Cooperative, Connected and Autonomous Mobility: A Simulation Framework for Evaluation of Coordination Mechanisms

Francisco Filipe

Zafeiris Kokkinogenis

Pedro d'Orey*

Rosaldo J. F. Rossetti

*CISTER Research Centre

CISTER-TR-230701

2023/09

Cooperative, Connected and Autonomous Mobility: A Simulation Framework for Evaluation of Coordination Mechanisms

Francisco Filipe, Zafeiris Kokkinogenis, Pedro d'Orey*, Rosaldo J. F. Rossetti

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: franciscocanelasfilipe@gmail.com, ore@isep.ipp.pt

https://www.cister-labs.pt

Abstract

This study proposes a simulation framework for researching MAS-based vehicle network coordinating mechanisms. The framework in question was required to provide a vehicular traffic simulator framework that allowed for near-real-world modeling of constraints within vehicular networks (both at the kinematic and network levels), as well as to equip each vehicle with advanced cognitive decision-making abilities via BDI-based agent modeling. We perform a preliminary analysis of the framework to assess its capability to cope with average complexity coordination mechanisms in connected and autonomous vehicle settings.

Cooperative, Connected and Autonomous Mobility: A Simulation Framework for Evaluating Coordination Mechanisms

Francisco Filipe[1], Zafeiris Kokkinogenis[2], Pedro M. d'Orey[3] and Rosaldo J. F. Rossetti[2], *Member, IEEE.*

***Abstract*— Autonomous driving and vehicular networking have gained traction during the last decade. Simulation is one of the major tools for assessing coordination mechanisms (e.g. intersection control), especially when considering autonomous vehicles. This study proposes a simulation framework for researching MAS-based coordination mechanisms. The framework includes three main components for accurately replicating the scenario, namely a traffic simulator for modeling vehicle dynamics and interactions, a network simulator modeling vehicular networks and a BDI-based agent simulator for modeling cognitive decision-making abilities. We perform a preliminary analysis of the framework to assess its capability to cope with average complexity coordination mechanisms in connected and autonomous vehicle settings.**

I. INTRODUCTION

Autonomous driving has gained traction in recent years. Similarly, recent developments in vehicular networks foster explicit collaboration between vehicles via vehicle-to-vehicle (V2V) communications and between vehicles and infrastructure via vehicle-to-infrastructure (V2I) communications. Cooperation is achieved through the periodic or event-driven transmission of static and dynamic data (e.g., location, trajectories) via wireless networks, such as through Cooperative Awareness Messages (CAM) [1]. However, this results in an information-sharing scheme that does not necessarily achieve a collective desired behavior and does not consider drivers'/vehicles' preferences in their decision-making process. Thus, the intention of having a group of vehicles collectively deciding over a set of goals and grouping together individual interests is not straightforward by only adopting CAM-like approaches.

On the one hand, the Multi-Agent Systems (MAS) community has developed a number of collective decision-making techniques for reaching an agreement on the aggregated preferences of autonomous systems. On the other hand, the vast majority of experiments and simulations considered in the literature and practice do not take into account the wide range of limitations that Connected Autonomous Vehicles (CAVs) will encounter in actual deployments (e.g unreliable and bandwidth-limited communication channels), which reduces the representation of the results.

We contend that practical limits need to be taken into consideration when evaluating collective decision-making. In order to hasten their development, MAS-based ITS solutions ought to be assessed in software (i.e. simulation) that captures both kinematic and communication restrictions. In order to assess the coordination of CAVs and analyze their effects (e.g., vehicle flow or pollution), simulation models and tools from multiple domains, programming paradigms, and resolutions must be integrated into a combined simulation framework.

The main contributions can be summarized as follows:

i) We propose a multi-domain simulation framework for testing and evaluating MAS-based Intelligent Transportation Systems (ITS) methods;
ii) We perform a preliminary analysis of the feasibility of proposed the simulation framework, by showcasing an intersection management scenario considering a multi-agent reservation-based approach as a potential vehicular agreement mechanism.

