國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master’s Thesis

分散式路口管理中車對車通訊協定設計之分析

Analysis of Vehicle-to-Vehicle Protocol Design for Distributed Intersection Management

黃品淳

Pin-Chun Huang

指導教授：林忠緯 博士

Advisor: Chung-Wei Lin, Ph.D.

中華民國 114 年 7 月

July 2025

doi:10.6342/NTU202502495

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Chung-Wei Lin. In our weekly meetings, Professor Lin consistently provided insightful and constructive feedback based on our progress reports. His guidance helped me overcome numerous challenges throughout the research process.

I am also very grateful to my lab mates. The work of my seniors served as a major source of inspiration for my thesis direction, and listening to presentations on related topics during group meetings sparked many new ideas and perspectives.

Last but not least, I am deeply grateful to my family. Their unwavering support and constant encouragement have been invaluable throughout this journey.

Pin-Chun Huang

National Taiwan University
July 2025

doi:10.6342/NTU202502495

分散式路口管理中車對車通訊協定設計之分析

研究生：黃品淳　　指導教授：林忠緯 博士

國立臺灣大學資訊工程學研究所

摘要

隨著聯網與自駕車輛的快速發展，交叉路口的高效與安全管理需求日益增加。雖然集中式系統可透過車輛與基礎設施間的通訊（V2I）提供穩定的協調機制，但主要以車對車通訊（V2V）為基礎的分散式方法展現了更高的適應性與較低的基礎設施成本。然而，分散式方法也常因頻繁廣播帶來高通訊負擔。

本研究提出一種類似防火牆的V2V協定框架，透過動態建立訊息過濾規則以降低通訊負擔，同時維持車輛的情境感知能力。我們進一步設計了一個分散式交叉路口管理框架，並對三個提出的協定與兩個基準方法進行比較與評估。實驗結果顯示，部分具備過濾功能的協定在效率上可與不做任何訊息過濾的基準測試相當，顯示許多訊息在該框架下是冗餘的。採用更細緻的衝突區域劃分時，即使在高交通密度情況下，系統仍能維持可接受的低延遲。此外，系統

doi:10.6342/NTU202502495

在高達55%的訊息遺失率下仍展現出良好的穩健性。這些結果凸顯了我們所提出的框架在基礎設施有限與通訊干擾等真實環境中，具備實際部署的潛力。

關鍵詞：交叉路口管理、分散式演算法、車對車通訊協定、訊息過濾、自動駕駛汽車

doi:10.6342/NTU202502495

ANALYSIS OF VEHICLE-TO-VEHICLE PROTOCOL DESIGN FOR DISTRIBUTED INTERSECTION MANAGEMENT

Student: Pin-Chun Huang **Advisor: Dr. Chung-Wei Lin**

Department of Computer Science and Information Engineering
National Taiwan University

Abstract

Keywords: Intersection Management, Distributed Algorithms, Vehicle-to-Vehicle Protocol, Message Filtering, Autonomous Vehicles.

The growing presence of connected and autonomous vehicles has accelerated the need for efficient and safe intersection management. While centralized systems offer reliable coordination through vehicle-to-infrastructure communication. In contrast, distributed approaches based on vehicle-to-vehicle communication offer greater adaptability and reduced infrastructure costs, but often incur high message overhead due to frequent broadcasting.

This work proposes a firewall-style Vehicle-to-Vehicle (V2V) protocol framework that dynamically constructs message filtering rules to reduce communication overhead while preserving situational awareness. A distributed intersection management framework is developed to evaluate three proposed protocols against two benchmarks. Experimental results demonstrate that some protocols with filtering

doi:10.6342/NTU202502495

can match the efficiency of non-filtering baselines, highlighting that many messages are redundant. The system achieves an acceptable low delay even under high traffic density when finer conflict zone granularity is applied. It demonstrates robustness against message loss rates of up to 55%. These findings underscore the potential of the proposed framework for deployment in real-world environments with limited infrastructure and communication interference.

doi:10.6342/NTU202502495

Table of Contents

doi:10.6342/NTU202502495

doi:10.6342/NTU202502495

List of Tables

doi:10.6342/NTU202502495

List of Figures

doi:10.6342/NTU202502495

Chapter 1

Introduction

The rise of connected and autonomous vehicles (CAVs) has drawn more attention to intersection management systems, which coordinate traffic flow near intersections safely and efficiently through vehicular network communication. Most existing approaches are centralized, relying on an intersection manager to collect data from vehicles or infrastructure and determine a global passing order of vehicles. Although effective in control and reliability, such systems depend heavily on vehicle-to-infrastructure (V2I) communication and require substantial infrastructure support. [8]

As a more flexible alternative, distributed approaches based on vehicle-to-vehicle (V2V) communication have recently gained traction. They reduce infrastructure costs and better adapt to dynamic environments. However, in real-time systems, vehicles must frequently broadcast messages to maintain situational awareness, often resulting in excessive message overhead due to redundant information.

To address this, we propose a firewall-style V2V communication protocol framework that dynamically generates lightweight filtering rules, allowing vehicles to discard irrelevant messages and reduce communication load while retaining essential awareness. We also develop a distributed intersection management framework to evaluate the performance of the protocols. Our design enables trade-offs between

doi:10.6342/NTU202502495

message overhead, efficiency, and computational cost, offering flexibility for diverse deployment scenarios. Furthermore, we analyze the robustness of the system in non-filtering conditions, with a focus on its resilience to message loss.

This thesis begins with a review of related work and foundational models, including a graph-based model [10] representation. Then it defines the problem, details our proposed methodology, presents experimental results, and concludes with key findings and directions for future research.

doi:10.6342/NTU202502495

Chapter 2

Related Works

Intersection management (IM) for connected and autonomous vehicles (CAV) is a critical component of future intelligent transportation systems. Using vehicle-to-everything (V2X) communication technologies such as DSRC and C-V2X, vehicles can coordinate their movements in real time to reduce delays, energy consumption, and collision risks—surpassing the performance of traditional signal-based traffic control. Existing works on non-signalized IM can be broadly categorized into centralized and distributed frameworks [7].

