Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in ICT For Internet And Multimedia

Using SUMO to Simulate V2V Control Algorithms and Evaluate the Impact of Communication Protocols on Packet Size

Master Candidate

Elnaz Dolati

Student ID 2071516

Supervisor

Prof. Andrea Zanella

University of Padova

Co-supervisor

Dott. Muhammad Sohail

University of Padova

Academic Year
2024/2025

Abstract

Autonomous vehicles (AVs) coordinate their movements through cooperative communication, especially at unsignalized intersections where traffic lights or other control infrastructure are absent. Two main paradigms are considered in this context: decentralized Vehicle-to-Vehicle (V2V) coordination, where vehicles exchange information directly, and centralized Vehicle-to-Infrastructure (V2I) management, where an intersection controller assigns priorities for each vehicle.

While both strategies have been studied in terms of safety and traffic efficiency, less attention has been given to their communication requirements. In particular, critical factors such as the size and frequency of exchanged messages determine whether existing Vehicle-to-Everything (V2X) technologies like Dedicated Short-Range Communication (DSRC), Cellular V2X (C-V2X), and 5G New Radio V2X (5G NR-V2X) can support large-scale deployment.

In this thesis, V2V- and V2I-based control algorithms are implemented in the Simulation of Urban MObility (SUMO) framework, and the communication overhead they generate is accounted for. Both normal traffic conditions and disrupted scenarios are considered, including full blockages caused by pedestrian crossings and partial blockages caused by congestion or accidents.

By examining the data generated in SUMO, the communication requirements of each cooperative intersection management approach are revealed. Comparing control strategies and communication protocols makes it possible to highlight the trade-offs between scalability, efficiency, and robustness. This information can serve as valuable guidance for the future deployment of cooperative autonomous traffic systems.

Keywords: autonomous vehicles, V2X communication, intersection management, SUMO simulation, packet size analysis, DSRC, C-V2X, 5G NR-V2X

Contents

List of Figures

List of Tables

List of Code Snippets

List of Acronyms

API	Application Programming Interface
AV	Autonomous Vehicle
BSM	Basic Safety Message
CARLA	Car Learning to Act (Open-source Autonomous Driving Simulator)
CSV	Comma-Separated Values
C-V2X	Cellular Vehicle-to-Everything
DSRC	Dedicated Short-Range Communications
ETSI	European Telecommunications Standards Institute
IEEE	Institute of Electrical and Electronics Engineers
IM	Intersection Manager
ITS	Intelligent Transportation Systems
JSON	JavaScript Object Notation
LLC	Logical Link Control
MAC	Medium Access Control
NR	New Radio
OSM	OpenStreetMap
PC5	Proximity Communication Interface 5
QoS	Quality of Service

Uu Cellular Uplink/Downlink Interface

3GPP 3rd Generation Partnership Project

PDCP Packet Data Convergence Protocol

QoS Quality of Service

RLC Radio Link Control

ROS Robot Operating System

RSU Roadside Unit

SCI Sidelink Control Information

SNAP Subnetwork Access Protocol

SPS semi-persistent scheduling

SUMO Simulation of Urban MObility

TraCI Traffic Control Interface

URLLC Ultra-Reliable Low-Latency Communication

UTF-8 8-bit Unicode Transformation Format

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

5G NR-V2X 5G New Radio Vehicle-to-Everything

1

Introduction

1.1 Background and Motivation

Intelligent Transportation Systems (ITS) are envisioned as a cornerstone of future mobility. ITS are expected to play an important role in the future of mobility. It aims to reduce accidents by integrating advanced communication, sensing, and control technologies, improve traffic efficiency, and enable new mobility paradigms such as shared autonomous fleets and mobility-on-demand services.

A crucial requirement for safe and coordinated operation in ITS is the ability to communicate effectively. This paradigm is referred to as V2X communication, which includes both V2V and V2I exchanges [9, 19]. In V2V, vehicles exchange their position, speed and intentions directly with nearby vehicles to coordinate maneuvers. In V2I, vehicles report to roadside units or an intersection manager, which can assign priorities and optimize flows.

Intersections are natural conflict points in road networks, where vehicles, pedestrians, and cyclists converge. Miscommunication or poor coordination at intersections often leads to accidents and congestion. International road safety reports repeatedly indicate that intersections account for a high proportion of traffic collisions involving both vehicles and pedestrians [24]. In the context of autonomous driving, where human negotiation signals are absent, reliable communication becomes a necessity to ensure both safety and traffic efficiency.

Much of existing research on intersection management has focused on developing control algorithms that allow autonomous vehicles to coordinate their movements in either decentralized (V2V) or centralized (V2I) settings [6, 5, 13]. However, a key practical question remains underexplored: What is the actual communication demand of these strategies in real traffic? In particular, the size and frequency of the packets exchanged directly determine whether these systems are scalable.

This question becomes even more relevant in the case of unexpected events. Pedestrians crossing unsignalized intersections or temporary lane blockages due to accidents can force vehicles to reassess priorities and exchange additional messages. These disruptions not only influence traffic flow, but may also considerably increase the communication load [15].

Candidate communication technologies include DSRC, C-V2X, and 5G NR-V2X. Each of these introduces different protocol overheads and bandwidth constraints. DSRC is based on Institute of Electrical and Electronics Engineers (IEEE) 802.11p and is valued for its low latency but faces congestion problems in dense traffic. C-V2X extends range and reliability by using cellular infrastructure, while 5G NR-V2X provides ultra-reliable low-latency communication and higher throughput for advanced cooperative applications [4, 7]. Understanding how these technologies perform under realistic conditions requires analyzing the actual size of exchanged packets and the communication overhead of control algorithms.

The motivation of this thesis is therefore to address this gap. The purpose of the work is to provide a clearer picture of the communication requirements of cooperative autonomous driving at unsignalized intersections. This is achieved by simulating realistic traffic scenarios with both V2V and V2I control strategies and by measuring the size of the messages exchanged under normal and interrupted conditions.

1.2 Problem Statement

The deployment of ITS in urban environments depends on the ability of cooperative algorithms to ensure safe and efficient intersection crossing. Most of the research has been carried out, with approaches ranging from decentralized V2V coordination to centralized V2I management to explore this challenge. Although these works have shown promising strategies for decision making and coordination, they often assume ideal communication channels and unlimited network capacity.

In practice, communication resources are constrained and the data exchanged between vehicles can become a bottleneck in large-scale deployments. Most existing studies evaluate intersection management in terms of safety, performance, or travel time. However, they rarely consider the communication load required to achieve these results. The feasibility of cooperative driving systems depends not only on the logic of the control algorithm but also on the size of the messages it generates. If communication overhead becomes excessive, theoretically sound algorithms may fail in real-world settings due to congestion, packet loss, or channel saturation.

Another limitation is that most of these evaluations are performed under idealized traffic conditions. Disruptions that frequently occur on real roads have not been considered. Events such as pedestrians crossing unsignalized intersections, lane blockages caused by stalled vehicles, or sudden congestion due to accidents can significantly increase communication requirements. During these situations, vehicles must re-negotiate their decisions and dynamically adapt their routes, requiring the exchange of additional messages. The absence of systematic studies measuring how these disruptions affect communication requirements leaves a critical gap in the literature.

Therefore, the problem addressed by this thesis has two aspects. First, there is a lack of systematic analysis of the communication overhead generated by cooperative V2V and V2I algorithms in realistic intersection scenarios. Second, there is limited information about how V2X technologies—such as Dedicated Short-Range Communications (DSRC), Cellular V2X (C-V2X), and 5G New Radio V2X (5G NR-V2X)—scale under dynamic and disrupted traffic conditions. Without this knowledge, it is difficult to evaluate which combination of control strategy and communication technology is most suitable for reliable deployment in ITS.

1.3 Objectives

In this thesis, we will measure the communication requirements of cooperative intersection management algorithms for ITS. We use two control methods: decentralized V2V coordination and centralized V2I management. The purpose of the work is to quantify the required message exchange of these methods, both in normal operation and during unexpected events such as congestion or partial lane blockages.

For measuring the size of the messages exchanged during simulation and mapping them to existing V2X technologies, we consider these protocols: DSRC, C-V2X and 5G NR-V2X.

More specifically, the research is guided by the following questions:

- What levels of communication overhead are introduced by cooperative intersection management algorithms in normal traffic conditions?
- How do unexpected interruptions, such as full or partial blockages, affect the size of exchanged messages?
- How different are the decentralized V2V and centralized V2I control strategies in terms of communication efficiency?
- Is there any overlap between the requirements of these algorithms and the capabilities and constraints of existing V2X communication standards (DSRC, C-V2X, and 5G NR-V2X)?

Answering these questions will help determine which strategies and technologies are most suitable to enable scalable and reliable cooperative driving at urban intersections.

1.4 Approach Overview

To serve the objectives described above, this thesis follows a simulation-based approach. The core idea is to establish a controlled, realistic environment in which both (V2V) and (V2I) control strategies can be tested, compared, and investigated with respect to their communication requirements.

The SUMO platform serves as the core simulation environment for modeling detailed traffic flows [14].

SUMO is a microscopic simulator capable of representing the movements of individual vehicles with high spatial and temporal resolution, which makes it well suited to analyze coordination at intersections. By interfacing SUMO through custom modules written in Python via the Traffic Control Interface (TraCI) API [23], it is possible to implement real-time autonomous decision logic and message exchanges.

Two different control models are integrated into this environment. The first approach is a decentralized V2V scheme, where vehicles share their local state (position, speed, and intended route) with nearby vehicles and make navigation decisions independently. The second approach is a centralized V2I scheme, in which vehicles periodically report their state to an intersection manager. The manager coordinates priorities and issues instructions. Implementing both in identical traffic scenarios ensures that their communication requirements can be compared fairly.

A standard communication structure is defined to represent the content of the messages exchanged. Each message is encoded and then assigned to different V2X technologies by adding protocol-specific overheads. For instance, the DSRC overhead is based on specifications from the European Telecommunications Standards Institute (ETSI) Cooperative Awareness Basic Service standard [7], while the bounds for C-V2X and 5G NR-V2X are derived from technical analyzes such as the MoVeOver Deliverable D3 report [17]. This abstraction allows for the estimation of realistic packet sizes while keeping the control logic independent of any specific protocol.

To test the robustness of the simulation, we introduce disruptions that reflect common real-world challenges. A full blockage is used to mimic a pedestrian crossing at the intersection and a partial blockage is used to represent congestion at the lane level or accidents. These events force vehicles to adapt their routes and increase the frequency of coordination messages, thereby illustrating how the communication load scales under stress.

Finally, the system is equipped with logging mechanisms that record the size of each packet transmitted. This dataset enables a detailed analysis of message size distributions, and protocol efficiency between scenarios. By combining realistic traffic modeling, implementation of the control algorithm, abstraction of protocol-sensitive communication, and injection of disruption, the approach provides a comprehensive framework to evaluate the scalability and feasibility of cooperative intersection management strategies.

1.5 Thesis Structure

The remainder of this thesis is organized into six chapters, each addressing a specific aspect of the research and gradually building toward the final conclusions.

Chapter 2 provides the theoretical background necessary to understand the scope of the work. It introduces the main concepts of V2X communication, including the technologies considered in this thesis (DSRC, C-V2X, and 5G NR-V2X), and describes the two coordination paradigms of interest: decentralized V2V and centralized V2I.

Chapter 3 reviews related work in the literature. It surveys cooperative intersection management strategies, existing studies on vehicular communication protocols, and prior uses of SUMO in traffic simulation. This chapter highlights the research gap regarding the quantification of communication overhead, which forms the motivation for this thesis.

Chapter 4 describes the design and implementation of the simulation framework. It explains the overall architecture, message formatting, and the logic behind the V2V and V2I control modules. It also details how disruptions such as pedestrian crossings and lane blockages are introduced into the system and how logging is performed to capture communication data.

Chapter 5 outlines the methodology followed during the experiments. It defines the simulation environment and parameters, explains the execution workflow, and describes the strategies adopted to ensure reproducibility and fair comparisons between scenarios.

Chapter 6 presents the results obtained from the simulations and discusses their implications. It analyzes message size distributions, compares protocol-specific overheads, and evaluates the differences between V2V and V2I strategies under normal and interrupted conditions.

Finally, Chapter 7 summarizes the main contributions of the work, acknowledges its limitations, and outlines possible directions for future research. The thesis concludes with reflections on the broader relevance of communication-aware design for cooperative autonomous driving.

2

Theoretical Background

This chapter provides the foundational concepts necessary to understand the broader context and methodology of this thesis. Covers the communication technologies behind cooperative driving, as well as the core control strategies used to manage vehicle movements at intersections. Factors influencing the communication needs in fast-changing traffic environments are also considered.

The purpose of this chapter is not to propose novel algorithms or breakthrough technology, but rather to establish the fundamental concepts upon which the later analysis is built. To start, the principal (V2X) communication frameworks are examined: DSRC, C-V2X, and 5G NR-V2X [4, 7, 9, 17]. Next, the chapter details the primary approaches to intersection management, comparing decentralized (V2V) coordination and centralized (V2I) control models [8]. Finally, crucial communication requirements for effective cooperative driving are outlined, focusing on message sizes and the importance of scalability in real-world deployments. Altogether, this chapter sets the stage for deeper analysis in subsequent sections by establishing the relevant technical groundwork.

2.1 V2X Communication Technologies

(V2X) communication refers to the set of technologies that allow vehicles to exchange information with their surroundings in real time. These technologies are the basis for cooperative and automated driving systems [9]. While onboard sensors allow vehicles to perceive their immediate surroundings, V2X enhances situational awareness by facilitating data sharing with both other vehicles and transportation infrastructure [8].

Two primary interaction types are typically considered. (i) V2V communication is used to share position, speed, acceleration, trajectory, and intent between vehicles [7, 9]. The purpose is to allow vehicles to anticipate one another's actions and coordinate their decisions accordingly. Such exchanges are especially critical for collision avoidance maneuvers, safe lane merging, and the negotiation of right-of-way at intersections. (ii) V2I communication, on the contrary, allows information exchange between vehicles and Roadside Unit (RSU)s or intersection managers [8, 13]. This form of communication supports centralized optimization tasks such as adaptive traffic signal control, congestion management, and dynamic assignment of right-of-way, ultimately improving overall efficiency and safety.

These functionalities are supported by several specialized wireless technologies. During the past decade, the advancement and deployment of standards such as DSRC, C-V2X, and most recently 5G NR-V2X have been essential for the realization of low latency and robust vehicular networks [4, 7, 17].

2.1.1 Dedicated Short-Range Communications (DSRC)

DSRC, built on the IEEE 802.11p standard and operating in the 5.9 GHz spectrum, represents the original backbone of vehicular wireless communication systems [9]. This technology enables direct, low-latency exchanges between vehicles and roadside infrastructure without reliance on cellular networks, typically supporting ranges of 300–1,000 meters and latencies under 100 ms [9]. A central component of DSRC is the Basic Safety Message (BSM) periodic status updates that broadcast a vehicle's position, speed, heading and other critical data, typically ten times per second (10 Hz) [7]. These transmissions allow surrounding vehicles to anticipate each other's movements and avoid collisions.

DSRC is valued for its reliably low latency, its independence from cellular infrastructure, and its mature standardization, with clearly defined message formats and interoperability guidelines [7, 9]. However, DSRC also faces challenges. Its performance can be degraded in dense traffic, as simultaneous transmissions from many vehicles cause channel contention [8]. Deployment has been fragmented across regions (e.g., U.S., EU, China), and industry momentum has increasingly shifted toward cellular-based alternatives [4]. Despite these limitations, DSRC remains an important reference technology in research, having established the foundation for cooperative awareness messages and application-layer protocols used in later standards [7].