The remainder of this document is organized as follows. Section II presents a related work analysis. Section III provides a description of the methodological approach for the simulation framework. Section IV includes showcase an intersection management application implemented using the simulation framework. In Section V we present and discuss some preliminary results about implementing the application in the simulation framework. Finally, Section VI presents the conclusions and the future work.

II. RELATED WORK

Individual simulators often fail to provide the necessary resolution to assess a traffic system [2]. For instance, vehicle platooning enabled by V2V/V2I communications requires a distributed simulation architecture that tightly couples (vehicular) network simulation with microscopic traffic simulation, including platooning capabilities. In the following, we review well-known integrated simulation platforms for evaluating ITS applications.

One of the first efforts to microscopically simulate autonomous vehicles is reported in [3], [4]. In [5] an integration between a microscopic traffic model and the NS-3 simulator is proposed to simulate heterogeneous vehicular networks. On the other hand, [6] discussed the integration of SUMO and OMNeT++. In a similar effort, Singh et al. in [7] proposed the VENTOS simulation framework, to implement a leader election protocol for platoons. In [8] QoS-CITS, a traffic simulator for heterogeneous autonomous vehicles dedicated to simulating service-oriented C-ITS with a focus

---

[1] F. Filipe is with the Faculty of Engineering, University of Porto, Porto, Portugal (`franciscocanelasfilipe@gmail.com`)

[2] Z. Kokkinogenis and R. Rossetti are with the Laboratory of Artificial Intelligence and Computer Science (LIACC), Department of Informatics Engineering, Faculty of Engineering, University of Porto, Porto, Portugal (`{kokkinogenis,rossetti}@fe.up.pt`)

[3] P. M. d'Orey is with CISTER, ISEP, Polytechnic Institute of Porto, Portugal, and the Department of Computer Sciences, Faculty of Sciences, University of Porto, Porto, Portugal (`ore@isep.ipp.pt`)

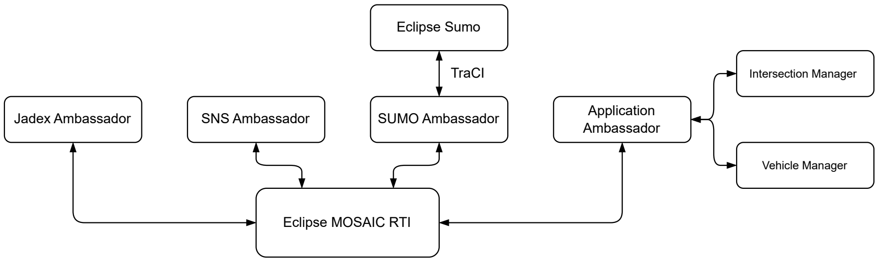


Fig. 1: System architecture

on Quality of Service (QoS), is discussed. A Service-oriented Cooperative ITS (SoC-ITS), with a focus on the assessment of the quality of service issues concerning vehicles and policies, is discussed in [9]. In [10] a simulator for assessing connected and autonomous mobility is proposed.

A similar rationale applies to modeling vehicles or other artifacts (e.g. traffic lights) as autonomous systems. This, usually, results in the integration of MAS frameworks with microscopic traffic simulators [11], [12]. Soares et al. presented the integration of JADE and SUMO in [13] to build an artificial transportation systems simulation framework where drivers and traffic control can be designed as MAS. In a similar approach, Görmer et al. combined JADE and AIMSUN [14]. A traffic management solution is evaluated in [15] using the Jason MAS framework and SUMO. In [16] a cost-benefit analysis of connected and autonomous freight vehicles on traffic efficiency was made. A methodology for building simulation testbeds for assessing the impacts of autonomous mobility is discussed in [17].

