FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Cooperative, connected and autonomous mobility: a case study of platoon coordination at intersections using market-based mechanisms

Francisco Filipe



Mestrado Integrado em Engenharia Informática e Computação

Supervisors: Pedro M. d'Orey, PhD

Second Supervisor: Zafeiris Kokkinogenis

July 29, 2021

Cooperative, connected and autonomous mobility: a case study of platoon coordination at intersections using market-based mechanisms

Francisco Filipe

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Ana Paula Rocha
External Examiner: Nélson Rodrigues, PhD
Supervisor: Pedro M. d'Orey, PhD
Second Supervisor: Zafeiris Kokkinogenis

July 29, 2021

Abstract

Recent advancements in Intelligent Transportation Systems have endowed vehicles with autonomous driving capabilities as well as V2X communication. Vehicular networks allow vehicles to augment their perception range beyond *Line-of-Sight* conditions, enabling novel strategies for effective conflict resolution in challenging scenarios (e.g. platooning) by making use of Multi-Agent Systems (MAS) coordination mechanisms (e.g. auctions). However, the deployment of such cooperative approaches should previously be tested in proper conditions, as in specific simulation environments. To this end, simulation-based evaluation of the coordination mechanisms should consider real-world constraints, such as kinematics and communication constraints. In this dissertation, a decentralised reservation-based mechanism is proposed for intersection management for connected autonomous vehicles in a platooning context. The designed approach introduces bidding as (i) an alternative to the First-Come-First-Serve policy for the decision making process of assigning reservations to vehicles and (ii) as a conflict resolution strategy between conflicting reservations. The distributed nature of the proposed mechanism forwards most of the computational cost from the intersection manager to the driving agents, thus improving scalability. Following a review of the state of the art and the proposed solution's modelling, an integration between a microscopic traffic simulator, a network simulator and a MAS framework is presented. The development of the designed intersection management strategy allows for an in-depth qualitative analysis of its performance against traditional intersection management strategies such as traffic light systems. Simulation results show promising improvements in terms of throughput with an increase of up to 75% when compared to traffic light systems while maintaining at least half of the experienced delay. Smaller improvements are also observed when comparing the designed negotiation based approach with a First-Come-First-Serve (FCFS) approach. Further improvements can however be made in the developed bidding strategy due to the developed bidding functions being prone to overbidding and inconsistencies in the returned values.

Resumo

Avanços recentes em Sistemas de Transporte Inteligentes dotaram veículos com capacidades de condução autónoma, bem como comunicação V2X. Redes de veículos permitem que os veículos aumentem o seu nível de percepção para além do seu campo de visão, permitindo novas estratégias para a resolução eficaz de conflitos em cenários mais complexos (i.e. *platoons*), recorrendo a mecanismos de coordenação Sistema Multi-Agente (MAS) (i.e. leilões). No entanto, o uso deste tipo de abordagens cooperativas deve ser testado previamente em condições adequadas, como em ambientes de simulação específicos. Para este fim a avaliação, com base em simulação, dos mecanismos de coordenação deve ter em consideração restrições cinemáticas e de comunicação realistas. Nesta dissertação, um mecanismo descentralizado baseado em reservas é proposto para a coordenação de veículos autónomos numa interseção em contexto de *platoons*. A abordagem pensada apresenta um mecanismo com base em leilões como (i) uma alternativa à política *First -Come-First-Serve* (FCFS) para o processo de atribuição de reservas a veículos e (ii) como uma estratégia de resolução de conflitos entre reservas de diferentes veículos. A natureza distribuída do mecanismo proposto reparte a maior parte do custo computacional do mecanismo de coordenação nos agentes de condução (veículos), reduzindo a carga computacional ao nível da interseção e melhorando assim a escalabilidade. Após uma revisão do estado da arte e da modelação da solução proposta, é apresentada uma integração entre um simulador de trânsito, um simulador de rede e uma framework multi-agente (MAS). O desenvolvimento da estratégia de coordenação definida permite uma análise qualitativa detalhada do seu desempenho em relação às estratégias atualmente utilizadas, como por exmplo os semáforos. Os resultados da simulação mostram melhorias promissoras em termos de volume trânsito dentro da interseção com um aumento de até 75 % quando comparado a semáforos. O atraso sentido ao nível da interseção também é alvo de uma redução significativa sendo no máximo metade do atraso existente com o uso de semáforos. A solução desenvolvida também demonstra resultados promissores em relação a uma abordagem FCFS, contudo com uma diferença de performance bastante menor do que aquela observada ao nível dos semáforos.No entanto, outras melhorias podem ser feitas na estratégia de *bidding* desenvolvida uma vez que as funções desenvolvidas podem dar aso a overbidding e inconsistências nos valores retornados.

Acknowledgements

It is not often that one has the opportunity to publicly express gratitude towards the people that impact our lives. Given the importance of this Chapter, I will proceed its writing in Portuguese, since I believe it conveys the message in the most meaningful way.

Aos meus orientadores Pedro d'Orey e Zafeiris Kokkinogenis pelo apoio prestado ao longo de todo o projeto. A prontidão das suas respostas no esclarecimento de dúvidas, bem como as palavras de positivismo e encorajamento dirigidas, em muito contribuíram para a realização desta dissertação.

A todos os meus amigos e colegas que me acompanharam ao longo deste percurso académico. As memórias dos bons momentos que partilhámos certamente não serão esquecidos.

À Liliana e à Mariana, por me mostrarem que há pessoas para as quais a definição de amizade parece ser pequena. Nesta jornada alimentada a cafeína e a quantidades de açúcar suficientes para levar qualquer pessoa a um estado pré-diabético, não podia ter escolhido melhores companheiras. Não sei o que o futuro nos reserva, mas se somos à prova de pandemia, somos à prova de tudo. Vocês fazem com que toda esta caminhada seja muito mais que um diploma.

Aos meus pais, por nunca duvidarem de mim e sempre me apoiarem ao longo de toda a minha vida. Sem vocês, nada disto seria possível, e serei eternamente grato por me terem dado esta oportunidade. Agradeço também ao meu irmão, que para além de companheiro de vida, partilhou comigo grande parte deste percurso académico e fez com que tudo ficasse mais fácil.

Em especial ao Ricardo Fazeres, pelo apoio e motivação constante ao longo deste percurso, principalmente nos momentos mais difíceis.

A todos, o meu muito obrigado

Francisco Filipe

"As long as there's a 1% chance,
we'll have 99% faith"

Neymar Júnior

Contents

1 Introduction **1**
- 1.1 Contextualization . . . 1
- 1.2 Motivation . . . 1
- 1.3 Problem Definition . . . 2
- 1.4 Contributions and Goals . . . 3
- 1.5 Document Structure . . . 3

2 State of the Art **5**
- 2.1 Introduction to Coordination Problems . . . 5
  - 2.1.1 Taxonomy . . . 6
  - 2.1.2 Coordination problems in transportation domain . . . 7
- 2.2 Coordination Approaches . . . 8
  - 2.2.1 Negotiation Mechanisms . . . 10
  - 2.2.2 Social choice - Voting . . . 11
- 2.3 Platooning . . . 12
  - 2.3.1 Platoon Formation . . . 13
- 2.4 BDI Agent Modeling . . . 14
  - 2.4.1 Vehicles as BDI Agents . . . 16
- 2.5 Related Works . . . 17
  - 2.5.1 Intersection Management . . . 17
  - 2.5.2 Lane Merging . . . 22
- 2.6 Summary . . . 23

3 Intersection Management **25**
- 3.1 Traffic Model . . . 25
  - 3.1.1 Intersection . . . 26
  - 3.1.2 Vehicles . . . 26
- 3.2 Bidding Protocol . . . 27
  - 3.2.1 Bid Calculation . . . 27
  - 3.2.2 Bid Payments . . . 29
- 3.3 Intersection Reservation Mechanism . . . 29
  - 3.3.1 Vehicle Protocol . . . 30
  - 3.3.2 Intersection Manager Protocol . . . 36
- 3.4 Summary . . . 38

4 System Architecture **41**
- 4.1 Simulation Architecture . . . 41
  - 4.1.1 Eclipse MOSAIC - mobility simulation framework . . . 42

4.1.2 SUMO - road traffic simulator . . . 43
4.1.3 SNS - simple network simulator . . . 44
4.1.4 Jadex - MAS Framework . . . 46
4.2 Communication . . . 47
4.2.1 Internal Communication . . . 47
4.2.2 V2X Communication . . . 51
4.2.3 Federate Communication . . . 53
4.3 Summary . . . 55

5 Results and Discussion **57**
5.1 Evaluation metrics . . . 57
5.2 Platooning . . . 58
5.2.1 Platoon Generation . . . 59
5.3 Scenarios . . . 59
5.3.1 Single Intersection . . . 60
5.3.2 Intersection Network . . . 60
5.4 Results Analysis . . . 61
5.4.1 Single Intersection . . . 61
5.4.2 Intersection Network . . . 66
5.5 Summary and Discussion . . . 68

6 Conclusions **69**
6.1 Contributions . . . 69
6.2 Future Development . . . 70
6.3 Future Research . . . 70

References **71**

List of Figures

2.1 Coordination Approaches according to degree of autonomy [MCZ20]. 9
2.2 Reasoning process of BDI agents [CGG+17]. 15
2.3 Auction process at an intersection [CBS13]. 19
2.4 Collision avoidance between a vehicle and an intersection obstruction 21

3.1 Intersection modelled as a matrix of reservation tiles 26
3.2 Intersection path connections example 31
3.3 Bresenham's algorithm result for path prediction 32
3.4 Conflict occurrence in a reservation-tile 33
3.5 Non-Conflict occurrence in a reservation-tile 34
3.6 Vehicle's conflict resolution process 35
3.7 Vehicle's BDI Flow 37
3.8 Reservation flow between the Intersection Manager and Vehicle Manager 39

4.1 System architecture 42
4.2 Eclipse MOSAIC Run-time Infrastructure 43
4.3 SUMO connected to Eclipse MOSAIC [FOK99] 45
4.4 SUMO connected to Eclipse MOSAIC 47
4.5 Message components and messages exchanged between vehicle applications . . . 49
4.6 Message flow between vehicle applications 50
4.7 Message flow between intersection applications 50
4.8 V2X message flow between vehicles and intersection manager 52
4.9 Interaction flow between a vehicle application and the Jadex Federate 54

5.1 Single intersection 60
5.2 Intersection network scenario layout 61
5.3 Vehicle demand analysis 62
5.4 Vehicle demand analysis 63
5.5 Time to acquire reservation (*milliseconds*) 64
5.6 Platoon maximum size analysis 65
5.7 Bid values with respect to spare time 65
5.8 Vehicle routes 67
5.9 Bid value distribution 67

List of Tables

2.1 Characterization of Coordination Problems 7

4.1 Communication channels and frequencies for V2X communication 45

5.1 Vehicle demand used in the experiments 61

Abbreviations

AIM	Autonomous Intersection Management
BDI	Belief,Desires and Intentions
CAV	Connected Autonomous Vehicle
CHV	Connected Human-operated Vehicle
DCOP	Distributed Constraint Optimisation Problem
ETA	Estimated Time of Arrival
FCFS	First Come First Serve
HDV	Heavy Duty Vehicle
IM	Intersection Manager
IEEE	Institute of Electrical and Electronics Engineers
ITS	Intelligent Transport Systems
IVC	Inter Vehicular Communication
MAS	Multi-Agent System
MOSAIC RTI	MOSAIC Run-Time Interface
MTA	Maximum Time of Arrival
PDC	Platoon-based Delay Minimization
PVM	Platoon-based Variance Minimization
RTI	Run-Time Infrastructure
SCA	Service Component Architecture
SNS	Simple Network Simulator
SUMO	Simulation of Urban MObility
TraCI	Traffic Control Interface
V2I	Vehicle to infrastructure
V2V	Vehicle to vehicle
VSimRTI	V2X Simulation Run-Time Infrastructure
XML	eXtensible Markup Language

Chapter 1

Introduction

1.1 Contextualization

Human errors are the leading cause of traffic-related accidents occurring nowadays. Every day, road accidents become a subject matter on the news as it bears vast repercussions. The rise of the number of vehicles in traffic through the years has only worsened this situation. Withdrawing the responsibility from the human driver could be the solution for safer roads. Hence, automation is slowly being introduced by the automobile industry as a more enjoyable and safer way of traveling.

Vehicles are already capable of attaining a certain degree of autonomy. Manufacturers began introducing autonomous features in their vehicles in the last years. This is a growing tendency, as more and more vehicles will be launched with new technologies capable of augmenting their levels of autonomy [ZSSL17]. However, autonomy at an individual scope does not improve traffic flow since it does not account for the rest of the vehicles in traffic unless when avoiding collisions. By adding vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication, vehicles can then be perceived as Connected Autonomous Vehicles (CAVs) capable of sharing information beyond *Line-of-Sight* conditions, thus improving the efficiency and safety of vehicular networks [LM18].

Vehicular coordination happens when vehicles account for others' actions when in the process of decision making. Cooperation between vehicles is one fundamental concept of vehicular coordination and can be used to have vehicles achieve their individual goals. Platooning is one form of coordination as vehicles keep a close and coordinated formation over time (at a safe distance) without any mechanical linkage. It is a useful mechanism for reducing traffic congestion and improving traffic throughput [MWK17]. Additionally, there are other ways of achieving coordination between vehicles, such as lane merging [SSSM19] or intersection management [DS08].

1.2 Motivation

The number of vehicles circulating on the road has increased throughout the years. As this number grows, the road becomes more congested, fuel consumption rises, and hours are unneces-

sarily spent in traffic. Furthermore, "Ninety percent of all accidents are caused by human factor. The remaining ten percent are other causes mostly combined with the human factor." [LB17]. Human-related errors are the most frequent cause of accidents. Having autonomous collaborative entities instead of humans controlling the vehicles could not only increase road safety but also improve fuel consumption, traffic flow and the overall traveling experience.

Previous studies in vehicular platoons [ZSFB95] claim a decrease in the drag coefficient by almost half when comparing to non-platooning vehicles. This drag reduction allows vehicles to travel at the same speed they would if not platooning, but consuming less fuel and producing fewer emissions. Another advantage of vehicle platoons is the increasing capacity and natural traffic flow from vehicles traveling in platoon formations.

Regarding negotiation mechanisms, market-based strategies have been studied in traffic research, but only to an extent. Research has only scratched the surface in what concerns platoon formation. Not only that, but some other scenarios such as intersection management have not yet been explored to their fullest potential. The contributions made fail to consider the existence of platoons at intersections, by only focusing on independent vehicles participating in the coordination process.

1.3 Problem Definition

- P - Platooning strategies are mostly explored in simpler scenarios with a low degree of complexity. Collaboration between vehicles in more challenging scenarios does not account for platooning. When it comes to intersection management, platooning is somewhat disregarded as most coordination strategies only take into account individual vehicles.

Tackling the problem at hand involves modeling intersections as infrastructures capable of accepting autonomous vehicles. Furthermore communication capabilities between autonomous vehicles as well as existing road side infrastructures need to be assessed. As platoons are established, the study of their impact in intersection management scenarios presents itself as another problem to tackle. Finally, the study of negotiation mechanisms in an intersection management context is addressed.

The research questions that will primarily be the focus of this dissertation are the following:

- Q_1 - How does platooning improve intersection management when compared to non-platooning approaches?
- Q_2 - How does the developed intersection management strategy perform against traditional forms of traffic coordination at intersections?
- Q_3 - How does applying negotiation mechanisms improve intersection management when compared to First Come First Serve (FCFS) approaches?

In Q_1, a thorough study is developed to asses how vehicle collaboration in platooning maneuvers affects the vehicle's performance at intersections. Various test scenarios are implemented

to evaluate how the time delay, traffic flow, safety and any other relevant metrics perform in the collaboration process. Moreover, Q_2 focuses on comparing the developed coordination strategy with traditional coordination mechanisms such as traffic lights. Lastly, Q_3 aims on comparing how negotiating strategies can provide some sense of fairness to the whole intersection management system. Each coordination strategy's performance will be assessed in similar scenarios. Once all the data is gathered from the simulations, a comparison is conducted based on the retrieved observations.

1.4 Contributions and Goals

This dissertation aims to implement, validate and analyze market-based (auction) mechanisms in a MAS transportation system. For the implementation, the dissertation focuses on one main coordination scenario, vehicle management at an intersection level. From the literature review made in chapter 2, we observed that the state-of-the-art vastly explored market-based negotiation strategies such as auctions when it comes to intersection management. However, its research focuses on managing independent vehicles in an intersection. Therefore, an introduction to market-based negotiation strategies in an intersection with both independent vehicles and platoons' co-existence is one contribution of this work.

To do so, another contribution of this work is the design and implementation of a simulation platform to evaluate the intersection management strategy proposed.

As a third contribution, this dissertation performs a comparative analysis of the developed coordination strategy when compared to both a traditional intersection management strategies and a FCFS approach, as well as the benefit of incorporating platoons in such scenarios.

1.5 Document Structure

The remainder of this document is organized as follows. In chapter 2, relevant concepts to the project are presented, as well as an overview of the current literature review. Research trends are identified and a gap analysis presents the purpose of this dissertation. Chapter 3 includes a detailed description of the developed intersection management strategy, giving insights on how the solution was modelled and the theory and logic behind it. Moreover, chapter 4 focuses on presenting the architecture and technologies used to implement the solution as well as describing how the different components communicate with each other. The results of the developed solution are presented and discussed in chapter 5. Lastly, chapter 6 summarizes the document's content and forecasts any expected outcomes from the work to be done.

Chapter 2

State of the Art

In the following chapter, a thorough presentation of the literature review in vehicle coordination through negotiation mechanisms in a MAS system is made. As such, an introduction to coordination problems is presented in section 2.1, which will focus on detailing key concepts about coordination problems as well as describing the different traffic problems that can benefit from the use of solutions based on the coordination between vehicles. Section 2.2 focuses on presenting the literature review on general coordination solutions to the problems uncovered earlier with particular attention to negotiation mechanisms based on auctioning strategies. Moreover, section 2.3 investigates the literature on vehicle platooning, focusing mainly in platoon formation strategies. Regarding 2.4 addresses BDI agent modeling and how vehicles have previously been modeled as BDI agents in a MAS system. Related works focusing on other coordination problems besides platooning, such as intersection management, lane merging and more, are detailed in section 2.5. A summary of what was reviewed and a gap analysis towards the literature is presented in section 2.6.

2.1 Introduction to Coordination Problems

Current research in autonomous vehicles has mainly focused on developing methods and tools that enable autonomy to reach the road safely at an individual level. In fact, some vehicles already contain a self-driving capability that does not require human interaction. Although still a work in progress, it is a step forward for the industry of autonomous vehicles. However, to fully take advantage of the benefits of autonomous vehicles (i.e., better traffic flow, less pollution, and fuel consumption), some scenarios may require coordinating vehicles' movements. Coordination problems include situations in which the interests of agents coincide, and the aim is to try to reach an outcome in which those interests are satisfied. Through communication and cooperation between the involved agents, a solution that meets every agent's requirements with the given information can be achieved. Some examples of coordination amongst a group of vehicles include intersection crossing, entering a freeway, platooning, ride-sharing, trying to improve parking occupancy, among many others. Effectively supporting such coordination needs implies devising effective mechanisms and strategies to support coordination activities.

In this section, a presentation of essential concepts to coordination problems is made. Furthermore, an analysis of the most critical coordination problems in the future is presented with particular attention to platoon formation and platooning section 2.3.

2.1.1 Taxonomy

Mariani et al. [MCZ20] define a two-dimensional taxonomy for vehicle coordination problems. In one dimension coordination problems can be split between resource-oriented and task-oriented problems.

- As the name states, resource-oriented coordination problems focus on gaining vehicle access to a shared resource for a limited period of time. Its behavior is similar to how semaphores work in concurrent systems to prevent critical sections from being accessed by multiple entities simultaneously. An excellent example of a resource-oriented coordination problem is access to a shared intersection where several vehicles can be requesting access at a time. By following previously defined strategies and rules, resource-oriented coordination problems allow access to a shared resource to be made safely.

- Task-oriented coordination problems concern allowing vehicles to achieve a particular task or goal. Although simple at a glance (i.e. driving a vehicle from point A to point B), these are complex problems and can be divided into simpler, smaller ones. Often these divisions include resource-oriented coordination problems (i.e. driving vehicle from point A to point B requires access to a shared intersection; thus a sub-problem of a task-oriented problem can be seen as resource-oriented).

On the other hand, coordination problems can also be perceived as either collaborative or competitive problems.

- Collaborative coordination problems require vehicles to work together to achieve a common goal or to accomplish their individual ones better. Behaviors like these can be observed by trucks when platooning on the highway.

- In competitive coordination problems, vehicles behave selfishly as they're only interested in fulfilling their own goals. These situations tend to include vehicles with conflicting goals; thus, a selfish behavior usually represents the vehicle's optimal solution.

Despite this two-dimensional taxonomy classification of coordination problems, when looking at possible solutions, Mariani et al. [MCZ20] outline two essential properties: safety and liveness. The former, as the name states, vows for protection during the coordination process. When it comes to vehicle coordination, guaranteeing safety during the coordination process should always be the primordial concern. If not achieved, consequences could include significant damage to the vehicles or even the loss of lives. The latter refers to how a coordination process should never allow vehicles to stay idle indefinitely and starve. Once again, vehicle coordination behaves similarly to a concurrent system, where processes should never be denied access to a resource indefinitely.

2.1.2 Coordination problems in transportation domain

Mariani et al. [MCZ20] present 6 different traffic problems as "key coordination problems" in the future: intersection management; smart parking; ride sharing; ramp merging; platooning; traffic flow optimisation.

	Intersection Management	**Smart Parking**	**Ride Sharing**	**Ramp Merging**	**Platooning**	**Traffic Flow Optimisation**
Resource vs Task	Resource	Resource	Task	Task & Resource	Task	Resource
Collaborative vs Competitive	Competitive	Collaborative & Competitive	Competitive	Collaborative & Competitive	Collaborative	Collaborative & Competitive

Table 2.1: Characterization of Coordination Problems

Table 2.1 presents each one of the coordination problems in relation to the two-dimensional taxonomy classification previously addressed.

1. **Intersection Management.** Involves the coordination of vehicles concurrently crossing an intersection. It is considered a competitive coordination problem since vehicles behave as self-interested agents "fighting" with each other for a shared resource, the intersection. In what comes to safety, the primary goal is to guarantee that no collisions occur between vehicles when crossing the intersection. Furthermore, no vehicle should wait indefinitely for permission to cross the intersection [RM17, MCZ20].

2. **Smart Parking.** Coordination of vehicles looking for a parking spot by taking into quality metrics such as the time it takes to find the parking spot, price, and distance to the desired destination [LRL17, PLP13]. When considering privately owned vehicles, it is regarded as a competitive coordination problem since there is no intention of cooperating with other vehicles as parking spots tend to be challenging to find. Looking at vehicles owned by private companies, smart parking can be perceived as collaborative. Vehicles within the same company can work together to better distribute their clients throughout the available parking spots; thus, contributing to a fairer allocation of vehicles to parking spots [MCZ20].