2.1 Centralized Intersection Management

Centralized approaches dominated early research efforts. These methods assume the presence of an intersection manager that gathers global information of the whole intersection and computes a crossing schedule to prevent conflicts.

Several works have extended the basic centralized model to incorporate more realistic assumptions. For example, Gholamhosseinian et al. [4] proposed a Plenary Centralized Intersection Management Scheme (PCIMS) that accounts for vehicle heterogeneity, road conditions, and safety constraints. Ahn et al. [1] formulated the safety verification problem as a job-shop scheduling task. They design a supervisory algorithm that takes control of vehicles only when a collision is imminent.

doi:10.6342/NTU202502495

To address the scalability limitations of traditional optimization-based coordination, Guan et al. [5] propose an RL training algorithm, model accelerated proximal policy optimization (MA-PPO), which integrates prior knowledge to reduce computation time. Pan et al. [13] also target computational efficiency through a hierarchical optimization framework. By leveraging space-domain modeling and reformulating the problem using second-order cone programming (SOCP), they enable fast scheduling for centralized coordination.

Another important class of work focuses on model-based methods with formal safety guarantees. Lin et al. [10] propose a generic graph-based model. Based on the proposed model, they derive verification techniques to identify and validate potential deadlocks. The intersection management challenge then becomes the problem of designing a centralized graph-based cycle removal algorithm. Their experimental results demonstrate the expressiveness of the proposed model and the effectiveness and efficiency of the algorithm. Based on the work, Lin et al. [9] further consider the impact of time violations, where vehicles may not strictly follow the controller's instructions. They propose a Robustness-Aware Greedy Scheduling (RGS) algorithm to mitigate the adverse effects of such violations. The results show that the algorithm improves the robustness of the system while maintaining acceptable traffic efficiency.

In summary, centralized methods provide high levels of controllability, efficiency, and stability. However, they rely heavily on infrastructure support, such as roadside units (RSU) and a central intersection manager, which introduces a single point of failure. Moreover, they may incur longer communication delay and significant computation overhead, particularly in dense or mixed traffic scenarios.

doi:10.6342/NTU202502495

2.2 Distributed Intersection Management

To overcome the limitations of centralized systems, distributed approaches have been proposed, where each vehicle makes local decisions based on on-board sensing and V2V communication, sometimes assisted by V2I, without a central intersection manager. Although these methods reduce infrastructure dependency and improve flexibility, they also introduce challenges in coordination, safety assurance, and traffic efficiency.

Liang et al. [8] propose DIMP, a fully V2V-based protocol where vehicles dynamically form clusters, elect leaders, and negotiate priorities. This improves coordination and driving comfort without RSU. Some works leverage decentralized model predictive control (MPC) [6]. Qian et al. [14] design an MPC framework under fixed priority rules to ensure safety while enabling smooth and efficient crossing.

Several studies optimize for specific objectives. Malikopoulos et al. [11] present an analytic framework to minimize energy use while maximizing throughput. Campos et al. [15] integrate reachability analysis and heuristics to manage collision risks, trading optimality for scalability. Gadginmath et al. [3] introduce a two-phase algorithm with feature-based precedence for scalability. For scalability and training efficiency, Cederle et al. [2] propose a multi-agent reinforcement learning (MARL) approach where vehicles select strategies using local 3D sensing and pre-trained models.

Communication reliability is another critical concern. Ormándi et al. [12] show that a CAM-only scheme can maintain safety even under degraded V2V conditions. Using co-simulation, they demonstrate the impact of message delay, noise, and loss on system performance.

doi:10.6342/NTU202502495

2.3 Motivation

Despite notable progress, several limitations remain. Centralized methods face practical deployment challenges, including the single point of failure due to the central controller, high infrastructure demands, and longer communication delays.

Distributed methods, while more flexible, often retain partial infrastructure dependence and mainly focus on algorithm design or modeling. Crucially, few studies examine how the design of the V2V protocol impacts the robustness and performance of the system. Important aspects such as levels of information exchange and resilience to message loss remain underexplored. To address these gaps, this work adopts a protocol-centric view of distributed intersection management. Our main contributions include the following.

- Propose a firewall-style V2V protocol design framework to filter out redundant messages.
- Design a distributed intersection management framework for evaluation.
- Analyze trade-offs among message overhead, delay, and computation cost.
- Demonstrate the robustness of the system under different message loss without filtering.

doi:10.6342/NTU202502495

Chapter 3

Background

We begin by reviewing the foundational concepts, particularly the graph-based model for intersection management introduced by [10], which serves as a basis for our approach. First, we describe essential definitions and restrictions.

3.1 Intersection Definitions

An intersection space $S = (I, \Delta)$ includes intersection $I = (\Xi, T, L)$ and a set of $\Delta = \{\delta_1, \ldots, \delta_m\}$. The basic components of the *intersection space* S are defined as follows:

- $\Xi = \{\xi_1, \ldots, \xi_{|\Xi|}\}$ is a set of *conflict zones*. Conflict zones are the division of the intersection space. No two vehicles can occupy the same conflict zone at the same time.

- $T = \{t_1, \ldots, t_{|T|}\}$ is a set of *trajectories*, where each trajectory $t \in T$ is a finite non-repeating sequence of conflict zones with $|t| > 0$. A trajectory consists of one or more conflict zones sequentially. All available trajectories are predefined and satisfy the rule: vehicles from the same source lane never merge after trajectory bifurcation.

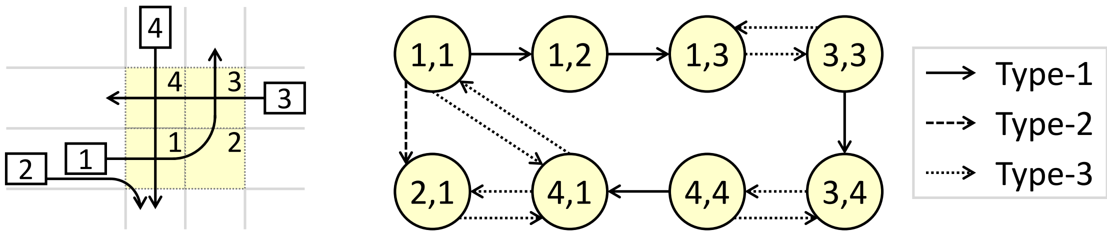
- $L = \{l_1, \ldots, l_{|L|}\}$ is a set of *source lanes* (before the intersection). Trajectories

doi:10.6342/NTU202502495

from the same source lane start with the same conflict zone.