III. SIMULATION ARCHITECTURE

To perform simulations with the real-world constraints that are expected of a vehicular network, the decision-making agents should interact within an environment that imposes both constraints on wireless communication and mobility. To achieve this goal, we have developed a hybrid simulation framework (Fig. 1), build on top of the Eclipse MOSAIC framework, integrating the following components:

- **agent-oriented platform** for the definition of the agent behaviors and responsible for the high-level decision making (i.e. auction-based negotiation).
- **microscopic traffic simulator** that simulates the vehicular traffic dynamics and satisfies all the requirements of traffic simulation (e.g. vehicle kinematics or road network).
- **network simulator** that implements the protocols of the communication stack.
- **application layer** that implements helper functions related to the application logic (e.g.vehicles and the intersection manager.

A. Eclipse MOSAIC - mobility simulation framework

Eclipse MOSAIC [18] is a multi-domain and multi-scale co-simulation framework for connected and automated mobility that enables the coupling of external simulators by employing the concept of Federates and Ambassadors. Any simulation model coupled with Eclipse MOSAIC is called a federate and connects to the MOSAIC federation through a proper interface called the 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.

Specifically, Eclipse MOSAIC uses a *publication-subscription* mechanism to handle the data exchange among the federates during the simulation execution. A published interaction is forwarded to each subscriber directly after it has been published. The interactions are handled by the *run-time infrastructure*.

B. SUMO - road traffic simulator

The road traffic simulation component of this implementation is SUMO (Simulation of Urban MObility) [19], an open-source simulator with rich features and transportation artifacts. A significant feature of SUMO is its TraCI API to allow for remote control of the simulation. 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.

C. SNS/OMNET++/... - network simulator

To allow messages to flow from one vehicle to another, as well as from vehicles to infrastructure nodes, a network component capable of mimicking wireless vehicular networks was added to the implementation. Several network simulators can be coupled to the simulation platform, such as the Simple Network Simulator (SNS) or OMNET++ with platooning and vehicular networking extensions. For instance, SNS aims to provide simple and fast capabilities for the transmission of V2X messages using two Ad-hoc communication: geographically and topologically-scoped transmissions. While geographical transmissions are limited only to broadcasting, topological transmissions allow also for unicasting.

D. Jadex - MAS Framework

The agents' high-level decision-making is modeled in the Jadex Active Component [20], which is a platform used to develop distributed and concurrent systems where agents follow the Belief, Desire, and Intentions (BDI) approach. To exemplify the agent behavior modeling, a simple auction-based intersection management experiment is considered in the following, with the primary goal of verifying that

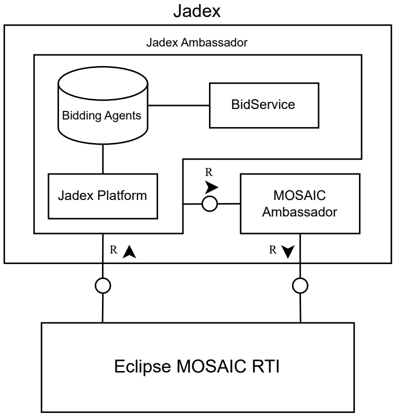


Fig. 2: SUMO connected to Eclipse MOSAIC

the decision-making agents can receive the environment state from the network simulator and deliberate on what actions (e.g. how to bid) to perform accordingly. The auction component's unique responsibility is the bid calculation.

Fig. 2 illustrates how the Jadex module is coupled with the Eclipse MOSAIC Run-time interface. 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. To generate new agents as vehicles spawn within the simulation, a *Jadex MAS* is constructed, which provides a specific service for the agent creation process (e.g. *ComponentManagementService*), every time an interaction (e.g. *VehicleRegistration*) is sent by MOSAIC's run-time interface, the agent creation process instantiates a new agent.