2.1.2 Cellular Vehicle-to-Everything (C-V2X)

C-V2X was launched in 3GPP Release 14 as an evolution of DSRC designed to leverage the cellular ecosystem [17]. It operates through two complementary mechanisms: the Proximity Communication Interface 5 (PC5) sidelink, which enables direct communication between vehicles, and the Cellular Uplink/Downlink Interface (Uu), which enables communication through cellular infrastructure. Compared to DSRC, the PC5 interface provides enhanced reliability and range, particularly under high mobility and in urban interference conditions [4, 15]. The Uu interface extends communication to the broader network and cloud services, benefiting from cellular coverage and integration with advanced applications [17].

Thus, C-V2X offers improved range and reliability, as well as a migration path toward 5G by integrating with existing cellular networks; these qualities have encouraged strong support from both the automotive and telecommunication sectors [17, 18]. At the same time, C-V2X introduces challenges. In Uu-based communication, it depends on cellular infrastructure, which may not always be available or consistent, and maintaining connections with the cellular core can increase signaling overhead [4, 17]. Limited backward compatibility with DSRC also complicates co-existence during transitions. However, empirical studies consistently show that C-V2X outperforms DSRC in packet delivery ratio, communication range, and robustness in high-density scenarios, positioning it as a strong candidate for large-scale deployment [4, 15].

2.1.3 5G New Radio V2X (5G NR-V2X)

With 3rd Generation Partnership Project (3GPP) Release 16 and beyond, 5G New Radio V2X was introduced as the next generation of vehicular communication [17]. Building on C-V2X, it incorporates advanced features of the 5G architecture, particularly Ultra-Reliable Low-Latency Communication (URLLC), with latency targets down to the order of 1 ms in ideal conditions [4]. This capability is essential for rapid response scenarios, such as emergency braking, collision avoidance, or collective decision-making among vehicles.

Another significant feature of 5G NR-V2X is its substantially increased throughput, allowing the exchange of richer data, such as raw sensor feeds, video streams and cooperative perception messages [4]. Managing communication on crowded roadways is also a strength of 5G NR-V2X: By dynamically allocating radio resources in real time, it achieves efficient use of the spectrum even when many vehicles transmit simultaneously [15]. Coupled with advanced error correction and redundancy mechanisms, the system provides the reliability required for safety-critical applications such as synchronized platooning or coordinated intersection management [15, 17].

These capabilities make 5G NR-V2X suitable for advanced cooperative driving scenarios, including cooperative perception (sharing sensor data to build a collective view of the environment), tight platooning with precise synchronization and intersection-scale coordination where multiple vehicles must negotiate complex trajectories simultaneously [4, 17]. Despite its potential, deployment faces obstacles: dense infrastructure is required to guarantee coverage, early-phase costs are high, and some features are still being finalized within ongoing 3GPP standardization efforts [17].

In summary, these three technologies represent successive stages in the evolution of vehicular communication. DSRC established the foundation with low-latency direct exchanges [7], C-V2X improved reliability and introduced cellular integration [4, 15], and 5G NR-V2X added the advanced features required for the most demanding cooperative driving applications [17]. Together, they outline the technological path to scalable and reliable V2X communication for autonomous transportation systems.

2.2 Intersection Management Strategies

Intersections are one of the most critical and risk-prone points in the modern transportation system. Multiple streams of traffic, including vehicles, cyclists, and pedestrians, all merge at these junctures, often resulting in complex encounters and a higher risk of accidents. While human drivers rely on nonverbal cues and social negotiation (for example, making eye contact or gesturing), autonomous vehicles must operate without access to such implicit forms of communication. Instead, they rely on explicit signaling and the use of sophisticated algorithms to determine the right-of-way and ensure safety. During the past two decades, academic and industry researchers have proposed a variety of automated intersection management strategies. These approaches can be categorized into two types: decentralized (V2V) control and centralized (V2I) control [8].

Decentralized Control (V2V) : In decentralized schemes, each vehicle operates as an independent agent that communicates directly with nearby cars. Vehicles periodically broadcast their current state, including position, velocity, acceleration, and intended route, and listen to messages from others. Based on this shared local information, each vehicle decides whether to proceed, yield, or adjust its route. In this model, road infrastructure is not required. This makes it more suitable for unsignalized intersections or areas where it is not possible to deploy an intersection manager. Dresner and Stone (2008) proposed a reservation-based intersection protocol in which vehicles negotiate access slots directly [6], while Colombo and Del Vecchio (2014) explored hybrid automata models for decentralized vehicle interactions [5]. More recent approaches based on consensus allow groups of vehicles to reach distributed agreements on crossing order. The main advantage of V2V systems is their robustness to infrastructure failures and their scalability in distributed settings. However, in situations such as dense traffic or when communication delays occur, it is difficult to ensure safety because vehicles can interpret the same situation differently.

Centralized Control (V2I) : Centralized strategies, on the other hand, assign the coordination task to a road Intersection Manager (IM). The intersection manager collects state information from all approaching vehicles. The IM then calculates an optimized route plan, assigns priorities, and sends explicit commands back to the vehicles. This centralized model allows for global optimization of intersection flow, reducing potential conflicts, and improving throughput. Studies such as Levin and Boyles (2016) have shown that centralized algorithms can reduce delays and improve efficiency compared to decentralized approaches [13]. The downside of this method is that centralized systems depend on reliable infrastructure and communication links. They also require a heavier communication load because every vehicle must exchange periodic status updates with the IM, and any failure of the IM can compromise the entire coordination process.

Comparative Insights : In practice, the V2V and V2I strategies represent two ends of a spectrum. Decentralized V2V systems reduce infrastructure dependency by allowing direct communication among vehicles. They may also be more efficient in areas with unreliable connectivity. However, when the density of participating vehicles increases, they face challenges such as maintaining consistent safety and equitable access, which becomes increasingly complex. On the other hand, centralized V2I approaches enable coordinated, deterministic scheduling but come at the cost of higher communication requirements and infrastructure dependency. Many researchers suggest that intersection management in the future may involve hybrid approaches, where a combination of V2V and V2I is used; for example, in normal conditions vehicles negotiate locally but rely on IM when traffic becomes too dense or when global optimization is required [8]. Although this thesis focuses specifically on pure V2V and V2I schemes, understanding their differences and trade-offs provides the foundation for analyzing the communication requirements they impose.

2.3 Communication Requirements in Cooperative Driving

For cooperative driving systems to function effectively, the communication infrastructure that supports them must meet a set of strict standards. Autonomous vehicles rely on continuous information exchange to anticipate the behavior of surrounding vehicles and to coordinate maneuvers in real time. For an efficient and reliable control algorithm that leads to safe situations, several requirements must be met.

One fundamental requirement is **low latency**. Messages must be delivered fast enough to support critical decisions such as accident avoidance or emergency braking. Studies in V2X communication highlight that delays even on the order of hundreds of milliseconds can compromise safety in dense traffic or at intersections, where vehicles often only have seconds to react [7].

Maintaining **consistent information** among vehicles is also crucial. Vehicles must have accurate and up-to-date knowledge of their environment. If some vehicles operate with delayed or incomplete data, the risk of conflicting decisions increases. This issue is particularly pronounced in decentralized systems, where there is no central controller to guarantee synchronization or oversight.

The importance of **scalability** cannot be denied, especially in urban intersections where there are not only vehicles but also pedestrians and cyclists. When all of these different entities attempt to communicate at the same time, the network must continue to operate reliably without crashing due to data overload. Technologies such as C-V2X and 5G NR-V2X have been designed with scalability in mind. They are specifically engineered to efficiently handle increasing communication loads, outperforming older standards like DSRC, particularly when it comes to managing heavy traffic and maintaining reliable connectivity [4, 15].

Finally, **efficiency** in communication is crucial to prevent unnecessary bandwidth consumption. The packets should contain enough information for decision making, but remain compact to avoid overhead. The balance between message richness and size is delicate: larger packets provide more context but increase transmission delays and vulnerability to interference, whereas smaller packets reduce overhead but may omit critical data.

Among these requirements, **packet size** emerges as a particularly relevant parameter to evaluate the feasibility of cooperative driving. Parameters such as latency and coverage are typically managed at the physical and network layers, but the packet size directly impacts the volume of data exchanged for coordination among vehicles. Measurement and comparison of packet sizes under different control models and protocols provides insight into bandwidth utilization and scalability of the V2V and V2I approaches, especially under abnormal traffic conditions.

2.4 Challenges in Measuring Communication Load

Although the requirements for cooperative driving are well recognized, accurately measuring the communication load generated by these systems is far from straightforward. One of the main problems is that many existing studies rely on overly idealized simulation environments [8, 12, 14]. These environments frequently presuppose flawless communication links, without taking into account critical real-world phenomena such as packet loss, channel contention, or interference from nearby technologies [7, 10]. Consequently, the resulting metrics on communication load tend to be unrealistic and overly optimistic, often failing to account for the overhead encountered in practical scenarios.

Another limitation is that evaluations are frequently performed under stable or average traffic conditions. However, real-world environments are not so predictable. Pedestrians can suddenly enter intersections, vehicles can stall and block lanes, or accidents can disrupt normal traffic flow [24]. Each of these events forces vehicles to re-negotiate priorities, adjust trajectories, and exchange additional messages. The result is a sharp increase in communication demand, which is difficult to capture in models that assume a uniform traffic flow. Without explicitly simulating such disruptions, spikes in network load that occur under stress remain hidden [18].

Adding further complexity, there is considerable diversity in the underlying communication protocols. Technologies such as DSRC, C-V2X, and 5G NR-V2X use different mechanisms for message formatting, scheduling, and error handling [1, 2, 3]. A decentralized system that performs well with DSRC may exhibit very different behavior when mapped to C-V2X or 5G NR-V2X, simply due to protocol-specific overhead. Therefore, making fair comparisons requires a framework that normalizes application-level content while accounting for the extra bytes added by each standard [7, 21].

Scalability further complicates the picture. As the number of participating vehicles grows, communication patterns increase non-linearly. In decentralized V2V systems, every new vehicle potentially broadcasts to all others, exponentially increasing the network load [9]. In centralized V2I systems, more vehicles mean more frequent updates to the intersection manager, which can overload the infrastructure and increase processing delays [13]. Capturing these scaling effects requires carefully designed experiments that extend beyond small pilot studies and consider dense, realistic traffic flows [5].

Finally, aligning simulation studies with real-world feasibility remains a challenge. Large-scale projects such as the European MoVeOver initiative have highlighted the importance of bridging this gap. The project deliverables stress that cooperative driving cannot be evaluated solely on the basis of algorithmic efficiency; communication overhead must be factored in to determine whether a given strategy is truly scalable [16, 17, 18]. However, modeling communication in realistic detail is resource-intensive, as it requires combining microscopic traffic simulators with network simulators or protocol-level abstractions [14, 23]. This added complexity explains why many studies still rely on simplified assumptions.

Together, these challenges underscore why communication load analysis remains a complex and underexplored aspect of cooperative driving research. Addressing them requires integrated simulation frameworks that combine realistic traffic behavior, protocol-specific modeling, and disruption scenarios to capture the true demands of V2V and V2I coordination under real-world conditions.

2.5 Modeling Unexpected Events

Real-world traffic environments are rarely stable. Even when the infrastructure functions correctly and vehicles follow standard traffic rules, unexpected events frequently occur. These disruptions, though temporary, can significantly influence traffic flow and the associated communication requirements of cooperative driving systems. For this reason, modeling such events within simulations is essential to produce realistic and meaningful results [17, 18].

One common type of disruption is the **full blockage**. This occurs when the entire intersection or a critical part of it becomes unavailable for a period of time. A typical example is a pedestrian crossing an unsignalized intersection that temporarily forces vehicles to stop and renegotiate their trajectories. In such cases, vehicles must transmit additional messages to update their neighbors or request new crossing slots. These bursts of communication increase the data load and may stress the underlying network more than steady traffic[16].

Another frequent scenario is the **partial blockage**, where only part of the road is blocked. Lane-level disruptions can occur due to stalled vehicles, minor accidents, or congestion that reduces available capacity. Unlike full blockages, partial disruptions often require vehicles to reroute dynamically while still allowing some traffic flow to continue. This situation creates continuous negotiation among vehicles and, in centralized approaches, increases the workload of the intersection manager. As a result, both the size and frequency of communication messages tend to increase [17].

Modeling these unexpected events in simulations is not trivial. Many existing studies still evaluate algorithms under smooth traffic conditions, which neglects the spikes in communication that occur during disruptions. Incorporating full and partial blockages into simulation frameworks makes it possible to stress-test cooperative algorithms under more realistic conditions. This is particularly relevant for comparing decentralized V2V and centralized V2I approaches, as their behavior under stress may diverge significantly.

For example, in V2V systems, each new disruption triggers a wave of direct vehicle-to-vehicle messages as agents attempt to resolve conflicts locally. In V2I systems, disruptions can overload the intersection manager, which must process an increased number of state updates and compute new passage schedules. By systematically modeling these scenarios, an approach also emphasized in the MoVeOver project deliverables, researchers can quantify not only average communication loads but also peak demands that are most likely to strain the network [16, 18].

2.6 Motivation for Packet Size Analysis

Among the different factors that influence the feasibility of cooperative driving, the size of the packet is the most critical factor, although sometimes underestimated [8]. Every message exchanged between vehicles or between vehicles and infrastructure must carry a certain amount of information. The size of these packets plays a direct role in determining the overall bandwidth usage within the system. Unlike latency or reliability, which are often influenced by the physical and network layers [1, 7], the packet size connects directly to how control algorithms are designed and the amount of information they require for effective operation. In essence, increasing the demand for richer data causes the pressure on system bandwidth to increase [10, 11].

For example, a decentralized V2V scheme may require vehicles to broadcast detailed trajectory predictions to avoid conflicts, resulting in relatively large packets [5, 6]. In contrast, a centralized V2I system may rely on more frequent but smaller status updates, shifting communication demand from packet size to message frequency [13]. However, in both cases, the size of the packet determines the minimum communication load that must be supported by the network [4].

The importance of packet size becomes even more evident when unexpected events occur. As highlighted in the MoVeOver deliverables, full blockages caused by pedestrians or partial blockages caused by congestion trigger additional coordination messages beyond those used under normal conditions [16, 17, 18]. These bursts of communication are not optional; they are necessary for vehicles to safely renegotiate priorities. In response, larger or more frequent packets can create temporary surges in bandwidth demand, which can overwhelm the capabilities of certain vehicular communication technologies, particularly in dense traffic situations [15].

Moreover, comparing packet sizes between DSRC, C-V2X, and 5G NR-V2X offers valuable information on how each standard manages identical cooperative tasks at the protocol level [2, 3, 7, 21]. Each technology introduces its own overhead, so a single application-layer message might result in varying total packet sizes depending on the protocol stack used. Without quantifying these differences, it is impossible to make a fair assessment of which technology is better suited to support a given control strategy.

For these reasons, analyzing the size of the packet is not a minor detail but a central step in understanding the scalability of cooperative driving systems. By systematically measuring packet sizes under both normal and disrupted traffic conditions, this thesis aims to provide insights into the real communication requirements of decentralized V2V and centralized V2I control, and to evaluate how well existing V2X technologies can support them.

2.7 Summary

This chapter introduced the theoretical background necessary to analyze the communication requirements of cooperative intersection management. It began with an overview of V2X communication technologies, including DSRC, C-V2X, and 5G NR-V2X, outlining their roles in enabling low latency and reliable data exchange. Following this, the two primary coordination paradigms for automated intersections were discussed: decentralized (V2V) schemes, which rely on direct exchanges among vehicles, and centralized (V2I) schemes, which depend on intersection managers to optimize traffic flow.