- $\Delta = \{\delta_1, \ldots, \delta_{|\Delta|}\}$ is the set of *vehicles*, and a vehicle δ_i is a triple (l_i, t_i, a_i) where $l_i \in L$ is the source lane of the vehicle, $t_i \in T$ is the trajectory of the vehicle and $a_i \in \mathbb{R}^+ \cup \{0\}$ denotes the *earliest arrival time* of the vehicle. In addition, vehicles do not change their trajectories, reverse, or overtake (i.e., they do not pass other vehicles from the same source lane).

- For each vehicle, the *earliest arrival time* is defined as the earliest possible time it can begin entering the first conflict zone along its trajectory, assuming that there is no delay caused by any other vehicle.



(a) An Example of Traffic Flow.

(b) The timing conflict graph derived from the left figure.

Figure 3.1: An example of the graph-based model [10].

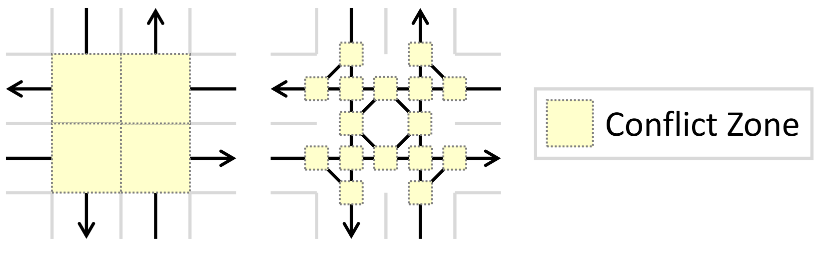


Figure 3.2: Different conflict zone granularities of an intersection [10]. In this thesis, we consider cases with 4 or 16 conflict zones.

doi:10.6342/NTU202502495

3.2 Timing Conflict Graph

Next, we introduce the timing conflict graph (TCG) defined in [10]. A timing conflict graph $G = (V, E)$ for a given intersection space I is a directed graph built using the following rules:

- There is a vertex $v_{i,j}$ if and only if conflict zone ξ_j is on the trajectory of vehicle δ_i. The vertex $v_{i,j}$ can be represented as (vid, cid), where *vid* denotes the ID of vehicle δ_i (i.e., i), and *cid* denotes the ID of conflict zone ξ_j (i.e., j).
- There is a Type-1 edge $(v_{i,j}, v_{i,j'})$ if and only if the next conflict zone of ξ_j on the trajectory of δ_i is $\xi_{j'}$.
- There is a Type-2 edge $(v_{i,j}, v_{i',j})$ if and only if δ_i and $\delta_{i'}$ are from the same source lane and the earliest arrival time of δ_i is earlier than that of $\delta_{i'}$, and ξ_j is on the trajectories of both δ_i and $\delta_{i'}$.
- There is a pair of Type-3 edges between $v_{i,j}$ and $v_{i',j}$ if and only if δ_i and $\delta_{i'}$ are from different source lanes, and ξ_j is on the trajectories of both δ_i and $\delta_{i'}$.

3.3 Distributed System

For a centralized intersection management system, the *intersection manager* receives information from vehicles or intersection infrastructure within its communication range, assigns a time window for each vehicle to enter each conflict zone along its trajectory, and sends messages to the controlled vehicles to inform them of their schedules.

In this thesis, we adopt a *distributed system* for intersection management, where each vehicle is equipped with a intersection manager that makes decisions

doi:10.6342/NTU202502495

based solely on its local view and locally available information. Each manager controls only the actions of its corresponding vehicle.

In addition, vehicles communicate with each other via V2V communication to exchange and update information, which is stored locally. Based on these locally maintained data, each vehicle determines whether it is allowed to proceed.

doi:10.6342/NTU202502495

Chapter 4

System Modeling

This section begins by introducing two newly proposed concepts: edge waiting time and conflict zone passing time. Then, it specifies the V2V message format adopted in the thesis. We then present the formal problem formulation, followed by a discussion of the key assumptions underlying our model.

4.1 Definitions

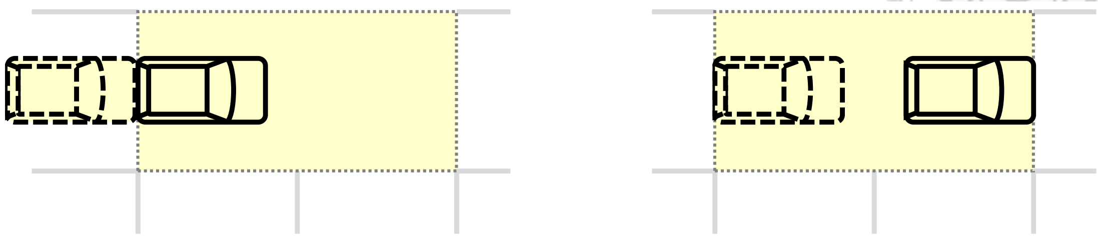
4.1.1 Edge Waiting Time

The *edge waiting time* refers to the time it takes for a vehicle to transition from the point it starts leaving a conflict zone (or its source lane) to the point it fully enters the next conflict zone on its path or exits the intersection entirely if no further conflict zones remain. As illustrated in Figure 4.1(a), it corresponds to the time taken for the dashed vehicle to reach the position of the solid vehicle.

4.1.2 CZ Passing Time

The *conflict zone (CZ) passing time* refers to the duration from when a vehicle fully enters a conflict zone to when it is ready to exit. As shown in Figure 4.1(b), it corresponds to the time taken for the dashed vehicle to reach the position of the solid vehicle.

doi:10.6342/NTU202502495



(a) Edge waiting time. (b) Conflict zone passing time.

Figure 4.1: Time-related definitions.