3. **Ride Sharing.** "Given a number of mobility requests expressed at a given time (the tasks to be performed) and a number of vehicles currently available to serve them, the system must assign vehicles to requests" [BMSZ17, MCZ20]. Once again, it is described as a competitive problem by privately owned vehicles as their only interest is to satisfy as many mobility requests as possible. Like smart parking, when vehicles are associated with a private company, they can cooperate to maximize the number of mobility requests of the company and the clients' overall satisfaction. Some examples already deployed in the real-world are Uber, Lyft, Bolt, Kapten, and Taxi companies.

4. **Ramp/Lane Merging.** Consists on coordinating vehicles when entering/exiting highway ramps or changing from one lane to another. Although it could be recognized as a resource-oriented problem, it is classified as task-oriented since the aim here is to ensure that vehicles coordinate between them, avoiding collisions and allowing the merge. Thus, this is primarily a collaborative coordination problem. Only when resources are scarce (congested traffic scenario) will this problem become resource-oriented and competitive [MCZ20].

5. **Platooning.**It occurs when vehicles behave as a single unit to achieve their own goals [JLW+16]. When in a platoon, vehicles travel at the same speed at a safe distance from each other. Platooning can be used in other coordination problems such as intersection management (i.e., permission to cross an intersection is given to an entire platoon). [MCZ20].

6. **Traffic Flow Optimisation.** Focuses on balancing the traffic to avoid congestion and jams by routing vehicles across the traffic network [SB11, ZDZ12]. It is a generic problem that aggregates all previously presented problems. Its competitive nature comes from individual vehicles trying to reach their desired destination. Nonetheless, collaboration can be integrated so that vehicles take into account that rerouting can favor everyone on achieving their goal quickly [MCZ20].

In section 2.3 we will dive into platooning coordination problems as they will be the focus of this dissertation.

2.2 Coordination Approaches

To solve most coordination problems, a global categorization can be made about the coordination solutions' architecture. Such classifications take primarily into account the degree of autonomy left to the vehicles in the decision making process. The degree of autonomy can be perceived as the level of freedom that vehicles have when deciding their action course. To make such decisions as thorough as possible, vehicles acquire information from their sensors, external infrastructures (V2I communications), or other vehicles in the coordination process (V2V communications). It is impossible to have absolute freedom in decision making as that would suggest vehicles to undergo with their course of action without accounting for the activities of other entities involved and their consequences. As such, the degree of autonomy can range from actions imposed externally (lowest degree of autonomy) to self-determined actions (highest degree of autonomy) [MCZ20]. Figure 2.1 identifies the four main classifications of coordination approaches according to their degree of autonomy.

1. **Centralized.** has the lowest degree of autonomy since the decision making in the coordination process is assigned to an external entity (i.e., a coordinator) whose conclusions on the course of action are inarguable. Vehicles do not hold any power over this entity and must abide by its decisions. A very well-known example that has been present throughout the years in the traffic system is the traffic lights. When in a traffic light, vehicles act

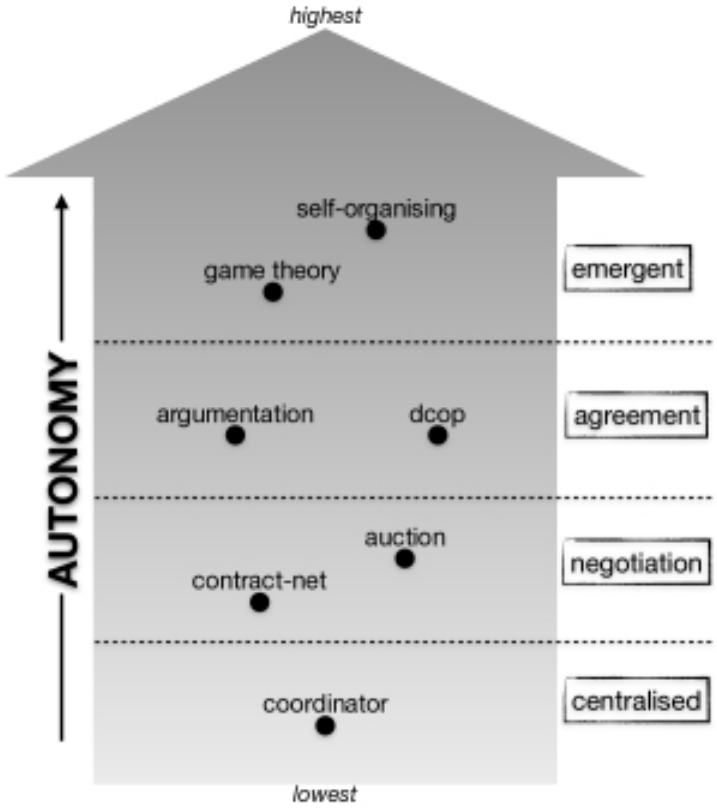


Figure 2.1: Coordination Approaches according to degree of autonomy [MCZ20].

according to its output. The traffic light does the coordination process's decision-making, and all that is left for vehicles to do is obey its decision. Although named "centralized", several distributed infrastructures could do the decision-making process. The centralized aspect of such coordination approaches is for having an external entity responsible for the coordination process's decision-making [MCZ20].

2. **Negotiation.** distributes the coordination process throughout the coordinating entities who use negotiations protocols during the decision making. In this negotiation protocol, vehicles propose a course of action according to their current information. If functioning correctly, the protocol should eventually converge towards a solution stating when and what should each vehicle do to solve the coordination problem [MCZ20].

3. **Agreement.** involves vehicles defining the goals and strategy to pursue to establish the course of action that better achieves their goals. Some examples of protocols used in agreement approaches include argumentation based approaches [RRJ+04] where vehicles argue together to reach a consensus on goals and solutions.

4. **Emergent.** defines vehicles as selfish entities who do not partake in a coordination protocol. Instead, they act based on maximizing utility functions defined either by their goals or

collected perceptions of other participants [MCZ20]. An example of such behaviour is the game theoretic approaches [NRTV07] where vehicles determine their own course of action based on informed guesses on the expected behaviour of other vehicles.

Of the four coordination approaches mentioned above, this dissertation focuses mainly on the negotiation coordination approaches, mainly focusing on auctioning strategies for vehicle coordination.

2.2.1 Negotiation Mechanisms

As previously stated, negotiation mechanisms involve a communication between vehicles following a pre-established protocol to agree upon a course of action. [Kra97]

Kraus [Kra97] identifies various parameters of primordial importance for choosing an appropriate technique to solve coordination problems. The first parameter outlined by Kraus covers the **level of cooperation** among the agents, whether they have common goals that they should work on or if they're self-motivated. Furthermore, the **use of regulations and protocols** should be agreed upon, depending on the environment. A third and fourth parameters to consider are the **number and type of agents** in the environment, whether automated agents only compose the environment or if there are people present. A last property introduced by Kraus is the **communication and computation costs**. The analysis of the previous properties in any given environment can guide the choice of the multi-entity technique applied to the specific task.

2.2.1.1 Bargaining

The traditional bargaining process consists of two main entities, a buyer and a seller, trying to reach an agreement on the value of a good or service offered by the seller. Once an agreement is made between them, a transaction can take place. In a bargaining scenario, the agreed-upon evaluation depends primarily on two factors: willingness to sell and buy. A seller may benefit from a buyer highly willing to buy a good as it is more likely to agree on higher prices. Similarly, buyers benefit from "desperate" sellers by buying at lower prices. Bargaining is more commonly seen when dealing with one-of-a-kind items (automobiles, antiques, jewelry, art, real estate, trade sales of businesses) and informal sales settings such as flea markets and garage sales. In some regions of the world, bargaining may be the norm, even for small commercial transactions.

2.2.1.2 Market Approach - Auctioning

Auctioning is a market-based coordination strategy that aims to allocate resources to agents. Auctions allow agents to specify their interest in resources through a value-based approach. It is a standard method used in real life as an economic mechanism for allocating goods and the formation of prices for those goods via a process known as bidding. Auctions are being used in a computational setting for resource allocation of the system's processing power. Auction architectures may vary according to the bidders' properties and the nature of the items to be auctioned.

Some auction variations may pose as more efficient or more profitable to the seller than others. Vickrey established a taxonomy for auctions based on how the auctioneer quotes prices and the bidder offers their bids. Four major auction types were defined [Fin].

- **Ascending-Bid Auction (English).** The most common of all, often used to sell art, wine, antiques, and other goods. An English auction starts with a reserve price established by the auctioneer, the lowest price willing to accept for the item. As bidders announce their interest at that price, the auctioneer solicits further bids by raising the price by a bid increment. This process repeats itself until no bidders are willing o up their bid any further. The auction is closed at this stage, and the final bidder receives the item at his bid price. It is described as a first-price auction as the winner pays his bid price.
- **Descending-Bid Auction (Dutch).** In contrast to the English auction, the bidder starts at a higher price than he believes the item will fetch. It then decreases the price by a decrement until a bidder claims the item. The bidder then receives the item at the cost he made the call, which consists of a first-price approach. If multiple items are up for auction, the process continues until all items are sold. Since there are never more bids than there are items, the auction process is speedy.
- **First-Price, Sealed-Bid Auction.** In this type of auction, information about other bidders' bids is not disclosed. The bidders submit bids in a sealed envelope, and the highest bid wins. It is mostly used in the financial setting.
- **Second-Price, Sealed-Bid Auction (Vickrey).** As in the previous sealed-bid auction, bidders submit sealed envelopes with their bid in one round of bid submission. However, in this case, the winner does not pay the price he bid for the item but instead pays it at the second-highest bid price.

2.2.2 Social choice - Voting

Computational social choice focuses in the combination of information from individuals into one collective perspective. Ideally, decision making through social choice strategies, such as voting, converges towards a decision that best reflects the interests or opinions of a group as a whole.

Pitt et al. [PKSA06] describe the following process of a voting procedure in a committee meeting. Initially, when a voting procedure occurs in a group, all of its members with a right to vote come together. A motion is then proposed by one of the present members and backed by another. The proposed motion is then debated among all members present so that a well formed opinion is developed within each agent about the motion. To cast a decision, the meeting **chair** calls for members in favour and against the proposed motion. According to the rules of the committee and the given votes of the present members, a decision is made towards carrying or not the motion.

Voting has seen some light in vehicle coordination. Ferreira et al. [FCVT10] explore leader election to have vehicles act as a temporary road junction infrastructure broadcasting traffic light messages to other vehicles. It nullifies the use of the traditional signalized intersection control,

increasing the traffic flow. A more tactical approach to vehicular autonomy has been developed by Sukthankar et al. [SBH02] through an agent-based framework for autonomous driving. An agent is responsible for monitoring a vehicle's sensor and can affect a set of tasks (i.e., velocity management, vehicle detection, lane detection). The agents then calculate utility functions towards the various courses of action the vehicle can take and cast a vote on the best action to take to the *voting arbiter*. The voting arbiter is then responsible for deciding what course of action to take based on the votes it receives.

2.3 Platooning

When it comes to vehicle coordination, communication between vehicles allows individual goals to be achieved through cooperating behaviors between vehicles. For example, coordinating vehicles through an intersection requires them to share their information with each other or even an external entity to decide how vehicles cross an intersection. During this process, the vehicles' movements and course of action are independent of each other as their goals within the intersection may differ (i.e., taking different exits). Although this makes sense in such a scenario, some others may benefit from vehicles joining forces and teaming up with one another.

Let's consider a group of vehicles traveling on a freeway independently from each other as they work towards reaching their destinations. Some vehicles in traffic may have similar routes or even the same destination. Due to the similarity of their objectives, joining these vehicles poses a desirable operation as it brings no harm to the involved vehicles and brings several benefits to the table. This is where platooning steps in. Platooning allows vehicles to travel together, adopting the desired speed and inter-vehicular distance within the platoon. It has been mostly used for freight transportation. However, research is now focusing on translating platooning into passenger transportation combined with Intelligent Transport Systems (ITS).

During the last decades, the number of vehicles have been increasing steadily, with a 3% growth a year of motor-vehicle sales [GHZ14]. As of 2015, individual car was considered the major transportation system with a total of 71% in comparison with the public transport system. Moreover, a growth by 8% on passenger transports occurred from 2005 to 2015 [fMT17, HD18]. These numbers not only increase the traffic congestion and road accidents but also endanger the environment as vehicles tend to be pollutant machines. In an attempt to manage such traffic growth and reduce the carbon footprint, platooning was developed.

When it comes to the literature, research highly focuses on Heavy Duty Vehicle (HDV) solutions. Most commonly, HDV solutions use a centralized entity capable of gathering information about its vehicles (i.e., the origin location, destination location, earliest departure time, and latest arrival time) and making decisions towards which vehicles should start platooning together, on what route should that happen, or what should the platoon ordering be [BAZ18]. One of the most common goals in HDV platooning is to minimize the overall consumption of fuel of all vehicles. In Liang et al. [LMJ16] solution, vehicles select platooning candidates in their neighborhood based on the potential fuel savings (closer vehicles require less fuel for the joining maneuver). For

vehicles to form a platoon, a leader must be elected and slow down its speed so that its followers can join the platoon by catching up to the leader. The goal is then to find a platooning speed that is optimal in terms of fuel consumption during the platooning stage.

Decentralized HDV platooning solutions have also been investigated . Wenzel et al. [WP17] investigate platoon formation in intersection scenarios where information is limited or missing. HDVs looking to form a platoon wait at intersections and search for incoming vehicles looking to form a platoon. Moreover, Larson et al. [LLJ15] coordinate platoon formation by adjusting arrival times at intersections. Dafflon et al. [DGC+11] use intersections as nodes for vehicles to merge, split and reconfigure platoons.

2.3.1 Platoon Formation

There's a tendency in most studies to assume the existence of platoons and, as such, research on platoon formation has only been very superficial or basic. Assuming a scenario such as freeway travelling, there is a need to define a platoon formation stage for when vehicles enter the freeway. At this stage, a vehicle travels alone until it finds another vehicle or group of vehicles to associate with, better known as en-route platoon formation. Moreover, platoon formation can also occur at the entrance of a freeway as vehicles group before leaving the ramp. The challenge here is to define how vehicles choose to partner with one another, which set of rules should be established to determine a candidate vehicle. After candidate vehicles are defined, the cars can begin performing join maneuvers to form a new platoon or link with previously established ones.

Heinovski et al. [HD18] try to find the best fitting platoon candidate for each car within a neighborhood while maximizing their similarity. To do so, they account for metrics such as speed and position of the vehicles to determine the vehicle's cost to join a platoon. Two different approaches were developed as optimization solutions. A *centralized* approach uses a central server to determine the optimal course of action. Using the cost information previously given by every vehicle, it greedily chooses which vehicles should perform a join maneuver. A *distributed* approach has vehicles executing that same greedy heuristic individually. An exchange of information between vehicles is required so that they become aware of other vehicles in their neighborhood. Given that information, vehicles can determine the optimal match in their join maneuver.

Segata et al. [SBJ+14] develop a protocol for the *JOIN* maneuvers in platooning scenarios. More specifically, joining maneuvers in the middle of existing platoons are explored. To this extent, Segata et al. divise 5 different *JOIN* scenarios, each of which accounting for various interferences that may occur during the joining stage (i.e., truck or human-driven vehicle interference obstructing the joining vehicle, communication interference - higher packet loss during communication). The implemented protocol focuses mainly on the active communication between 3 different vehicles: the leader of the platoon, the follower creating the gap in the platoon and the joining vehicle. Upon a request from the joining vehicle, the leader of the platoon assesses according to a pre-established criteria whether or not to accept the request. If so, communications between the leader and follower culminate in an opening of a gap for the joining vehicle to link with the platoon.

Hall an Chin state that "To maximize benefits, it is desirable to form platoons that are reasonably large (five or more vehicles), and it is also desirable to ensure that platoons remain intact for considerable distances." [HC05]. To do so, they modeled a platoon formation process at freeway ramps centered on grouping vehicles by destination so that the distance that a platoon travels without splitting apart is maximized. This process involves only vehicles at the freeway ramp. Hall and Chin examine the performance of 3 different strategies for forming platoons on ramps: destination groups, where platoons are grouped by lanes based on their destination; dynamic grouping, having vehicles grouped by destination but not restricted by lanes; dynamic grouping and platoon splitting, which allows platoons to split by sorting the platoon vehicles, non-increasingly, according to their destination, and thus ensuring steady leadership of the platoon. From all the strategies, destination group had the best performance with respect to platoon ratio (ratio of vehicle miles traveled to platoon miles traveled) and throughput. However, it showed higher waiting time at entrance. Furthermore, the study made by Hall and Chin elucidate on how difficult it is to maintain large platoon sizes when platoon formation is solely limited to entrance ramps.

2.4 BDI Agent Modeling

BDI architecture agents have been introduced in several agent-based modeling and simulation platforms. It was first introduced by Bratman [Bra87] as a model of human practical reasoning with the following attitudes: Belief, Desire and Intention. [GPP+70].

- **Belief (what it thinks) -** the internal knowledge the agent has about the environment. Informative component that represents environment characteristics [CGG+17].
- **Desire (what it wants) -** objectives that the agent wishes to accomplish, as well as the properties, priorities and costs of achieving them. Frequently updated during the simulation, a desire becomes fulfilled when present in the Belief base. A motivational state of the system [CGG+17].
- **Intention (what it is doing) -** The action chosen by the agent. A deliberative component of the system [CGG+17].

These three attitudes represent the knowledge base of the BDI agent model. Complementing the knowledge base, BDI models define a behavior base consisting of a **perception**, a function called every iteration responsible for updating an agent's belief and desires, and **plans**, the behaviors defined to achieve specific desires (similarly to Desires, they can have a priority).

Figure 2.2 illustrates the process of practical reasoning made by BDI agents.

The reasoning process starts by collecting the perceptions about the environment. At this stage, Beliefs and Desires may be updated. Once perceptions are gathered, the agent abides to the following set of rules [CGG+17]:

- **Intentions achieved?** If a intention is achieved, then both the intention and corresponding desire must be removed from the intention and desire bases, respectively. Furthermore, a

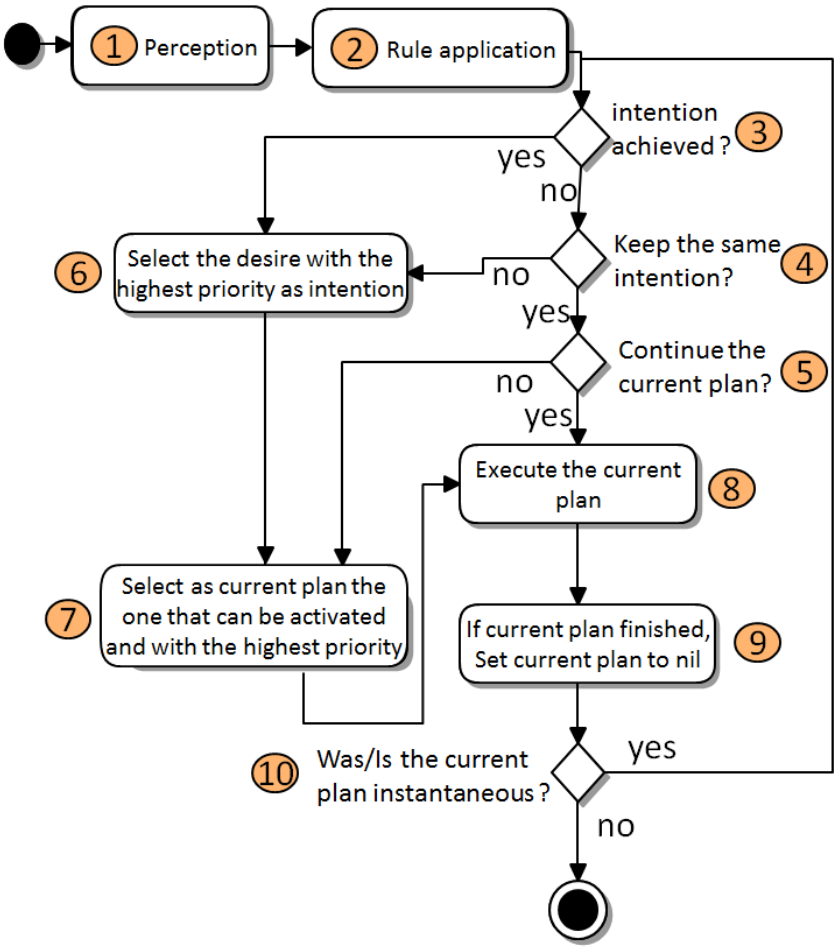


Figure 2.2: Reasoning process of BDI agents [CGG+17].

new desire must be chosen as the new current intention from the updated desire base. Since desires possess a priority, the chosen desire is usually the one with the highest priority. Based on the newly defined intention a plan is established.

- **Keep the current intention?** As the environment evolves, agents might not benefit with pursuing the current intention. The intention persistence of an agent, a coefficient applied in the reasoning process, dictates whether or not an agent should continue trying to achieve its current intention. If not, the intention is removed and, similarly to the previous rule, a new intention is chosen from the desire base and a corresponding plan is defined.

- **Continue the current plan?** A plan can also be dropped depending on a persistence coefficient. If so, once again there is a definition of a new plan based on the current intention. Otherwise, the plan can be executed.

- **Plan finished?** Persistent plans that are not dropped due to a high persistence coefficient are maintained during iterations until a termination condition is met.

To summarize the architecture and vocabulary defined, lets use the following example: a vehicle has a general desire to reach a destination. Since it is the only thing it wants at the start, it is its initial intention (what it is doing). The vehicle establishes an itinerary (its plan to reach the destination). Once it perceives intersections, it stores a new belief and creates a new desire to cross the intersections safely. As soon as the vehicle reaches an intersection, the initial intention (to reach its destination) is put on hold. A new intention is selected based on the desire to cross the intersection. To achieve this, it designs a plan with two steps, i.e. two new desires: wait for permission to cross and cross the intersection once approval is received. And so on.

2.4.1 Vehicles as BDI Agents

This section will explore modeling vehicles with a BDI agent architecture. However, some related work in this area has already been developed. Rüb et al. present an attempt to design BDI architectures for modeling a single self-driving car and a whole system of such vehicles [RDK19]. The BDI approach designed for vehicles distinguishes goals to be pursued from obligations that must be fulfilled at any instance. Furthermore, the interactions between vehicles imply the fulfillment of two rules: a CAV should try to optimize its travel time; it must only help others in their attempt to drive faster if no conflict of interests is found. Such is the platooning case, where all of the vehicles but the leader benefit from traveling together. Since leading a platoon has no disadvantages, a vehicle must agree to it when asked. For the BDI vehicle model to become a reality, Belief, Desires, and Intentions must exist as the name implies.

