Voxel grid filter reduces point density by 50% to meet real-time constraints.
- Resulting cloud (~5,000 points) retains obstacles above 0.2 m in height.

11.4.2 Image Enhancement

- Undistort: Lens distortion correction using pre-computed calibration map.
- Histogram Equalization: Adaptive equalization improves contrast under varying lighting

11.5 Object Detection & Classification

LIDAR Clustering: Euclidean Cluster Extraction segments points into clusters. Clusters with bounding box dimensions matching typical vehicle size (length 1.5–4.5 m, width 1.2–2.2 m) are labeled as "vehicle candidates."

Camera-Based Classification: A lightweight CNN (e.g., MobileNet SSD) runs on the Raspberry Pi, classifying vehicles, pedestrians, or static obstacles in the image ROI (Tesla AI Day, 2022)800 corresponding to LIDAR cluster projections. For each LIDAR cluster, the system projects its 3D bounding box into the image plane; if the CNN confidence > 0.6 for "vehicle," the cluster is confirmed.

11.6 Sensor Fusion & Risk Assessment

Kalman Filter Tracking: Each confirmed obstacle feeds into a multi-object tracker using an Unscented Kalman Filter, maintaining state vectors $[x, y, \dot{x}, \dot{y}]$ (Singh & Patel, 2024)
Time-to-Collision (TTC) Computation: or the closest tracked object, TTC is computed as

$$TTC = \frac{d}{v_{ego} - v_{obs}}$$

where d is current range, v_{ego} is vehicle speed, and v_{obs} is obstacle relative speed.
Threshold Check : If TTC < 1.8 s (configurable parameter), the Risk Assessment Node flags an imminent collision.

11.7 ADAS Decision & Actuation (Vehicle A)

- Brake Command Generation: The control law maps TTC to a brake torque command via a lookup table:

  Tbrake= f(TTC)

  where shorter TTC yields higher torque.
- Brake Actuation: The Raspberry Pi publishes a /brake_cmd ROS topic; a CAN-bus interface (or PWM to a brake actuator) converts this to hydraulic pressure.
- Confirmation Feedback: Wheel-speed sensors confirm deceleration; if deceleration rate < target, torque is increased iteratively until target is achieved or wheels lock.

11.8 V2V Warning Generation

11.8.1 Event Packaging

Upon brake command send, the ADAS node composes an Emergency Warning Message (EWM)

```
{
  "vehicle_id":"A123",
  "event":"EMERGENCY_BRAKE",
  "timestamp":1700000000123,
  "lat":50.7753,
  "lon":6.0839,
  "speed":24.5,
  "heading":1.57,
  "ttc":1.2,
  "hop_count":0
}
```

Fig. 11.2 Code 3: Emergency Warning Message (EWM)

11.8.2 ROS Bridge

The EWM is published on /ewm_out. A v2v_comm_node serializes this JSON and sends it via UART to the ESP32.

11.9 Alert Reception & Verification (Vehicle B)

- UART Reception: ESP32 on Vehicle B forwards incoming packets to the Raspberry Pi's serial port.
- Signature & Timestamp Check: The v2v_input_node verifies ECDSA signature and ensures message age < 500 ms.
- Alert Publication: A validated EWM is republished on /ewm_alert.

11.10 ADAS Response (Vehicle B)

- LIDAR: Confirms obstacle at expected bearing/rough range ±2 m.
- Camera: Verifies lane position is free for braking (no current lane-change underway).
- Brake Actuation : Identical control law as Vehicle A triggers /brake_cmd. Because Vehicle B's TTC would have been ~1.5 s, this intervention reduces effective TTC to ~1.2 s, providing extra margin.
- Relay Logic: After initiating braking, v2v_comm_node increments hop_count to 1, rebroadcasts the EWM to ensure trailing vehicles longer chain also receive the warning.

11.11 Performance & Safety Gains

1) Latency Breakdown:

- Sensor detection → TTC flag: ~12 ms
- Brake command execution: ~8 ms
- EWM broadcast: ~5 ms
- Reception & verification on Vehicle B: ~15 ms
- Total end-to-end ≈ 40 ms.

Compared to human reaction (~800 ms), this system affords an additional 760 ms of warning, translating to ~15 m of stopping distance at 18 m/s (65 km/h) (Yang et al.,2004). By tightly integrating high-frequency LIDAR and camera perception with real-time V2V alerts, the ADAS pipelines on both vehicles act cooperatively to mitigate collisions in a fraction of the time required by conventional approaches.