4.1.3 V2V Message Format

In this work, the V2V messages exchanged between vehicles are represented as tuples of the form $(vid, fromCid, toCid, createdTime)$. For example, a message (i, j, j', τ) indicates that at time τ, the vehicle δ_i is located in the conflict zone ξ_j, and the next conflict zone along its trajectory t_i is $\xi_{j'}$. A special case is when $j = -1$, which indicates that $\xi_{j'}$ is the first conflict zone on t_i; conversely, $j' = -1$ means that ξ_j is the last conflict zone on t_i.

4.2 Problem Statement

This work aims to investigate an V2V-only distributed approach to non-signalized intersection management based on the graph-based model [10]. Specifically, we design and implement different V2V communication protocols of varying levels of information exchange, without relying on a central intersection manager.

The objective is to analyze and compare the performance of these protocols through simulation, under a fixed scheduling rule. We seek to evaluate how communication strategies impact system efficiency and safety in a distributed setting.

To comprehensively assess protocol performance and robustness, we further conduct extended experiments considering:

doi:10.6342/NTU202502495

- **Message loss**: Evaluate the robustness of each protocol under different message drop rates.
- **Conflict-zone granularity** (n_{CZ}): Investigate the effect of different levels of spatial partitioning within the intersection.
- **Traffic density** (λ): Analyze how traffic arrival rates influence the behavior of the system.

4.3 Assumptions

To focus the scope of this thesis on the design of V2V communication protocols, the following assumptions are made:

- **Vehicle**
  - All vehicles are connected and autonomous.
  - Each vehicle is equipped with a collision avoidance mechanism.
  - All vehicles operate in synchronized global time.
  - Each communication–decision cycle is assumed to take 0.1 seconds.
- **Movement**
  - Each vehicle follows a predefined fixed trajectory that does not change or move in reverse during the simulation.
  - Vehicles from the same source lane do not overtake one another; their order is fixed.
- **V2V Communication**

doi:10.6342/NTU202502495

– Vehicles exchange V2V messages at a frequency of 10 Hz, following the SAE J2735 standard guideline for the transmission of the Basic Safety Message (BSM).

– Each vehicle initially knows the planned trajectory and the earliest arrival time of all other vehicles in the simulation round.

– A vehicle stops broadcasting messages 2 seconds after it fully exits the intersection.

In addition, after initially receiving the planned trajectories and earliest arrival times of other vehicles, each vehicle constructs the corresponding TCG and applies a unified Type-3 edge removal algorithm. This algorithm removes all Type-3 edges directed from a vehicle with a later earliest arrival time to one with an earlier one. Specifically, the edge from $v_{i,j}$ to $v_{i',j}$ is removed if and only if $a_i > a_{i'}$ or $(a_i = a_{i'} \wedge i < i')$.

As illustrated in Figure 4.2, assuming that vehicle IDs are assigned in ascending order of the earliest arrival time, the red edges indicate the Type-3 edges selected for removal from each pair of conflicting edges. The algorithm ensures that the resulting graph contains no cycles.

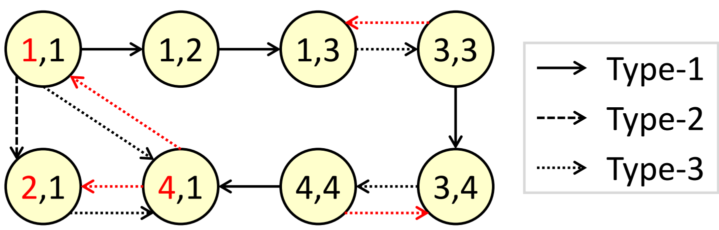


Figure 4.2: Illustration of the Type-3 edge removal algorithm on a TCG.

doi:10.6342/NTU202502495

Chapter 5

Methodology

In this section, we present the methodology for addressing the problem of distributed intersection management under different V2V communication protocols. We begin by defining five vehicle states and the transitions between them. Next, we describe the data each vehicle maintains locally, how this information is updated upon message reception, and the criteria a vehicle uses to decide whether to proceed. Finally, we introduce the concept of a V2V protocol as defined in this thesis, followed by detailed definitions of the three proposed and implemented protocols.

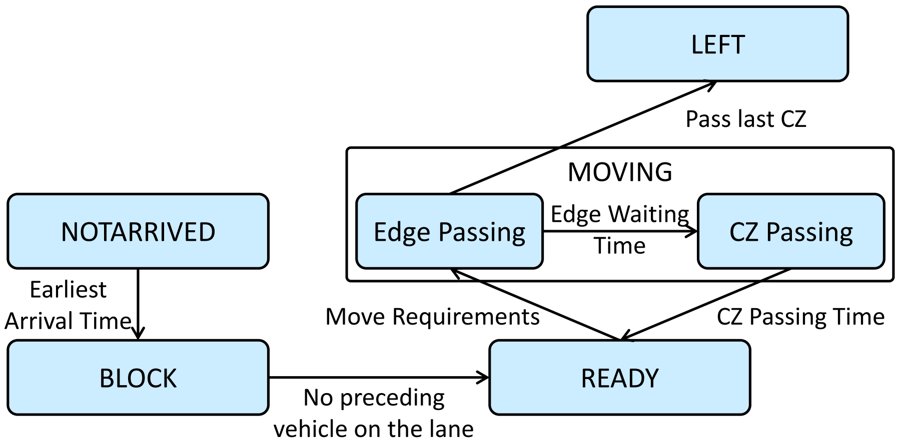


Figure 5.1: Vehicle state transitions modeled as a Finite State Machine (FSM).

doi:10.6342/NTU202502495

5.1 Vehicle States

We define five vehicle states and provide their definitions along with explanations of the transitions between them. At any given time, each vehicle is in exactly one of these states. The finite state machine illustrating these transitions is shown in Figure 5.1.

- **NOTARRIVED**: The vehicle has not reached the intersection, which means that its earliest arrival time a_i is later than the current time. Transition to **BLOCK** if current time $\geq a_i$.

- **BLOCK**: The vehicle has arrived at the intersection but cannot move because the preceding vehicles in the same source lane with earlier earliest arrival times have not yet entered any conflict zone. Transition to **READY** once there are no such preceding vehicles in the source lane.