Regarding the Beliefs of a CAV, Rüb et al. present them in four different categories. At first, a vehicle must define its initial knowledge about its current environment (i.e. information about the map and kinematics). Vehicles can update their knowledge by contacting other CAVs or infrastructures such as traffic lights and their current state. The third type of knowledge consists of random events such as accidents or obstacles in traffic. A final type of knowledge is the vehicle's self-awareness (i.e., its dimensions and position) [RDK19].

When it comes to Desires, Rüb et al. divide them into basic and meta desires. **Basic desires** are always valid and propel a vehicle to go along the given path safely and optimally (drive-fast, follow the route, obey the traffic rules, and do not crash). These desires are crucial and need to be taken into account for every vehicle's action. As desires have different priorities, fulfilling a lower priority desire might not be possible if it compromises a higher priority. From the examples above, it might be better not to obey the rules if it prevents vehicles from crashing in an extreme situation. **Meta desires** make CAVs reason and react to some external changes to the environment (i.e., replan the route if the road is blocked and cooperate with platoons). Triggering such desires results in a long-term modification of a vehicle: coordination of traffic influences beliefs about the environment; joining a platoon replaces basic desire "follow the route" with a desire "stay in a platoon"; finally, replanning the route changes the vehicle's goals [RDK19].

Intentions are modeled as a response to a mixture of various constraints and obligations. They consist of commitments to an action based on the desires and their priorities. Rüb et al. [RDK19]

diverge from the standard BDI model. In their architecture, a commitment to an action is based not only on one but on all basic desires.

One of the first studies of BDI agent modelling in a context of traffic regulation is the work of Rossetti et al [RBB+02]. In their approach,the traffic domain is modelled as a multi-agent world and drivers are endowed with mental attitudes, which allow rational decisions about route choice and departure time.

2.5 Related Works

Even though the focus of this dissertation lies primarily in the study of platoon formation strategies based on negotiation mechanisms, many other coordination problems have already been thoroughly researched in a transportation domain. This section details the current state of the art of some of the more relevant coordination problems that can be found nowadays in a transport network.

2.5.1 Intersection Management

Eliminating the high impact of intersection bottlenecks in the traffic flow has been one of the major research concerns. Nowadays, the mechanisms in use, such as stop signs or traffic lights, were designed with human drivers in mind. Research is now shifting their attention to autonomous driving agents to achieve higher levels of efficiency. When in the presence of said agents, traffic lights and stop signs pose an obstacle since they fail to take advantage of computerized driver agents' increased sensitivity and precision and do not contribute towards a more efficient traffic flow during intersection management. Autonomous driving agents introduce the necessity of rethinking some of the system's coordinating elements [DS08]. The burst of CAV technologies helps take advantage of the connectivity between vehicles and infrastructure (V2V and V2I communications). The following section describes many of the proposed solutions for Autonomous Intersection Management (AIM) in an autonomous MAS system.

Dresner Stone [DS08] define a list of properties that should be taken into account when coordinating traffic at intersections.

- **Autonomy -** Each vehicle should be an autonomous agent.
- **Low Communication Complexity -** Minimizing the number of messages exchanged between vehicles allows for more communication reliability measures.
- **Sensor Model Realism -** Each vehicle should access sensors only available with current-day technology (no fictional technology that may never become a reality).
- **Protocol Standardization -** Standard protocol allows for new agents to be introduced later in the system. Easier for vehicle manufacturers to adopt.
- **Deadlock/Starvation Avoidance -** Should not occur in the system. Guarantees that every vehicle approaching an intersection eventually crosses.

- **Incremental Deployability -** The system should function only in selected intersections and have room for further expansion. Additionally, it should also perform with few or no autonomous vehicles. Improvement of the performance needs to be guaranteed at each deployment.

- **Safety -** As long as vehicles follow the protocol, no collision is to occur within the intersection. Moreover, the system should be safe upon a total communication failure (i.e., dropped or corrupted messages). If a collision is unavoidable, the system should minimize the damage and casualties.

- **Efficiency -** Vehicles should cross the intersection and continue pursuing their goals in as little time as possible (minimizing the delay within the intersection).

By taking into consideration the previous properties, Dresner Stone [DS08] developed a solution for the intersection management of autonomous vehicles using the concept of reservation.

Dresner Stone [DS08] developed a multi-agent approach to guide vehicles through an intersection in a more efficient way. Their approach uses an intersection manager that receives requests from incoming vehicles attempting to reserve a space-time block within the intersection. To establish a reservation, a vehicle must provide the intersection manager with relevant information such as arrival time, velocity, acceleration/deceleration capabilities, and size. Upon receiving such requests, the intersection manager processes its information and transfers it to the control policy, which will decide whether it is safe to cross the intersection according to the control policy and parameters received from the vehicle. If safety is met, the intersection manager communicates to the vehicle the acceptance of its request accompanied by any relevant information to the driver to guarantee a safe crossing of the intersection. If a safe intersection crossing is not guaranteed, a message is sent by the intersection manager to the vehicle rejecting the reservation. During the decision process, it is not required that the vehicle obtains any information regarding the intersection manager. Besides accepting or rejecting the vehicle's offer, the intersection manager can also respond with a counter-offer to the vehicle's request. In no circumstances should the vehicle proceed to the intersection without having first a reservation confirmed by the reservation manager.

Moreover, Cabri et al. [CGM19] analyze the benefits of using auctions to regulate the traffic at intersections. Their proposition includes both competitive and collaborative strategies in an environment of both autonomous and human-driven vehicles. Furthermore, Cabri et al. defined an enhancement mechanism to consider the number of vehicles in the lanes to speed up longer lanes in the auction process. Instead of having vehicles communicating with each other, Cabri's solution establishes a centralized V2I communication as vehicles forward their bids using a virtual coin to a physical management device placed at an intersection. Bids were considered using two different methods; a first method in which all vehicles pay regardless of the outcome; and a second one where only auction winners are charged for their bid. Results showed better overall performance using a collaborative approach, with lower vehicle average waiting. The competitive strategy revealed a higher variability in the same metric.

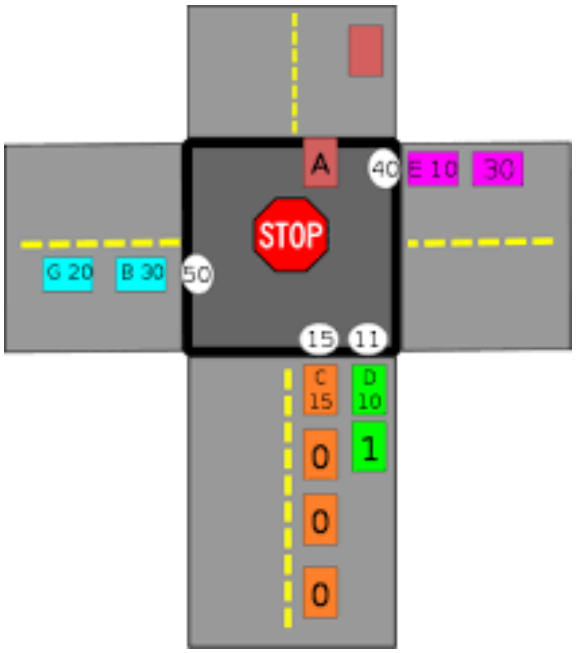


Figure 2.3: Auction process at an intersection [CBS13].

Carlino et al. [CBS13] propose a decentralized auction-based autonomous intersection management. Vehicles automatically bid credit through a "wallet agent" for crossing intersections sooner or later, depending on their value of time. Furthermore, Carlino et al. try to mitigate two problems; a steady stream of wealthy vehicles competing against others without enough funds; evenly distributed auction winners causing a stop sign like behavior. To do so, Carlino et al. implemented a benevolent wallet system, which intervenes to maintain various fairness properties.

An example of how the proposed intersection auction works is depicted in Figure 2.3. Such scenario only allows a single vehicle at a time through the intersection. In the figure, vehicle A is about to leave the intersection, which introduces the need to select a new vehicle to cross the intersection after. To select the new vehicle, an auction between vehicles B, C, D, and E is made as these are the only vehicles at the beginning of the intersection. Each driver bids for the driver at the front of its lane since each is self-interested. While the vehicles behind C opt to not bid anything, the vehicle behind E helps it win the auction and benefits by moving up in the queue. Auction winners split the cost of the second-highest bid with a proportional payment, a method inspired by the Clarke Groves tax mechanism [Cla71, Gro73]. For each candidate vehicle the total bids favor vehicle B with a value of 50 and E as a runner-up with 40. The winners, B, and G, must pay the runner-up's bid proportionally. Since B comprised 60% of the winning bid and G made up the other 40%, B pays 60% of the 40 and G pays the remaining 40%. Once B crosses the intersection, a new auction is made, with G now being a candidate.

Hausknecht et al. [HAS11] applies Dynamic Traffic Assignment (DTA) to a AIM-based grid of intersections. Once again based on the work of Dresner Stone [DS08] in AIM for a single intersection, through the introduction of a navigation policy, an adaptation is made to determine

which route should a vehicle take towards its destination. By determining their own path through a network of intersections, vehicle's are only required to make local reservations with the Intersection Manager of the next intersection they plan to traverse, which allows a revision of the planned path as new information is gathered regarding current traffic conditions.

Vasirani and Ossowski [VO14] propose a distributed, market-inspired approach for intersection management in urban road traffic networks. Their work is based on the reservation based model proposed by Dresner Stone [DS08]. Following a similar approach, the intersection is modelled as a set of space-time slots over which a vehicle will perform reservations before crossing. Opposite to Dresner Stone's work, reservations are not assigned using a FCFS policy. Instead, bids are placed by vehicles and the winner is chosen based on the placed bid values. The bidding system allows vehicles to represent in a quantitative measure how valuable an intersection is. High bids highlight the importance of crossing an intersection, whilst low bids suggest that crossing a given intersection is of little importance/urgency. Since the auction is performed in real-time, the process of collecting bids and determining winners is time-bounded.

In a single intersection environment, results show that the winner determination algorithm delivers better results the longer the processing time given. Moreover, an analysis of the average delays as a function of the bid value show that the designed auction policy rewards higher bids with lower average delays. When compared with the FCFS approach developed by Dresner Stone, the proposed auction-based policy shows higher average delay values when traffic demand increases due to the nature of its main goal not being the maximization of granted requests but rather to grant a reservation to the vehicles that value it the most. A higher number of rejected reservation requests is also observed in the auction policy with its values averaging four times those of the FCFS approach.

Worrawichaipat et al. [WGKR21] propose a decentralised agent-based mechanism for road intersection management of CAVs. Their work focuses mainly on solving delays caused by road obstructions, at the entrance or in the middle of an intersection. Improvements to the proposal of Dresner Stone are presented through a decentralised approach. Essentially, the developed algorithm transfers most of the computation cost from the intersection manager to the vehicles themselves, improving scalability. Instead of having the intersection manager assess the safety of a vehicle's reservation request by computing its path, the vehicle's themselves compute their own valid path and send it to the intersection manager for further validation, due to the concurrency of vehicles requests. Thus, the path prediction overhead in Dresner Stone's model is removed from the intersection manager.

Moreover, their work further improves the state of the art with the proposal of an intersection path prediction algorithm with collision avoidance. Given that an intersection manager shares information regarding the obstructions it currently has (i.e. position, length, safety radius surrounding the obstruction), a vehicle can detect whether or not following its normal path through the intersection can end up in a collision with one or more obstructions. If such is the case, the collision avoidance algorithm illustrated in 2.4 is applied: (1) avoid the blockage by steering the vehicle to a tangent line along the safety ring; (2) follow the curve that moves along the safety ring

until the obstruction is no longer detected; (3) continue to steer and move forward to the entry of the target lane.

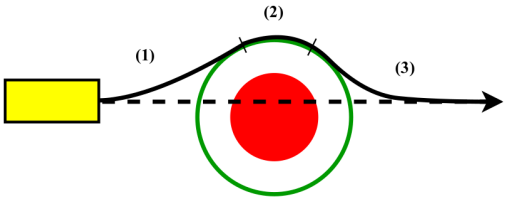


Figure 2.4: Collision avoidance between a vehicle and an intersection obstruction

Results show that when an obstruction is present at the entrance or in the middle of an intersection, existing state of the art algorithms and traffic lights show a reduced throughput between 65 and 90% from its optimal point while our mechanism can maintain the throughput up to 94 to 99%.

Shen et al. [SMWW19] also propose a coordination strategy for both human and autonomous connected vehicles that aims to increase vehicle capacity and a better traffic flow at intersections, while ensuring vehicle safety. A centralized controller replacing what would be a traffic light, is responsible for determining the priority and passing conditions of each vehicle. Once given permission, vehicles can cross the intersection. Regarding CAVs, besides receiving a pass from the centralized controller, a target vehicle is also received so that the receiving vehicle's knowledge base contains conflicting vehicles to avoid colliding with.

Jin et al. [JWBB13] propose an intersection management system that allows the formation of platoons to cross an intersection. Their approach has vehicles communicate with a centralized intersection manager responsible for validating incoming vehicle requests. Vehicles are classified as leaders of a platoon or platoon followers if a platoon leader is within range. Only platoon leaders communicate with the intersection manager by sharing their estimated time of arrival to the intersection and estimated clearance time (time to cross the intersection) according to their platoon size. The intersection manager then accepts or rejects a vehicle's request depending on its current "bookings".

Results of Jin et al. [JWBB13] approach shows promising results when it comes to travel-time , with a reduction of 30% and 4% to 8% when compared compared with the traffic light control system and the non-platoon-based system, respectively. Regarding fuel consumption and emissions, both a platoon and non-platoon based systems show a 10% reduction compared to the traffic light control system. Important to notice that the platoon approach achieves slightly higher fuel consumption and emissions than a non-platoon approach due to the platoon formation stage. Finally, communication load was evaluated through the number of communicating vehicles. A comparison between a non-platoon and a platoon approach shows a significant reduction of communicating entities, with a decrease of up to 90% in more congested traffic.

Another intersection management approach considering platoons was proposed by Bashiri and

Fleming [BF17]. Two distinct policies were developed for further evaluation: Platoon-based Delay Minimization (PDC) and Platoon-based Variance Minimization (PVM). In each policy, similarly to the approach of Jin et al. [JWBB13] a centralized intersection manager listens for platoon requests. If a PDM measure is adopted, the intersection manager will try to schedule the platoon crossings with the goal of minimizing the overall waiting times of the incoming platoons. On the other hand, PVM makes a trade-off between minimizing average delay and minimizing the maximum wait time.

The two proposed policies where compared against the stop sign policy regarding average delay and variance in delay per vehicle. Results show great improvements in average delay when compared to stop signs. In low traffic volumes, PVM outperforms PDM. However, as vehicle demand increases the delay minimization effects of the PDM show better results in terms of average delay. However, when it comes to variance minimization, the greedy nature of the PDM approach show significantly higher results compared to the PVM and stop sign policies.

2.5.2 Lane Merging

A lane merging scenario involves merging a vehicle or a group of vehicles onto a carriageway [SSSM19]. A lane merge operation can occur between vehicles travelling in two different adjacent lanes. For a lane merging operation to be successful and safely performed, a safety distance is required between the merging vehicle and the rest of the vehicles, ensuring that no collisions occur during the operation. The way this safety distance is achieved can vary depending on the lane merging scenario. Some approaches focus on controlling the joining vehicles while others work primarily on the vehicles in the main lane.

Lane merging coordination approaches have already been studied. Nilsson et al. [NSB+16] details a low-complexity lane merging algorithm that determines if a lane manoeuvre is desirable or not by calculating longitudinal and lateral control trajectories. Similarly, Zhu et al. [ZHK+13] attempts to reduce additional road space before a lane merging operation takes place. With this view, a two-lane road was divided into cells, that can be empty or contain a vehicle. Four different actions were proposed to manage vehicles (i.e, acceleration, slowing down, randomization, and vehicle motion). Based on this, three types of lane change were investigated: tail to head, head to tail and random. The tail to head approach showed better performance compared to a random lane merge.

Real world information has been used, but some of these works are at a primal stage. Wang and Chan [WC17] presented an on-ramp merge driving policy. To do so, they considered a scenario involving three vehicles - a merging vehicle and two vehicles on the main lane. A total of 9 variables were used, where 5 variables described the merging vehicle's driving state. These included heading angle, position, speed, distance to the right lane, and distance to the left lane. For the other two vehicles, only the speed and position are known. However, the algorithm has not been verified or validated.

Sequeira et al. [SSSM19] present a lane merge coordination model based on a centralised system. The system delivers trajectory recommendations to connected vehicles on the road. It makes

use of real data from two highways to train and validate a various machine learning algorithms.

2.6 Summary

This previous chapter provided a literature review on vehicle coordination through MAS systems' negotiation mechanisms. Specifically, market-based strategies were introduced in platoon formation and intersection management scenarios. The study of BDI agent architectures applied to vehicles in an ITS system was also considered.

The reviewed literature in this chapter helped to identify two main research gaps that will comprise the bulk of the work developed in this dissertation.

- Little studies have been made on platoon formation strategies using negotiation mechanisms.
- Research lacks when it comes to coordination between platoons and independent vehicles at intersections using negotiation mechanisms.

Platoon formation strategies studied in the literature primarily consisted of algorithmic approaches in which vehicles do not negotiate the possible outcomes. Instead, they are "told" what to do as central entities take care of the decision-making process. Other solutions, although using negotiation mechanisms [SBJ+14] cannot be considered "true" platoon formation strategies as they describe a reconfiguration of existing platoons through the insertion of external independent vehicles, thus assuming the existence of a pre-defined platoon.

Intersection management has seen a more significant application of negotiation mechanisms through market-based approaches, auctions. Collaboration between vehicles within the same lane is considered in some of the work by having non-eligible vehicles (not at the start of the intersection) contributing towards a bid to decrease the time spent at an intersection by moving up the queue [CBS13]. However, the benefits of platooning strategies in the context of intersection management are somewhat disregarded, which opens an opportunity to strengthen this research subject by tackling it in this dissertation.

Chapter 3

Intersection Management

Following the critical review of the state-of-the-art given in Chapter 2, we herein propose a new resource reservation intersection management mechanism using auctioning strategies, that reduces the reliance on the intersection manager (IM) to compute paths by a decentralised computation of trajectories for individual vehicles. The basis of this approach is to have vehicles communicate with an intersection manager and make reservations to portions of the intersection according to the path they take while crossing it. These reservations consist of the time-space slots needed for a vehicle to cross the intersection. Due to the competitive nature that intersections have - multiple vehicles competing for a resource (the intersection itself) - reservations are accompanied by a bid. In this context a bid consists of a value that the vehicle must pay to cross the intersection. In its essence, it describes how urgent it is for a vehicle to go through the intersection. The bidding process can be seen as a first-price, sealed-bid approach since the bidding entities pay their bidding price and do not know other's bidding prices. This approach also significantly reduces the high computational cost at the intersection manager existing in the centralised FCFS (First-Come-First-Serve) approach by Dresner and Stone [DS08] .

Thus, this thesis advances the state-of-the-art in the following ways:

1. Using bids as a conflict resolution mechanism in the IM request validation algorithm.
2. Improvement of the cell-based conflict resolution algorithm introduced by Dresner and Stone [DS08] to support platooning.

The following sections provide a description of the model as well as any relevant definitions and assumptions. Furthermore, an explanation of the intersection reservation mechanism is presented.

3.1 Traffic Model

To model the intersection coordination strategy, a reservation-based mechanism was designed. This mechanism is inspired in Dresner and Stone's [DS08] First-Come-First-Serve (FCFS) approach. Before going through the reservation-based mechanism itself, the two main entities that take part in it must first be introduced: the intersection and the vehicles.

3.1.1 Intersection

As previously stated, this approach focuses on having vehicles make reservations on portions of the intersection, which consist on the time-space slots needed for a vehicle to cross it. To mimic this behavior the intersection was converted to a matrix of reservation tiles (cells). When compared to modelling the intersection as a whole, dividing the intersection into reservation tiles can improve the traffic throughput, while reducing the experienced delays since it allows several vehicles to cross the intersection at the same time as long as their paths do not have conflicting cells.

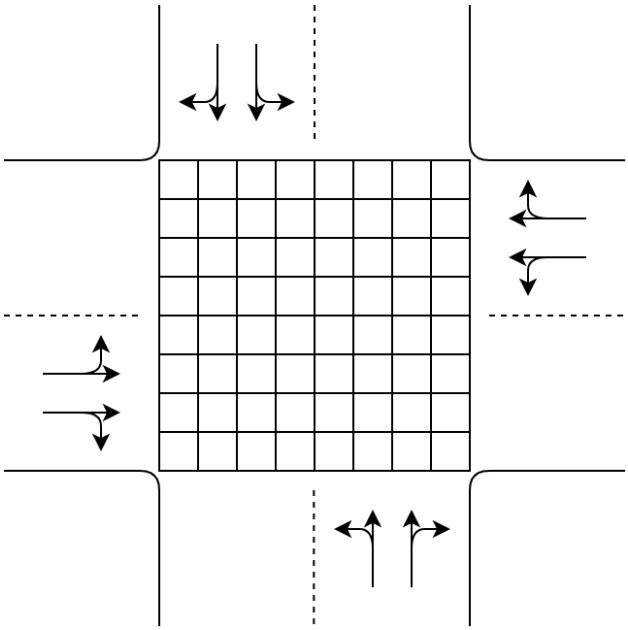


Figure 3.1: Intersection modelled as a matrix of reservation tiles

Figure 3.1 illustrates how an intersection can be divided into reservation tiles. Each reservation tile of the matrix then holds a list of all the reservations it currently has in place. A reservation can be characterized by the following properties:

- **begin-time:** time-step at which a vehicle/platoon enters a reservation-tile
- **end-time:** time-step at which a vehicle/platoon leaves a reservation-tile
- **bid:** value of the vehicle/platoon's bid for its reservation
- **vehicle identifier:** identifier of the vehicle/platoon leader that made the reservation

3.1.2 Vehicles

They represent the entities that communicate through V2X networks with the intersection manager by performing reservations to the intersection depending on their route. The designed implementation allows for three different types of vehicles: (i) *independent vehicles*, (ii) *platoon leaders*

and (iii) *platoon followers*. Independent vehicles, as the name suggests, consist of individual vehicles that travel alone and do not depend on any other vehicle to dictate their course of action. In a platooning setting, platoon leaders co-exist with platoon followers. A vehicle is considered a platoon leader if it travels in a platoon and is the vehicle ahead in it. On the other, for a vehicle to be considered a platoon follower, it must be travelling in a platoon but not be the vehicle ahead.

A vehicle's behavior differs depending on its type. Independent vehicles only communicate with the IM. On the other hand, platoon leaders have the responsibility of sharing the best interest of the platoon to the IM. As such, platoon leaders communicate with both their followers and the IM. As a consequence of having platoon leaders acting as the bridge between their followers and the IM, platoon followers only communicate their interests to their respective leader. Communication between platoon followers and IMs is not considered in this study.