E. Application layer

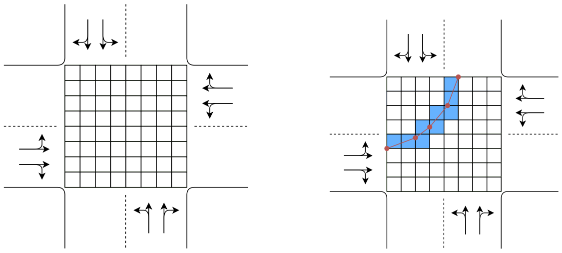
This layer provides the logic behind the underlying mechanism (e.g. intersection control) and thus needs to be tailored for each application. Due to this, an example implementation is presented in the following section.

IV. INTERSECTION MANAGEMENT APPLICATION

We assess coordination mechanisms among autonomous vehicles and the infrastructure in a traffic environment. The environment is composed of resources that need to be distributed among the vehicles with some coordination. Several resources can be considered, such as traffic lanes, intersections, or slots in platoon formations. For the sake of simplicity, we consider the case of coordinating trajectory reservation using auctions for the crossing of an intersection. The computation is decentralized in the sense that each vehicle computes a time-slot path that crosses the intersection. Each vehicle will communicate to the intersection manager the resource that wants to access, accompanied by a bidding value. The coordination strategy for resource allocation is based on a reservation-based mechanism. This strategy has similarities with the First-Come-First-Serve (FCFS) approach by Dresner and Stone's [21].

A. Environment Representation

Traffic Model. The traffic model main two entities are the *intersection* and *vehicles*. The intersection is represented as an occupancy grid, where vehicles reserve a sequence of time-space slots (cells) to cross it. Fig. 3a illustrates how an intersection can be divided into a matrix of reservation tiles. A reservation can be characterized by the parameters: i) begin-time, ii) end-time, and iii) bid value.



a) Intersection model

b) Trajectory prediction result

Fig. 3: Intersection modeled as a matrix of reservation tiles

We consider three different types of vehicles, namely i) *independent vehicles*, ii) *platoon leaders*, and iii) *platoon followers*. For the vehicles to perform reservation requests, information on the upcoming intersection must be received by the intersection manager. A successful reservation request is composed of four steps: i) trajectory prediction, iii) conflict detection, iii) conflict resolution, and iv) bid calculation.

Bidding Protocol. Bids represent the value of time for crossing the intersection. The bids are paid to the intersection manager, once a vehicle starts crossing an intersection. When considering the planning strategy of a vehicle, four parameters must be known for every vehicle: i) Maximum Time of Arrival (MTA), ii) Estimated Time of Arrival (ETA), iii) number of intersections still to transverse (nI), and iv) wallet credits for bidding (W). The MTA and ETA metrics are estimated assuming constant velocity. Only vehicles with available credit are eligible to bid.

B. Application Layers

We consider two main types of application layers: vehicle and intersection managers, which are described in the following:

- **Vehicle Manager**: associates a vehicle in SUMO with an agent in Jadex in a delegation-based fashion: the vehicle's operational level actions (e.g break, accelerate) are delegated to SUMO, vehicle's tactical level