- **READY**: The vehicle is ready to move. Transition to **MOVING-Edge Passing** once all movement requirements are satisfied.

- **MOVING**: This state contains two substates:

  - Edge Passing: The vehicle has begun leaving a conflict zone (or its source lane) but has not yet fully entered the next conflict zone along its trajectory. Transition to **MOVING-CZ Passing** once

    $$\text{current time} \geq \tau + \text{edge waiting time},$$

    where τ is the time the vehicle entered this state. If there is no subsequent conflict zone (i.e., the current one is the last in its trajectory), the vehicle transitions to the **LEFT** state instead.

doi:10.6342/NTU202502495

- CZ Passing: The vehicle is moving inside a conflict zone (CZ). Transition to **READY** once

$$\text{current time} \geq \tau + \text{cz passing time},$$

where τ is the time the vehicle entered this state.

- **LEFT**: The vehicle has exited the intersection.

5.2 State-based Decisions

In addition to vehicle state, each vehicle maintains a list that represents, from its own perspective, the states of all vertices on the TCG. Each vertex state can be either `EXECUTED` or `UNKNOWN`. Initially, all vertex states are set to `UNKNOWN`. Upon receiving a message of the form

$$\text{message} = (i, j, j', \tau),$$

the vehicle marks the vertex (i, cid) as `EXECUTED` for all $\text{cid} \in t_i$ such that cid appears before j in the trajectory t_i, which is a conflict zone sequence.

Each vehicle sends a message when it is in the **READY**, **MOVING**, or **LEFT** state. In the **LEFT** state, broadcasting lasts at most two seconds (Section 4.3).

Given the vehicle and vertex states, we now describe how a vehicle makes a state-based movement decision. δ_i proceeds to move to ξ_j when the following conditions are satisfied:

- The vehicle state of δ_i is **READY**.
- In the local information of δ_i, all vertices that have a type-3 edge pointing to (i, j) in the TCG are marked as `EXECUTED`.

doi:10.6342/NTU202502495

Under this movement requirement, the execution order of the vertices will strictly follow the topological order.

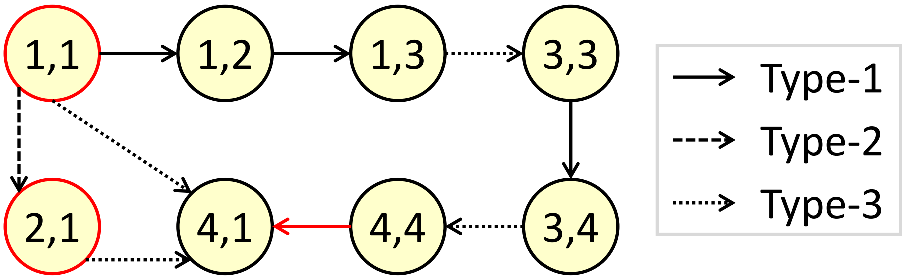


Figure 5.2: An example where δ_4 intends to move from ξ_4 to ξ_1. The vertices marked by red circles are considered, as they have type-3 edges pointing to $v_{4,1}$.

5.3 Protocol

In this thesis, we define a *firewall-style V2V protocol* as a set of rules that determine how a vehicle uses its own state and locally stored information to establish and update the criteria for filtering V2V messages. Based on the predefined protocol and the initial TCG, each vehicle can compute two sets: *targetVids* and *targetCids*. These sets are dynamically updated during vehicle movement according to the protocol rules. As illustrated in Figure 5.3, a vehicle accepts a message (i, j, j', τ) if and only if $(i \in targetVids \vee j' \in targetCids)$. This firewall-like design effectively reduces message overhead while enabling efficient and distributed coordination.

As a general assumption for all protocols, we consider that messages sent from vehicles in the **LEFT** state are always accepted. It ensures that, under ideal conditions (i.e., no message loss), no vehicle remains permanently stalled. Permanent stalling can occur when a vehicle leaves the intersection and stops broadcasting after a certain period, while another vehicle, the one that may eventually stall, fails to receive its messages during that time. As a result, the vehicle may never obtain

doi:10.6342/NTU202502495

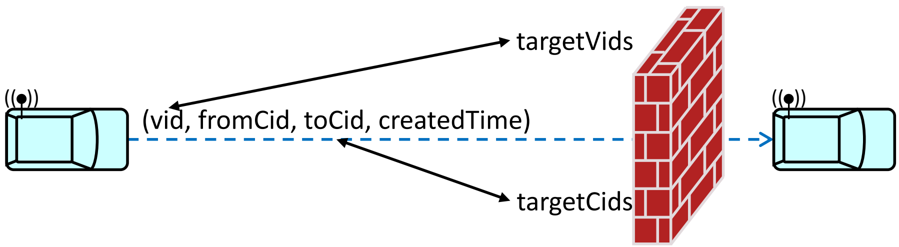


Figure 5.3: A firewall-like design that filters messages based on the dynamically maintained sets *targetVids* and *targetCids*.

the information necessary to meet its movement requirements. In particular, it may be unable to mark certain vertices (i, j) as `EXECUTED` because vehicle δ_i has already ceased communication.

We propose three protocols and one benchmark protocol without filtering for comparison.

Protocol-0 (Benchmark-2): Receive messages from all vehicles.

- **Construction of target sets:**

$$targetVids = \mathcal{V}, \quad targetCids = \emptyset,$$

  where $\mathcal{V} = \{1, 2, \ldots, |\Delta|\}$.

- No filtering is applied.

Protocol-1: Receive messages from all vehicles currently competing with the ego vehicle for the same conflict zone.

- **Construction of target sets:**

$$targetVids = \emptyset, \quad targetCids = \{cz_{\text{next}}\},$$

  where cz_{next} is the next conflict zone on the trajectory of the ego vehicle.

doi:10.6342/NTU202502495

- Update *targetCids* when the vehicle enters the **MOVING-CZ Passing** state.

Protocol-2: Receive messages from all vehicles that pose a potential collision risk to the ego vehicle.