3.2 Bidding Protocol

As previously stated, a reservation is accompanied by a bid. Bids represent the urgency of a vehicle to cross an intersection. High bid values mean that crossing the intersection as fast as possible is very important for a vehicle, while low bid values mean that a vehicle is not in a hurry and can afford to wait at an intersection. These bids, once a vehicle starts crossing an intersection, are payed to the IM. This bidding approach, in theory, allows for a more fair-minded distribution of reservations with the common goal of reaching the destination in time being its main priority. The following paragraphs will focus on detailing how bids are calculated, both in an independent and platooning context. Furthermore, the implemented payment strategies are also described in a platooning context.

3.2.1 Bid Calculation

To represent the sense of urgency felt by a vehicle, three assumptions are made:

- every vehicle has a Maximum Time of Arrival (MTA) previously established at the beginning of its trip. This value refers to the latest arrival time a vehicle can arrive. Arriving after the MTA is considered arriving late.
- every vehicle is capable of calculating its ETA (Estimated Time of Arrival) at any given time.
- every vehicle knows how many intersections its current route has left at any given time.

Both the MTA and ETA estimated values assume a constant velocity during their calculation. More complex estimations would take into account the volume of traffic present in each segment of its route. However, to validate the implemented mechanism a simpler approach was chosen. Furthermore, for a vehicle to be eligible to bid, it needs to possess some sort of credits. The meaning of these credits was not subject of any study. It is assumed that each vehicle has some amount of credits prior to the start of its trip (the term "wallet" will be used to refer to a vehicle's

current credits). We perform a study on how a credits distribution affects the intersection crossing mechanisms. Two different credit distributions are studied:

- **Uniform:** every vehicle starts its trip with the same amount of credits in their wallet (e.g., 500 credits).
- **Non-uniform:** the credit's distribution follows a normal distribution with a certain mean (e.g., 250 credits) and a given standard deviation (e.g., 100 credits).

These four vehicle parameters can then be used in the bid calculation process to estimate a bid value. To do so, two different bid calculation formulas are studied. A first approach focuses on the difference between the ETA and MTA values. The smaller the difference, the greater the urgency of a vehicle. This results in the following formula:

$$Bid = \frac{W}{nI} * \frac{1}{MTA - ETA} \tag{3.1}$$

where:

$W =$ current wallet credits
$nI =$ number of intersections left in the vehicle's trip

The first quotient of the formula represents the maximum available credits to bid at a given intersection. If a vehicle with 300 credits in its wallet still has two intersections left in its route, upon arriving at the first intersection only 150 credits will be available for bidding. From these 150 credits, the second part of the formula decides how much to use on the vehicle's bid. Let's assume the time values of t=50 and t=100 for both the ETA and MTA, respectively. With the 150 available credits from the first half of the formula, the bid results in a 1/50 share of that value, which equals a bid value of 3 credits. Using this formula, bids increase exponentially as the ETA gets closer to the MTA.

A similar behavior can be obtained by using the quotient between the ETA and MTA values. The result of such quotient is the percentage of available credits that will be used in a bid. Since the ETA is always smaller or equal than the MTA, their quotient is always less than 1, which means that the percentage will always be less than 100% of the available bidding credits.

$$Bid = \frac{W}{nI} * \frac{ETA}{MTA} \tag{3.2}$$

Despite portraying the same behavior, the two approaches have clear differences. Using the same values of 150 available credits, t=50 and t=100 for both the ETA and MT respectively, the resultant bid value is of 75 credits as a share of 50% of the available credits is used for bidding. This formula increases linearly as the ETA and MTA values get closer to each other.

In a platooning context, each vehicle still calculates its own bid using the previous process, since it represents a good estimation on how valuable an intersection is regardless of the platooning context the vehicle is in. A platoon bid is represented as the average of its vehicle's individual bids. A vehicle in a hurry while travelling in a platoon, boosts the platoon's bid as its individual bid is

high. The opposite happens if a vehicle that's travelling in a platoon has all the time in the world to finish its trip.

3.2.2 Bid Payments

The bid payment approach for independent vehicles is straightforward as we use a first-price rule [HZF+20]. If a vehicle wins a reservation with a bid of x, it pays the IM x credits. However, that is not the case in platooning vehicles. Since every vehicle of the platoon benefits from successfully crossing an intersection, every vehicle must pay the price for it. This prevents vehicles from overbidding credits that they do not possess. That could be the case if, for example, only the platoon leader payed the bid to the IM. Two different bid payments were studied when it comes to platoon bids:

- **Non-Weighted:** every vehicle pays the same share of the final bid ($finalBid/platoonSize$). For instance, if a platoon of 5 vehicles bids 50 credits, each vehicle must pay exactly 10 credits to the IM.

- **Weighted:** each vehicle's share depends on how much the vehicle contribute to the final bid. Vehicles that contribute more to the final bid also pay more than others that contribute less.To calculate each vehicle's share, the following formula is used:

$$share_n = \frac{bid_n}{\sum_{n=1}^{plat_{size}} bid_n} * bid_{plat} \tag{3.3}$$

where:

$share_n$ = bid paying share of the nth vehicle in the platoon
bid_n = bid of the nth vehicle in the platoon
$plat_{size}$ = size of the platoon
bid_{plat} = platoon's final bid

The later method can be seen as a more fair-minded payment method than the *Non-Weighted* as vehicles that are not in a hurry are not prejudiced by vehicles that are.

3.3 Intersection Reservation Mechanism

This section introduces the reservation-based intersection mechanisms which supports both independent and platooning vehicles. In addition, the developed mechanism reduces the computation load of the IM when compared to the Dresner and Stone's [DS08] FCFS approach by having the vehicles perform the necessary computation to successfully make a reservation. The following paragraphs will dive deeper into the protocols for the IM and vehicles as well as the interactions between them.

3.3.1 Vehicle Protocol

As a consequence of relieving the IM's computation load, the decentralised approach makes the vehicle responsible for computing all the required information in a vehicle's request. Looking back at Dresner and Stone's proposed solution, the IM has a more proactive role since it is responsible for assigning each vehicle's reservations based on the information that the latter share with it. On the other hand, a decentralised solution requires the IM to share its information with vehicles instead of the vehicles sharing their information with the IM.

For a vehicle to be able to perform a reservation request, it requires some information regarding the upcoming intersection to be shared by the IM. More specifically, the intersection manager broadcasts its current reservation map as well as details about the different routes the vehicle can take to cross the intersection. Upon receiving such information from the IM, four main stages must be completed for a vehicle to successfully perform a reservation request:

- **path prediction** (3.3.1.1): compute the reservation map tiles intersected by the vehicle when crossing the intersection. Furthermore, calculate the respective timestamps that the vehicle enters and leaves each reservation tile.
- **conflict detection** (3.3.1.2): evaluate the predicted path with the received reservation map by comparing whether or not the computed tiles and respective timestamps generate conflicts, resulting in vehicle collisions.
- **conflict resolution** (3.3.1.3): re-calculate the reservation timestamps to delay the vehicle's departure if any conflict occurs in the previous stage.
- **bid calculation** (3.3.1.4): calculate the bid value based on the bidding protocol presented in section 3.2.

3.3.1.1 Path Prediction

The main form of information the IM shares with a vehicle is the reservation map. However, that is not the only information the vehicle receives. One additional important piece of information received are the different possible routes through the intersection. Such information is relative to the SUMO traffic simulator. To represent a route trough the intersection, SUMO only stores some key points within that route. The route is then represented by straight lines that are defined by two consecutive key points. Figure 3.2 shows a visual representation of what a possible route through an intersection can be. The problem at hand now is to find the reservation tiles that intersect the given path so that the vehicle can request reservations only to the tiles it uses. Furthermore, for each reservation tile in its path, the vehicle must estimate an initial and final time-step representing when the vehicle enters and leaves each tile.

To determine which reservation tiles are intersected by a vehicle's predetermined path, an adaptation on the **Bresenham's algorithm** [Bre65] can be made. Bresenham's algorithm is a line drawing algorithm that determines the points of an n-dimensional raster that should be selected

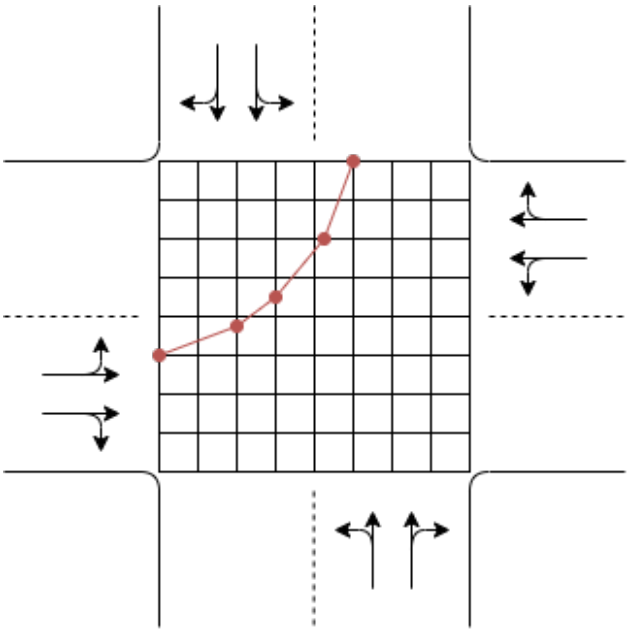


Figure 3.2: Intersection path connections example

in order to form a close approximation to a straight line between two points. It was originally developed in the field of computer graphics. Its advantage in comparison with other algorithms is the use of cheap operations such as addition, subtraction and bit shifting using only integers. However, the implemented variation of Bresenham's algorithm in this work makes use of such operations with float values since the dimensions of the reservation tiles might not be integers depending on the granularity of the reservation map (matrix dimension).

Lets consider an example of how the Bresenham's algorithm works, only taking into account first quadrant values ($x > 0$ and $y > 0$). In Bresenham's algorithm, we move across the x-axis in unit intervals. In every iteration, after increasing the x value, a choice must be made regarding the y-value. There are two different possible outcomes: in a positive slope line y can either increase or remain the same, in a negative slope line y can decrease or remain the same. From these two different outcomes, one is closer to the "true" line. For this reason, Bresenham's approach creates a decision variable that helps decide whether to maintain the y value or change it. The idea is to keep track of slope error from previous increment of y. If the slope error becomes greater than 0.5, we know that the line has moved upwards/downwards one pixel, and that we must increment our y coordinate and readjust the error to represent the distance from the top of the new pixel – which is done by subtracting one from the current error. This algorithm continues until x reaches the final point.

To adapt the Bresenham's algorithm to the reservation map, an important change to take into account is the fact that, contrary to pixels, the reservation tile's width and height might not be unitary. After that, to find which reservation tiles intersect a vehicle's path, the Bresenham's algorithm can be used between every two consecutive key points of a vehicle's route. By adding

the previous adaptation to the original Bresenham's algorithm, an end result as the one in figure 3.3 can be expected. The resultant path includes the reservation tiles a vehicle intersects in order to cross the intersection and thus, those will be the reservation tiles considered in the vehicle's reservation.

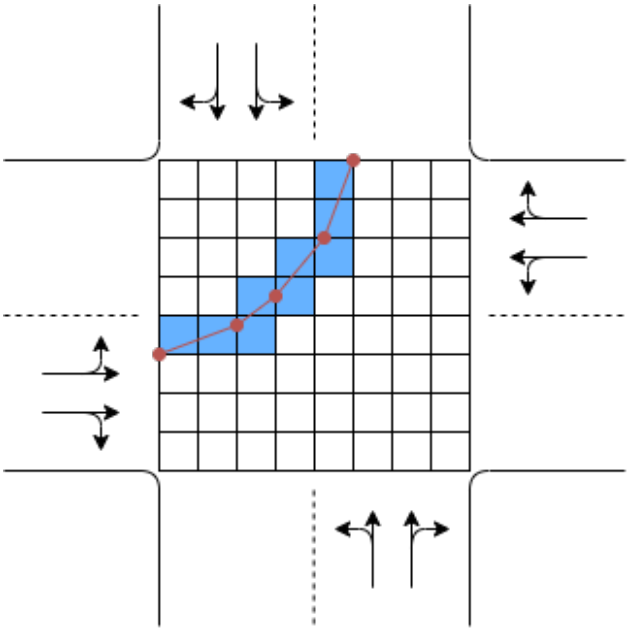


Figure 3.3: Bresenham's algorithm result for path prediction

As soon as the vehicle finishes calculating all of the reservation tiles in its path, a time interval for each tile needs to be determined. Such time interval specifies the time-steps at which a vehicle enters and leaves a specific reservation tile. The definition of these time intervals is particularly useful when considering platoons as they typically may take an higher amount of time to go through a reservation tile. For this computation, a constant velocity is considered and as such, the formula for deriving speed from distance and time can be used in this situation to calculate the time interval a vehicle spends in a reservation tile.

Since there is no easy way to know how much distance a vehicle travels in a reservation tile (SUMO does not provide enough information to compute that measure) a maximum distance is assumed in order to estimate how much time it takes for a vehicle to cross a reservation tile. That distance is equivalent to the diagonal of the reservation tile as that is the maximum possible distance any vehicle can travel inside a particular reservation tile. This approximation slightly decreases performance but also increases safety as each vehicle will end up reserving each tile for a longer period than it needs. When in the presence of a platoon, this time interval is multiplied by the number of vehicles in a particular platoon. The complete formula for calculating the time interval required to cross a reservation tile is presented in 3.4

$$t_{tile} = \frac{d}{v} * plat_{size} \tag{3.4}$$

where:

t_{tile} = time to cross a reservation tile
d = distance
v = velocity
$plat_{size}$ = platoon size (set to 1 if an independent vehicle)

Furthermore, for a vehicle/platoon to compute its reservation tiles' time intervals, it also needs to specify the time-step at which it will start crossing the intersection. To do so, a vehicle just needs to compute the time it takes to travel between its current location and the beginning of the intersection by using Equation 3.5 for deriving time from distance and speed.

$$t = \frac{d}{v} \tag{3.5}$$

By combining a vehicle's starting time with the time interval computation in Equation 3.4, it is possible to determine the corresponding time-steps at which a vehicle/platoon enters and leaves each reservation tile.

3.3.1.2 Conflict Detection

Once a vehicle finishes calculating its path through the intersection, it still has to test it against the reservation map information previously sent by the IM. This evaluation takes each reservation-tile from the computed path and checks if the corresponding reservation-tiles in the reservation map already contain reservations during the requesting reservation timestamps.

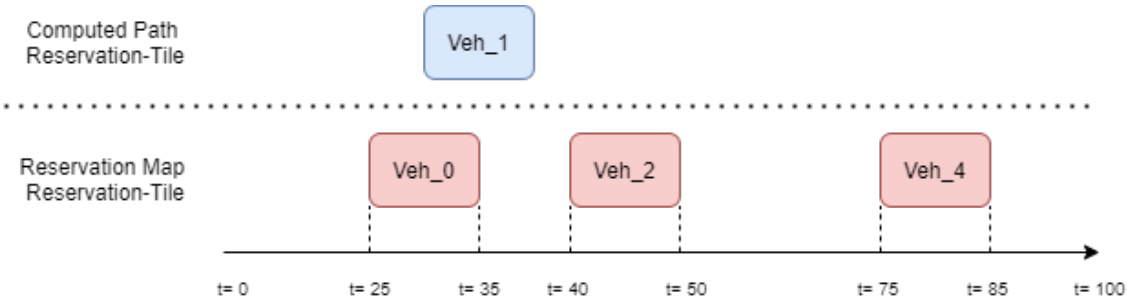


Figure 3.4: Conflict occurrence in a reservation-tile

Figure 3.4 depicts a conflict scenario in a given reservation-tile that may occur during the conflict detection phase. In the given reservation-tile, the reservation map contains three different reservations from three different vehicles. The reservation intervals are [t=25, t=35], [t=40, t=50], [t=75, t=85]. Which means that, in order to avoid conflicts, the path's reservation-tile timestamp must fall inside the following intervals: [t=0, t=24], [t=36,t=39], [t=51,t=74] and [t=86, ∞]. However, with the given timestamps, Veh_1 enters the reservation-tile sometime between[t=25, and t=35, which is already reserved by Veh_0 and thus resulting in a conflict.

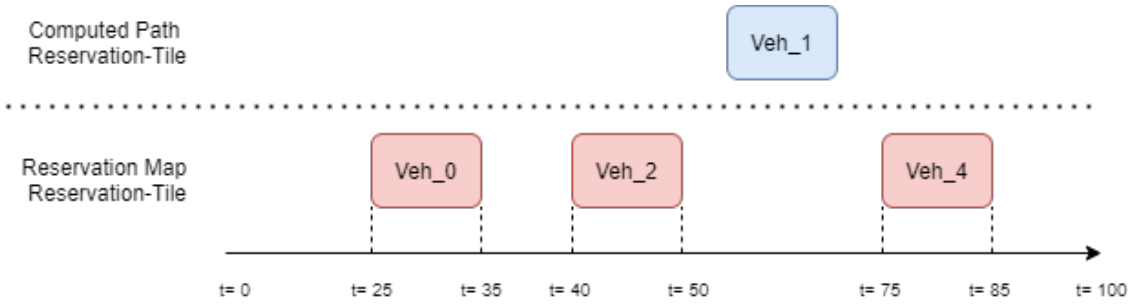


Figure 3.5: Non-Conflict occurrence in a reservation-tile

Opposite to 3.4, in figure 3.5 no conflict occurs since the vehicle enters and leaves the reservation-tile between t=51 and t= 74. As previously observed, this is one of the available time-intervals with no reservations in the reservation map.

For a vehicle's computed path to be eligible for a reservation request, its reservation-tiles must all be available during the specified time-intervals in the reservation map. If at least one conflicting reservation-tile occurs, then the path is invalid as this means that vehicles would collide while crossing the intersection.

3.3.1.3 Conflict Resolution

If conflicts occur, a strategy must be adopted to solve them. The implemented strategy consists in delaying the departure of the vehicle until no conflicts exist. The core concept for this approach lies on the following premise: if a conflict occurs with a duration time of t, regardless of the reservation tile, at least a delay of t time needs to be added to the departure time in order for the conflict to disappear. In the presence of multiple conflicts, t is used to represent the conflict with the biggest duration.

Figure 3.6 depicts how the previous premise works in a practical example. Initially a requesting vehicle's reservation has conflicts with two other reservations in the reservation map from Veh_0 and Veh_2. These conflicts endure for $t1 = 10s$ and $t2 = 5s$, respectively. Delaying the departure by $t2$ seconds would not solve both existing conflicts since the conflict with Veh_0 would still exist but with a lower duration time of $t1 = 5s$. In order to have a chance of resolving all the conflicts, the delay used should be of $t1$ seconds as that solves all the currently existing conflicts. However, delaying the departure by exactly the same amount of time as the conflict lasts does not guarantee that further conflicts cease to exist. For that reason, a new conflict detection iteration is required. If any conflicts persist then another delay is added on top of that. Eventually, no conflicts will occur within the defined time intervals for the all reservation tiles, at which point the vehicle's path is considered valid.

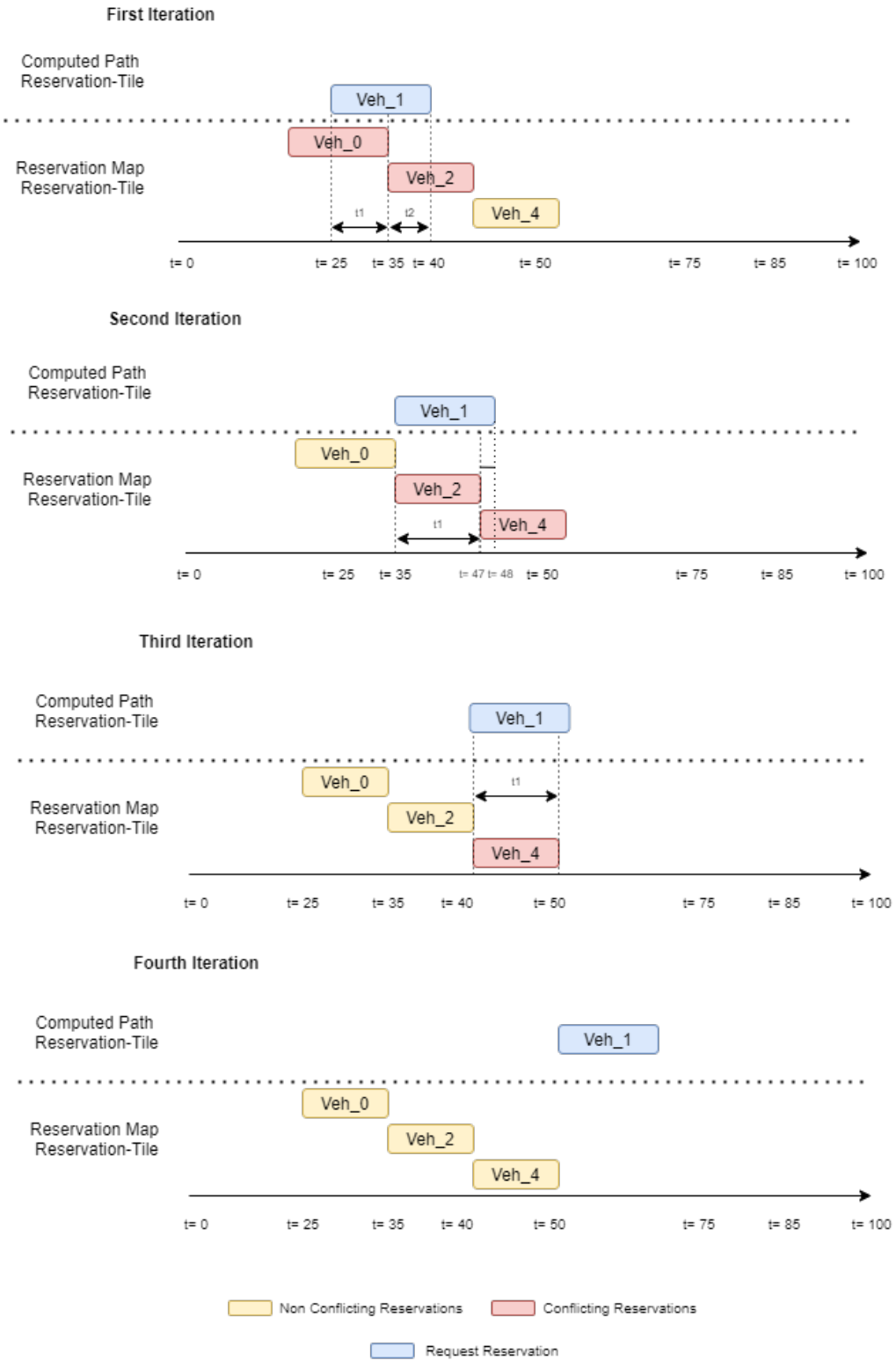


Figure 3.6: Vehicle's conflict resolution process

3.3.1.4 Bid Calculation

Once a vehicle concludes the first three stages previously detailed, all that is left is for the vehicle to calculate its bid by using the bidding protocol in 3.2. The designed approach does so using a BDI architecture when it comes to the bidding stage to represent a human-like reasoning to estimate a bid value for a given reservation request.