Table-11.1 V2V + ADAS Use Cases

Use Case	V2V Role	ADAS Enhancement
Emergency Braking Ahead	Alert via V2V	Preemptive AEB
Intersection Collision Warning	Position exchange	Alert + Lane Correction
Blind Curve Warning	NLOS vehicle alert	Braking + Lane Assist
Cooperative Adaptive Cruise Control (C-ACC)	Synchronized acceleration/deceleration	Platooning
Lane Change Warning	Position & speed awareness	Safe lane decisions
Do Not Pass Warning	Oncoming vehicle info	Warning + Acceleration Cutoff

12. SECURITY & PRIVACY CONSIDERATIONS

Security is a non-negotiable requirement in any Vehicle-to-Vehicle (V2V) communication system. Since V2V enables autonomous or semi-autonomous actions based on messages from other vehicles, any compromise in the integrity or authenticity of this data can result in dangerous or even catastrophic outcomes (European Commission, 2021). This section outlines how we handle message authentication, data integrity, anonymity, revocation, and defence against spoofing, all while maintaining the low-latency requirements of safety-critical applications.

12.1 Threat Model in V2V Communication

V2V systems face multiple security threats, including

1. Message Forgery: A malicious actor generates fake EWMs to disrupt traffic or trigger unnecessary evasive actions.
2. Replay Attacks: Old but valid messages are retransmitted to confuse nearby vehicles.
3. Eavesdropping: Adversaries listen in on unencrypted V2V traffic to gather sensitive information.
4. Location Tracking: Continuous reception of messages tied to a single vehicle allows adversaries to track movement patterns.
5. Denial-of-Service (DoS): Spamming the channel with bogus messages can delay or prevent delivery of legitimate warnings.

13. FUTURE SCOPE

The fusion of ADAS with V2V communication opens many avenues for advancement in autonomous and connected mobility:

13.1 AI-Driven Decision-Making

The use of machine learning and neural networks in ADAS can allow for real-time adaptation to driving styles, environments, and behavioral trends, making the system more personalized and efficient (Tesla AI Day, 2022).

13.2 Cloud-Integrated Fleet Learning

V2V communication can be extended to vehicle-to-cloud (V2C), enabling fleet-wide learning where one vehicle's experience benefits all connected vehicles through data sharing and over-the-air updates.

13.3 Expansion to V2X Ecosystem

V2V is just a subset of the broader V2X (Vehicle-to-Everything) framework. Integration with infrastructure (V2I), pedestrians (V2P), and networks (V2N) will create a fully connected traffic ecosystem.

13.4 Real-World Deployment and Smart City Integration

ISSN Number: 2454-2024(Online)

International Journal of Technical Research & Science

Governments and municipalities are increasingly investing in smart infrastructure, and future ADAS-V2V systems can be directly synchronized with traffic signals, roadworks, and dynamic speed limits.

CONCLUSION

The combination of Advanced Driver Assistance Systems (ADAS) and Vehicle-to-Vehicle (V2V) communication represents a critical leap toward enhanced road safety, accident prevention, and autonomous mobility. While ADAS is capable of real-time decision-making using onboard sensors like LiDAR and cameras, the integration of V2V adds a new dimension: proactive awareness beyond line-of-sight and direct perception.

Through realistic scenarios such as emergency braking, lane changes due to obstacles, and coordinated accident mitigation, this paper demonstrated how V2V alerts act as an extension of the vehicle's senses. Vehicles with ADAS not only react to their immediate surroundings but also anticipate events shared by nearby vehicles—thereby improving reaction time, reducing collisions, and optimizing traffic behavior.

However, several challenges still need to be addressed, including infrastructure readiness, standardization, and cybersecurity. Looking forward, the synergy between AI, cloud connectivity, and evolving V2X technologies will continue to strengthen the bridge between autonomy and safety. The path to collaborative intelligence on the road is now clearer than ever.

REFERENCES

[1] 5G Automotive Association. (2022). C-V2X use cases and deployment guidelines.

[2] Bosch Mobility Solutions. (2023). ADAS technology overview. Retrieved from https://www.bosch-mobility-solutions.com

[3] Continental AG. (2023). Vehicle environment perception using LiDAR and cameras: Technical whitepaper.

[4] European Commission. (2021). Strategy on cooperative intelligent transport systems (C-ITS).

[5] Kenney, J. B. (Year). Dedicated short-range communications (DSRC) standards in the United States. Journal/Publisher, Volume (Issue), Page Numbers.

[6] Lyu, F., Zhu, H., Cheng, N., Zhou, H., Xu, W., Li, M., & Shen, X. (Year). Characterizing urban vehicle-to-vehicle communications for reliable safety applications.

[7] National Highway Traffic Safety Administration (NHTSA). (2021). Advanced driver assistance systems (ADAS) guidelines.

[8] SAE International. (2021). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles – SAE J3016_202104.

[9] Singh, S., & Patel, K. (2024). A comparative study on sensor fusion approaches in ADAS. Journal of Vehicle Intelligence, 11(3), 213–230.

[10] Tesla AI Day. (2022). Vision-based perception and full self-driving stack. Retrieved from https://www.tesla.com.

[11] Thunberg, J., Lyamin, N., Sjöberg, K., & Vinel, A. (Year). Vehicle-to-vehicle communications for platooning: Safety analysis.

[12] U.S. Department of Transportation. (2020). Vehicle-to-vehicle communications: Readiness of V2V technology for application.