- **Construction of target sets:**

$$targetVids = \{i \in \mathcal{V} \setminus \{i_{\text{self}}\} \mid t_i \cap t_{i_{\text{self}}} \neq \emptyset\}, \quad targetCids = \emptyset,$$

  where i_{self} denotes the vehicle ID of the ego vehicle.

- No update required.

Protocol-3: Receive messages from all vehicles that potentially pass through the next conflict zone of the ego vehicle.

- **Construction of target sets:**

$$targetVids = \{i \in \mathcal{V} \setminus \{i_{\text{self}}\} \mid cz_{\text{next}} \in t_i\}, \quad targetCids = \emptyset,$$

  where cz_{next} and $\mathcal{V}$ are defined as previously.

- Update *targetVids* when the vehicle enters the **MOVING-CZ Passing** state.

doi:10.6342/NTU202502495

Chapter 6

Experimental Results

6.1 Experimental Settings

The experimental process begins with specifying the intersection structure and the input data of the vehicles. Each vehicle then uses its onboard intersection manager to construct a TCG and applies the fixed Type-3 edge removal algorithm, as described in Section 4.3. During simulation, all vehicles operate in synchronized time steps of 0.1 seconds and follow the same protocol. In each step, every vehicle sequentially performs all or part of the following tasks:

1. Update its internal state, including vehicle state, *targetVids*, *targetCids*, etc.
2. Broadcast a message.
3. Receive and filter messages from other vehicles.
4. Update its local vertex state list based on the accepted messages.
5. Determine whether to move.

We design three experiments based on the proposed framework and protocols.

- Evaluate the performance of the proposed protocols on various scenarios.
- Examine the impact of message loss on each protocol.

doi:10.6342/NTU202502495

- Investigate the robustness of our distributed intersection management framework in the presence of message loss.

Our experimental settings include the combinations of the following parameters:

- **Simulation Duration:** The earliest arrival times are distributed over a range of 60 seconds.
- **Traffic Density (λ)** [8]: Vehicles are generated uniformly across all source lanes. The vehicle arrival process on each lane follows a Poisson distribution with parameter λ, which reflects traffic density and can be considered approximately as the average number of vehicles arriving per second.
- **Conflict-Zone Granularity (n_{CZ}):** We consider two levels of granularity in conflict-zone division, called the 4CZ and 16CZ configurations, as shown in Figure 3.2.
- **(Edge Waiting Time, CZ Passing Time):** Based on the assumption that the passing times of straight-crossing vehicles remain consistent across different n_{CZ} values, the pair is set to (0.4, 1.4) for the 4CZ configuration and (0.4, 0.32) for the 16CZ configuration.
- **Drop Rate:** From the perspective of a given vehicle, each message broadcast by any other vehicle has a probability p_{drop} of being lost.
- **Protocols:** In addition to Protocols 1–3, we define two benchmarks:
  - **Benchmark-1 (BM-1):** All vehicles cross the intersection strictly in the order of earliest arrival time. Each vehicle may enter the intersection only after the previous has completely exited.

doi:10.6342/NTU202502495

– **Benchmark-2 (BM-2):** A baseline protocol (Protocol-0) used for comparison. No filtering is applied.

We categorize our evaluation metrics into two groups: robustness and performance. The following metrics are used [8]:

- **Leaving Time of the Last Vehicle (T_L):** The total time required for all vehicles to pass the intersection.
- **Average Delay Time (T_D):** Average delay experienced by all vehicles, calculated as the difference between the actual passing time and the ideal (unblocked) passing time. The ideal passing time is defined as the time it would take a vehicle to pass through the intersection without any interference.
- **Accepted Message Count (AMC) per vehicle:** The average number of V2V messages accepted (i.e., not filtered) and processed by each vehicle during simulation.
- **Failure Rate:** The percentage of simulations in which permanent stalling occurs.

Each parameter setting is evaluated using 100 simulated traffic flows based on the arrival rate (λ) and simulation duration. The results are averaged over all runs. Simulations are implemented in Python and run on a 64-bit Windows system with an AMD Ryzen 7 5700U (1.80 GHz) and 16 GB RAM.

6.2 Simulation Results

6.2.1 Study 1: V2V Protocol Comparison

This study evaluates three V2V communication protocols and two benchmarks (Protocols 1–3 and Benchmarks 1–2) under ideal conditions without loss of

doi:10.6342/NTU202502495

message. We consider three traffic densities (0.8, 1.2 and 1.6 vehicles/sec) and two CZ granularities (4 and 16 zones).

For each setting, we generate 100 traffic flows. For each flow, we construct the corresponding TCG, apply the type-3 edge removal algorithm, configure the message filtering based on the protocol, and run a single simulation. The metrics are then computed based on the results aggregated over the 100 simulations.

Tables 6.1, 6.2, and 6.3 summarize the results for T_L, T_D, and the average AMC per vehicle, respectively. As discussed in Section 5.3, permanent stalling is prevented in ideal conditions, ensuring that all vehicles eventually pass through the intersection in a finite time. The results for T_L and T_D show the following trend:

- In BM-1, both T_L and T_D are significantly higher compared to other methods, indicating that our intersection management framework effectively improves efficiency by finer CZ division.

- Protocols 2 and 3 perform equivalently to BM-2 in terms of time-related metrics, despite applying message filtering. It suggests that many messages under this framework may be non-essential to improving efficiency.

- Protocol 1 has obvious higher T_L and T_D and the lowest AMC per vehicle, reflecting the trade-off between efficiency and message overhead.

- At the same arrival rate λ, 16CZ significantly outperforms 4CZ in T_L and T_D, suggesting improved traffic efficiency with a finer CZ division.

- With the exception of BM-1, T_L increases much slower than T_D as λ increases, demonstrating that the spatial utilization of the intersection improves alongside the higher traffic density.

doi:10.6342/NTU202502495

Table 6.1: Average T_L (in seconds) under different protocols and benchmarks. μ_m denotes the average number of vehicles.