The chapter then described the key communication requirements that cooperative driving systems must satisfy, including low latency, consistency, scalability, efficiency, and the careful balance of packet size. It also highlighted several challenges in measuring communication load, including the reliance on idealized simulation models, the diversity of communication protocols, and the difficulties of capturing scalability effects. To address these limitations, the importance of incorporating unexpected events, such as full and partial blockages, into simulation studies was emphasized, since disruptions significantly influence communication patterns.

Among the identified requirements, packet size is recognized as a critical factor for the viability and scalability of cooperative driving systems. Unlike latency or coverage, which are typically addressed at lower communication layers, packet size is more closely linked to the architectural design of control algorithms and has a direct impact on bandwidth consumption. Consequently, given its central role in optimizing system performance, the packet size is the analytical focus of this thesis.

By consolidating these theoretical foundations, this chapter prepares the basis for the methodology that follows. The next chapter details the simulation framework, control models, and experimental setup used to systematically measure the communication load under both normal and disrupted traffic conditions.

3

Literature Review

3.1 V2X Communication in Autonomous Traffic Systems

A considerable amount of research has explored the role of V2X communication in enabling autonomous vehicle coordination, particularly at intersections [8, 9, 19]. Within this domain, two main paradigms can be distinguished: decentralized V2V approaches and centralized V2I approaches.

Decentralized control strategies treat each vehicle as an autonomous agent that exchanges information directly with nearby vehicles. Dresner and Stone [6] introduced one of the earliest reservation-based intersection models, where vehicles negotiate access slots to avoid conflicts. Colombo and Del Vecchio [5] later extended this work by applying hybrid automata models, which formalized decentralized interaction rules without relying on intersection managers. These models highlight the potential of V2V systems to operate robustly even without roadside infrastructure. However, they often assume ideal communication environments, overlooking issues such as message loss, channel congestion, or the actual size and frequency of transmitted packets [4, 7].

In contrast, centralized V2I strategies rely on an external intersection manager that collects state information from all approaching vehicles and computes optimized passage plans. Levin and Boyles [13] demonstrated that such centralized scheduling can significantly improve throughput and reduce waiting times compared to decentralized methods. However, these systems introduce higher communication loads because every vehicle must exchange frequent updates with the manager and are highly dependent on the reliability of the infrastructure [8]. Most evaluations of V2I control focus on safety and efficiency metrics, often neglecting the volume of data required to maintain coordination in practice [17].

Together, previous work has shown that both V2V and V2I approaches have clear benefits but also important limitations [8]. What remains underexplored is the detailed analysis of their communication overhead, particularly at the packet level [16, 18].

3.2 Communication Protocol Studies

Research on vehicular communication has also examined the underlying protocols that support V2V and V2I exchanges. The most studied technology has been (DSRC), based on IEEE 802.11p. DSRC supports direct broadcasting of Basic Safety Messages (BSMs), which typically include attributes such as vehicle ID, position, speed, and direction, transmitted at a frequency of around 10 Hz [7]. While DSRC offers low-latency communication, scalability concerns have been widely observed in dense urban environments, where simultaneous transmissions cause channel contention [9].

The limitations of DSRC motivated the development of Cellular V2X (C-V2X) in 3GPP Release 14. This technology introduced both direct sidelink communication (PC5) and infrastructure-based communication (Uu). Studies such as Molinaro et al. [15] and Campolo et al. [4] show that C-V2X provides superior reliability and range compared to DSRC, particularly in scenarios with high traffic density or interference. More recently, 5G New Radio V2X (5G NR-V2X) has been standardized in Release 16, offering ultra-reliable low-latency communication (URLLC) and higher throughput capabilities. These features make it suitable for advanced cooperative applications, including cooperative perception and platooning. However, most evaluations remain at the simulation level, with limited consideration of the packet size implications of application-layer messages.

Overall, protocol studies provide important insights into performance at the physical and network layers but rarely connect these findings to the communication demands of specific cooperative intersection algorithms.

3.3 Traffic Simulation with SUMO

Simulation tools play a key role in evaluating intelligent transportation systems. The SUMO platform is one of the most widely used microscopic traffic simulators, providing fine-grained modeling of vehicle interactions. SUMO supports integration with external control logic through the TraCI, allowing researchers to implement real-time decision-making and V2X communication processes.

Several studies have demonstrated the adaptability of SUMO when combined with external tools for traffic research. Krajzewicz et al. [12] introduced standard scenarios for traffic signal optimization, establishing a reference for later evaluations. Beyond traffic control rules, reinforcement learning techniques have also been applied to intersection management within SUMO, as summarized in recent surveys on cooperative control [8]. The simulator has further been coupled with communication frameworks to capture the interaction between traffic flow and wireless networking. Examples include its integration with OMNeT++ [22] and the TraCI interface [23], which enable joint evaluation of traffic dynamics and network behavior. These extensions highlight the adaptability of SUMO across different domains. However, most of these works prioritize mobility or protocol performance and rarely quantify the communication requirements of cooperative driving. In particular, message sizes and bandwidth consumption are often overlooked, and the effects of disruptions such as pedestrian crossings or lane blockages are generally not considered.

This gap is particularly relevant for the present work, as SUMO's modular design makes it possible to measure communication load directly while simulating realistic traffic flows.

3.4 Gap in the Literature

The existing body of work can be broadly divided into two groups. On one side, algorithm-centric studies investigate how vehicles can coordinate their behavior through decentralized or centralized strategies, usually focusing on safety, throughput, or delay metrics. These works, however, tend to abstract communication away, assuming perfect message delivery with negligible cost [5, 6, 13]. On the other side, communication-centric studies evaluate protocols such as DSRC, C-V2X, and 5G NR-V2X in terms of latency, reliability, and coverage, but often using synthetic traffic models or fixed message rates that do not reflect the variability of real traffic coordination [4, 7, 9, 15, 19].

Only a limited number of studies have examined the communication overhead of cooperative intersection management in detail, and even fewer studies provide explicit measurements of packet sizes. The impact of unexpected disruptions, such as pedestrian crossings at unsignalized intersections or lane blockages caused by accidents has also received little attention. Recent large-scale initiatives, including the MoVeOver project, have stressed the need to take into account these events when evaluating cooperative driving strategies [16, 17, 18]. However, there is still a lack of systematic evaluations that quantify packet-level behavior under such conditions.

3.5 Positioning of This Thesis

This thesis addresses the identified gap by explicitly focusing on packet size as a key metric to evaluate the scalability of cooperative intersection management. The work implements both decentralized V2V and centralized V2I control algorithms in SUMO, ensuring a consistent message structure between approaches to allow fair comparison. Packet sizes are calculated with protocol-specific overheads for DSRC, C-V2X, and 5G NR-V2X.

Furthermore, simulations incorporate realistic disruption scenarios, including full blockages caused by pedestrians and partial blockages due to congestion or accidents. By measuring and analyzing packet sizes under both normal and disrupted traffic conditions, this research provides new information on the communication efficiency of cooperative strategies. In doing so, it contributes to bridging the gap between algorithm design and communication feasibility in autonomous traffic systems.

4

System Design and Implementation

4.1 Overview of Simulation Environment

The system developed in this thesis is built on top of the SUMO [14], an open source microscopic traffic simulator that provides fine-grained control over vehicle behavior and road interactions. SUMO is particularly well suited for research on cooperative driving systems because it allows the modeling of individual vehicles with precise spatial and temporal resolution. Within this simulator, users can configure parameters such as speed, acceleration, and route for each vehicle. Furthermore, complex scenarios such as intersection interactions and coordinated maneuvers can be explored under tightly controlled conditions, making SUMO an optimal tool to advance studies in the field of intelligent transportation systems[12].

To enable dynamic control and communication between vehicles, SUMO is coupled with Python scripts through the TraCI [23]. TraCI is a socket-based Application Programming Interface (API) that allows external programs to interact with the simulation while it is running. Through this interface, vehicles can be queried for their current state (position, speed, direction, and route) and issued new commands (such as braking, stopping, or rerouting). This integration transforms SUMO from a static traffic simulator into a flexible experimentation environment where cooperative control algorithms can be tested in real time.

In the context of this thesis, TraCI serves as the backbone for implementing the two coordination schemes under investigation: Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). For the V2V case, Python modules enable vehicles to broadcast their state and make local decisions based on received information from peers. For the V2I case, vehicle states are instead transmitted to a centralized intersection manager, which computes priorities and issues commands back to vehicles.

Another important feature of the environment is the ability to inject dynamic disruptions. Using TraCI, it is possible to introduce events such as full blockages (e.g., pedestrian crossings) or partial blockages (e.g., lane reductions or congestion). These disruptions force vehicles to re-coordinate, thereby generating additional communication load. Such event injection makes it possible to evaluate how control algorithms and communication protocols scale under stress, rather than only under idealized conditions.

Finally, the system includes a logging and analysis framework. Every message exchanged–whether between vehicles in V2V or between vehicles and the intersection manager in V2I–is recorded, capturing details such as message size and precise timestamps. These detailed logs serve as the foundation for subsequent analyses, supporting the calculation of key communication metrics, including packet size distribution, message frequency, and aggregate communication load observed in various operational scenarios.

In summary, the simulation environment integrates SUMO, TraCI, and a set of Python modules into a cohesive platform. This infrastructure is able to model nuanced vehicle behaviors, implement advanced cooperative control strategies, simulate disruption events, and collect granular communication data. As such, it provides a robust foundation upon which the system's architecture and experimental methodology are developed.

4.2 System Architecture

The system architecture developed for this thesis integrates traffic simulation, control logic, communication modeling, and data logging into a unified framework. Its design reflects the need to simulate both decentralized and centralized coordination strategies while preserving adaptability for experimenting with various communication protocols and introducing disruption scenarios. The architecture offers the flexibility needed for comprehensive analysis in dynamic traffic environments.

At the highest level, the system consists of five core components:

1. **SUMO Simulator**
   SUMO provides the microscopic traffic environment, where vehicles are instantiated, moved along their routes, and interact at intersections. It handles the physical dynamics of vehicle motion and ensures realistic traffic flow.

2. **TraCI Interface**
   The Traffic Control Interface acts as the bridge between SUMO and the external Python control layer. It enables two-way communication: Python scripts can extract vehicle states from the simulator and issue commands that influence their behavior in real time.

3. **Python Control Layer**
   This layer implements the decision-making logic for the two control models under investigation:

   - **V2V control module**: Vehicles exchange messages directly and make local right-of-way decisions.
   - **V2I control module**: Vehicles report to a centralized intersection manager, which computes and distributes crossing instructions.

   Each module is implemented as a separate set of scripts, but follows a common structure to ensure fair comparison.

4. **Event Injector**
   To simulate realistic disruptions, an event injection mechanism is included. It introduces disturbances such as full blockages (e.g., pedestrian crossings) or partial blockages (e.g., lane closures or congestion). These events trigger additional coordination efforts, which affects the communication load.

5. **Logger and Analysis Tools**
   All communication events are systematically logged, capturing key parameters such as message size, sender and receiver identities, timestamps, and protocol-specific overhead. This comprehensive dataset is subsequently exported for detailed analysis, serving as the primary empirical foundation for the quantitative evaluation presented in Chapter 6.

The system architecture follows a modular paradigm that allows for independent modification or extension of individual components without substantial disruption to other subsystems. For example, a consistent traffic scenario can be used to evaluate both V2V and V2I coordination trials simply by substituting relevant control modules. Additionally, the architecture accommodates the integration of new communication protocols via adjustments to the message abstraction layer, leaving the core traffic logic unaffected.

As depicted in the high-level system diagram (Figure 4.1), the architecture supports continuous state updates from simulated vehicles in SUMO, transmitted via TraCI into the Python-based control layer. Depending on which coordination module—V2V or V2I—is active, messages are exchanged either directly between vehicles or through the intersection manager. The logger records every communication event, while an event injector introduces controlled disturbances to the traffic scenario, thereby increasing communication demands.

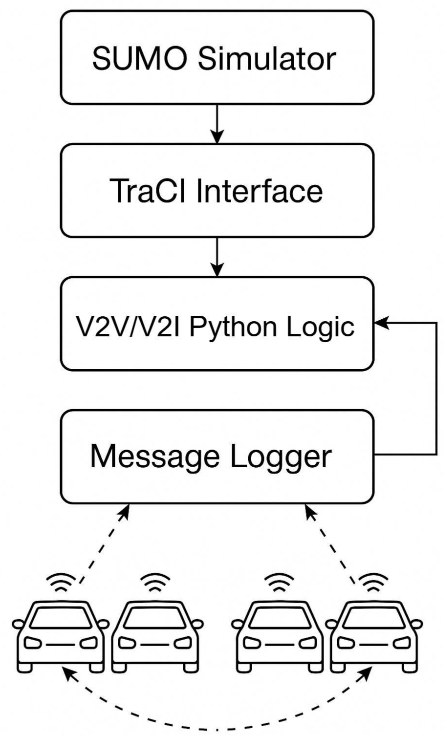


Figure 4.1: Architecture of the V2V/V2I Simulation Framework

This layered, modular approach ensures both realism and adaptability in simulation, affording researchers the ability to conduct controlled experiments. The design supports the isolation and measurement of the impact that control strategies, communication protocols, and disruptive events have on the overall communication load.

4.3 Message Packet Format and Protocol Abstraction

A central aspect of this system is the way in which communication between vehicles—and between vehicles and infrastructure —is represented. Since SUMO itself does not natively simulate wireless communication, a *message abstraction layer* was developed to model the data exchanged in cooperative driving. This layer defines the structure of the packet, serializes the content into byte streams, and applies protocol-specific overheads to approximate realistic packet sizes.

At the application level, each communication event is encapsulated in a *message packet* object. This structure contains the essential information vehicles require to coordinate at intersections:

- **vehicle_id**: a unique identifier for the transmitting vehicle.
- **position**: the current Cartesian coordinates (x, y).
- **speed**: the vehicle's instantaneous velocity in meters per second.
- **heading (angle)**: the current orientation of the vehicle in degrees.
- **route**: a list of upcoming road segments (edge IDs) that define the intended path of the vehicle.

The `MessagePacket` is serialized into a JavaScript Object Notation (JSON)-formatted string and then encoded into 8-bit Unicode Transformation Format (UTF-8) bytes. The raw payload size is measured directly in Python, ensuring precise control over how much data is exchanged during each simulation step.

Protocol Overhead Definition

To enable a consistent and reproducible comparison across DSRC, C-V2X, and 5G NR-V2X, we explicitly define the overhead of the protocol as the sum of link/Medium Access Control (MAC) headers, control/scheduling information, and security fields specified in the corresponding standards and technical reports. Table 4.1 reports the *minimum* and *maximum* overhead assumed for each protocol based on the cited sources.

Given a payload of size P (bytes) and a protocol-dependent overhead of size H (bytes), the *overhead percentage* reported later in the results (e.g., Table 6.1) is

Protocol	Overhead Components (sources)	Min (B)	Max (B)
DSRC	IEEE 802.11p MAC header 30–40 B [10]; LLC/SNAP 8 B; IEEE 1609.2 security 60–100 B [11]	98	148
C-V2X	Sidelink Control Info (SCI) 16–24 B [1]; MAC/PDCP/RLC 20–40 B [21, 1]; Security 48 B [21]	84	112
5G NR-V2X	Scheduling/control info 24–40 B [2, 3]; NR MAC/PDCP stack 48 B [2]; Security + QoS 72 B [20]	144	160

Table 4.1: Protocol overhead bounds used in packet sizing. Min/Max reflect typical ranges from standards and vendor/tech reports; see citations.

defined as:

$$\text{Overhead\%} = \frac{H}{P + H} \times 100.$$

This definition makes protocol comparisons payload-aware and aligns the simulated packet construction with the documented header/security fields for each technology.