The first step of defining a BDI architecture is to identify the beliefs, desires and intentions required.

- **Beliefs:** since the defined bidding protocol only makes use of two different parameters in the bidding calculation process, as of now, two beliefs can be identified: MTA and ETA.

- **Desires:** the only desire or goal identified in an intersection setting is to simply cross the intersection

- **Intentions:** specifies the vehicle's course of actions. Once again, only one course of action is identified, which is used to fulfill the vehicle's desire to cross the intersection. This course of action or intention is to simply calculate the vehicle's bid value using the bid calculation process specified in 3.2.

Figure 3.7 illustrates the BDI process implemented for the bid calculation process. Upon a vehicle's initialization, the MTA parameter is initialized in the given vehicle's belief base. After the vehicle finishes the initialization stage, no ETA values are defined and no intentions are adopted by the vehicle to achieve its desire of crossing the intersection. After the vehicle finishes the previous stages of path prediction, conflict detection and conflict resolution, an event is triggered which prompts the vehicle to update its belief base, more specifically its ETA value. The update of the ETA parameter triggers a plan. Plans are sequences of actions that an agent can perform to achieve one or more of its intentions. In this case the plan itself only contains one action consisting of the bid calculation, which now becomes the vehicle's intention. After the vehicle finishes calculating its bid value, it is then able to send a valid reservation request containing the computed trajectory through the intersection, the reservation timestamps, and the vehicle's estimated bid value.

3.3.2 Intersection Manager Protocol

The job of the IM is to validate incoming requests and update its reservation map information if needed. This approach relieves the IM's computation load as each vehicle handles its own path computation. However, to validate a request, a second conflict detection stage must be performed (the first one being performed by the vehicle before sending its request) and, if needed, conflict resolution strategies are used to determine whether to accept or reject a conflicting request, based on its bid value. Section 3.3.2.1 explains how the IM validates incoming reservation requests.

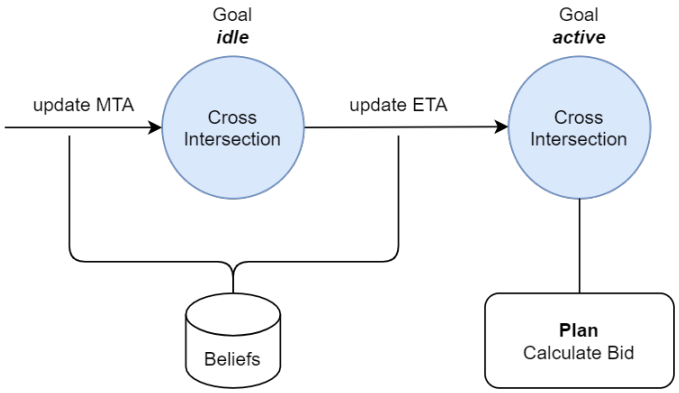


Figure 3.7: Vehicle's BDI Flow

3.3.2.1 Conflict Resolution

Even though a vehicle validates its own request before forwarding it to the IM, there is a chance that this evaluation is not justifiable once it reaches the IM. Due to the concurrency of requests being made, from the moment a vehicle validates its own request to the moment that same request reaches the IM, the reservation map can be updated by other vehicles' successful reservation. This means that the vehicle's local evaluation is outdated and thus not valid. The following example helps understand the problem at hand. In this scenario two vehicles(veh_0 and veh_1) need to cross the intersection.

1. veh_0 receives the most updated reservation map from the IM.
2. veh_0 calculates and evaluates its trajectory against the received reservation map.
3. veh_0 calculates its bid and sends a reservation request to the IM.
4. veh_1 receives the most updated reservation map from the IM (which is the same one that veh_0 received).
5. IM accepts veh_0's request, updates the reservation map and notifies veh_0.
6. veh_1 calculates and evaluates its trajectory against the received reservation map (which does not contain veh_0's reservation)
7. veh_1 calculates its bid and sends a reservation request to the IM.

The previous example perfectly illustrates the problem at hand as veh_1's local evaluation does not take into account the fact that veh_0 already has a successful reservation in the reservation map. This can be rather problematic as veh_0 and veh_1 might have conflicts between them. To mitigate this problem, a second evaluation is made by the IM upon receiving a new request.

The conflict evaluation made by the IM is very similar to the local evaluation that vehicles make prior to sending their request. Here, each reservation tile of the request's path is tested

against the correspondent reservation tile in the reservation map using the same approach presented in section 3.3.1.2. However, an additional measure is taken into account to resolve any existing conflicts. That measure is the reservation's bid value.

There are two possible outcomes to the conflict detection algorithm: either conflicts are detected or no conflicts occur between the requesting vehicle and the reservations in the reservation map. By introducing the bid value in the conflict resolution stage, vehicles can be allowed to successfully make a reservation even after conflicts occur. As such, the entire process of conflict detection and resolution in the IM has a total of three different outcomes:

- **No conflicts occur:** in this case, since no conflicts occur between a vehicle/platoon's request and the previous successful requests in the reservation map, the requesting vehicle/platoon's reservation can simply be added to the reservation map.
- **Conflicts occur:**
  - **Request's bid is higher than all conflicting bids:** the requesting vehicle/platoon higher bid means that crossing the intersection is more important for it than to the conflicting vehicle/platoon(s). As such, the reservation is granted to the requesting vehicle/platoon and, to ensure that safety is guaranteed while crossing the intersection, the conflicting vehicle/platoon(s) in the reservation map loose their respective reservations. The IM, besides notifying the requesting vehicle, now needs to inform the conflicting vehicles as well of their lost reservations.
  - **Request's bid is not higher than all conflicting bids:** since there is at least one conflicting vehicle whose crossing through the intersection is more important when compared to the requesting vehicle, the request is rejected and no changes are made to the reservation map.

Having both the vehicle and intersection manager protocols described, as well as the main interactions between each other, Figure 3.8 summarizes and illustrates the reservation flow observed by the reservation mechanism between the IM and the vehicle entities.

3.4 Summary

In this chapter the theoretical model behind the implemented solution was presented. This model has focuses on two main entities: the vehicles and the intersection. Vehicles can be of three different types: independent, platoon leaders and platoon followers. As for intersections, they are modelled following a reservation based approach in which they are perceived as a matrix of reservation tiles that can be booked by incoming vehicles.

Furthermore, the bidding strategy was described. For a vehicle to be able to bid it firstly needs to instantiate one of two types of wallets (Uniform or Non-uniform). After that, two different bidding calculation strategies were presented: a first approach with an exponential behavior over time, and secondly a linear approach. The bidding strategy for platooning vehicles is also presented in

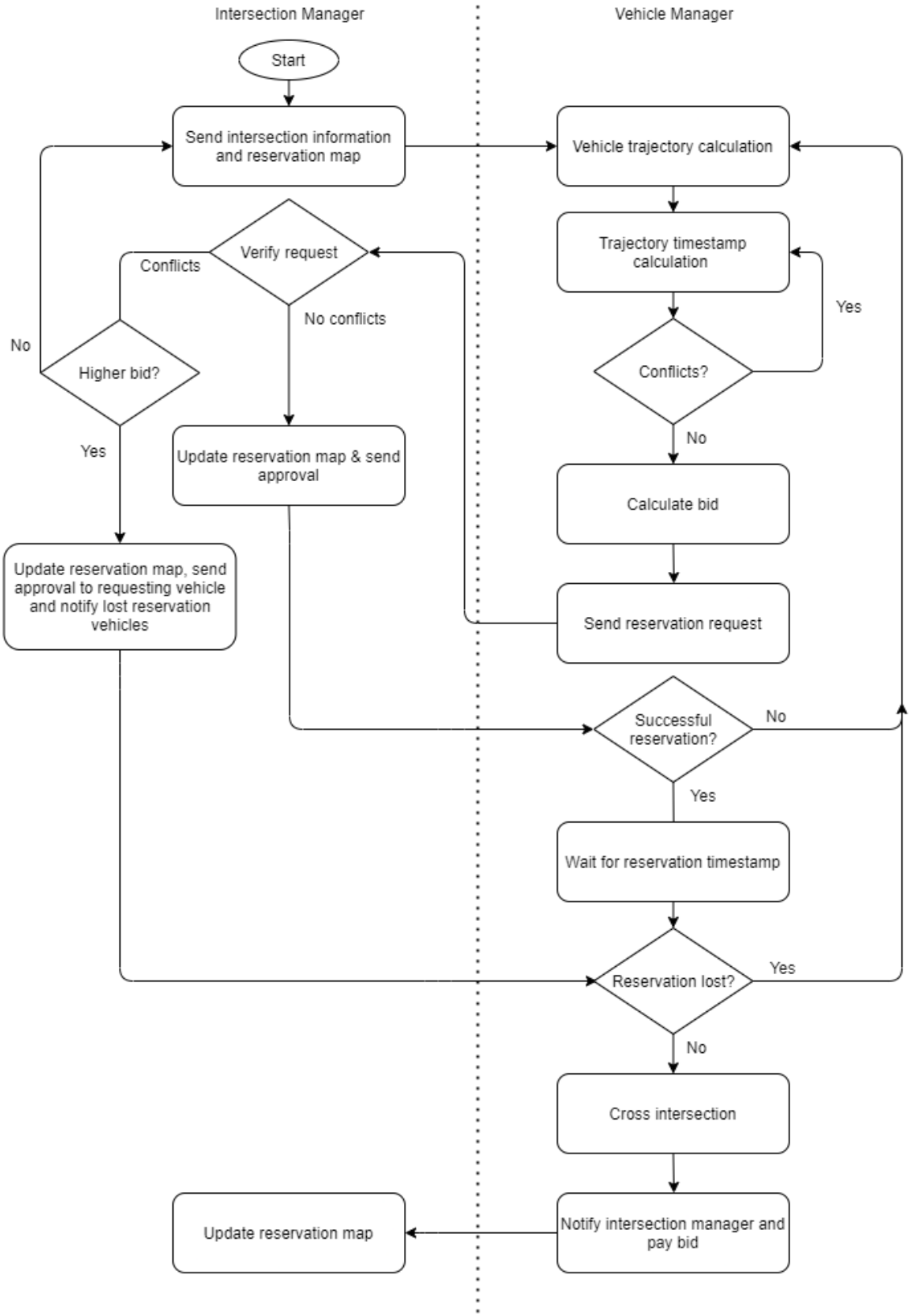


Figure 3.8: Reservation flow between the Intersection Manager and Vehicle Manager

this section. To finish the bidding protocol, a more in-depth description was given detailing the two different types of platoon payments designed: Non-Weighted and Weighted.

Moreover, the vehicle's reservation process was described and each of its three stages were thoroughly presented. Firstly, a description on how a vehicle's path is computed using the reservation map was made. Following that, the conflict detection algorithm was presented followed by the conflict resolution mechanism based on departure delay.

In what concerns the IM, the conflict detection and resolution designed solutions were described with the bidding strategies in mind. The decentralised aspect of the implemented solution allows for computation costs to be taken away from the IM. To finish of the chapter, a flow map depicting the different stages of each entity and their interactions was presented.

In the next chapter, an overview of the architecture of the implementation is presented as well as the different technologies and components it uses. Furthermore, the next chapter focuses on explaining how the different components of the implemented solution communicate with each other in order to successfully achieve the final results of the implementation.

Chapter 4

System Architecture

This chapter presents a high-level description of the implemented solution for the strategy designed in Chapter 3. Section 4.1 an overview of the system's architecture is presented. Furthermore, sections 4.1.1, 4.1.2, 4.1.3 and 4.1.4 describe the components integrated in the solution. The communication between components is specified in section 4.2. Finally section 4.3 provides a summary of this chapter alongside some concluding thoughts.

4.1 Simulation Architecture

To implement the designed solution three main components are required: a traffic simulator, a network simulator and a MAS framework for bidding process.

Proper communication and synchronization between these components must be ensured for the implemented solution to behave as expected. Fortunately, a framework by the name of Eclipse MOSAIC (Sub-Section 4.1.1) greatly helps such integration. Eclipse MOSAIC, previously known as VSimRTI (V2X Simulation Run-time Infrastructure), is a multi-domain and multi-scale co-simulation framework for connected and automated mobility. It couples a variety of traffic, communication simulators together to create complex simulation scenarios. Furthermore, it allows external simulators to be coupled as well if needed. Its Run-Time Interface (MOSAIC RTI) handles all management tasks for the life-cycle of a simulation, such as, the federation management to start and stop simulators, the interaction management to exchange data between simulators, and the time management to synchronize the event processing in all simulators.

The chosen simulator for the microscopic traffic simulation is SUMO (Sub-Section 4.1.2), an open-source simulator able to replicate the kinematics of multiple vehicles, customize the traffic environment and control the entities within a simulation using its TraCI (Traffic Control Interface) interface. For the network simulator the Simple Network Simulator (SNS) was chosen (Sub-Section 4.1.3). Besides already being coupled with Eclipse MOSAIC, SNS allows the transmission of V2X messages using Ad-hoc communication. The communications established make use of Ad-hoc communication channels that obey to the IEEE802.11p standard protocol communication. As for the MAS framework, the Jadex Active Components Framework was selected. It allows for a BDI compliant agent modelling that is Java based, thus facilitating the coupling with Eclipse

MOSAIC. Furthermore, Eclipse MOSAIC's Application Simulator allows the modelling of the application logic for different simulation units (e.g. vehicles, road side units (RSUs), traffic lights, and others) as well as possible interaction attempts between the units via different communication links. Figure 4.1 presents a high level system architecture diagram describing how the different components are connected. Eclipse MOSAIC RTI is used to communicate between simulators. As such, each individual simulator is decoupled from one another. Sharing information between them is done through Eclipse MOSAIC's interactions.

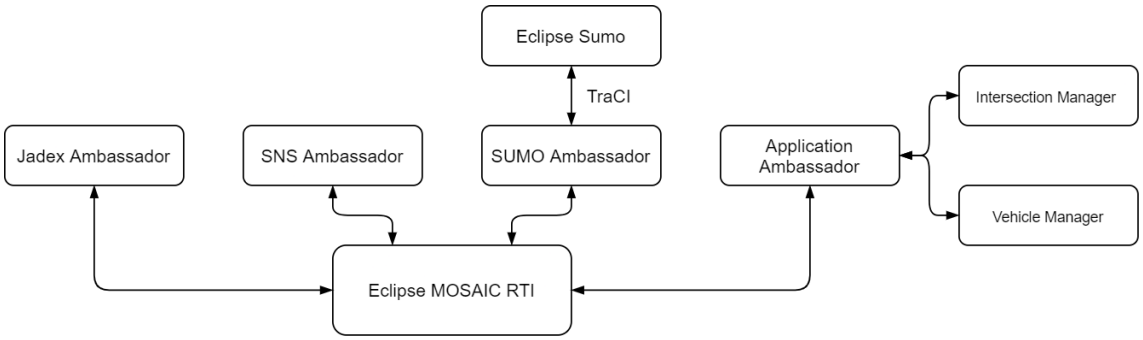


Figure 4.1: System architecture

4.1.1 Eclipse MOSAIC - mobility simulation framework

Eclipse MOSAIC [SHH+15], previously known as VSimRTI, is a multi-domain and multi-scale co-simulation framework for connected and automated mobility. It couples a variety of simulators together to create complex simulation scenarios. Eclipse MOSAIC also enables the coupling of external simulators by employing the concept of Federates and Ambassadors. For any simulator coupled with Eclipse MOSAIC, a federate It consists of the original simulator that is connected to its federate ambassador and an instance of an Eclipse MOSAIC ambassador. The communication between the federates and the run-time infrastructure is enabled by ambassadors that provide such services. This is the strategy used for the already coupled traffic and communication simulators and other evaluation tools supported by the framework. As of today, the set of simulators already coupled with Eclipse MOSAIC are:

- **Traffic simulators** - SUMO and PHABMACS.
- **Communication simulators** - NS-3, OMNeT++, MOSAIC SNS, and MOSAIC Cell.
- **Application simulator** - MOSAIC Application.
- several visualization and analysis tools.

The MOSAIC RTI (Run-time Infrastructure) handles all management tasks for the lifecycle of simulation, such as, the federation management to start and stop simulators, the interaction management to exchange data between simulators, and the time management to synchronize the

event processing in all simulators. Figure 4.2 shows a schematic illustration on how the MOSAIC's run-time interface interacts with any given simulator.

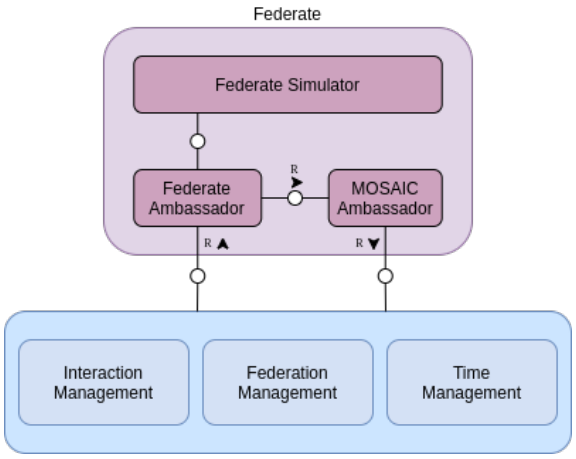


Figure 4.2: Eclipse MOSAIC Run-time Infrastructure

Besides allowing the coupling of external simulators, the Application simulator provides the ability to model the application logic for different simulation units (e.g. vehicles, road side units (RSUs), traffic lights, and others) as well as possible interaction attempts between the units via different communication links. Available interfaces implemented by MOSAIC allow control of a simulation in a quick and easy way, with convenient methods that are able to read and edit data regarding the different simulation units.

When setting up a MOSAIC scenario, the folder structure must be taken into account. Each component has a respective folder with an adequate name (i.e.sumo, sns, application) and a configuration file which consists of a JavaScript Object Notation (JSON) file containing useful data for the different simulators to use upon the simulation's startup.

Choosing Eclipse MOSAIC mobility framework reduces the complexity of the implementation as it already couples a traffic and a network simulator that only require further configuration to be ready to use. Furthermore, MOSAIC's Application Simulator simplifies the process of developing custom behaviors for the simulation units (vehicles and RSUs). Finally, Eclipse MOSAIC simplifies the process of coupling any tools or simulators if needed, which is the case here with Jadex MAS framework.

4.1.2 SUMO - road traffic simulator

The road traffic simulation component of this implementation is SUMO (Simulation of Urban MObility). SUMO is an open-source simulator with multiple features, namely from different models of vehicles (including public transports, bikes and even pedestrians) to tools capable of estimating pollutant emissions, importing existing networks, and graphical interfaces that make the process of building custom traffic scenarios much easier.

The justification for the selection of the SUMO for vehicular simulation is due to meeting the following criteria:

- Criteria 1 – Microscopic traffic simulator, allowing for the simulation of individual vehicles and infrastructures.
- Criteria 2 – Open-source.
- Criteria 3 – Fast execution speeds.
- Criteria 4 – Control of the simulation from external sources using a Traffic Control Interface (TraCI) API.
- Criteria 5 - Coupled with Eclipse MOSAIC.
- Criteria 6 - Support for simulating vehicle platooning.

A significant feature of SUMO is its TraCI API to allow for remote control of the simulation. SUMO is already coupled with Eclipse MOSAIC as one of its traffic simulators and provides an easy to use interface obeying to the TraCI command syntax. Figure 4.3 shows how SUMO is connected to MOSAIC's run-time interface. In their implementation, control is done using TCP sockets. A TraCI server offers an interface to exchange commands using a byte protocol. To communicate with the server, a TraCI client is used which implements the TraCI protocol, allowing SUMO to be used with no modifications.

For further configuration of road traffic networks, vehicle properties, traffic demand and any additional features, SUMO offers built-in tools and the option of defining custom XML configuration files obeying to the SUMO's predefined syntax.

4.1.3 SNS - simple network simulator

In order to allow messages to flow from one vehicle to another, as well as from vehicles to infrastructures, a network component capable of mimicking wireless vehicular networks was added to the solution implementation. The Simple Network Simulator (SNS) aims to provide simple and fast capabilities for the transmission of V2X messages using Ad-hoc communication. Using the SNS simulator, transmission models can easily be configured through JSON configuration files. Furthermore, SNS is already built-in Eclipse MOSAIC, meaning that integrating wireless communication is as easy as setting up a new configuration file with the respective parameters.

Communication via a specific Ad-hoc channels in an Ad-hoc network can be achieved using the standardized 5.9 GHz frequency band (5.85-5.925 GHz) established in the IEEE 802.11p standard communication protocol for Intelligent Transportation Systems(ITS). Table 4.1 shows the possible channels on the 5.9 GHz band used for V2X communication.

Two different types of Ad-hoc transmissions are differentiated by SNS: geographically and topologically-scoped transmissions. Geographical transmissions are limited to broadcasting, no explicit receivers are defined, only a destination area is specified for this type of communication.

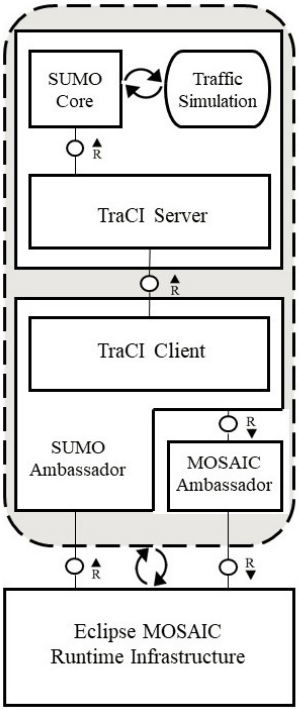


Figure 4.3: SUMO connected to Eclipse MOSAIC [FOK99]

Channel Number	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Channel Name	SCH1	SCH2	SCH3	CCH	SCH4	SCH5	SCH6
Frequency Band	5.86	5.87	5.88	5.89	5.9	5.91	5.92

Table 4.1: Communication channels and frequencies for V2X communication

On the other hand, topological transmissions allow for both braodcasting an unicasting transmissions.

The transmission models used for topological transmissions is the *SimpleAdhocTransmissionModel*. This model only checks, whether a potential receiver is inside the destination area and has enabled Ad-hoc capabilities. If those conditions are met it will simulate the transmission by calculating an actual delay value and saving it into a transmission-result. Such a result holds information of the success of the transmission, the delay-value, the amount of hops (always 1 for this model), and the number of attempts.

4.1.4 Jadex - MAS Framework

The framework chosen for the job was Jadex Active Components Framework, an agent platform used for distributed and concurrent systems. The Active Components design considers systems to be composed of components that act as service providers and consumers. It extends the Service Component Architecture (SCA) approach in the direction of agents with the possibility of abstract business logic implementation based on BDI Agents or BPMN Workflows.