λ	μ_m	BM-1		BM-2		Protocol-1		Protocol-2		Protocol-3	
		4CZ	16CZ	4CZ	16CZ	4CZ	16CZ	4CZ	16CZ	4CZ	16CZ
0.8	47.1	201.4	187.0	110.2	72.4	113.2	79.8	110.2	72.4	110.2	72.4
1.2	70.6	296.3	278.5	167.0	100.2	171.6	116.0	167.0	100.2	167.0	100.2
1.6	97.4	403.5	381.5	229.0	136.4	235.0	158.6	229.0	136.4	229.0	136.4

Table 6.2: Average T_D (in seconds) under different protocols and benchmarks.

λ	μ_m	BM-1		BM-2		Protocol-1		Protocol-2		Protocol-3	
		4CZ	16CZ	4CZ	16CZ	4CZ	16CZ	4CZ	16CZ	4CZ	16CZ
0.8	47.1	67.8	62.0	22.9	5.7	24.3	8.8	22.9	5.7	22.9	5.7
1.2	70.6	116.9	108.1	49.7	17.7	51.8	24.8	49.7	17.7	49.7	17.7
1.6	97.4	172.4	160.6	80.3	34.9	83.3	45.5	80.3	34.9	80.3	34.9

Regarding the average AMC per vehicle, 16CZ generally results in lower message counts than 4CZ. However, as λ increases, the difference narrows. The reduced waiting time before intersection is offset by a more complex TCG, which increases the time required to meet movement requirements. Consequently, vehicles spend more time in conflict zones under the 16CZ configuration. Protocols 1 and 3 are more sensitive to n_{CZ} than Protocol-2, probably because Protocol-2 uses fixed filtering, while the others rely on state-based updates. Protocol-1 achieves the lowest AMC per vehicle, and although Protocol 3 has a lower AMC than Protocol 2, it requires dynamic updates, increasing computational cost.

Overall, the distributed intersection management system ensures that all vehicles complete their tasks under ideal conditions. Each protocol shows different trade-offs: Protocol 1 minimizes message overhead; Protocols 2 and 3 match the efficiency of no-filtering benchmarks in T_L and T_D, but involve trade-offs between message overhead and filtering update cost.

doi:10.6342/NTU202502495

Table 6.3: Average AMC per vehicle under different protocols and benchmarks.

λ	μ_m	BM-1		BM-2		Protocol-1		Protocol-2		Protocol-3	
		4CZ	16CZ	4CZ	16CZ	4CZ	16CZ	4CZ	16CZ	4CZ	16CZ
0.8	47.1	-	-	3243	2387	1524	852	2700	1920	2160	1139
1.2	70.6	-	-	5364	4845	2399	1440	4414	3750	3497	2017
1.6	97.4	-	-	7622	7456	3370	2757	6222	5647	4936	2948

6.2.2 Study 2: Impact of Message Drop

The previous study assumes an ideal environment with a drop rate of zero, i.e., no message loss. Although this setting is useful for highlighting the characteristics of different configurations and V2V protocols, it does not reflect real-world conditions. In this study, we introduce message loss by varying the drop rate to simulate environments with different levels of communication reliability and observe how each protocol performs under these conditions.

We focus on the 4CZ configuration and fix the vehicle arrival rate λ at 1.2. Message loss is modeled by setting a drop rate p_{drop}, which means that from a vehicle's perspective, any message has a probability p_{drop} of being lost. We test four drop rate settings: 0, 0.25, 0.5, and 0.75. All other parameters and settings are consistent with those used in Study 1.

The results are presented in Tables 6.4 and 6.5. Since message loss may lead to permanent stalling, table entries are gray if any of the 100 simulations in that setting resulted in stalling. Furthermore, any simulations affected by permanent stalling failures are omitted from the metric computations.

In general, drop rates between 0 and 0.5 cause only slight increases in T_L and T_D, indicating that moderate message loss has a limited impact on efficiency. However, at sufficiently high drop rate (e.g. 0.75), permanent stalling may occur, leading to task failure.

doi:10.6342/NTU202502495

Table 6.4: Average T_L and T_D (in seconds) at different drop rates.

p_{drop}	BM-1		BM-2		Protocol-1		Protocol-2		Protocol-3	
	T_L	T_D	T_L	T_D	T_L	T_D	T_L	T_D	T_L	T_D
0.0	296.3	116.9	167.0	49.7	171.6	51.8	167.0	49.7	167.0	49.7
0.25	-	-	168.2	50.3	172.8	52.4	168.2	50.3	168.3	50.3
0.50	-	-	170.6	51.5	175.4	53.6	170.7	51.5	170.8	51.5
0.75	-	-	165.3	48.5	151.5	43.3	172.0	51.7	171.7	55.2

Table 6.5: Average AMC per vehicle at different drop rates.

p_{drop}	BM-1	BM-2	Protocol-1	Protocol-2	Protocol-3
0.0	-	5364	2399	4414	3497
0.25	-	4043	1805	3328	2635
0.50	-	2727	1211	2240	1774
0.75	-	1277	504	1099	907

For any protocol, the AMC per vehicle approximately follows the relationship:

$$\text{AMC} \approx \text{AMC}_0 \times (1 - p_{\text{drop}}),$$

where AMC_0 is the AMC value in the same setting but $p_{\text{drop}} = 0$. In addition, variations in the drop rate affect all protocols in a similar manner, with no obvious differences observed.

6.2.3 Study 3: Robustness of the System

We also evaluate the relationship between the drop rate and the failure rate. In this study, we focus solely on BM-2 (a no-filtering protocol) and vary the drop rate from 0.55 to 0.75 in increments of 0.025 to examine its effect on system robustness.

Figure 6.1 shows the failure rate as a function of the drop rate. Permanently stalled vehicles begin to appear when the drop rate reaches approximately 0.6. From 0.65 to 0.75, the failure rate increases almost linearly. These results suggest that the system remains robust without filtering up to a drop rate of around 0.55.

doi:10.6342/NTU202502495

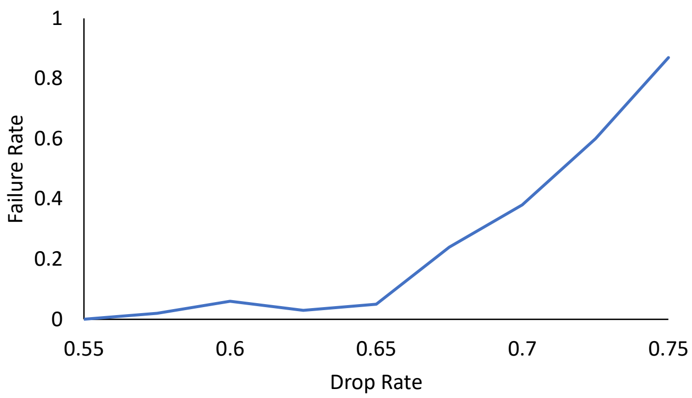


Figure 6.1: Failure rate vs. drop rate under the BM-2 protocol.