By combining the payload and the overhead of the protocol, the system estimates the *packet size in the air* that would be transmitted under each communication standard. This abstraction allows the same control logic to be tested across multiple technologies without requiring low-level network simulation, while still providing meaningful comparisons of communication load.

The uniform message format ensures fairness between the V2V and V2I implementations. In both cases, the same set of fields is transmitted; only the communication direction differs, peer-to-peer between vehicles in V2V, or uplink and downlink with an intersection manager in V2I. This consistency guarantees that any observed differences in communication load stem from the coordination strategy itself rather than message design.

In summary, the packet format and protocol abstraction layer serve as the *translation bridge* between cooperative control logic and communication models. It ensures that message sizes are realistic, comparable across standards, and measurable within the simulation framework.

4.4 V2V Control Implementation

The V2V control model implements a fully decentralized coordination strategy, where each vehicle acts as an autonomous agent responsible for making local decisions based on shared information from its neighbors. No central authority exists; instead, vehicles rely on direct message exchanges to determine right-of-way at intersections.

The V2V control logic is implemented as a continuous loop running in the Python control layer. At each simulation step, the following sequence of operations is executed:

State retrieval

Each vehicle obtains its current state variables from SUMO via the TraCI interface. This includes position, speed, direction, and planned route.

Message broadcasting

The vehicle constructs a *MessagePacket* containing its state and broadcasts it to all nearby vehicles within communication range. In the simulation, this is modeled as the delivery of a message to all active vehicles that share the same intersection context.

Message reception

The vehicle listens for and processes incoming *MessagePacket*s from surrounding agents. These packets provide information on the intentions, trajectories, and estimated arrival times of other vehicles at the intersection.

Right-of-way determination

Using the shared information, each vehicle evaluates potential conflicts. Right-of-way is determined based on relative distances and estimated arrival times:

- A vehicle closer to the intersection or with an earlier arrival estimate is typically given priority.
- In case of ties, predefined rules (e.g., yielding to the right) are applied.

Behavior adaptation

When a vehicle does not have the right of way at an intersection, it must adjust its behavior by decelerating, yielding, or adjusting its intended path to avoid potential conflicts. Priority vehicles, in contrast, continue along their planned route unless an unexpected disruption occurs.

Re-coordination under disruptions

If an obstacle occurs (such as a full or partial blockage), all vehicles within the affected vicinity are required to renegotiate their movements. This process generates increased communication activity, as each agent transmits updated intentions and attempts to secure the right of way once again. The result is a significant, if temporary, increase in digital "conversation" among participants.

A fundamental aspect of this decentralized V2V approach is its inherent scalability; there is no reliance on external infrastructure, making it highly deployable across various environments. Yet, this comes at a cost: as the number of vehicles converging on a given intersection rises, so too does the volume of exchanged messages. In dense traffic, this "all-to-all" communication model results in substantial overhead compared to centralized alternatives.

However, observing these distributed negotiation processes highlights the capacity of autonomous vehicles to self-organize using only local awareness and direct peer communication. This illustrates the delicate trade-offs such as simplicity versus communication efficiency, and scalability versus control. These insights are valuable for understanding potential structures of future autonomous traffic management systems.

4.5 V2I Control Implementation

The V2I control model introduces a centralized coordination strategy, in which a dedicated *Intersection Manager* (IM) assumes responsibility for organizing the vehicle passage through the intersection. Unlike the decentralized V2V scheme, where vehicles must negotiate among themselves, the V2I approach reduces local decision making by shifting coordination to an external authority.

The V2I implementation is structured around a continuous exchange of state reports and instructions between vehicles and the IM. At each simulation step, the following sequence of operations is executed:

1. **State reporting**

   Each vehicle periodically collects its state information—position, speed, heading, and intended route—and sends a *MessagePacket* to the IM.

2. **Data aggregation**

   The IM receives these messages from all vehicles within its communication range and stores them in a centralized data structure indexed by vehicle ID. This provides the IM with a global view of the current intersection state.

3. **Priority evaluation**

   Once per simulation step, the IM analyzes the collected data to detect potential conflicts and compute a safe and efficient passage schedule. Priorities are typically determined by comparing estimated arrival times and applying global optimization rules, such as minimizing delay or maximizing throughput.

4. **Command dissemination**

   The IM transmits instructions back to each vehicle. These may include directives to proceed, yield, stop, or reroute according to the computed schedule.

5. **Vehicle adaptation**

   Vehicles receive and execute IM instructions. This may involve adjusting their speed, stopping before the intersection, or rerouting to avoid a blocked path. Vehicles no longer make independent crossing decisions, but instead rely entirely on the commands issued by the IM.

6. **Disruption handling**

   In the event of full or partial blockages, the IM recalculates its global schedule and sends updated commands to affected vehicles. This central control allows for a more coordinated response to disruptions, but also generates increased communication traffic due to frequent updates and instructions.

The V2I control approach effectively acts as a centralized authority, analogous to having a traffic coordinator that oversees the entire intersection environment. In contrast to the decentralized V2V model, where vehicles individually negotiate their maneuvers, V2I enables an IM with comprehensive situational awareness to orchestrate scheduling and movements. This global perspective allows the IM to mitigate conflicts and avoid deadlocks, thereby improving traffic safety and efficiency.

However, these advantages introduce notable trade-offs. Achieving continuous and coordinated operation requires persistent higher communication overhead between each vehicle and the IM, increasing bandwidth consumption considerably. Furthermore, reliance on a singular IM creates a single point of failure: any disruption or latency in this system can lead to decreased reliability and potential bottlenecks at the intersection.

In essence, V2I centralizes coordination and streamlines negotiation among vehicles, promoting smoother intersection management. However, these improvements are based on robust infrastructure and reliable data exchange; without them, the benefits will not be realizable.

4.6 Simulation of Dynamic Disruptions

To evaluate the robustness of both V2V and V2I coordination mechanisms, it is essential to simulate realistic traffic disturbances. In urban mobility, unexpected disruptions, such as pedestrian crossings, temporary lane blockages, and congestion, often occur and directly affect coordination efficiency.

In this work, two representative types of disruption were injected into the simulation environment:

1. **Full blockage:** modeled as a pedestrian crossing event that stops all vehicle flow through the intersection for a fixed duration.
2. **Partial blockage:** modeled as traffic congestion or lane obstruction, reducing the available road capacity without fully closing the intersection.

Both scenarios were implemented in SUMO through the TraCI interface, which enabled dynamic modification of traffic conditions during runtime. These events were strategically introduced mid-simulation to observe adaptive responses from both control models (V2V and V2I).

4.6.1 Full Blockage (Pedestrian Crossing)

In the full blockage scenario, all incoming approaches to the intersection were temporarily blocked to replicate a pedestrian crossing event. Implementation in SUMO was achieved by setting the traffic light phase to red or by directly closing the lanes via TraCI commands.

Observations:

- **V2V:** Vehicles attempted to renegotiate locally. While this avoided collisions, the negotiation overhead increased significantly and led to temporary deadlocks until the blockage was cleared.
- **V2I:** The Intersection Manager recalculated a global schedule immediately after detecting the blockage. Although throughput decreased during the disruption, recovery was faster once the blockage was lifted.

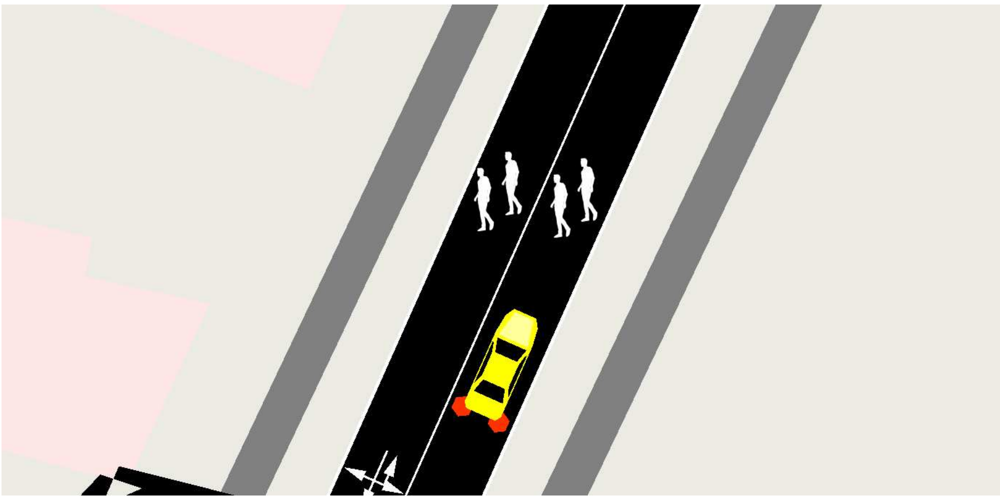


Figure 4.2: Traffic response under full blockage (pedestrian crossing).

- In both models, the number of exchanged packets spiked sharply during the event, reflecting the need for greater coordination. This will be analyzed quantitatively in Section 4.7.

4.6.2 Partial Blockage (Traffic Congestion)

In the partial blockage scenario, one or more lanes approaching the intersection were restricted to simulate congestion, accidents, or temporary obstacles. Vehicles were still able to cross, but at reduced capacity.

Observations:

- **V2V:** Local negotiation enabled vehicles to adjust routes, but this caused uneven traffic distribution and localized queueing. Deadlocks were less frequent than in the full blockage case, but overall efficiency decreased.
- **V2I:** The Intersection Manager dynamically rerouted vehicles and adjusted priorities. While effective in minimizing congestion buildup, this required frequent command updates, generating substantial communication overhead.
- Compared to full blockage, the system maintained partial throughput; however, the persistent congestion amplified communication demands across longer time intervals.

Summary

The simulation of dynamic disruptions demonstrates that both control models experience performance degradation under unexpected events, but in different ways:

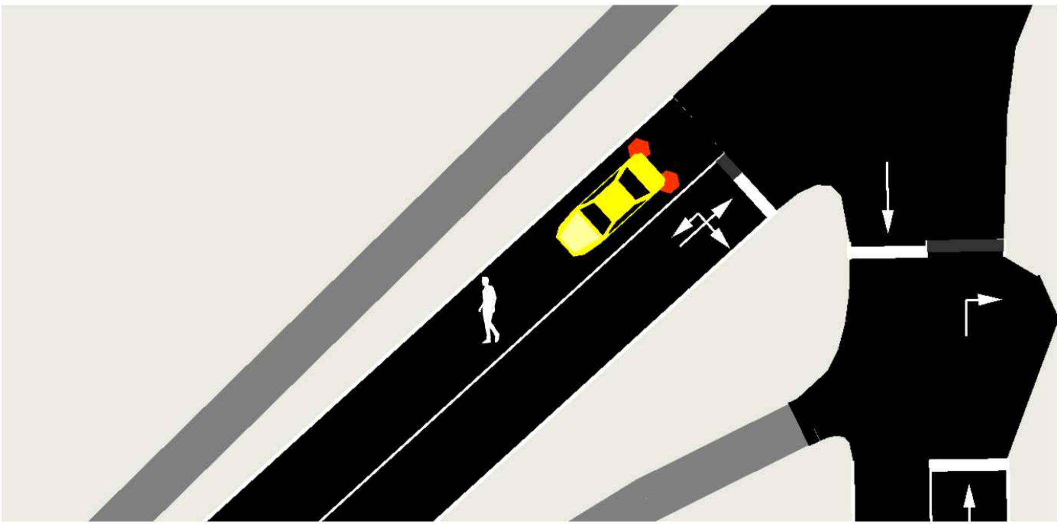


Figure 4.3: Traffic response under partial blockage (congestion scenario).

- V2V favors distributed resilience but risks negotiation deadlocks under full blockages.
- V2I ensures global recovery with fewer conflicts, but at the cost of higher communication load during disruptions.

These findings establish a foundation for the communication analysis presented in Section 4.7, where packet sizes during disruptions are examined quantitatively.

4.7 Packet Size Logging and Analysis

To understand the communication requirements of cooperative intersection management, a logging framework was developed to capture every message exchanged during the simulation. This framework records metadata such as message type, size in bytes, sender and receiver, and timestamps with millisecond resolution. The resulting dataset provides the basis for analyzing communication overhead under both V2V and V2I control schemes.

4.7.1 Logging Framework

The logging system was implemented as a Python module connected to the TraCI interface. Each time a vehicle or the Intersection Manager generated a message, a corresponding entry was appended to the log file. Each log entry contains the following fields:

- **Message ID:** a unique identifier for each transmission.
- **Sender and receiver IDs:** the vehicle or intersection manager involved in the exchange.
- **Timestamp:** precise time of message generation.
- **Message type:** state report, command, or negotiation message.
- **Payload size:** the total number of bytes in the packet.

This structured logging enables complete reconstruction of the communication history for each simulation run.

4.7.2 Analysis Procedure

Once the logs were collected, a post-processing pipeline was applied to compute the following metrics:

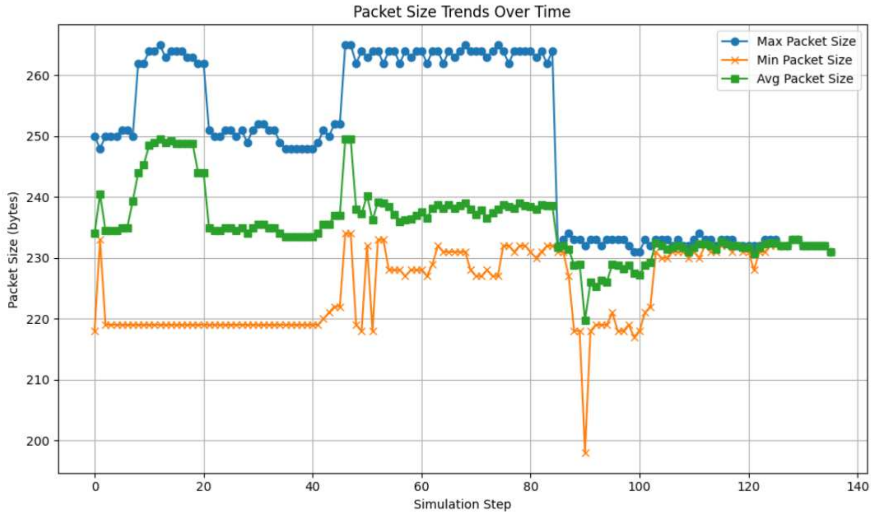
1. **Average packet size** (in bytes).
2. **Packet size distribution** across all messages.
3. **Message frequency:** messages transmitted per vehicle per second.
4. **Total communication load:** aggregate number of bytes exchanged during the run.

These metrics were separately evaluated for the coordination of V2V and V2I, and were further distinguished by traffic conditions: normal flow, full blockage (pedestrian crossing) and partial blockage (congestion).