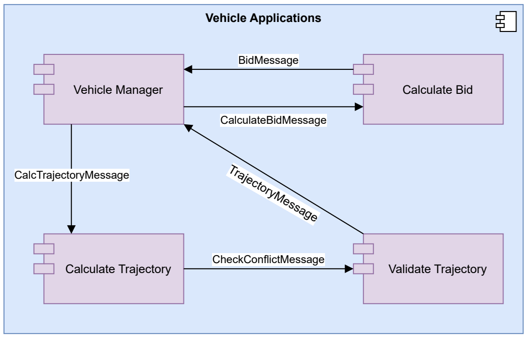


Fig. 4: Message components and messages exchanged between vehicle applications

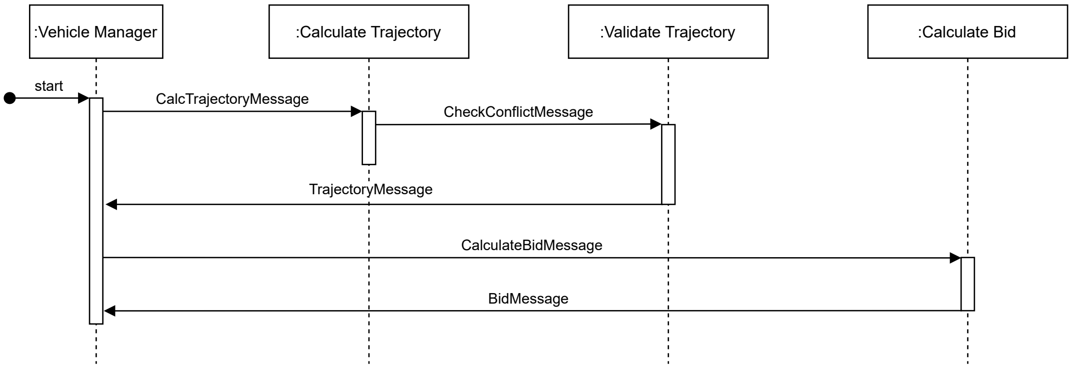


Fig. 5: Message flow between vehicle applications

decisions (e.g trajectory computation) are delegated to the vehicle manager, while higher level decisions (e.g. bidding strategy) are delegated to the Jadex agent. This module will handle the communication of the vehicle with the others and with the intersection manager.

- **Intersection Manager**: simply approves or rejects incoming requests due to the decentralized nature of the solution. Furthermore, it constantly updates and broadcasts the reservation map to incoming vehicles

The component diagram in Fig. 4 depicts the implemented vehicle applications and the communications between them. Other sub-modules composing the application ecosystem are:

- **Calculate Trajectory:** responsible for computing a vehicle's trajectory through the intersection
- **Validate Trajectory:** elicits the validity verification of a trajectory against the reservation map. Resolves conflicts if any occur by delaying the departure of the current trajectory.
- **Calculate Bid:** elicits the Jadex agent for the computation of a vehicle (or platoon) bid.

Fig. 5 depicts the interaction flow among the different sub-modules. Upon receiving the reservation map information from the intersection manager, the *Vehicle Manager* (either of a single vehicle or platoon) invokes the calculate trajectory sub-module by sending a *CalculateTrajectoryMessage*. Once computed the desired trajectory with all the reservation tiles the vehicle needs to bid on, it shares both the trajectory and the reservation map with the *Validate Trajectory* sub-module through a *CheckConflictMessage* to detect potential conflicts. If any conflict occurs, the reservation timestamps are delayed for each reservation tile of the trajectory, and a new validation is made. Otherwise, the trajectory is considered valid and a *Trajectory Message* is dispatched to the *Vehicle Manager*. The final step of the logic is the calculation of a reservation bid through the sub-module *Calculate Bid*. For this to happen the application communicates with a Jadex agent responsible for computing the vehicle's bid. After the bid is computed, the Jadex agent shares its value with the *Vehicle Manager* application through a *Bid Message*.

Fig. 6 illustrates the interaction between the validity trajectory sub-module and intersection manager as well as the communications established between them. After receiving a vehicle's reservation request, the *Intersection Manager* dispatches a *CheckConflict* message to the *Validate Trajectory* application. This application then communicates a response to the *Intersection Manager*. Depending on its evaluation, there are three different responses possible:

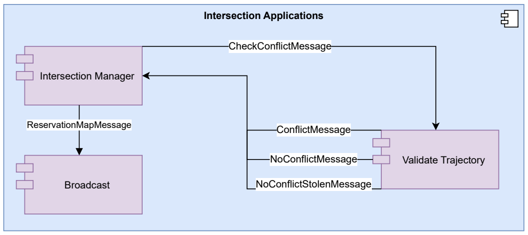


Fig. 6: Message components of the intersection application

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

C. *Communication*

Two types of data communication are considered, namely *V2X Communications* for simulating message exchange mainly between vehicles and the IM in ad-hoc mode and *Federate Communication* for data exchange between the different simulation models running through the Eclipse MOSAIC federation.