The designed architecture for the Jadex component is rather simple as no communication between agents is required. The bidding component's unique responsibility is the bid calculation. Figure 4.4 illustrates how the Jadex module is coupled with Eclipse MOSAIC Run-time interface. Contrary to SUMO and MOSAIC's coupling,the Jadex Ambassador implements the agent's logic itself as no external communication channels are required, thus acting as both the ambassador and federate of the coupling. In order to generate new agents as vehicles spawn within the simulation, a Jadex Platform is constructed, which provides a specific service for the agent creation process by the name of *ComponentManagementService*. In the designed implementation, agents are represented by components and every time a *VehicleRegistration* interaction is sent by MOSAIC's run-time interface, the *ComponentManagementService* instantiates a new agent.

As for the agent logic itself, only one agent type was implemented, the *BidingAgent*, which implements the bid service where the vehicle calculates its bid value. As of now, that is the only service that is implemented, however it can easily be extended as adding services is a rather quick and straight forwards process. A BDI approach was used to model the agents with Jadex. As of now, the BDI design is very simple. A bidding agent only contains two different beliefs: the MTA which is known immediately after a vehicle spawns, and the current ETA which is a dynamic belief that is updated once a *RequestBidCalculation* interaction forwarded by MOSAIC's RTI is received by the Jadex Ambassador. Such interation updates the current ETA belief which triggers a plan with the goal of calculating a bid with the set of MTA and ETA values known.

The BDI modelling approach is rather simple as vehicles do not communicate with each other using the MAS framework and due to the bidding process only depending on two different variables: MTA and ETA. Nonetheless, extending the current integration should not pose as a challenge if the bidding process evolves and other beliefs are required.

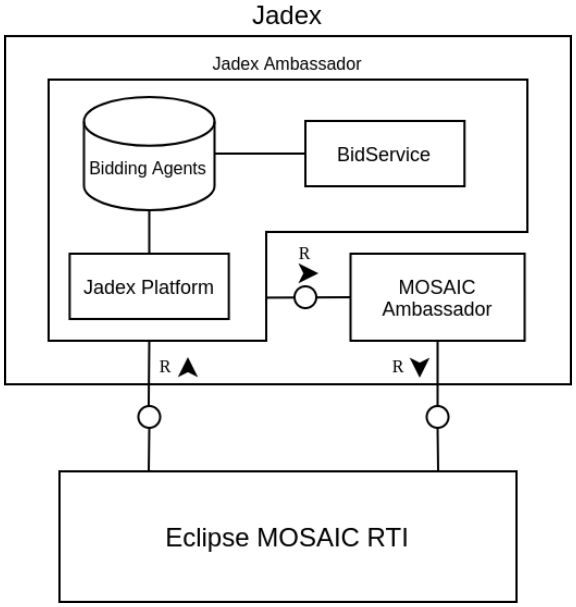


Figure 4.4: SUMO connected to Eclipse MOSAIC

4.2 Communication

This section presents a description of the messages used to exchange data between the different components of the simulation. As far as communication goes, the developed solution encompasses three main forms of communication.

- **Internal Communication:** Due to Eclipse MOSAIC's ability to deploy more than one application in a simulation entity (vehicle or intersection manager), internal communication concerns the message exchange between applications of the same entity.
- **V2X Communication:** involves the exchange of messages between vehicles and the intersection manager. Also, in platooning vehicles, messages are exchanged between a platooning leader and its followers.
- **Federate Communication:** The previous two forms of communication occur within Eclipse MOSAIC's application simulator. Federate communication concerns the exchange of data between the different simulator running through Eclipse MOSAIC.

The following sections present each form of communication mentioned above, and detail their implementation in the designed solution.

4.2.1 Internal Communication

As previously described, to customize any entity's behavior (vehicle or RSU), Eclipse MOSAIC allows the development of one or more custom applications to be deployed in a simulation entity at

run-time. Communication between the different developed applications is required for exchanging any relevant data between any entity's running applications.

Eclipse MOSAIC allows this type of communication through the use of events by implementing an event queue. On the sending side, whenever an application decides to send information to another application, it first gathers a list of all the running applications in its entity. It then constructs the message to be sent. Every message sent contains a message type which identifies the type of message that is being sent. According to this message type, the message payload (data to be exchanged with another application) will also differ. Once a message is fully constructed the application iterates over the list of running applications previously retrieved and adds an event to the event queue of each application. In its essence, it works as a broadcast as every application running will receive every message sent by another application.

On the receiving end, each application implements a listener method that listens for new events added to the event queue. Once an event is received, each application processes its message accordingly. To process a message, an application simply has to analyse the message type. For example, the vehicle application implemented to estimate a bid value acts upon the reception of an event containing a message of type CALC_BID(signaling that it is time to calculate the vehicle's bid). However, if the same application receives any other event unrelated to bid calculation, it simply ignores it. The implemented solution focuses on constructing such messages for two different entities: vehicles and intersection managers.

4.2.1.1 Vehicle Messages

The component diagram in figure 4.5 shows the implemented vehicle applications and the communications between them. Four different applications were developed for controlling the vehicle.

- **Vehicle Manager:** does all the communication between the vehicle and the intersection manager.
- **Calculate Trajectory:** responsible for computing a vehicle's trajectory through the intersection
- **Validate Trajectory:** verifies the trajectory computed by the previous module against the received reservation map. Resolves conflicts if any occur by delaying the departure of the current trajectory.
- **Calculate Bid:** calculates a vehicle/platoon bid. Communicates with the Jadex Federate so that the vehicle's assigned agent computes the bid based on the criteria presented in 3.2.1. In case of platooning vehicles, it establishes vehicle-to-vehicle communications with a platoon leader and its followers.

Figure 4.6 depicts how the different vehicle applications communicate with one another through messages. Upon receiving the reservation map information from the intersection manager, the *Vehicle Manager* of platooning leaders as well as independent vehicles share such information with

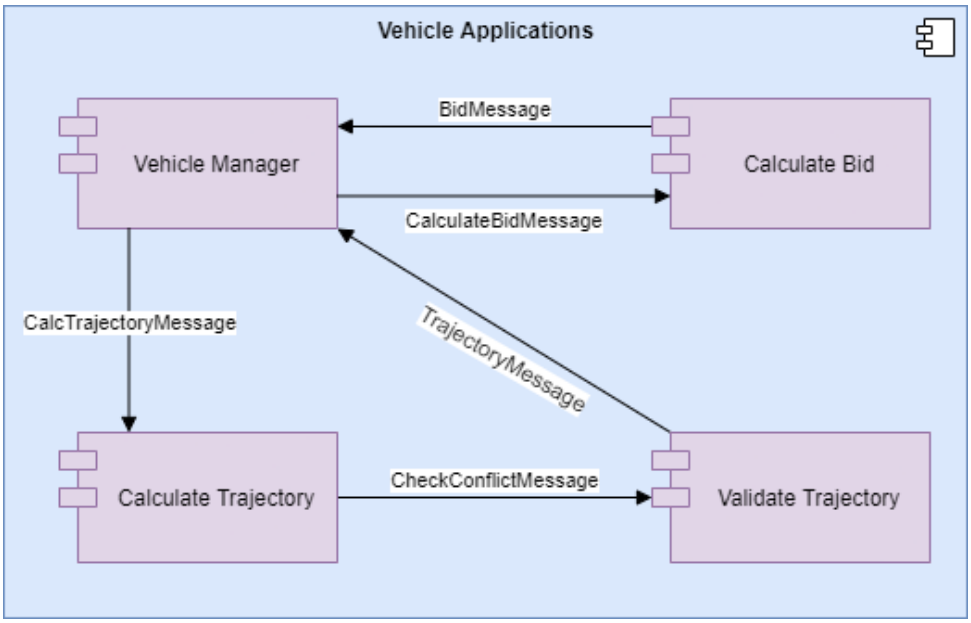


Figure 4.5: Message components and messages exchanged between vehicle applications

the calculate trajectory module by sending a *CalculateTrajectoryMessage* to the *Calculate Trajectory* application (platooning follower vehicles ignore all the communication between the vehicle and intersection manager).

The *Calculate Trajectory* module then computes the trajectory according to the algorithm presented in 3.3.1.1. Once finished computing its trajectory with all the reservation-tiles the vehicle needs to bid on, it shares both the trajectory and the previously received reservation map with the *Validate Trajectory* module through a *CheckConflictMessage*.

After receiving the information sent by the *Calculate Trajectory* module, the Validate Trajectory application checks if any conflicts occur by testing each reservation-tile of the trajectory against its current reservations in the reservation map. If one or more conflicts occur, the reservation timestamps are delayed for each reservation-tile of the trajectory and a new validation is made. Once no conflicts are found, the trajectory is considered valid and a *Trajectory Message* is dispatched to the *Vehicle Manager*.

To finish the vehicle's logic, after computing the trajectory, the vehicle calculates its reservation bid. To do so, it first communicates with the *Calculate Bid* application through a *CalculateBidMessage* containing the vehicle's estimated time of arrival at its destiny. This application then communicates with a Jadex Federate responsible for computing the vehicle's bid based on the criteria presented in 3.2.1. After receiving the vehicle's bid evaluation from the Jadex Federate, it shares its value with the *Vehicle Manager* application through a *Bid Message*. This message flow occurs every time a vehicle desires to cross an intersection.

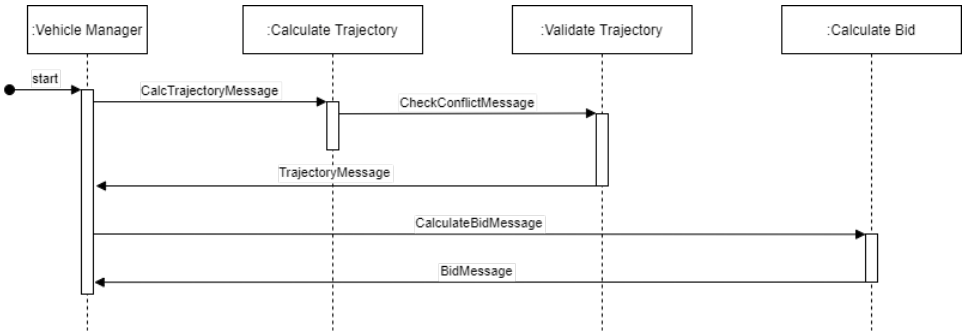


Figure 4.6: Message flow between vehicle applications

4.2.1.2 Intersection Manager Messages

Due to the decentralised aspect of the designed approach, the intersection manager's logic becomes simpler when compared to the previously presented vehicle message flow. Since the reservation calculation is performed at a vehicle level, all the intersection manager needs to do is approve or reject incoming requests. Furthermore, it constantly broadcasts the reservation map to incoming vehicles.

To mimic this behavior, the three following applications were implemented:

- **Intersection Manager:** Dispatches incoming reservation requests to the *Validate Trajectory* for conflict examination. It's where the information regarding the reservation map is located. Furthermore, it broadcasts the most updated version of the reservation map.
- **Validate Trajectory:** Responsible for validating incoming reservation requests. After assessing their validity it dispatches a response to the *Intersection Manager* application.

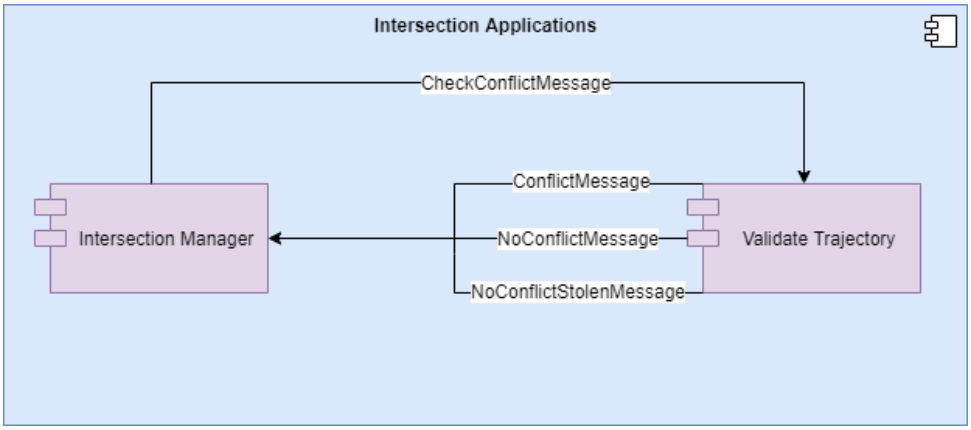


Figure 4.7: Message flow between intersection applications

In figure 4.7 illustrates the different modules of the intersection manager as well as the communications established between them. During the initialization of the intersection manager, as

soon as the reservation map is ready for broadcasting, the *Intersection Manager* is responsible for sharing its information with incoming vehicles. It does so every 100 milliseconds by broadcasting the reservation map.

Moreover, after receiving a vehicle's reservation request, the *Intersection Manager* dispatches a *CheckConflict* message to the *Validate Trajectory* application. The conflict evaluation is then performed according to the criteria presented in 3.3.1.2. This application then communicates a response to the Intersection Manager. Depending on its evaluation, there are three different responses possible:

- *ConflictMessage:* Conflicts occurred and requesting vehicle could not outbid conflicting vehicles.
- *NoConflictMessage:* No conflicts occurred and reservation was successful.
- *NoConflictStolenMessage:* Conflicts occurred and requesting vehicle outbid conflicting vehicles.

The Intersection Manager acts according to the message received. Every time any incoming request modifies the reservation map, a new *ReservationMapMessage* is sent to the *Broadcast* application to ensure that the broadcasting information is up to date.

4.2.2 V2X Communication

When it comes to exchanging data between simulation entities, the developed implementation allows for communication between vehicles and intersection manager or between a platooning leader and its followers. Together with the Simple Network Simulator (SNS), the communication between entities is done via an Ad-hoc network. Depending on the communicating entities, different Ad-hoc channels will be used for exchanging data. The designed implementation makes use of three different Ad-hoc channels:

- **CCH:** used by the *Intersection Manager* application of the intersection to periodically broadcast the reservation map information. The messages transmitted through this channel will roughly travel 80 meters as this is the configured travel distance of this Ad-hoc channel.
- **SCH1:** used by the *Vehicle Manager* and Intersection Manager applications to exchange messages regarding reservation requests and responses between them. Moreover, used by platooning vehicles to communicate with each other during the bid calculation process.

The sequence diagram in figure 4.8 describes the message flow between an independent vehicle and an intersection manager. Initially, the Intersection Manager is responsible for broadcasting the most updated version of the reservation map information. As presented in the previous section, this is done by the *Intersection Manager* application which dispatches a *BroadcastMessage* every 100 milliseconds through the *CCH* Ad-Hoc channel.

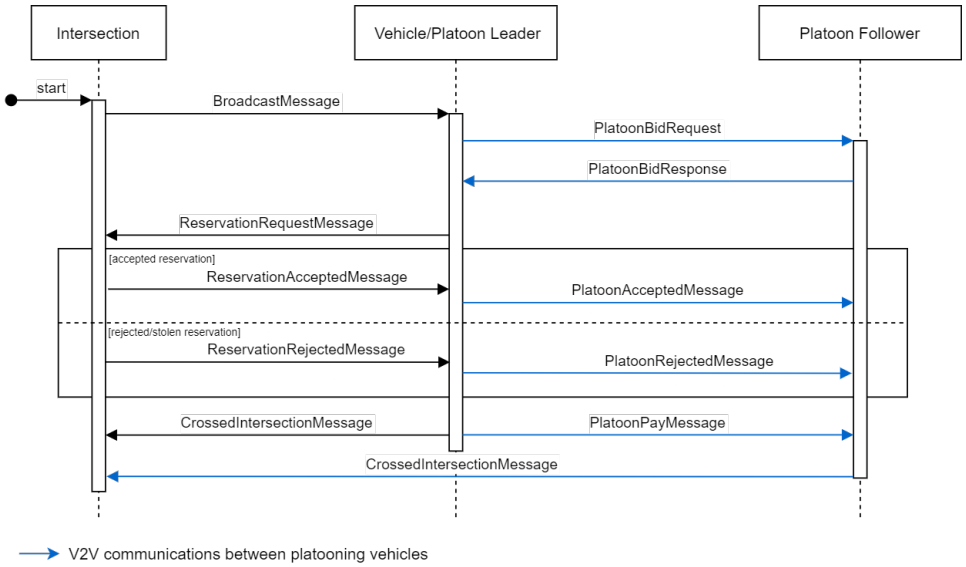


Figure 4.8: V2X message flow between vehicles and intersection manager

Upon receiving a *BroadcastMessage* by the intersection, the vehicle will follow the message flow detailed in figure 4.6 to calculate the reservation-tiles used by the vehicle when crossing the intersection, the time interval it requires to cross each tile and the estimated bid value. With all this information in hand, the vehicle sends a *ReservationRequestMessage* through the *SCH1* Ad-hoc channel to the intersection manager.It is important to notice that as soon as a vehicle receives a BroadcastMessage, it stops listening to messages of this type until it receives a response from the intersection manager to its reservation request.

After processing the received message, the intersection manager acts by evaluating the received request against its most updated reservation map. After evaluating the request, one of three outcomes can occur:

- **Reservation Accepted without Conflicts:** no conflicts occurred during the validation of a vehicle's request and thus the reservation was successfully made. Notify the requesting vehicle with a ReservationAcceptedMessage.

- **Reservation Accepted with Conflicts:** conflicts occurred and the vehicle outbid the conflicting vehicles. This results in the conflicting vehicles losing their reservation to the requesting vehicle. Notify the requesting vehicle with a *ReservationAcceptedMessage* and the conflicting vehicles with a *ReservationRejectedMessage*.

- **Reservation Rejected:** conflicts occurred and the vehicle did not outbid the conflicting vehicles. Notify the requesting vehicle with a ReservationRejectedMessage.

After identifying the correct outcome, the intersection manager notifies the involved vehicles using the previous messages. If a vehicle receives a *ReservationAcceptedMessage*, it simply stores

information regarding its reservation departure time and waits until it is time to cross the intersection. On the other hand, when receiving a *ReservationRejectedMessage*, the vehicle removes any instance of a reservation departure time (if exists) and starts listening again for new *BroadcastMessage* messages.

The described process also applies to platooning leader vehicles. However, when a platoon leader receives information about a successful reservation or a reservation that was lost, it needs to broadcast that information to its followers. In the presence of a successful reservation, the platoon leader uses the SCH1 Ad-hoc channel to broadcast a *PlatoonAcceptedMessage* to its followers who will update their intersection departure time accordingly. If a *PlatoonRejectedMessage* is received, the follower removes the departure time instance if present.

In addition, platooning leader vehicles also communicate with their followers in the bidding process so that their bid value estimations are taken into account in the platoon's final bid. When a platooning leader starts the bid calculation process, it first notifies its followers by sending a *PlatoonBidRequest* message through the SCH1 Ad-hoc communication channel containing the vehicle's ETA (Estimated Time of Arrival) necessary for calculating its bid. After receiving such message, a platoon follower simply communicates with the Jadex Platform through the Jadex Federate to calculate the vehicle's bid according to the received ETA. Once a response from the Jadex Federate containing the vehicle's bid is received, the follower forwards its value to the leader by dispatching a PlatoonBidResponse message. The leader, waiting for its followers replies, after receiving all of its followers bids, continues the reservation process as specified in 3.3.1.

Lastly, once a platoon starts crossing the intersection, each vehicle must pay its share of the bid according to the previously defined payment method from the payment methods specified in 3.2.2. During the paying stage, the platooning leader vehicle notifies its followers with a *PlatoonPayMessage* who, upon its reception, communicate with the Jadex Platform through its federate to withdraw the needed amount of credits from its current total credits.

It is important to notice that platooning follower vehicles only communicate with their respective platooning leader since there is no need for them to make requests to the intersection manager, as this is the responsibility of the platooning leader vehicle.

4.2.3 Federate Communication

Eclipse MOSAIC uses a specific mechanism for handling communication between the different federates running during a simulation. This mechanism is particularly useful when coupling new simulators with Eclipse MOSAIC as any simulator that is not already integrated with Eclipse MOSAIC requires the definition of a custom federate.

The exchange of data among federates is offered by the Interaction Management using interactions. Eclipse MOSAIC and its federates are decoupled through a publish-subscribe paradigm provided by the Interaction Management. A published interaction is forwarded to each subscriber directly after it has been published. Before a federate can receive interactions, it has to subscribe to them. The Interaction Management offers two different ways to subscribe to certain interactions. A federate can either define a interaction type only to receive all interactions of this type

or, additionally, define conditions to filter interactions by their content. If a federate is not longer interested in subscribed interactions, it can rescind its interest. Each federate is allowed to publish any interaction at any valid time of the simulation. After a interaction is published, the Interaction Management forwards the interaction to each federate that has subscribed to this type of interaction.

This dissertation's developed solution required the implementation of a custom federate to establish communication between the developed vehicle applications with the Jadex Platform used for the bidding estimation of every vehicle. That communication is made through custom developed interactions. The sequence diagram in figure 4.9 depicts the process of managing interactions between the vehicle application and the developed federate to calculate bids.

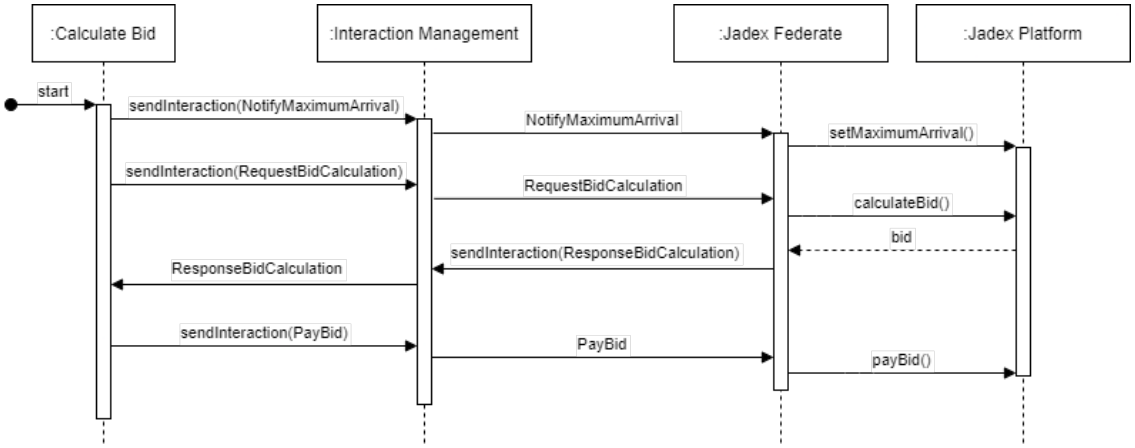


Figure 4.9: Interaction flow between a vehicle application and the Jadex Federate

The Jadex Federate developed subscribes to four different interactions and publishes one other. Based on the bidding strategy described in 3.2, to calculate a given vehicle's bid, two parameters must be known: the vehicle's MTA and ETA. Upon a vehicle's initialization, its MTA is calculated based on the distance it has to travel and the velocity used to travel such distance (assumed as constant velocity in the MTA estimation). After calculating the MTA, the *Vehicle Manager* application sends a *NotifyMaximumArrival* interaction to the Interaction Management. Since the Jadex Federate is subscribed to this type of interaction, it will process it and communicate with the Jadex Platform which will then initialize the respective agent with the given MTA.