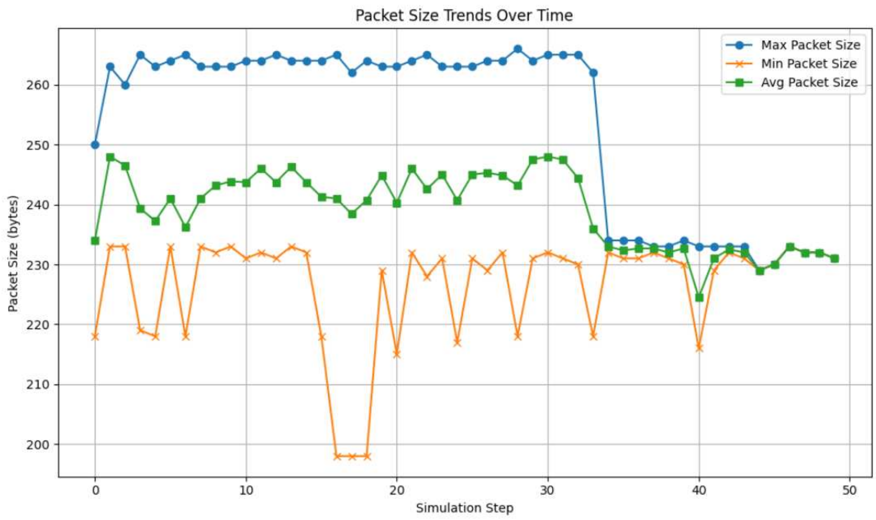
4.7.3 Results Overview

The analysis highlights fundamental differences between V2V and V2I coordination:

- **V2V:** Produced smaller but more frequent packets, since vehicles continuously broadcast state reports to neighbors. The packet size distribution remained narrow, but message frequency grew rapidly with traffic density.
- **V2I:** Generated larger packets, as each state report contained complete vehicle state information. Command messages from the Intersection Manager further contributed to load. Message frequency was lower than in V2V, but the average packet size was substantially higher.
- **Dynamic disruptions:** Both schemes experienced communication spikes. V2V required additional negotiation messages, while V2I issued frequent schedule updates, leading to increased command traffic.



V2V



V2I

Figure 4.4: Example packet size distribution for V2V and V2I control under normal traffic condition.

4.7.4 Implications

The results illustrate the trade-offs between distributed and centralized communication:

- V2V scales poorly in dense traffic due to the high frequency of messages, increasing channel contention.

- V2I requires more bandwidth due to larger packet sizes and frequent updates during disruptions.

These findings underscore the importance of selecting communication protocols that match the expected deployment environment. In particular, bandwidth limitations and latency constraints of DSRC, C-V2X, and 5G NR-V2X technologies must be considered when deploying either scheme.

4.8 Software Structure Summary

The framework is structured around distinct Python modules, each designated for a specific function within the simulation environment. Table 4.2 outlines these core modules alongside their primary responsibilities. This kind of modular design clarifies the roles of V2V and V2I coordination mechanisms, supports reproducibility, and makes it easy to adapt the framework to new control strategies or communication protocols as needed.

File	**Model**	**Functionality**
`vehicle.py`	V2V	Vehicle autonomy and peer-to-peer decision-making
`vehicle_v2i.py`	V2I	Vehicle behavior under central authority
`intersection_manager.py`	V2I	Coordinates passage priorities at intersections
`sumo_run.py`	V2V	Controls V2V simulation and logs messages
`sumo_run_v2i.py`	V2I	Controls V2I simulation and logs messages
`MessagePacket`	Shared	Unified message structure and protocol size estimation

Table 4.2: Summary of Python modules and their responsibilities.

The modular structure separates vehicle logic, intersection management, simulation control, and message abstraction into distinct components. This design not only supports the experiments presented in this thesis, but also facilitates extension toward future cooperative control strategies and protocol studies.

4.9 Conclusion

This chapter presented the design and implementation of the simulation framework used to evaluate the communication requirements of cooperative intersection management. The system integrates SUMO as a microscopic traffic simulator, TraCI for real-time control, and Python modules implementing both decentralized V2V and centralized V2I coordination schemes.

To reflect realistic urban conditions, mechanisms for dynamic disruption injection were developed, including full blockages caused by pedestrian crossings and partial blockages simulating congestion or accidents. In addition, a logging framework was implemented to capture every communication event, enabling detailed analysis of packet sizes, message frequency, and aggregate load.

The modular software structure, summarized in Table 4.2, ensures flexibility, reproducibility, and extensibility of the framework. Together, these components establish a robust testbed for systematically comparing V2V and V2I control under normal and disrupted traffic conditions.

The next chapter builds on this foundation by defining the simulation scenarios, parameter settings, and methodology used to generate the results discussed in Chapter 6.

5

Simulation Scenarios and Methodology

5.1 Simulation Environment

To evaluate the communication efficiency of V2X control strategies, a simulation environment was constructed using the SUMO platform. SUMO is a widely adopted microscopic traffic simulator capable of handling thousands of individually controlled vehicles [14]. It allows researchers to implement realistic traffic flow logic, road geometry, routing strategies, and dynamic events such as lane closures and pedestrian interference. These features make it particularly suitable for evaluating cooperative driving strategies in controlled but realistic conditions.

The SUMO environment is connected to a Python-based control framework through the TraCI [23], which provides real-time access to the simulation. TraCI enables external scripts to query vehicle data such as position, speed, and lane identifiers, and to issue commands that directly influence vehicle behavior. Examples include modifying acceleration, imposing stops, rerouting vehicles, or injecting disruption events. This interactive capability transforms SUMO from a static simulator into a dynamic testbed where V2V and V2I control algorithms can be executed in real time.

For this project, a simplified urban map was generated from OpenStreetMap (OSM) data. The selected area consists of a three-way unsignalized intersection, representative of common urban traffic scenarios where vehicles must negotiate right-of-way without traffic lights. The raw OSM data were processed using SUMO's network generation tools to create road networks, junction geometry, and vehicle routes. This ensured that the experimental environment maintained both realism and reproducibility, providing a consistent basis for the evaluation of communication requirements in later chapters.

5.2 Simulation Parameters

To ensure reproducibility and capture realistic interactions among vehicles, the simulation was configured with a set of controlled parameters. Each experiment was carried out under identical environmental conditions, so that variations in the outcomes could be directly attributed to the coordination strategies and disruption events under investigation. The simulation step length was fixed to provide high temporal resolution, while traffic flows and vehicle insertion rates were chosen to guarantee consistent interaction at the intersection without saturating the network.

Table 5.1 summarizes the core parameters adopted in all simulation runs. The total simulation time was set long enough to observe the effects of both baseline operation and disruption events, while the vehicle generation interval was calibrated to produce moderate but sustained traffic flow. The type of intersection was defined as a three-way unsignalized junction, representative of many urban scenarios. Disruption events—full and partial blockages—were introduced at predefined intervals within the run to evaluate the system's responsiveness.

Parameter	**Value**
Simulation duration	200 seconds
Time resolution (step length)	0.1 seconds
Vehicle generation interval	1 vehicle per 2 seconds
Intersection configuration	Unsignalized 3-way
Vehicle flow volume	50–70 vehicles per simulation
Control models tested	V2V and V2I
Disruption events	Full blockage (steps 20–30) and partial blockage (steps 40–80)

Table 5.1: Key simulation parameters used across all test scenarios

The chosen parameters strike a balance between computational feasibility and realism. They generate sufficient vehicle density to trigger frequent intersection interactions while avoiding extreme congestion that would mask the specific impact of control strategies and communication requirements.

5.3 Control Algorithms Under Test

The experiments evaluate two contrasting approaches to cooperative intersection management: decentralized V2V coordination and centralized V2I management. Implementing both models within the same simulation environment ensures a fair comparison, since the traffic demand, intersection geometry, and disruption events remain identical across runs.

Decentralized Control (V2V) The decentralized control strategy models each vehicle as an autonomous agent that communicates directly with nearby vehicles and makes its own right-of-way decisions. This approach was adapted from time-driven coordination methods proposed in the literature [6, 5], but was re-implemented in a **step-based** form to operate synchronously with SUMO's simulation cycle through the TraCI interface. Each simulation step represents one decision interval, during which vehicles exchange information, check for conflicts, and update their speed or lane according to local rules.

At every step, each vehicle updates its state using TraCI commands (`getPosition`, `getSpeed`, `getAngle`, etc.) and constructs a compact JSON message through the `MessagePacket` class. This packet includes the vehicle's ID, position, speed, angle, and planned route. The payload size of this message forms the basis for evaluating communication requirements. Protocol-specific overheads for DSRC, C-V2X, and 5G NR-V2X are later computed analytically by adding standardized header and security sizes according to the IEEE 802.11p, IEEE 1609.2, 3GPP TS 36.331, and 3GPP TS 38.300 specifications [1, 2, 10, 11].

The main decision loop is implemented in the `communicate()` method of the `Vehicle` class and follows three stages:

1. **Pre-entry intersection check:** Each vehicle determines whether it is approaching or already within the intersection area, defined by bounding coordinates. If the vehicle is outside but detects another vehicle already inside within a predefined safe distance (15 m), it stops to prevent entering an occupied conflict zone.

2. **In-intersection behavior:** Once inside the intersection, vehicles are never stopped by this controller to avoid deadlocks. Vehicles that have entered are expected to clear the intersection uninterrupted.

3. **Rear-end and lane-level safety:** On each edge, vehicles identify neighbors traveling on the same edge within a radius of 20 m.

   - If a full blockage event is active (`handle_unexpected_event("full_block", ...)`), the vehicle performs an immediate stop and waits until cleared.
   - If a lane blockage occurs and the edge has multiple lanes, the vehicle changes to an unblocked lane; otherwise, it treats the case as a full blockage and stops.
   - If no leaders are detected, the vehicle accelerates gradually (by 20%) to regain speed.
   - If a leader is detected within the safe distance, the vehicle slows to 80% of the leader's speed; otherwise, it maintains its current speed.

Both full and partial blockage events are part of the algorithm itself and can be toggled for specific experiments by commenting or uncommenting the corresponding lines in the simulation script. Each vehicle's message payload size (in bytes) is returned by `communicate()` and recorded per simulation step. The main `run_simulation()` loop manages vehicle creation and removal, triggers events (e.g., steps 41–79 for a full blockage), collects per-step packet sizes, and generates statistical plots showing minimum, maximum, and average message sizes. The resulting data are mapped to protocol-specific packet sizes to evaluate the overhead imposed by different V2X technologies under both normal and disrupted conditions.

```
for each simulation step:
    update all vehicle states from SUMO via TraCI
    for each vehicle v:
        detect nearby vehicles within radius R
        if v outside intersection and another vehicle inside within safe_distance:
            stop(v)
        else if v in full_blockage:
            stop(v)
        else if v in lane_blockage:
            if multiple lanes: change_lane(v)
            else: stop(v)
        else:
            if no leader: accelerate(v)
            else if leader too close: slow_down(v)
```

```
            else: maintain_speed(v)
        send MessagePacket(v)
```

Code 5.1: Simplified pseudocode of the V2V control logic

Each simulation step therefore acts as a discrete-time update of the algorithmic behavior, ensuring that all decision-making, communication, and event handling remain synchronized with SUMO's internal time progression.

Centralized Control (V2I) The centralized strategy coordinates vehicles through a single *Intersection Manager* that collects status reports from vehicles and issues `go/wait` decisions each simulation step. As with the V2V scheme, the logic is adapted from time-driven scheduling approaches in the literature [13] but implemented here in a **step-based** form to run synchronously with SUMO via TraCI. Each step acts as a decision interval: vehicles report their state, the manager resolves priorities, and vehicles apply the assigned action.

Vehicle reporting and packet sizing. Every vehicle (`VehicleV2I`) refreshes its state from SUMO using TraCI (`getPosition`, `getSpeed`, `getLanePosition`, etc.) and constructs a compact JSON message through `MessagePacket` (fields: `id`, `position(x,y)`, `speed`, `angle`, `route`). The payload size (bytes) is recorded per step; protocol-specific packet sizes for DSRC, C-V2X, and 5G NR-V2X are then obtained analytically by adding standardized header and security components per IEEE 802.11p, IEEE 1609.2, 3GPP TS 36.331, and 3GPP TS 38.300 specifications [1, 2, 10, 11].

Manager-side decision logic. The `IntersectionManager` maintains a dictionary of recent vehicle reports. At each step, it identifies the subset of *approaching* vehicles as those with a decreasing lane position (i.e., `lane_pos` < `last_lane_pos`) and still within the 30 m approach zone (`lane_pos` < 30). If none are approaching, all vehicles receive `go`. Otherwise, the manager sorts approaching vehicles by `lane_pos` (closest first) and allows only the nearest one to `go`, instructing the rest to `wait`. Vehicles not in the approaching set also receive `go`. This simple first-come, closest-first policy mirrors a reservation-style scheduler in step form and avoids deadlock by ensuring only one conflicting approach gets priority at a time.

Vehicle-side compliance and events. Each vehicle applies the received decision via `apply_decision`: if `wait`, it stops; if `go`, it restores the SUMO controller speed (`setSpeed` to `-1.0`). To prevent mid-intersection stalls, vehicles with `lane_pos` > 30 m (i.e., already past the conflict zone) are never forced to stop. The algorithm also includes disruption handling: a *lane blockage* triggers a lane change when possible; if only one lane exists, it is treated as a full block and the vehicle stops. In experiments, full/partial blockage events are *part of the algorithm* and are activated by uncommenting the relevant lines for the desired scenario.

```
for each simulation step:
    # --- vehicle side ---
    for each vehicle v:
        v.update_state_from_SUMO()
        payload_bytes = build_MessagePacket(v).payload_size()
        send_report_to_manager(v.id, v.state)   # position, speed, lane_pos, last_lane_pos, edge

    # --- manager side ---
    approaching = { v in reports | v.lane_pos < v.last_lane_pos and v.lane_pos < 30 }
    if approaching is empty:
        decisions = { vid: "go" for all vehicles }
    else:
        # closest to the stop line gets priority
        allowed = argmin(approaching, key = lane_pos)
        decisions = {}
        for each vehicle vid:
            if vid not in approaching: decisions[vid] = "go"
            else if vid == allowed:    decisions[vid] = "go"
            else:                      decisions[vid] = "wait"

    # --- broadcast and compliance ---
    for each vehicle v:
        # optional disruptions (toggled per experiment)
        if full_blockage_active_on(v): v.stop()
        else if lane_blockage_active_on(v):
            if multi_lane(v.edge): v.change_lane()
            else: v.stop()
        else:
            v.apply_decision(decisions[v.id])   # wait => stop; go => resume controller speed

    # log payload bytes; map to DSRC / C-V2X / 5G NR-V2X totals via
```

```
    standard overheads
```

Code 5.2: Simplified pseudocode of the centralized V2I control logic

As in the V2V case, the simulation script aggregates per-step payload sizes to produce min/avg/max trends and then maps the payloads to total packet sizes for each V2X technology. This allows a direct comparison of communication load under normal operation and under disruption scenarios in a centralized scheduling regime.

Both models were implemented as custom Python modules, ensuring a consistent structure and comparable message formats. This guarantees that observed differences in communication load stem from coordination logic rather than implementation artifacts.

5.4 Communication Protocol Modeling

The communication system is abstracted using a shared Python class called *MessagePacket*, which encapsulates each exchange in a JSON-formatted object containing the vehicle identifier, position, speed, heading, and intended route. The payload is serialized into UTF-8 bytes, and its size is measured directly. To approximate realistic transmission, protocol-specific overheads are then added.

For DSRC, the overhead is modeled based on IEEE 802.11p and IEEE 1609.2 standards, which together contribute between 98 and 148 bytes depending on security fields [10, 11]. For C-V2X, overhead values are taken from 3GPP Release 14 specifications and technical reports, yielding a range between 84 and 112 bytes [1, 21]. For 5G NR-V2X, scheduling and security fields defined in Release 16 and subsequent specifications contribute 144–160 bytes of overhead [2, 20]. These values were defined in Section 4.6 and are summarized in Table 4.1.

By maintaining a consistent payload structure across protocols, this abstraction ensures that the comparison of communication loads reflects only protocol-specific characteristics, without introducing bias from message content.

5.5 Event Injection: Blockage Scenarios

Realistic urban environments frequently experience disruptions, which can significantly alter communication requirements [8]. To test robustness, the simulation includes two event types introduced at predefined intervals.