V2X Communications. There exists ad-hoc communication between the following entities: i) IM ↔ vehicle (or platoon leader), ii) platoon leader ↔ platoon followers and iii) platoon follower ↔ IM. Six different messages types are generated by these three entities:

- **Intersection Manager**:
  - i) reservation map information (*BroadcastMessage*)
  - ii) reservation responses to vehicles:
    - ∗ accept (*ReservationAcceptedMessage*)
    - ∗ reject (*ReservationRejectedMessage*)

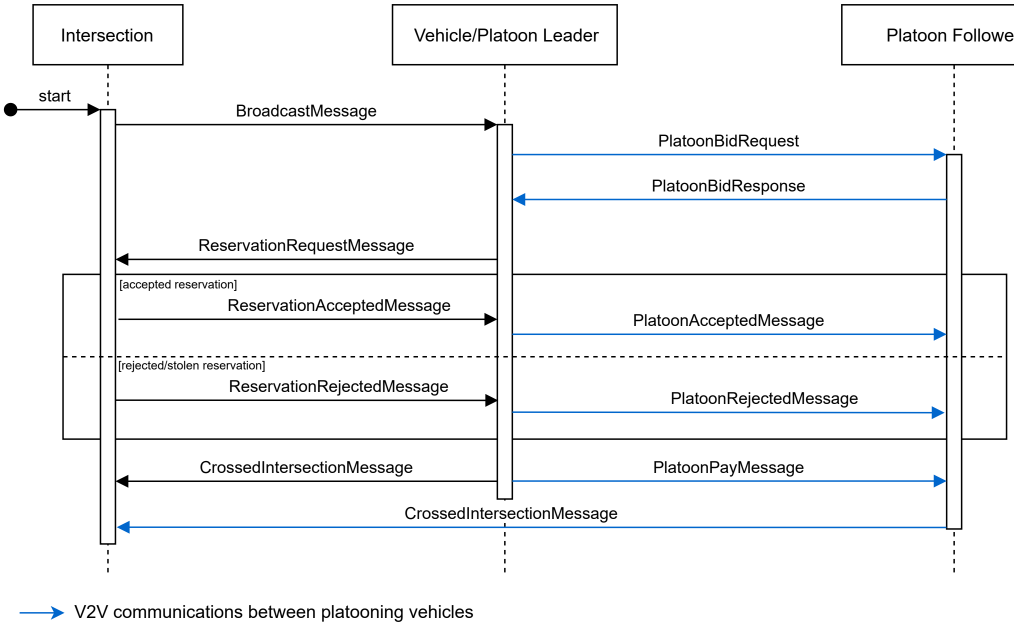


Fig. 7: V2X message flow between vehicles and intersection manager

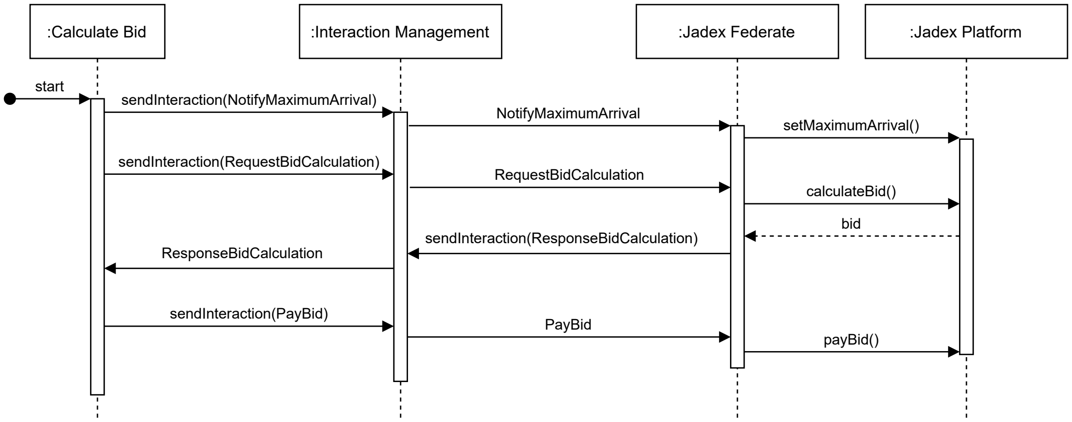
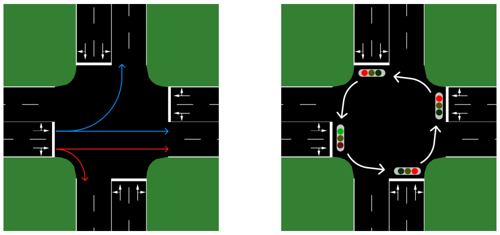


Fig. 8: Interaction flow between a vehicle application and the Jadex Federate

- **Vehicle Manager**:
  - iii) reservation requests to the intersection manager, termed *ReservationRequestMessage*
  - iv) (*platooning only*): messages for determining the bid of the platoon (*PlatoonBidRequest*) and informing of request state: accept (*PlatoonAcceptedMessage* or deny (*PlatoonRejectedMessage*)
- **Platoon Follower**:
  - v) platoon vehicle bid (*PlatoonBidResponse*)
  - vi) finalization of intersection crossing to the intersection manager (*CrossedIntersectionMessage*)

Fig. 7 depicts the message flow between the three entities. Initially, the IM broadcasts up-to-date information on the reservation map, which is triggered by the *IM* application that dispatches a *BroadcastMessage* with a given frequency (e.g. every 100 ms). Upon receiving a *BroadcastMessage*, the vehicle calculates the trajectory (tiles), the time required to cross the intersection, and the corresponding bidding value (see Fig. 5). Following, the vehicle sends a *ReservationRequestMessage* to the IM. Note that as soon as a vehicle receives a *BroadcastMessage*, it stops listening to messages of this type until it receives a response from the IM to its reservation request. After