Every time a vehicle needs to calculate a bid value, upon receiving a message containing the ETA (similarly to the MTA it's calculated based on the distance yet to travel and the vehicle's travel speed), it dispatches a *RequestBidCalculation* to the Intersection Management module. Once again the Jadex Federate will process such interaction and tell the respective agent within the Jadex Platform to calculate the bid's value according to the bid calculation formulas presented in 3.2.1. In response, the Jadex Federate forwards a *ResponseBidCalculation* containing the bid's estimated value.

After receiving such message, the vehicle's *Calculate Bid* application continues the reservation

process previously detailed. As soon as a vehicle starts crossing the intersection and a green light is sent for it to pay its bid (or bid share in the case of platooning vehicles) a *PayBid* interaction is sent to the Intersection Management module and subsequently forwarded to the Jadex Federate. The federate then communicates with the Jadex Platform and the respective bid value is withdrawed from the vehicle's agent wallet.

4.3 Summary

In this chapter the implementation of the solution presented in chapter 3 was presented. Three main components compose the architecture: a traffic simulator (SUMO), a network simulator (SNS) and a MAS framework (Jadex). To bind together each one of these components, Eclipse MOSAIC is used, which already couples the traffic and network simulators. The Jadex framework is coupled with MOSAIC using the same Federate/Ambassador concept implemented in all other already coupled simulators.

Moreover, the communication between the different simulation components is described. Three different types of communications were observed: *Internal Communication* for exchanging information within a simulation unit, *V2X communication* to exchange data between simulation units, and *Federate Communication* to exchange information between the several coupled simulators. In the next chapter (Chapter 5), the tested simulation scenarios are described and the obtained results are presented.

Chapter 5

Results and Discussion

In this chapter, two different scenarios are implemented to validate the designed approach. These scenarios intend to not only evaluate the performance of the model presented in chapter 3, but also compare it with current traditional forms of traffic management at intersections such as traffic lights as per Q_2 of the research questions specified in Chapter 1. In addition, the proposed negotiation approach is tested against a FCFS approach based on the designed solution to answer Q_3 of the research questions specified in Chapter 1. Furthermore, and to evaluate the behavior of the previous model in a more complex environment, a scenario containing several intersections was developed to emulate a urban setting.

To evaluate how platoons can impact the performance of vehicle coordination at intersections, both scenarios are evaluated in platooning and non-platooning environments to help answer yet another research question of this work, Q_1, presented in Chapter 1.

5.1 Evaluation metrics

This section presents a brief description of the evaluation metrics collected for the performance comparison of the different different coordination strategies in the implemented scenarios. Most of the parameters explained in this section are common to the scenarios that will be detailed in the following sections, unless otherwise stated. Any metric or parameter specific to an individual scenario is specified in their respective sections.

- **Throughput:** assessed at an intersection level, it is used to measure the rate at which vehicles cross any given intersection (in $vehicles/hour$). To measure this parameter, a SUMO detector was used. This detector consists of a series of sensors placed at the entrances and exits of the intersection capable of retrieving some data regarding the traffic that goes through, including the number of vehicles that traveled through the intersection area.

- **Average Delay:** the average waiting time experienced by the vehicles at an intersection (in *seconds*). Only vehicles that crossed a given intersection are taken into account for computing the average delay of said intersection. The average delay can be given by SUMO

using vehicle detectors. These sensors are capable of measuring traffic parameters such as the mean time loss within any given closed stretch of road.

- **Travel Time:** the time taken by a vehicle to complete its route (in *seconds*).
- **Bid Value:** the bid value sent by a vehicle to the intersection manager. Only values regarding successful reservations are accounted for (in *credits*).
- **Reservation Time:** the time it takes for a vehicle to successfully schedule a reservation (in *milliseconds*). Reservations stolen by other vehicles further ahead are also taken into account.
- **Reservation Requests:** the number of reservation requests received by the intersection manager from incoming vehicles/platoons (in *occurrences*).
- **Delay-Bid Correlation:** the correlation between the estimated bid values received by an intersection manager, and the delay experienced at the intersection by incoming vehicles.
- **Rejeted Requests Percentage:** the ratio of requests that are rejected from the number of total requests received by the intersection manager(in $rejected_requests/total_request$).

5.2 Platooning

During the development of the system architecture described in chapter 4, Eclipse MOSAIC proved to be unprepared for executing traffic simulations in SUMO that required more than one TraCI client to be connected to SUMO's TraCI server. This posed as a major inconvenience given that a second external TraCI client was a must for platooning to be part of the simulation, apart from the one initialized by Eclipse MOSAIC itself.

To benefit from SUMO's platooning capabilities, one of two different tools currently available can be used: Simpla or Plexe. Simpla is a platoon plugin for SUMO's TraCI Python client. It manages the spontaneous creation of platoons. Using Simpla, different platooning modes can be defined through the standard SUMO vehicle types. As of today, it does not allow for any custom control over platooning vehicles. Nonetheless, these features are not essential for the designed implementation since the only control required for a platoon is based on stopping at the beginning of an intersection if necessary, and resuming its route once allowed by the intersection manager.

Another existing tool used for platooning is Plexe, a cooperative driving framework that allows for the use and extension of various car-following systems, which, among others, includes multiple cruise control models, platooning network protocols and cooperative maneuvering within platoons. A recent addition to the SUMO official release were the car following models used for platooning by Plexe. These car following models can be controlled in SUMO through TraCI by using the Plexe Python API.

For the developed solution Simpla was chosen. The great advantage of Plexe is the ability to control a platoon's behavior, which in this scenario is not at all necessary. Furthermore, using

Simpla proves to be a quick and easy solution since only the definition of a configuration file is required.

Due to the inability of Eclipse MOSAIC to allow for the connection of two or more TraCI clients with the SUMO server, the coordination strategy described in Chapter 3 was implemented without the use of Eclipse MOSAIC. As such, and to be able to meet the delivery deadline, no network simulator was considered in this new integration and any information exchanged between vehicles and intersection managers is under the assumption of eventually being delivered successfully to its destination.

5.2.1 Platoon Generation

When it comes to generating platoons there is no specific SUMO vehicle type or configuration setting specifically designed to achieve a platooning behavior. Platoons are generated in a similar way as regular vehicles. SUMO allows the definition of traffic demand via extensible markup language (XML) files. Vehicles can be spawned spontaneously or periodically through the use of vehicle flows. In the case of platooning vehicles, defining vehicle flows is not an option as SUMO does not allow any variability to be inserted into a flow. Every platoon periodically spawned would have exactly the same amount of vehicles as the ones spawned prior to it. As a result, to generate platoons in a given simulation all of the spontaneous individual vehicle spawning must be specified in the respective XML file.

To allow for arbitrary size platoons to be spawned with custom vehicle demands ($vehicles/hour$) a script was developed to generate a SUMO route file with all of the platoons to be used in the simulation being deployed periodically. This script takes three major input parameters: simulation time ($seconds$), vehicle demand ($vehicles/hour$) and platoon maximum size. As a result, it calculates the number of vehicles to be spawned using equation 5.1 and generates platoons until the vehicle demand goal is met. The generated platoon's departure times are distributed evenly throughout the simulation with a period of $\frac{sim_time}{num_platoons}$, where sim_time is the simulation time in seconds and num_platoons is the number of generated platoons. Furthermore, the generated platoons follow a normal distribution regarding their number of vehicles, so that platoons are evenly distributed according to their size.

$$n_{vehicles} = \frac{sim_time}{3600} * demand \tag{5.1}$$

The platooning behavior that these vehicles display is achieved once Simpla is loaded into the simulation. With Simpla, regular vehicles defined in a SUMO route file can behave as platooning vehicles and adopt platooning car following models. Telling Simpla which vehicle types to act as platoons in its configuration file is all that is needed.

5.3 Scenarios

This chapter focuses on presenting the scenarios used to analyse the performance of the developed intersection management strategy. Sections 5.3.1 and 5.3.2 introduce the designed scenarios'

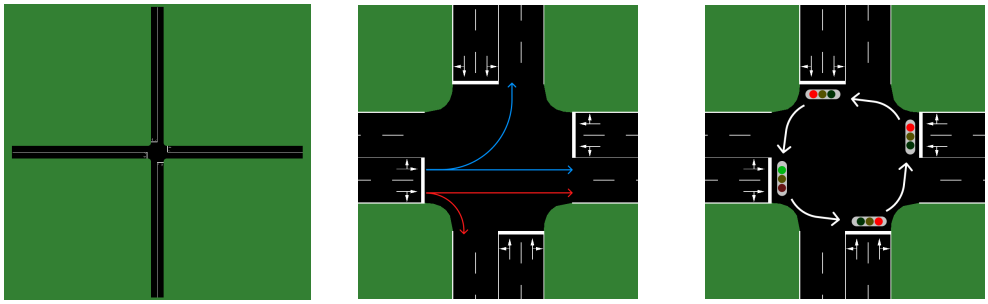
layout, the infrastructures used and the vehicle demand parameters used in the simulation.

5.3.1 Single Intersection

The single intersection scenario contains a four-way intersection. Each edge of the intersection is composed of two lanes. In total, the intersection has eight incoming and eight outgoing lanes. The length of the intersection is 150 meters long to allow bigger platoons to be fully inserted into the simulation prior to its leader handling any information from the intersection manager. Figure 5.1a presents the specified intersection layout. The defined vehicle types for the simulation are 4 meters long with a maximum speed of 10 m/s which translates to approximately 36km/h. The minimum safety gap established between vehicles is of 2 meters, which corresponds to the distance between stopped vehicles when waiting at the intersection.

Furthermore, the vehicle trajectories to be adopted while crossing an intersection are presented in figure 5.1b. Two different trajectories were defined for the two different existing lanes of every incoming edge. The defined behavior is standard to today's road design. The inner-lane allows a vehicle to turn left whilst the outer-lane allows it to turn right. To simply continue its travel path a vehicle can choose from any lane. This definition allows any incoming vehicle to continue its path through any of the three possible outgoing edges.

To compare the coordination strategy with traditional intersection management systems, one additional identical scenario was developed with the addition of a traffic light system. For the traffic light system, since any incoming vehicle is able to choose from one of the three different outgoing edges, in order to avoid conflicts, at most one edge can have a green phase at any given time. This traffic light definition means that access to the intersection is exclusive to the edge that has been given a green light. Figure 5.1c illustrates how the traffic light phases evolve over time.



(a) scenario layout (b) lane trajectories (c) traffic light phases

Figure 5.1: Single intersection

5.3.2 Intersection Network

To understand how the developed solution behaves and how the bidding mechanism affects traffic throughput and congestion of a road network, a sample intersection network scenario was designed. Figure 5.2 illustrates the developed road network made of nine different four-way intersections. Each individual intersection follows the structure presented in 5.3.1. The vehicle types

and parameters defined previously are also applied in this scenario, only the route definition differs. The generated vehicle routes for the executed simulation runs start and end in the outer edges of the simulation scenario. Similarly to the Single Intersection simulation scenario, a comparison with the traffic light management system is made, using the same traffic light phase definition illustrated in figure 5.1c.

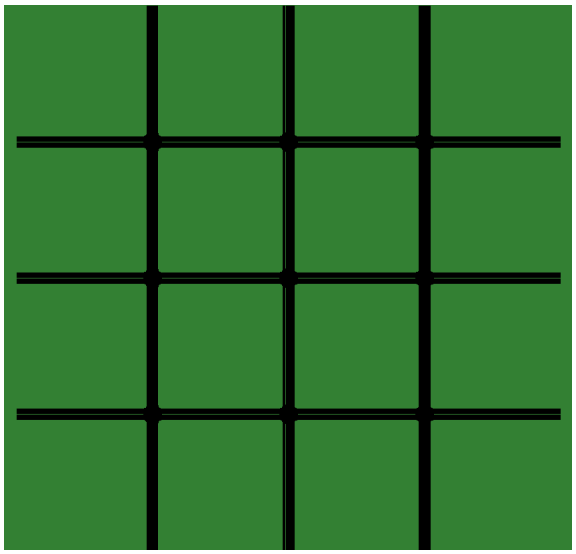


Figure 5.2: Intersection network scenario layout

5.4 Results Analysis

In sections 5.4.1 and 5.4.2 the results obtained by simulating the implemented scenarios described in 5.3 under different conditions are analysed.

5.4.1 Single Intersection

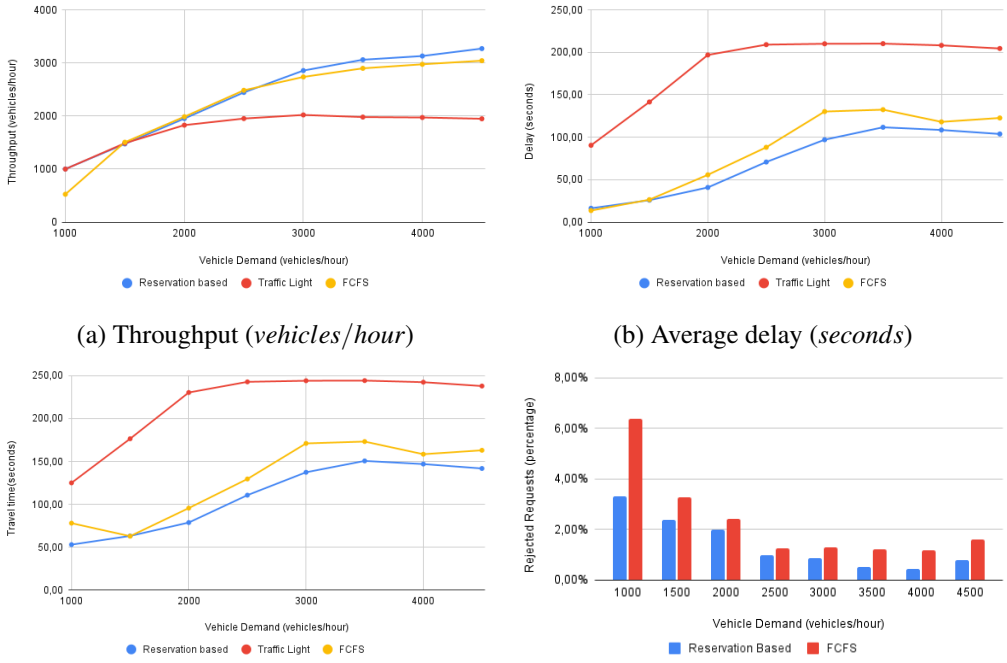
A first analysis focused on observing the performance of each strategy as the vehicle demand increases. Different vehicle demands were generated to emulate low, moderate and high traffic volume environments. Table 5.1 lists the traffic demand volumes for each experiment. A total of 8 different experiments were made, ranging from 1000 $vehicles/hour$ to 4500 $vehicles/hour$ with increments of 500 $vehicles/hour$.

Experiment Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Vehicle Demand	1000	1500	2000	2500	3000	3500	4000	4500

Table 5.1: Vehicle demand used in the experiments

For the reservation based strategy, an intersection map with granularity of 5 was used (meaning that the reservation map consists of a 5 by 5 matrix). As for the traffic light strategy, a default green light phase length of 20 seconds was specified. Moreover, regardless of the strategy used, all of

the experiments emulated a total of 30 minutes of simulation time with 50% of the traffic demand being made of platooning vehicles.



(a) Throughput (*vehicles*/*hour*)

(b) Average delay (*seconds*)

(c) Average travel time (*seconds*)

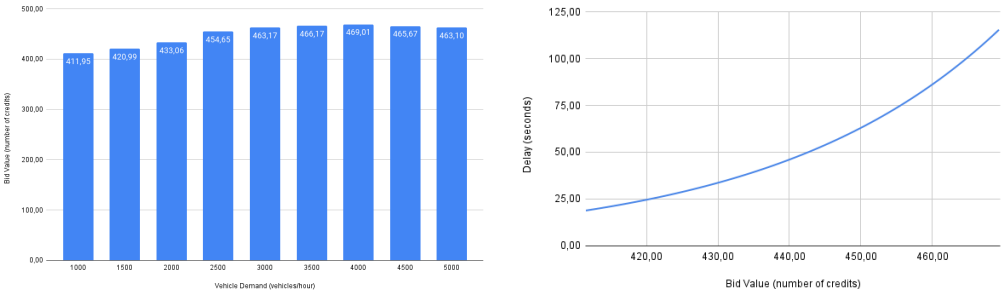
(d) Number of rejected requests

Figure 5.3: Vehicle demand analysis

Figure 5.3a depicts the intersection throughput as a function of vehicle demand. For low traffic demands, the performance difference is not very significant between the autonomous approaches and the traditional approach as the traffic light green phase is long enough to flush all of the waiting vehicles in an intersection edge with time to spare. However, when increasing the traffic demand it is clear that the traffic light strategy has a hard time handling vehicle demands higher than 2500 *vehicles*/*hour*. At this stage, 20 seconds of green light phase are not enough to flush all waiting vehicles in a given intersection edge, due to the vehicle build up in between a traffic light cycle 5.1c. On the other hand, the designed reservation based approach shows a better ability of handling higher vehicle demands by allowing vehicles from multiple lanes to cross at the same time. This behavior results in a much smaller build up of vehicles at each intersection edge. In return, an improvement of up to 75,8% can be obtained when compared to the traffic light management system. The comparative analysis between the negotiation based approach and the FCFS approach shows very similar throughput values and behavior as vehicle demand increases. The same behavior is present when observing the average delay. Even though the developed negotiation based approach shows slightly better results in both of those metrics, the differences are rather small when compared to the ones observed between these approaches and the traffic light system. The small improvements provided by the negotiation based approach can be a result of the number of conflicts occurring being higher too. Since the FCFS does not try to solve any conflicts at the intersection manager (it only accepts or rejects vehicle requests based on the

conflicts found), a higher percentage of rejected requests arises, as suggested in figure 5.3d. It is interesting to note that as vehicle demand increases the percentage of requests that are rejected by the intersection manager decreases too. Moreover, the analysis of the average travel times in figure 5.3c shows a similar behavior to the experienced delay due to the scenario's nature of only having one intersection.

When it comes to the average delay experienced by the vehicles, the reservation based approach shows, once more, promising results by reducing, at least in half, the delay experienced in a traffic light system. The analysis of both Figures 5.3a and 5.3b shows direct correlation between the average delay and throughput experienced at an intersection in the reservation based approach. As the throughput increases, so does the delay experienced due to the increasing vehicle build up at the intersection.



(a) Average bid value

(b) Delay-Bid correlation

Figure 5.4: Vehicle demand analysis

In figure 5.4a the variation of the average bid value received by the intersection is presented as a function of the vehicle demand. A slight increase in the average bid can be observed as vehicle demand increases as a result of the increasing delay experienced by the vehicles waiting at an intersection. As time goes by, the vehicle's urge to cross the intersection increases resulting in higher bid values from requesting vehicles. Such behavior is proven through the analysis of 5.4b which shows the exponential correlation between the average delay values obtained with the increasing demand and the average bid values estimated. An exponential correlation was chosen instead of a linear or polynomial correlation since it posed as a better approximation to the true delay values. It is possible to observe a growth in delay as the average vehicle bid value increases, which is prompted by the increasing vehicle demand.

The analysis of figure 5.5 shows how the average reservation time varies as vehicle demand increases. The reservation time is defined as the time interval between the moment a vehicle receives the information from the IM and the moment it receives a confirmation of a successful reservation being made. As seen in figure 5.5 even at high volumes of traffic demand (5000 vehicles per hour), the average reservation time takes less than 100 milliseconds to perform the vehicle and IM protocols detailed in sections 3.3.1 and 3.3.2, respectively.

The study of the reservation time poses as a network analysis that helps assess if the developed solution is feasible in real-time situations. Results show that even in high volumes of traffic

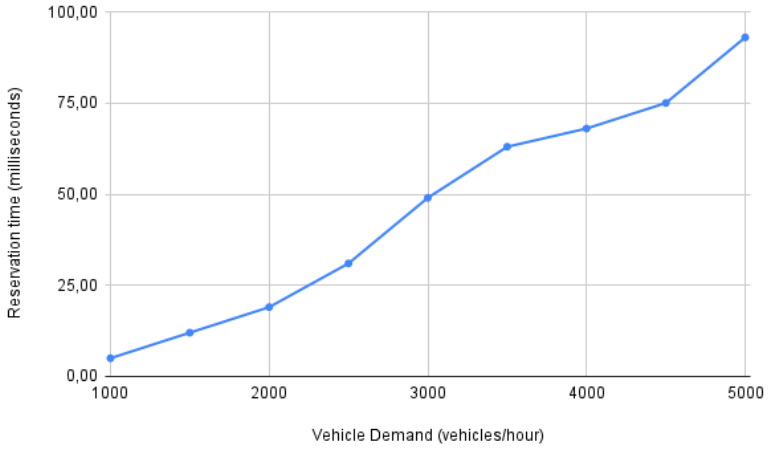
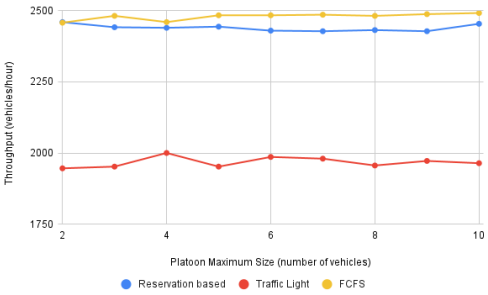


Figure 5.5: Time to acquire reservation (*milliseconds*)

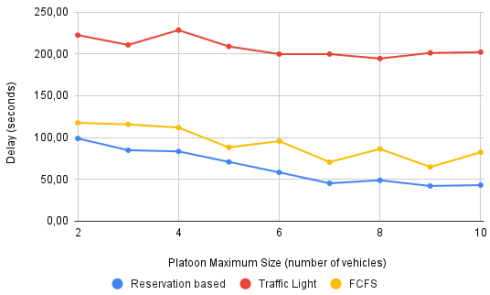
demand, the network is able to deliver reasonable reservation time values. Besides the reservation process taking very little time from start to finish, the approach is fairly quick due to few requests being rejected, regardless of the traffic demand. Furthermore, since only the first vehicle of a lane is allowed to perform reservations, the messages exchanged are fairly limited. At most, in the scenario defined in 5.1a, 8 vehicles can perform reservation requests at a time.

Moreover, the performance impact that platooning vehicles have at an intersection level is discussed. For such analysis a vehicle demand of 2500 *vehicles*/*hour* was used, with 50% of the demand consisting of platooning vehicles. Similarly to the vehicle demand analysis, a default green light phase length of 20 seconds is provided.