Full Blockage (Pedestrian Simulation). Between simulation steps 20 and 30, a complete blockage is imposed on one approach to the intersection, forcing vehicles to stop. This event simulates the presence of pedestrians crossing or other full interruptions of traffic flow. In the V2V case, vehicles detect the blockage through their local state exchange and must re-negotiate right-of-way with neighbors. In the V2I case, the IM identifies the disruption through vehicle reports and recalculates a global crossing schedule. This type of event is critical for testing whether systems can avoid deadlocks and maintain safety when traffic is temporarily interrupted.

Partial Blockage (Congestion or Accident). Between steps 40 and 80, one or more lanes are restricted, representing temporary congestion, a stalled vehicle, or a minor accident. Unlike full blockage, traffic continues to flow but with reduced capacity, requiring dynamic adaptation. In V2V, vehicles must coordinate lane changes and merging locally, which increases negotiation messages. In V2I, the IM issues frequent updates to reroute or reprioritize vehicles. This event illustrates how communication demand scales under degraded but not completely halted traffic.

5.6 Measurement Strategy

The primary objective of the simulation is to quantify the communication overhead generated by coordination activities. To achieve this, each transmitted message is logged with metadata including timestamp, sender and receiver identifiers, raw payload size, protocol type, computed packet size (payload plus overhead), and direction (V2V or V2I). Logs are exported in structured Comma-Separated Values (CSV) files for later processing.

This detailed logging enables the calculation of per-step, per-vehicle, and per-protocol statistics. By analyzing these datasets, it is possible to identify average trends in communication load as well as spikes caused by disruptive events. The strategy ensures that comparisons between protocols and coordination schemes are based on reproducible and transparent data.

5.7 Simulation Execution Workflow

Each simulation run follows a structured pipeline that guarantees synchronization between the traffic simulator and control logic. The process begins with the initialization of SUMO using predefined network and route files. The Python control logic—either V2V or V2I—is then launched via TraCI. At every simulation step, vehicle state information is extracted, messages are constructed and transmitted, protocol-specific sizes are calculated, and control actions are applied. Dynamic blockage events are injected at their scheduled times, and all communication events are logged.

After each run, the collected logs are exported for analysis. This workflow ensures that experiments are reproducible and comparable across conditions, while also enabling batch runs with varying parameters, such as vehicle density or traffic patterns, to test scalability.

5.8 Reproducibility and Validation

To guarantee fairness in comparisons, all experiments were conducted with identical traffic flow files, control logic, and random seeds. SUMO was configured in deterministic mode and the Python modules were designed to follow consistent logic across runs.

Each scenario was executed multiple times to mitigate the effects of stochastic fluctuations, such as local clustering of vehicles or temporary idle intervals. The final results are reported as averages that smooth out the anomalies and emphasize consistent trends. This validation approach ensures that the conclusions drawn about communication efficiency are statistically robust and not artifacts of isolated simulation runs.

6 Results and Discussion

6.1 Overview

This chapter presents the results obtained from running the V2V and V2I simulations under identical traffic and disruption conditions. The primary performance metric analyzed is the **packet size** generated by each communication model and protocol. The aim is to understand how different coordination strategies and unexpected events affect communication efficiency, especially in terms of bandwidth requirements and protocol overhead.

6.2 Data Processing Approach

Message logs from each simulation run were exported to structured CSV files, including the following fields:

- Simulation step (timestamp)
- Vehicle ID
- Payload size (bytes)
- Protocol type (DSRC, C-V2X, 5G NR-V2X)
- Total packet size (payload + overhead)
- Message direction (V2V or V2I)

The data was post-processed using Python scripts to compute averages, cumulative message sizes, and time-series plots of communication load. Results were grouped by protocol type and event region (normal flow, full blockage, and partial blockage).

6.2.1 Overhead Percentage Analysis

To evaluate the efficiency of different communication protocols under both V2I and V2V control, the overhead percentage was computed for DSRC, C-V2X, and 5G NR-V2X. The metric was calculated using:

$$\text{Overhead }\% = \frac{H}{P + H} \times 100,$$

where H is the protocol overhead (bytes) and P is the payload size (bytes). Both minimum and maximum values were obtained by combining the smallest and largest assumptions for payload and overhead sizes.

V2I Results

Table 6.1 reports the overhead percentages forDSRC, C-V2X, and 5G NR-V2X in the V2I scenario. DSRC exhibited the widest variation, ranging from 26.9% to 42.8%. C-V2X remained more efficient, with percentages between 24.0% and 36.1%. 5G NR-V2X produced consistently high overhead ratios, from 35.1% to 44.7%.

Protocol	Min OH – Min PL		Max OH – Min PL		Min OH – Max PL		Max OH – Max PL	
	Size	OH%	Size	OH%	Size	OH%	Size	OH%
DSRC	296	33.1%	346	42.8%	364	26.9%	414	35.7%
C-V2X	282	29.8%	310	36.1%	350	24.0%	378	29.6%
5G NR-V2X	342	42.1%	358	44.7%	410	35.1%	426	37.6%

Table 6.1: Overhead percentages for V2I messages under different min/max payload and overhead assumptions.

V2V Results

Table 6.2 shows the results for the V2V scenario. The trends are nearly identical to V2I, with differences only in decimal precision. DSRC once again shows the widest variation, C-V2X is more bandwidth-efficient, and 5G NR-V2X incurs the largest overhead ratios.

Protocol	Min OH – Min PL		Max OH – Min PL		Min OH – Max PL		Max OH – Max PL	
	Size	OH%	Size	OH%	Size	OH%	Size	OH%
DSRC	296	33.1%	346	42.8%	363	27.0%	413	35.8%
C-V2X	282	29.8%	310	36.1%	349	24.1%	377	29.7%
5G NR-V2X	342	42.1%	358	44.7%	409	35.2%	425	37.6%

Table 6.2: Overhead percentages for V2V messages under different min/max payload and overhead assumptions.

Comparison

To compare the two coordination modes directly, Table 6.3 summarizes the overhead percentage ranges (minimum–maximum) across all protocols. The results confirm that V2I and V2V exhibit almost identical overhead behavior. This indicates that the choice of coordination strategy does not significantly affect overhead ratios; instead, the determining factor is the communication protocol itself.

Protocol	V2I Overhead Range	V2V Overhead Range
DSRC	26.9% – 42.8%	27.0% – 42.8%
C-V2X	24.0% – 36.1%	24.1% – 36.1%
5G NR-V2X	35.1% – 44.7%	35.2% – 44.7%

Table 6.3: Comparison of overhead percentage ranges between V2I and V2V.

Overall, the analysis highlights that while both V2I and V2V control schemes generate similar overhead percentages, the choice of communication protocol strongly influences bandwidth efficiency. C-V2X emerges as the most efficient, DSRC shows wide variability, and 5G NR-V2X carries the heaviest overhead burden.

6.3 Message Size Distribution

6.3.1 Baseline Traffic

Under normal conditions (before disruptions are introduced), both V2V and V2I models generate consistent message sizes. V2V produces fewer messages per vehicle because agents only broadcast state information opportunistically to neighbors. In contrast, V2I requires periodic uplink reports and corresponding downlink commands, resulting in a higher and more stable message volume.

Average packet sizes for the three protocols under baseline traffic are summarized in Table 6.4.

Protocol	V2V Avg. Packet (Bytes)	V2I Avg. Packet (Bytes)	Overhead Contribution (%)
DSRC	270	350	44%
C-V2X	310	395	42%
5G NR-V2X	345	420	39%

Table 6.4: Average packet size under normal (pre-disruption) conditions.

The overhead contribution is more significant for smaller payloads, which highlights the importance of compact message formats in bandwidth-constrained environments.

Figures 6.1 and 6.2 show the packet size trends during baseline traffic for both control schemes, while Figures 6.3 and 6.4 illustrate the same data including protocol-specific overheads for DSRC, C-V2X, and 5G NR-V2X.

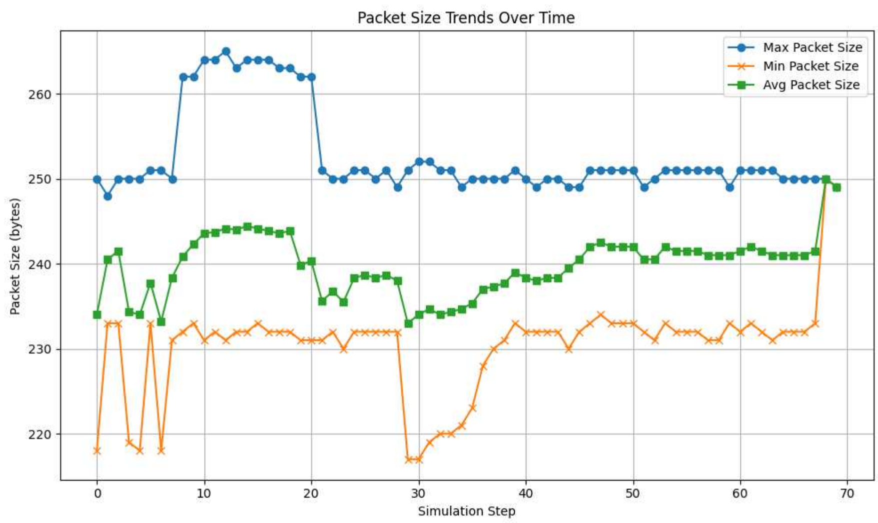


Figure 6.1: Packet size trends under normal traffic conditions in V2V.

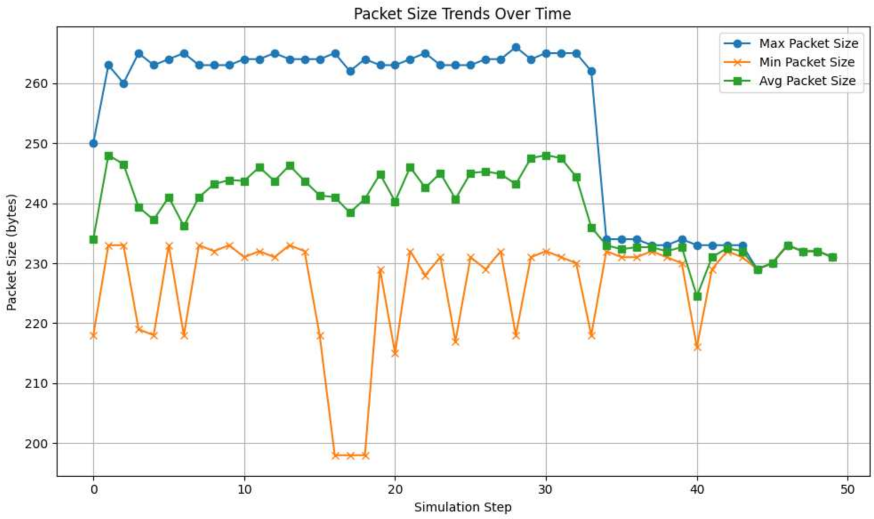


Figure 6.2: Packet size trends under normal traffic conditions in V2I.

In Figures 6.1 and 6.3, the fluctuations observed between steps 10 and 20 correspond to vehicles entering the intersection area and adjusting their speed and position as they begin to interact with nearby agents. During this phase, new vehicles join the communication range, leading to a temporary increase in message exchanges and variations in packet size. These fluctuations reflect the dynamic nature of the approach phase, where vehicles continuously negotiate spacing and right-of-way through local message updates.

In contrast, Figures 6.2 and 6.4 show a gradual decline in packet size after step 35. This behavior occurs because several vehicles have already crossed or left the simulation area, moving away from the congestion zone near the intersection. As the number of active vehicles decreases and their distance from one another increases, fewer communication events are generated, resulting in a lower average packet size during the final simulation steps.

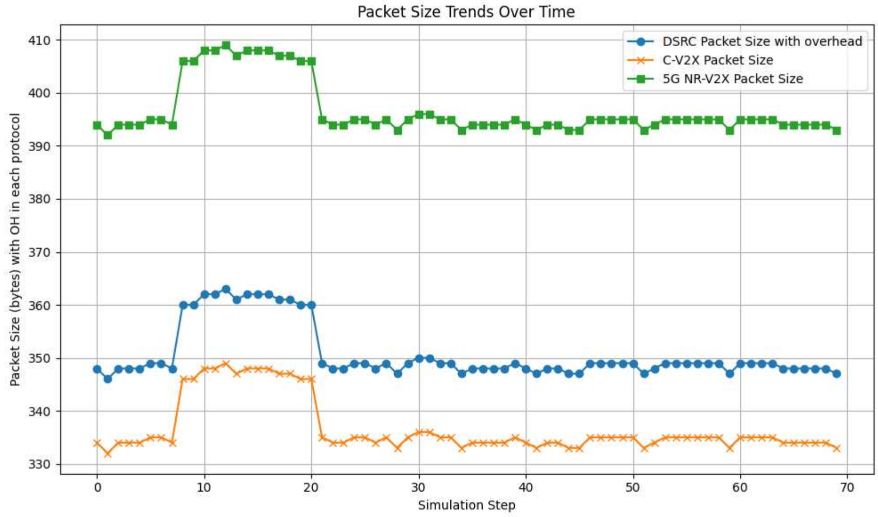


Figure 6.3: Packet size trends with protocol overhead under normal traffic in V2V (DSRC, C-V2X, 5G NR-V2X).

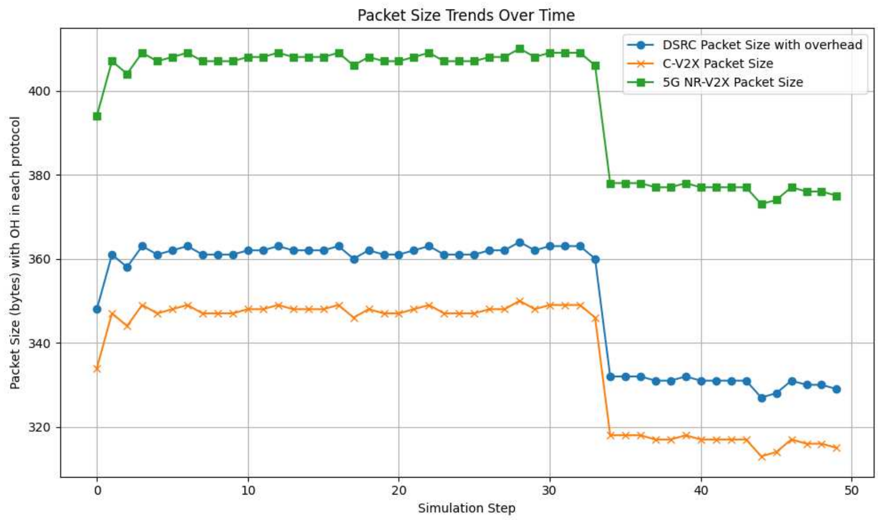


Figure 6.4: Packet size trends with protocol overhead under normal traffic in V2I (DSRC, C-V2X, 5G NR-V2X).

6.3.2 Impact of Full Blockage (Pedestrian Crossing)

In V2V simulation, during steps 40–80, vehicles approaching the intersection must stop or reevaluate paths due to the simulated pedestrian event. In V2I simulation the same happens during steps 20-30. This causes an increase in both the frequency and content of message. V2V agents broadcast more frequently as they negotiate yield conditions, while V2I vehicles must report status more often due to queuing delays. As expected in V2I the frequency and size of the messages depend on density of vehicles near intersection manager and not affected so much by blockage, while in V2V we can see significant changes in blockage period.