6.3 Discussion

The experimental results demonstrate that the distributed intersection management framework exhibits robustness against message drop rates. Under the 16CZ configuration, the protocols achieve acceptable delay performance, although Protocol 1 shows lower efficiency. Each protocol presents trade-offs among message overhead, delay, and computational cost.

The results confirm that the CZ granularity strongly affects the system performance. As expected, 16CZ offers greater flexibility and higher throughput, leading to a shorter delay. However, the total message count grows more rapidly in 16CZ than in 4CZ, as the time that vehicles spend in the intersection grows faster with increasing traffic density. This leads to a faster increase in the broadcast duration and the message count with increasing λ. The increased message overhead reduces scalability. We conclude that n_{CZ} is a key factor influencing traffic efficiency in distributed intersection management.

doi:10.6342/NTU202502495

Chapter 7

Conclusion

In summary, this thesis presents novel frameworks for distributed intersection management with only V2V and for the design of a firewall-style V2V protocol. Based on these frameworks, we propose and implement three V2V communication protocols along with two baseline benchmarks. A comprehensive evaluation is conducted to analyze their performance in terms of traffic efficiency, message overhead, and computational cost.

Our results show that some filtering-based protocols deliver time-related performance comparable to the non-filtering benchmark, indicating that many messages may have limited impact on traffic efficiency in the proposed framework. Moreover, finer CZ granularity consistently improves performance, enabling higher throughput and lower delays. Specifically, with the 16CZ configuration, the proposed protocols maintain acceptable delays even at high traffic density (e.g., 1.6 vehicles/sec), while in 4CZ, the delay is often too high to be considered acceptable by drivers. The framework also exhibits robustness to message loss, tolerating drop rates up to 0.55, highlighting its potential in less-than-ideal communication environments.

In future work, the proposed framework can be extended to support real-time or online scheduling with minimal modifications, as each vehicle's movement requirements depend only on vehicles with earlier earliest arrival times. This property

doi:10.6342/NTU202502495

also enables the development of more stringent rules to filter messages. Although the lack of standardized assumptions (e.g., edge waiting times or CZ crossing durations) limits direct comparisons with centralized approaches, existing scheduling algorithms could be integrated and evaluated within this distributed framework. As another future extension, relaxing movement constraints—for example, allowing moves as long as no target vertex is `NON_EXECUTED` instead of requiring all to be `EXECUTED`—may violate the topological order for vertex execution. In such cases, deadlock detection and cycle-removal algorithms can be introduced to maintain safe and efficient coordination.

doi:10.6342/NTU202502495

Bibliography

[1] H. Ahn and D. Del Vecchio, “Safety verification and control for collision avoidance at road intersections,” *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 630–642, 2018.

[2] M. Cederle, M. Fabris, and G. A. Susto, “A distributed approach to autonomous intersection management via multi-agent reinforcement learning,” 2024. [Online]. Available: https://arxiv.org/abs/2405.08655

[3] D. Gadginmath and P. Tallapragada, “Data-guided distributed intersection management for connected and automated vehicles,” in *2022 American Control Conference (ACC)*, pp. 767–774, 2022.

[4] A. Gholamhosseinian and J. Seitz, “Pcims: Plenary centralized intersection management scheme for heterogeneous connected vehicles,” in *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pp. 1–6, 2023.

[5] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, “Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.

[6] K.-D. Kim and P. R. Kumar, “An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic,” *IEEE Transactions on*

doi:10.6342/NTU202502495

Automatic Control, vol. 59, no. 12, pp. 3341–3356, 2014.

[7] R. Krajewski, P. Themann, and L. Eckstein, "Decoupled cooperative trajectory optimization for connected highly automated vehicles at urban intersections," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 741–746, 2016.

[8] X. Liang, T. Yan, J. Lee, and G. Wang, "A distributed intersection management protocol for safety, efficiency, and driver's comfort," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1924–1935, 2018.

[9] K.-E. Lin, K.-C. Wang, Y.-H. Chen, L.-H. Lin, Y.-H. Lee, C.-W. Lin, and I. H.-R. Jiang, "Graph-based deadlock analysis and prevention for robust intelligent intersection management," *ACM Trans. Cyber-Phys. Syst.*, vol. 8, no. 3, Jul. 2024. [Online]. Available: https://doi.org/10.1145/3632179

[10] Y.-T. Lin, H. Hsu, S.-C. Lin, C.-W. Lin, I. H.-R. Jiang, and C. Liu, "Graph-based modeling, scheduling, and verification for intersection management of intelligent vehicles," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, Oct. 2019. [Online]. Available: https://doi.org/10.1145/3358221

[11] A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang, "A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections," *Automatica*, vol. 93, pp. 244–256, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109818301511

[12] T. Ormándi, B. Varga, and T. Tettamanti, "Distributed intersection control based on cooperative awareness messages," in *2021 5th International Conference on Control and Fault-Tolerant Systems (SysTol)*, pp. 323–328, 2021.

doi:10.6342/NTU202502495

[13] X. Pan, B. Chen, S. Timotheou, and S. A. Evangelou, “A convex optimal control framework for autonomous vehicle intersection crossing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 163–177, 2023.

[14] X. Qian, J. Gregoire, A. de La Fortelle, and F. Moutarde, “Decentralized model predictive control for smooth coordination of automated vehicles at intersection,” in *2015 European Control Conference (ECC)*, pp. 3452–3458, 2015.

[15] G. Rodrigues de Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sjöberg, “Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 8–21, 2017.

doi:10.6342/NTU202502495