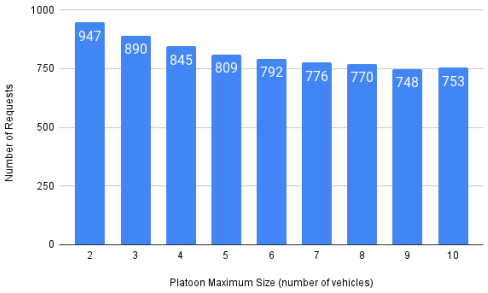
Figure 5.6a shows the vehicle throughput at an intersection as a function of the platoon's maximum size. For both the reservation based and traffic light systems, the throughput remains constant regardless of the platoon's size. This behavior is expected as the traffic light system isn't prepared to handle platoons at all, treating each vehicle as independent. As soon as a green light phase finishes, its vehicles must stop as part of the traditional driving rules, even if it means that a platoon will be broken of. As for the reservation based approach, the constant throughput results are based on two different causes: firstly, the intersection throughput is almost equal to the vehicle demand, meaning that the intersection is able to handle every vehicle's request; secondly, the platoon generation tool adapts the number of vehicles being generated to the vehicle demand defined. As such, no "extra" platoons are generated in order to maintain vehicle demand. Once again, the FCFS approach displays similar behaviors to the ones observed during the vehicle demand analysis. Very similar in terms of throughput values and slightly worse than the negotiation based approach in terms of experienced delay due to, once again, the higher percentage of rejected requests due to conflicts. Figure 5.6b analyses the experienced delay with the increasing maximum platoon size. As expected, the traffic light system displays a nearly constant behavior for the same reason as before: traffic lights are not prepared for vehicles in a platooning context. However, the reservation based approach is able to reduce the experienced delay in nearly half its



(a) Throughput (*vehicles*/*hour*)



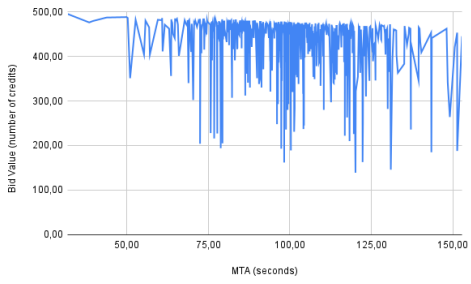
(b) Average delay (*seconds*)



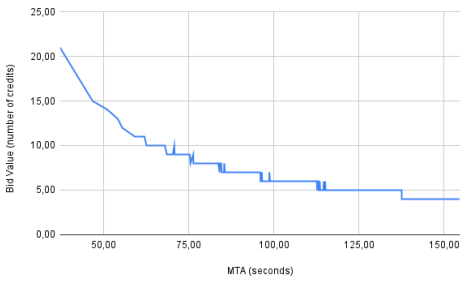
(c) Number of requests

Figure 5.6: Platoon maximum size analysis

initial value as the platoon's maximum size increases. The main reason behind this is the fact that increasing the platoon size reduces the number of platoons needed to achieve the same vehicle demand. As less platoons are generated throughout the simulation time, less reservation requests will be made to the intersection manager. As a result, a greater amount of vehicles can cross the intersection at a time, without the need of any of the extra reservation requests made by smaller platoons in a low maximum platoon size environment. Figure 5.6c shows the number of requests received by the intersection manager as a function of the platoon maximum size, with a reduction of 200 requests being observed by the intersection over a span of 30 minutes of simulated time by allowing platoons with a higher number of vehicles.



(a) Linear



(b) Non-linear

Figure 5.7: Bid values with respect to spare time

To assess the designed bid calculation functions' behaviors, random maximum times of arrival

(MTA) were assigned to the simulated vehicles following a normal distribution with a mean of 100 seconds and a standard deviation of 20 seconds. This allows for vehicles with different MTAs to be spawned and more variability to be added to the bidding process. For the evaluation of the MTA's influence in the bid calculation to be valid, each vehicle starts the simulation with the 500 credits, as the amount of credits directly influences the estimated bid value. The analysis of figures 5.7 shows the bid values as a function of the random additional time that is added as the time a vehicle can waste at the intersection. For the linear bid calculating function 3.2, although the generated bid values follow a linear distribution, their estimation is not consistent. In figure 5.7a,the steep fluctuations observed represent close spare times that output very different bid values due to the time at which they were calculated. For example, a vehicle that has a current ETA at t=50 and MTA of t= 70, using the bid function 3.2 calculates a bid of approximately 70% the total amount the vehicle is allowed to spend in that intersection. However, another vehicle with a current ETA of t=70 and MTA of t=90, using the same function outputs a bid value of approximately 78% the total amount allowed. Both vehicles have exactly 20 seconds to spare however their bid values differ. As time goes by this difference only becomes more obvious. Despite the desirable linear behavior of this bid function, the fairness of the output value is questionable. Additionally, depending on the time as a relative measure poses as a poor design for a bid calculation function.

On the other hand, figure-5.7b illustrates how 3.1 displays a more desirable behavior. As spare time increases the bid values tend to decrease, adopting the desirable behavior of having bids evolve according to their urgency. The advantage of the non-linear bid function designed is the fact that it solely depends on the time differences observed between a vehicle's MTA and current ETA. As such, the above example that was given to represent the inconsistence of the linear function would ouput the same values if using the non-linear approach. However, a non-linear approach to the bid function still has its drawbacks has vehicles display a rather desperate behavior. Combining both the linear behavior of function 3.2 and consistency and fairness of function 3.1 could be the next step to take in the pursuit for the most desirable bid calculation function.

5.4.2 Intersection Network

Regarding the intersection network, a first analysis focuses on the vehicle's bidding behaviour along its route. To prepare the simulation a couple of vehicle routes were defined. Figure 5.8 illustrates the four different routes established having vehicles start their trip at the top left corner of the simulation scenario and finishing in the scenario's bottom right corner. This particular definitions is ideal to assess the bidding behavior at each intersection since it avoids having the same intersection in a position where it can be both the first and last intersection in different routes. In this case, intersections i0 and i8 are always the first and last intersection that the vehicles cross and the order is maintained throughout the route's intersections.

Figure 5.9 presents the bid distribution values at each intersection. Note that intersections i2 and i6 do not have any bid distribution values since no vehicle crosses i2 and i6 given the specified routes in 5.8. The analysis of the box charts in 5.9 reveals that as a vehicle progresses through its

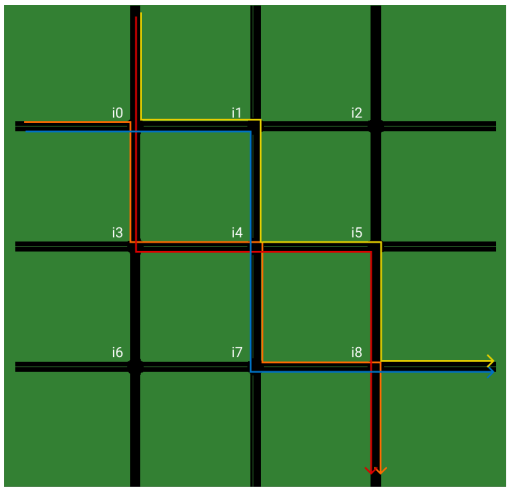


Figure 5.8: Vehicle routes

route, vehicle bids tend to increase. Such behavior is a direct consequence of the natural reduction of spare time (time that a vehicle can waste at intersections while still arriving on time - before its MTA) as time goes by. However, this growth is attenuated since the vehicles' bids depend on the number of intersections yet to cross along its route. One major drawback of the designed bidding calculation strategy comes to light through the analysis of figure 5.9. Even though it is a desirable behavior to have vehicles increase their bid as their spare time decreases, an unnecessary natural growth is also observed during a vehicle's route. If we were to spawn a single vehicle in the simulation environment, with one of the routes defined in 5.8, the same ascending behavior in the vehicle's bids would be observed, even though no time wasting would occur at intersections as no other vehicles compete for reservations. The developed bidding strategy shows a tendency to overbid in situations where the spare time is long enough for a vehicle to complete its route with ease or when in low traffic volume conditions.

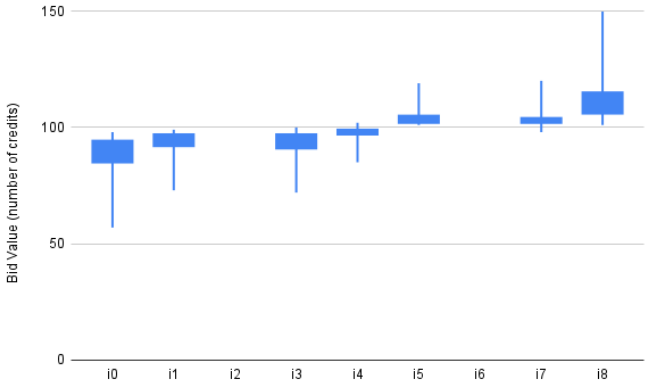


Figure 5.9: Bid value distribution

5.5 Summary and Discussion

The analysis of the obtained results in section 5.4 shows a promising performance when compared to traditional intersection management systems. The designed solution can significantly increase throughput while maintaining lower average delays than traffic light systems, regardless of the vehicle demand. When it comes to platooning vehicles, the increase of its size shows a considerable reduction in the delay experienced at the intersection caused by the decrease in the number of requests received by the intersection manager, thus relieving the intersection manager of some computational load. When comparing to a FCFS approach, although showing better results, the improvements observed are much smaller with a very similar throughput and a small difference in the average delay experienced. As such, the designed reservation based approach considering bidding mechanisms as a way of solving conflicts, can be considered a good starting point that could be worked upon for further improvements in the management of autonomous vehicles or platoons at intersections.

On the other hand, the results from the designed bidding protocol, although displaying some desirable behaviors, such as the increasing nature of a bid as spare time decreases, could be improved to ensure that inconsistencies concerning the outputted bid values are not present in the designed bid functions. The implemented bid calculation functions showed that applying a bid calculation formula based solely on a vehicle's spare time throughout its trip might result in overbidding as this value tends to decrease naturally over time. Another form of representing a vehicle's delay might be required to avoid overbidding in low traffic situations or when a vehicle has enough time to spare.

Given the analysis of the results, the developed implementation can be considered a successful approach to coordinating vehicles through intersections through a bidding mechanism. The developed solution can fulfill its primary goal of allocating reservations to vehicles. Better performance than traditional intersection management strategies is achieved. It is, however, essential to note that despite the performance gains, improving the designed bidding protocol, regardless of whether it boosts the mechanism's performance or not, is critical to ensure that a correct, fair, and consistent bidding strategy exists within the implementation.

Chapter 6

Conclusions

The developed work focused on the design and implementation of an intersection management strategy based in negotiation mechanisms in a platooning context. A literature review was made to better understand the current state of the art of negotiation mechanisms in ITS systems, explore intersection management strategies and platooning capabilities. An attention to detail was given to platooning scenarios and intersection management research as it is the focal point of this dissertation. Regarding intersection management, its study has primarily been centered around auctioning strategies and the impact of having vehicles collaborating at crossing sites. However, further optimisation through the use of platoons at intersections has been disregarded.

As a result, and as a way to fill in the gaps identified during the review of the literature, a multi agent implementation of an intersection management was designed, considering the presence of both platoons and independent vehicles. The strategy itself acts as a reservation based bidding mechanism in which vehicles request reservations to portions of the intersection by assessing how valuable an intersection appears through

6.1 Contributions

A proposal of contributions was described in section 1.4, and all tasks set forward to achieving can be marked as completed. As a first contribution, the development of a simulation platform for the proposed intersection management strategy was made, considering realistic constraints. Such implementation helps validate the designed solution and evaluate its performance in different scenarios.

With the simulation platform implemented, an analysis on the modeled bidding protocol regarding both independent and platooning vehicles is another contribution of this work. From this study, many possible paths for future research were made clear, as some benefits and drawbacks of the implementation were discussed.

The last contribution of this work was made through a comparative analysis of the implemented solution with both a traditional intersection management system (traffic lights) and a FCFS approach. The developed solution, even though in an initial stage of implementation, shows great potential with room for improvement.

6.2 Future Development

Some potential improvements to the work developed during this dissertation can be worked on as future development. In what concerns the implementation architecture in Chapter 4, a first improvement to be made is the addition of platooning capabilities when using the mobility framework Eclipse MOSAIC to execute the simulation scenarios. As referenced in section 5.2, Eclipse MOSAIC is not prepared to handle multiple TraCI client connections to a SUMO server, which is crucial for platooning capabilities to be added to the developed integration. Extending Eclipse MOSAIC's functionalities to support multiple TraCI clients could be the first step towards this goal.

Furthermore, regarding the bidding protocol established, as observed in 5.4.1, consistency issues arise due to the designed bidding functions. Implementing a consistent bidding function that can represent urgency in respect to time could pose as future research. Alternatively, estimating the bid value can be designed as a more complex problem that not only depends on a vehicle's MTA and ETA, but takes into account the state of affairs of the vehicle's route for a more thoughtful decision.

6.3 Future Research

The developed worked acts a starting point to build upon and expand from with some future research. In what concerns the developed approach, no fairness strategies were evaluated to ensure a balanced flow and distribution of reservations throughout the intersection's incoming edges. Future research on this topic could include bid appreciation with respect to a vehicle's waiting time, vehicle starvation avoidance (a vehicle never winning a reservation), and cooperation between in-lane vehicles to avoid any waiting vehicles of being delayed excessively due to a "poor" vehicle at the lane's front.

A final suggestion for future research would focus on improving and expanding the BDI agent model. As of now, the set of beliefs explored is very simplistic as vehicles do not take into account any exogenous information in the decision making process. One possible extension to the BDI model could include the representation of the SUMO scenario as a graph (set of edges and nodes) with information regarding the current traffic volume experienced. As a consequence, more accurate arrival times could be estimated by a vehicle during the bidding process. Moreover, the intersection manager itself can also be modelled, following a BDI architecture, to adapt the prices of reservations based on the bids received from each lane to enforce fairness into the management system.

References

[BAZ18] Anirudh Kishore Bhoopalam, Niels Agatz, and Rob Zuidwijk. Planning of truck platoons: A literature review and directions for future research. *Transportation Research Part B: Methodological*, 107:212 – 228, 2018.

[BF17] Masoud Bashiri and Cody H. Fleming. A platoon-based intersection management system for autonomous vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 667–672, 2017.

[BMSZ17] N. Bicocchi, M. Mamei, A. Sassi, and F. Zambonelli. On recommending opportunistic rides. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3328–3338, 2017.

[Bra87] Michael Bratman. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.

[Bre65] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[CBS13] D. Carlino, S. D. Boyles, and P. Stone. Auction-based autonomous intersection management. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 529–534, 2013.

[CGG+17] Philippe Caillou, Benoit Gaudou, Arnaud Grignard, Quang TRUONG Chi, and Patrick Taillandier. *A simple-to-use BDI architecture for agent-based modeling and simulation*. 01 2017.

[CGM19] Giacomo Cabri, Luca Gherardini, and Manuela Montangero. Auction-based crossings management. In *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, GoodTechs '19, page 183–188, New York, NY, USA, 2019. Association for Computing Machinery.

[Cla71] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[DGC+11] Baudouin Dafflon, Franck Gechter, Jean-Michel Contet, Abdeljalil Abbas-Turki, and Pablo Gruer. *Intelligent Crossroads for Vehicle Platoons Reconfiguration*, pages 203–214. 01 2011.

[DS08] Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *J. Artif. Intell. Res. (JAIR)*, 31:591–656, 01 2008.

[FCVT10] Michel Ferreira, Hugo Conceição, Wantanee Viriyasitavat, and O.K. Tonguz. Self-organized traffic control. pages 85–90, 01 2010.

[Fin] Leslie R Fine. Auctions.

[fMT17] European Commission. Directorate General for Mobility and Transport. *EU Transport in Figures: Statistical Pocketbook 2017*. EU transport in figures. Publications Office, 2017.

[FOK99] Fraunhofer FOKUS. Eclipse mosaic, a multi-domain and multi-scale simulation framework for connected and automated mobility., 1999.

[GHZ14] P. Gao, R. Hensley, and A. Zielke. A road map to the future for the auto industry. pages 43–53, 01 2014.

[GPP+70] Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. The belief-desire-intention model of agency. 02 1970.

[Gro73] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[HAS11] M. Hausknecht, T. Au, and P. Stone. Autonomous intersection management: Multi-intersection optimization. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4581–4586, 2011.

[HC05] Randolph Hall and Chinan Chin. Vehicle sorting for platoon formation: Impacts on highway entry and throughput. *Transportation Research Part C: Emerging Technologies*, 13(5):405 – 420, 2005.

[HD18] J. Heinovski and F. Dressler. Platoon formation: Optimized car to platoon assignment strategies and protocols. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, 2018.

[HZF+20] Yanjun Han, Zhengyuan Zhou, Aaron Flores, Erik Ordentlich, and Tsachy Weissman. Learning to bid optimally and efficiently in adversarial first-price auctions, 2020.

[JLW+16] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen. A survey on platoon-based vehicular cyber-physical systems. *IEEE Communications Surveys Tutorials*, 18(1):263–284, 2016.

[JWBB13] Qiu Jin, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew Barth. Platoon-based multi-agent intersection management for connected vehicle. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1462–1467, 2013.

[Kra97] Sarit Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1):79 – 97, 1997. Economic Principles of Multi-Agent Systems.

[LB17] Jan Lizbetin and Ladislav Bartuska. The influence of human factor on congestion formation on urban roads. *Procedia Engineering*, 187:206 – 211, 2017. TRANSBALTICA 2017: TRANSPORTATION SCIENCE AND TECHNOLOGY: Proceedings of the 10th International Scientific Conference, May 4–5, 2017, Vilnius Gediminas Technical University, Vilnius, Lithuania.

[LLJ15] J. Larson, K. Liang, and K. H. Johansson. A distributed framework for coordinated heavy-duty vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):419–429, 2015.

[LM18] J. A. Leon Calvo and R. Mathar. Connected vehicles coordination: A coalitional game-theory approach. In *2018 European Conference on Networks and Communications (EuCNC)*, pages 1–6, 2018.

[LMJ16] K. Liang, J. Mårtensson, and K. H. Johansson. Heavy-duty vehicle platoon formation for fuel efficiency. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1051–1061, 2016.

[LRL17] T. Lin, H. Rivano, and F. Le Mouël. A survey of smart parking solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3229–3253, 2017.

[MCZ20] Stefano Mariani, Giacomo Cabri, and Franco Zambonelli. Coordination of autonomous vehicles: Taxonomy and survey. *ArXiv*, abs/2001.02443, 2020.

[MWK17] Santa Maiti, Stephan Winter, and Lars Kulik. A conceptualization of vehicle platoons and platoon operations. *Transportation Research Part C: Emerging Technologies*, 80:1 – 19, 2017.

[NRTV07] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[NSB+16] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, and J. Fredriksson. If, when, and how to perform lane change maneuvers on highways. *IEEE Intelligent Transportation Systems Magazine*, 8(4):68–78, 2016.

[PKSA06] Jeremy Pitt, Lloyd Kamara, Marek Sergot, and Alexander Artikis. Voting in multi-agent systems. *Comput. J.*, 49:156–170, 01 2006.

[PLP13] E. Polycarpou, L. Lambrinos, and E. Protopapadakis. Smart parking solutions for urban areas. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–6, 2013.

[RBB+02] Rosaldo JF Rossetti, Rafael H Bordini, Ana LC Bazzan, Sergio Bampi, Ronghui Liu, and Dirck Van Vliet. Using bdi agents to improve driver modelling in a commuter scenario. *Transportation Research Part C: Emerging Technologies*, 10(5-6):373–398, 2002.

[RDK19] Inga Rüb and Barbara Dunin-Kplicz. *BDI Model of Connected and Autonomous Vehicles*, pages 181–195. 08 2019.

[RM17] J. Rios-Torres and A. A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2017.

[RRJ+04] Iyad Rahwan, Sarvapali Ramchurn, Nicholas Jennings, Peter Mcburney, and Simon Parsons. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18, 01 2004.

[SB11] M. Seredynski and P. Bouvry. A survey of vehicular-based cooperative traffic information systems. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 163–168, 2011.

[SBH02] Rahul Sukthankar, Shumeet Baluja, and John Hancock. Multiple adaptive agents for tactical driving. *Applied Intelligence*, 9, 07 2002.

[SBJ+14] M. Segata, B. Bloessl, S. Joerer, F. Dressler, and R. L. Cigno. Supporting platooning maneuvers through ivc: An initial protocol analysis for the join maneuver. In *2014 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 130–137, 2014.

[SHH+15] Christoph Sommer, Jérôme Härri, Fatma Hrizi, Björn Schünemann, and Falko Dressler. *Simulation Tools and Techniques for Vehicular Communications and Applications*, pages 365–392. Springer International Publishing, Cham, 2015.

[SMWW19] Zhaokun Shen, A. Mahmood, Yafei Wang, and Lin Wang. Coordination of connected autonomous and human-operated vehicles at the intersection. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1391–1396, 2019.

[SSSM19] L. Sequeira, A. Szefer, J. Slome, and T. Mahmoodi. A lane merge coordination model for a v2x scenario. In *2019 European Conference on Networks and Communications (EuCNC)*, pages 198–203, 2019.

[VO14] Matteo Vasirani and Sascha Ossowski. A market-inspired approach for intersection management in urban road traffic networks. *CoRR*, abs/1401.5851, 2014.

[WC17] Pin Wang and Ching-Yao Chan. Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge. pages 1–6, 10 2017.

[WGKR21] Phuriwat Worrawichaipat, Enrico Gerding, Ioannis Kaparias, and Sarvapali Ramchurn. Resilient intersection management with multi-vehicle collision avoidance. *Frontiers in Sustainable Cities*, 3:44, 2021.

[WP17] S. Wenzel and T. Peter. *Analysis of decentral platoon planning possibilities in road freight transportation using an agent-based simulation model*, page 159. 2017.

[ZDZ12] D. Zhao, Y. Dai, and Z. Zhang. Computational intelligence in urban traffic signal control: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):485–494, 2012.

[ZHK+13] M. Zhu, J. Hu, L. Kong, R. Shen, W. Shu, and M. Wu. An algorithm of lane change using two-lane nasch model in traffic networks. In *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 241–246, 2013.

[ZSFB95] Michael Zabat, Nick Stabile, Stefano Farascaroli, and Frederick Browand. The Aerodynamic Performance Of Platoons: A Final Report. Institute of Transportation Studies, Research Reports, Working Papers, Proceedings qt8ph187fw, Institute of Transportation Studies, UC Berkeley, January 1995.

[ZSSL17] Betina Zanchin, Rodrigo Silva, Max Santos, and Kathya Linares. On the instrumentation and classification of autonomous cars. Available at http://dx.doi.org/110.1109/SMC.2017.8123022, November 2017.