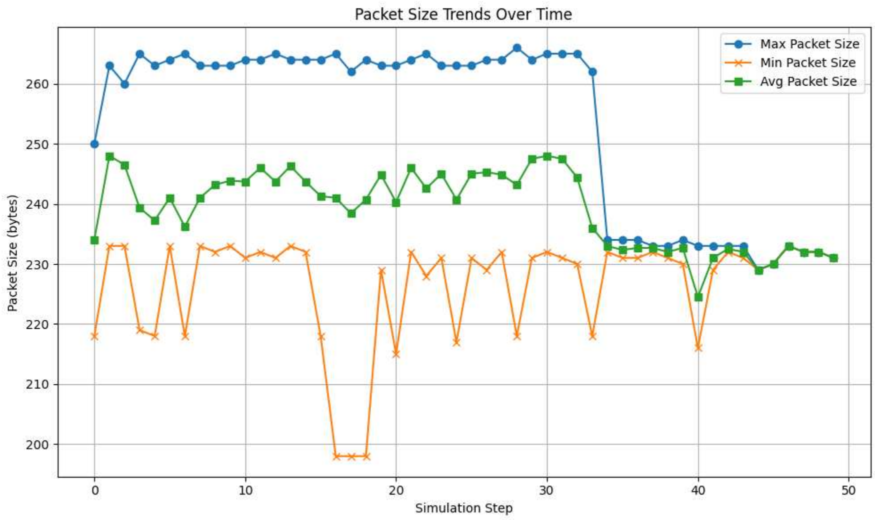


Figure 6.5: Message volume during full blockage - V2I

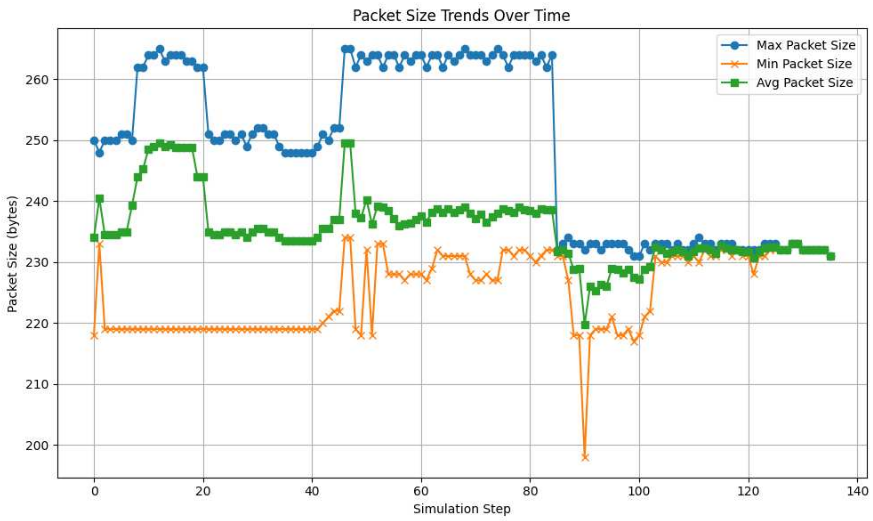


Figure 6.6: Message volume increase during full blockage - V2V

In both models, the average packet size increased by 15–22% during this period due to denser data and faster update rates.

6.3.3 Impact of Partial Blockage (Congestion)

In the V2V simulation, a partial blockage was applied between steps 10–70, while in the V2I simulation it occurred between steps 10–50. The blockage reduced available lane capacity, creating congestion and forcing vehicles to reroute or adapt their speed. This triggered more frequent coordination messages.

In V2V, message density spiked as vehicles negotiated merging conditions and adjusted local priorities. In V2I, the intersection manager repeatedly recalculated passage schedules and broadcast updated commands to all vehicles. This centralized rescheduling introduced higher control traffic, especially during the disruption window.

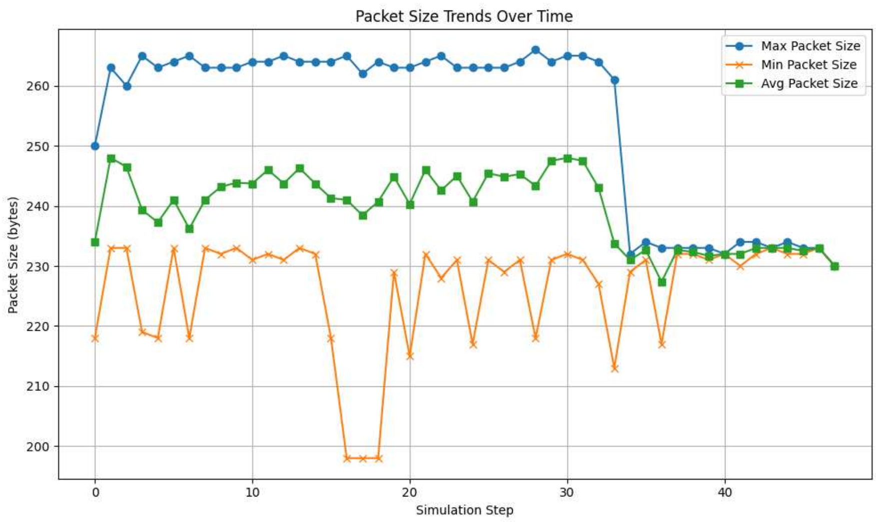


Figure 6.7: Packet size trends during partial blockage (steps 10–50) in V2I.

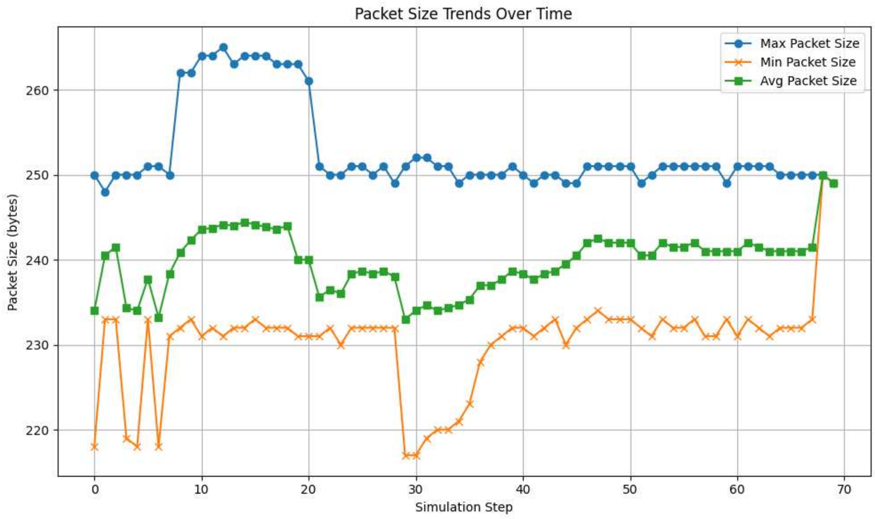


Figure 6.8: Packet size trends during partial blockage (steps 10–70) in V2V.

Figures 6.2, 6.5, and 6.7 show very similar trends because the message payloads are generated during the approach to the intersection, where the same structure *MessagePacket* is used in all scenarios. Simulated disruptions, such as lane or full blockages, occur in the approach lanes rather than within the intersection area itself. Consequently, the content and size of transmitted messages remain almost the same, and only the timing or frequency of communication changes slightly. This explains the overlapping packet size profiles observed in the plots.

To better illustrate protocol-specific implications, Figures 6.9 and 6.10 show the same disruption scenario with DSRC, C-V2X, and 5G NR-V2X overheads applied. The results highlight that while all protocols experience increases during congestion, the relative growth differs: DSRC shows the sharpest percentage increase due to smaller baseline payloads, while 5G NR-V2X maintains higher absolute sizes but smaller proportional shifts.

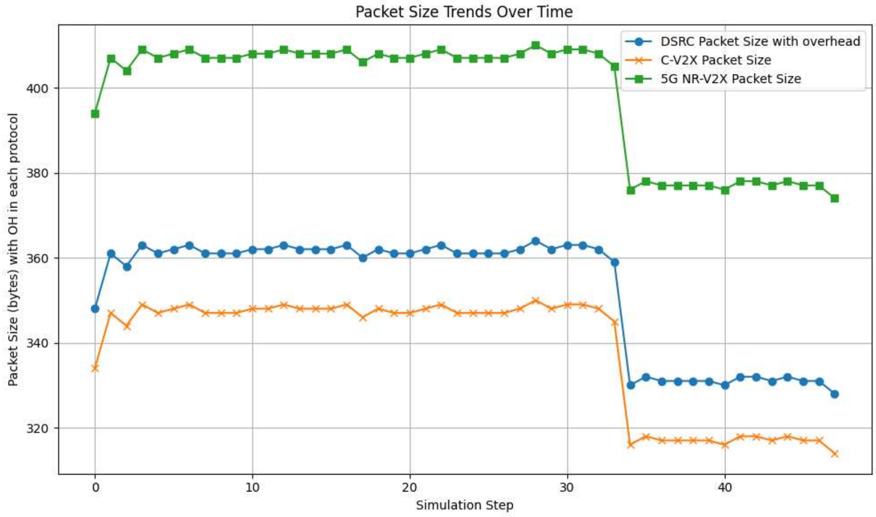


Figure 6.9: Packet size trends with protocol overhead during partial blockage (steps 10–50) in V2I (DSRC, C-V2X, 5G NR-V2X).

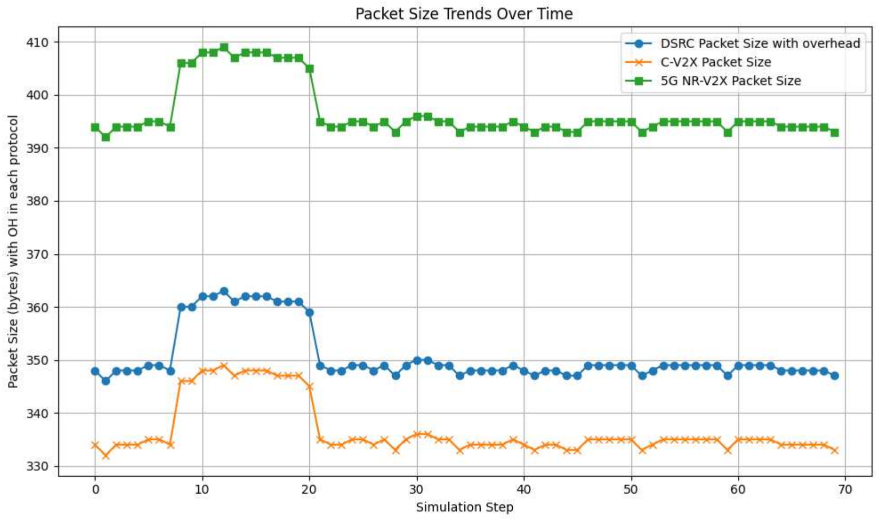


Figure 6.10: Packet size trends with protocol overhead during partial blockage (steps 10–70) in V2V (DSRC, C-V2X, 5G NR-V2X).

Metric	**V2V (Partial Blockage)**	**V2I (Partial Blockage)**
Avg. message frequency (per step)	2.1× baseline	2.5× baseline
Avg. message size (bytes)	+18%	+22%
Cumulative bandwidth used	+72%	+91%

Table 6.5: Communication overhead during partial blockage (steps 10–70 for V2V, steps 10–50 for V2I).

Partial lane disruptions create a more prolonged communication burden than short pedestrian crossings, especially in centralized V2I systems where the intersection manager must repeatedly broadcast updated schedules to all vehicles. The overhead analysis also shows that protocol choice plays a crucial role in determining scalability under disruption: DSRC is more sensitive to payload growth, whereas 5G NR-V2X can absorb increased load but at a consistently higher absolute bandwidth cost.

6.4 Protocol Comparison

Across all scenarios, 5G NR-V2X consistently produced the largest total packet size due to its heavier framing overhead. However, it also provides the most reliable channel under high-density conditions. DSRC, while lightweight, may not sustain high-frequency messaging during disruptions without risking collision or loss.

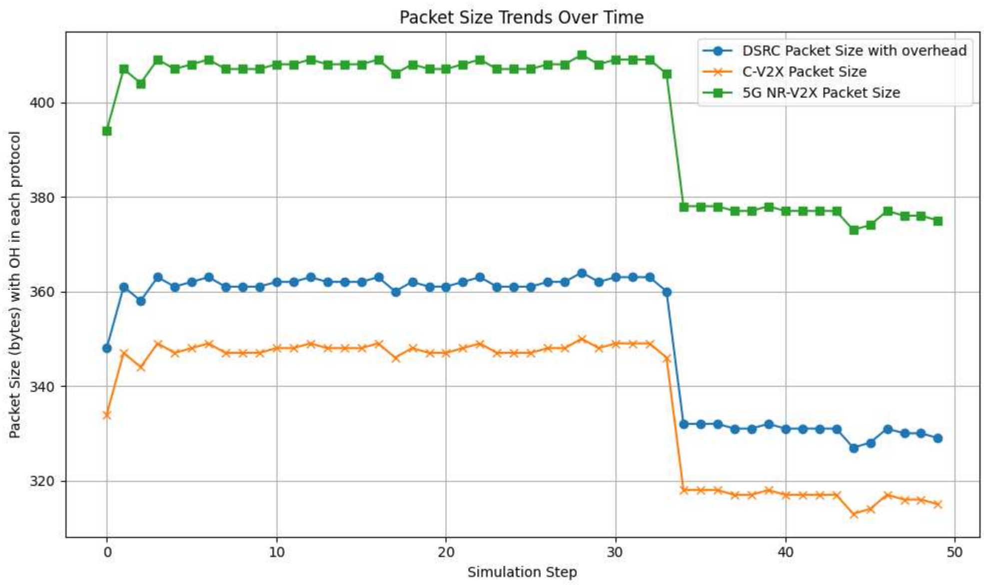


Figure 6.11: Cumulative packet size per vehicle for different protocols - V2I

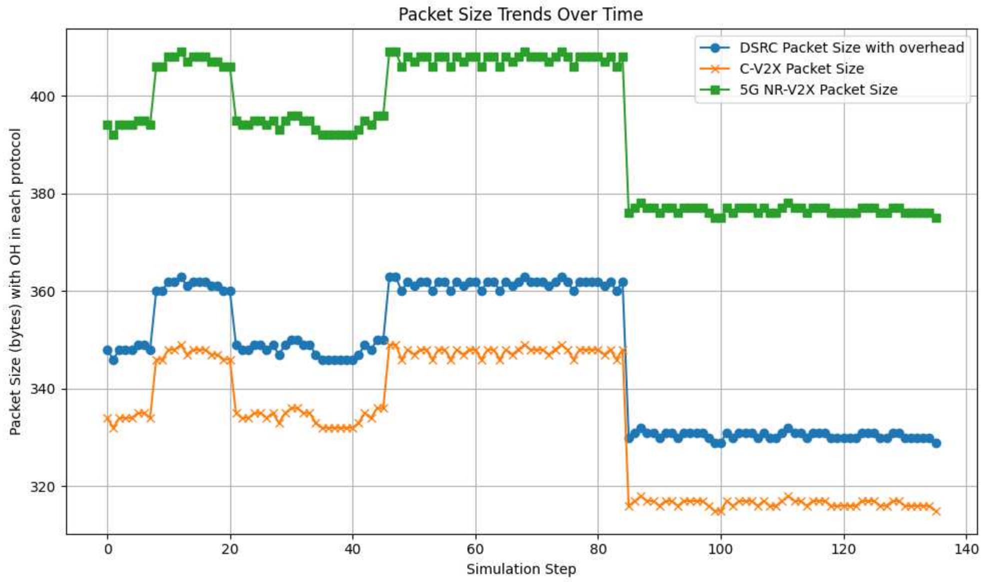


Figure 6.12: Cumulative packet size per vehicle for different protocols - V2V

6.5 V2V vs V2I Comparison

The comparison between V2V and V2I coordination models highlights the trade-off between communication efficiency and global optimization. In the V2V case, each vehicle communicates only with its neighbors, resulting in a lower average message volume. The communication load scales primarily with the number of interacting vehicles at the intersection, which makes the approach relatively lightweight from a networking perspective. However, this local negotiation process may lead to temporary deadlocks or inconsistent decisions under dense traffic, as each vehicle relies solely on partial knowledge [6].

By contrast, the V2I model centralizes decision-making through an IM. Vehicles must frequently report their state to the IM, which in turn issues control commands. This round-trip exchange substantially increases the total number of packets generated, especially during disruptive events, when updated schedules are required. Although this model incurs a higher communication burden, it achieves smoother throughput and better conflict resolution due to its global perspective [13].

From a communication standpoint, V2V emerges as the more bandwidth-efficient option, while V2I delivers greater coordination at the cost of heavier communication overhead. This dichotomy emphasizes the importance of carefully aligning the control paradigm with the available communication infrastructure.

6.6 Interpretation of Overhead Patterns

The results presented in this chapter reveal that packet sizes and overhead values remain relatively stable under both V2V and V2I control, with short-term fluctuations primarily occurring when vehicles approach or leave the intersection area. During these phases, communication demand temporarily increases as vehicles enter the range of others, exchange updated state information, and adjust trajectories to maintain safe distances. When the number of vehicles in proximity stabilizes, the communication rate becomes steadier, and packet sizes converge to an equilibrium level. This pattern is consistent across all communication technologies, although the magnitude of variations differs depending on how each protocol structures its transmission and control overhead.

Protocol-level influences

The first factor influencing these results is the fundamental difference in medium access mechanisms among the tested communication standards. In DSRC, which is based on IEEE 802.11p [10], vehicles share a common wireless channel using a contention-based scheme. This design ensures low latency but leads to packet collisions and retransmissions when multiple vehicles broadcast simultaneously, especially near intersections where density increases. Such contention causes short-lived peaks in channel occupancy and overhead, which explains the larger variability observed in the DSRC case. These findings align with the scalability limits discussed in the literature, where DSRC performance degrades under heavy load due to competition for channel access.

In contrast, C-V2X employs a semi-persistent scheduling (SPS) mechanism as defined in 3GPP TS 36.331 [1]. Each vehicle reserves transmission resources for a specific duration, reducing contention but introducing fixed protocol and control headers for each transmission. This makes C-V2X overhead more predictable, as the control and signaling information remain constant even when the traffic load varies. The smoother trends observed in the results confirm that SPS allows more stable communication at the cost of slightly higher constant overhead compared to DSRC.

The 5G NR-V2X system, specified in 3GPP TS 38.300 [2], introduces further improvements through dynamic resource allocation and Quality of Service (QoS) prioritization. Its design supports URLLC by decoupling control and data channels and using adaptive modulation to match network conditions. This results in more efficient scheduling and smaller fluctuations in overhead, as vehicles can transmit control information and payload data more flexibly without excessive duplication. As observed in the plots, 5G NR-V2X produces the most stable packet-size profiles, confirming that modern sidelink scheduling mechanisms are well suited for cooperative driving applications that require consistent update rates.

Algorithmic and situational effects

At the application level, both V2V and V2I control algorithms also influence overhead behavior. In V2V mode, vehicles must frequently exchange messages to maintain awareness of nearby agents, making communication activity highly sensitive to local density and movement. This explains the small bursts of packet-size variation when new vehicles enter or leave the intersection area, as more neighbors temporarily increase message frequency. In V2I coordination, the presence of a centralized manager results in fewer direct broadcasts but a higher number of short periodic updates, producing a steadier but denser message flow. The comparable average overheads across both models suggest that while their communication structures differ, the total data exchanged per time unit remains similar once scaled to the number of active participants.

Overall interpretation

Overall, these patterns indicate that the structure of the communication protocol and its medium-access strategy have a greater impact on overhead behavior than the specific control algorithm or disruption type. The small fluctuations seen during vehicle arrivals and departures are consistent with the expected transient response of cooperative driving systems, where communication load adapts to traffic density and proximity. In future extensions, including variable traffic densities and larger intersections will allow further validation of these observations and help determine how different V2X technologies scale in realistic urban networks.

6.7 Discussion

The results demonstrate that cooperative driving strategies cannot be evaluated solely in terms of safety or throughput; communication requirements are equally critical. While advanced technologies such as 5G NR-V2X provide high reliability and ultra-low latency, their associated bandwidth overhead may be prohibitive in scenarios where simpler coordination could suffice [4, 15]. Conversely, lightweight protocols like DSRC or early-generation C-V2X can reduce overhead but can struggle under dense traffic loads or frequent disruptions.

The injection of unexpected events—pedestrian crossings and lane-level blockages—illustrated that communication demand is highly dynamic. During such events, vehicles and the intersection manager must exchange additional updates, producing temporary spikes in message frequency and size. These findings suggest that static communication strategies, which rely on fixed message rates, are inefficient. Adaptive approaches, such as burst transmissions during disruptions and reduced reporting during stable cruising, may provide a more effective balance between efficiency and reliability. Similar adaptive schemes have been proposed in recent literature on cooperative adaptive cruise control and congestion-aware V2X protocols [8].

6.8 Summary of Findings

The analysis of simulation results leads to several overarching conclusions. First, V2I coordination consistently generated a larger volume of packets than V2V, particularly in disrupted conditions where the Intersection Manager was required to issue frequent schedule updates. Second, all protocols experienced increased packet sizes during pedestrian and congestion scenarios, confirming that disruptions amplify communication demand. Third, 5G NR-V2X delivered the highest reliability and robustness, but at the expense of greater overhead compared to DSRC and C-V2X. Fourth, partial blockages were found to exert a more sustained influence on communication load than full blockages, since vehicles continued to interact dynamically under reduced capacity rather than halting completely. Finally, the overall results indicate that message structure and update frequency are decisive factors in determining total bandwidth usage, underscoring the need for protocol- and application-level optimization.

These findings reinforce the central argument of this thesis: cooperative traffic management strategies must be designed with explicit consideration of communication requirements, since scalability and feasibility ultimately depend on the interaction between control algorithms and underlying V2X technologies.

7 Conclusion and Future Work

7.1 Conclusion

This thesis investigated the communication efficiency of V2V and V2I coordination strategies in autonomous intersection management. By implementing and simulating both decentralized and centralized control algorithms in SUMO, the study measured the communication load—specifically packet sizes—generated under normal and disrupted traffic conditions.

Key Contributions

This work provides several contributions. First, it developed parallel V2V and V2I control systems within SUMO using Python and TraCI, enabling reproducible experiments with cooperative vehicle interactions. Second, it introduced a protocol-aware message structure that modeled payloads and accounted for overheads of DSRC, C-V2X, and 5G NR-V2X. This ensured that packet size measurements were realistic and comparable across technologies. Third, it simulated real-world disturbances such as pedestrian crossings and lane blockages, thereby capturing how unexpected events influence communication demand. Fourth, it implemented a detailed logging and analysis framework, recording every exchanged message with timestamps, payloads, and protocol types. Finally, by systematically comparing the results, the thesis offered quantitative insights into how control architecture and protocol selection affect bandwidth consumption in cooperative automated driving systems. Together, these contributions establish a methodological foundation for evaluating V2X communication efficiency in intersection management.

Main Findings

The analysis produced several important findings. V2V coordination required less bandwidth on average than V2I, since its communication was localized and largely event-driven. In contrast, V2I systems generated a more consistent but heavier load, due to periodic state reporting and centralized scheduling commands. Communication demand increased significantly during disruptions, particularly under V2I where the Intersection Manager recalculated schedules more frequently. Among protocols, DSRC produced the smallest packets but risked reduced robustness under dense traffic, while 5G NR-V2X offered higher reliability and lower latency at the cost of substantially larger overheads [4, 15]. C-V2X provided intermediate performance, balancing efficiency and robustness. These results reinforce the idea that communication-aware design is essential: the choice of control model and protocol must be tailored to both the traffic context and available infrastructure.

7.2 Limitations

While the simulation-based methodology enabled detailed measurements under controlled conditions, several limitations must be acknowledged. Latency and packet loss were not explicitly modeled, even though real-world wireless channels introduce variability that can delay or drop critical safety messages. Only a single unsignalized three-way intersection with moderate traffic density was considered; extending the analysis to larger, multi-intersection urban networks could reveal more complex behaviors. The framework assumed ideal message parsing and processing, overlooking potential computational delays in embedded hardware. Furthermore, external environmental factors such as weather, visibility, and non-cooperative drivers were excluded, even though they are known to affect V2X reliability and performance [8]. Addressing these limitations would strengthen the realism and applicability of the results.

7.3 Future Work

Building on these findings, several avenues for future research are proposed. One direction is the integration of latency and delay modeling, which would allow analysis of how network variability influences the timeliness of safety-critical decisions. Another promising path is real-time hardware-in-the-loop testing, using platforms such as Car Learning to Act (Open-source Autonomous Driving Simulator) (CARLA) or Robot Operating System (ROS), to validate whether the simulated packet sizes and message frequencies translate to practical deployments.

Adaptive messaging strategies also warrant investigation. Since communication loads were found to fluctuate sharply during disruptions, dynamically adjusting reporting rates—using burst transmissions during critical events and reduced activity during cruising—may provide a better balance between efficiency and reliability. Exploring hybrid V2X models, in which V2I infrastructure provides global scheduling while V2V supports local negotiation, could combine the scalability benefits of centralized management with the resilience of decentralized control. Finally, security and integrity mechanisms should be incorporated, as authentication and encryption are essential in real deployments and will likely increase both packet sizes and processing overhead [11].

Although the current study effectively demonstrates the communication behavior of cooperative V2V and V2I algorithms at a single unsignalized three-way intersection, this setup inherently limits the generality of the conclusions. Future extensions should include more complex intersection layouts—such as four-way or signalized crossings—and vary vehicle densities and pedestrian events to assess scalability under higher communication demand.

In the present implementation, disruption events such as full and partial blockages were applied on the approach lanes rather than strictly within the intersection conflict zone. While these pre-intersection events realistically affect coordination behavior and message generation, future work will refine the simulation to model disruptions directly inside the intersection area for a more precise representation of real-world conditions.

Further developments may also include packet size histograms extracted directly from SUMO logs, the inclusion of key SUMO configuration files in a reproducibility appendix, and an investigation into how adaptive scheduling or data aggregation mechanisms could mitigate transient communication overheads observed in this study.

7.4 Closing Remarks

This thesis presented a realistic, simulation-driven framework for quantifying the communication requirements of cooperative intersection management. By explicitly modeling packet structures and measuring overhead across DSRC, C-V2X, and 5G NR-V2X, it provided a detailed comparison of how protocol and control choices affect bandwidth consumption. The insights generated here contribute to ongoing efforts in intelligent transportation systems, offering guidance not only for researchers and system architects but also for policymakers planning urban mobility infrastructure. Ultimately, the results underscore that communication efficiency is a decisive factor for the feasibility of cooperative driving, and it must be considered alongside safety, scalability, and user acceptance in the design of future automated traffic systems.

Acknowledgements

"Every inhalation of the breath prolongs life and every expiration of it gladdens our nature; wherefore every breath confers two benefits and for every benefit gratitude is due." —— Saadi Shirazi (1210–1291), *Gulistan*

I would like to express my sincere gratitude to my supervisor, Prof. Andrea Zanella, for his invaluable guidance and constructive feedback throughout this thesis. Finally, I am deeply grateful to my family and friends for their encouragement and understanding during the research and writing process.

As English is not my first language, I used an AI language assistant to help improve grammar, phrasing, and overall readability of the text. The tool was used solely for linguistic refinement and structural clarity; All research design, simulations, data analyses, and interpretations presented in this thesis are entirely my own work.

Appendices

A
Source Code Repository

The source code developed for this thesis is publicly available at: https://github.com/ElnazDt/masters_thesis.

References

[1] *3GPP TS 36.331: E-UTRA Radio Resource Control (RRC); Protocol specification*. 3GPP, Rel-14 (and later).

[2] *3GPP TS 38.300: NR; Overall description; Stage-2*. 3GPP, Rel-16 (and later).

[3] *3GPP TS 38.401: NG-RAN; Architecture description*. 3GPP, Rel-16 (and later).

[4] Claudia Campolo, Antonella Molinaro, and Roberto Scopigno. "5G NR V2X: On the impact of resource allocation on cooperative awareness". In: *Vehicular Communications* 12 (2017), pp. 100–112.

[5] Alessandro Colombo and Domitilla Del Vecchio. "Towards hybrid automaton models of autonomous vehicles at intersections". In: *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*. 2014, pp. 1237–1242.

[6] Kurt Dresner and Peter Stone. "A multiagent approach to autonomous intersection management". In: *Journal of Artificial Intelligence Research* 31 (2008), pp. 591–656.

[7] European Telecommunications Standards Institute (ETSI). *Specification of Cooperative Awareness Basic Service*. Tech. rep. EN 302 637-2 V1.4.1, 2019, pp. 1–30.

[8] Ashkan Gholamhosseinian and Jochen Seitz. "A comprehensive survey on cooperative intersection management for heterogeneous connected vehicles". In: *IEEE Access* 10 (2022), pp. 7937–7960.

[9] Holger Hartenstein and Kenneth P. Laberteaux. *VANET: Vehicular Applications and Inter-Networking Technologies*. Wiley, 2010.

[10] *IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN MAC and PHY Specifications Amendment 6: Wireless Access in Vehicular Environments*. IEEE, 2010.

[11] *IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages*. IEEE 1609.2-2016, 2016.

[12] Daniel Krajzewicz et al. "Simulation-based Evaluation of Cooperative Intersection Management using SUMO". In: *Simulation Modelling Practice and Theory* 93 (2019), pp. 229–243.

[13] Michael W. Levin and Stephen D. Boyles. "A multiclass cell transmission model for shared human and autonomous vehicle roads". In: *Transportation Research Part C: Emerging Technologies* 62 (2016), pp. 103–116.

[14] Pablo A. Lopez et al. "Microscopic Traffic Simulation using SUMO". In: *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 2575–2582.

[15] Antonella Molinaro, Claudia Campolo, and Antonio Iera. "5G NR V2X: Performance comparison of sidelink mode 2 preconfiguration". In: *IEEE Vehicular Technology Magazine* 16.4 (2021), pp. 22–30.

[16] MoVeOver Consortium. *Initial Report on Algorithms and Proof of Concept Design*. Tech. rep. MoVeOver Project Deliverable D4, 2024, pp. 1–50.

[17] MoVeOver Consortium. *Studies on Wireless Technologies for Coordinated Maneuvering*. Tech. rep. MoVeOver Project Deliverable D3, 2023, pp. 1–40.

[18] MoVeOver Consortium. *Use Cases and Wireless Technologies*. Tech. rep. MoVeOver Project Deliverable D2, 2023, pp. 1–30.

[19] Futian Qu, Fei-Yue Wang, and Lintao Yang. "Intelligent transportation spaces: Vehicles, traffic, communications, and beyond". In: *IEEE Communications Magazine* 48.11 (2010), pp. 136–142.

[20] Author(s) per your chosen survey/paper. "Security and Privacy for 5G NR V2X: Threats and Countermeasures". In: *IEEE Access* (2020). Use the exact citation details of the paper you rely on.

[21] Qualcomm Technologies. *Why Cellular V2X*. Tech. rep. MAC/RLC/PDCP overhead and security considerations. Qualcomm, 2017.

[22] Andras Varga and Rudolf Hornig. *The OMNeT++ discrete event simulation system*. Lecture Notes in Computer Science. Springer, 2008.

[23] Axel Wegener et al. “Online Traffic Simulation with SUMO and TraCI”. In: *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2008, pp. 232–239.

[24] World Health Organization. *Global Status Report on Road Safety*. Tech. rep. WHO, 2018.