evaluating the request the IM will answer with a *ReservationAcceptedMessage* if the evaluation is positive, otherwise will transmit a *ReservationRejectedMessage*. If the reservation is accepted with conflicts, the vehicle that lost its reservation will also be notified using a *ReservationRejectedMessage*. A vehicle receiving a *ReservationAcceptedMessage* will store the reservation departure time and wait until it is time to cross the intersection. On the other hand, a vehicle receiving a *ReservationRejectedMessage* will remove any instance of a reservation departure time (if existing) and start listening again for new *BroadcastMessage*s. In the former case, the vehicle will inform the IM that the intersection is empty using a *CrossedIntersectionMessage*.

A similar process also applies in the case of platoon formations with small adjustments. First, the platoon leader



a) Trajectories b) traffic light phases

Fig. 9: Intersection

communicates with their followers in the bidding process so that their bid value estimations are taken into account in the platoon's final bid. The notification starts with a *PlatoonBidRequest* message containing the vehicle's Estimated Time of Arrival (ETA) for calculating the bid. The followers forward their values to the leader by dispatching a *PlatoonBidResponse* message. When a platoon leader receives information about the result of the reservation, it broadcasts it to its followers. In case of a successful reservation, the platoon leader broadcasts 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. Lastly, once a platoon starts crossing the intersection, each vehicle must pay its share of the bid according to the previously defined payment method. The platoon leader notifies with *PlatoonPayMessage* the other platoon members who, upon its reception, communicate with the Jadex Platform through its federate to withdraw the needed amount of credits from its wallet.

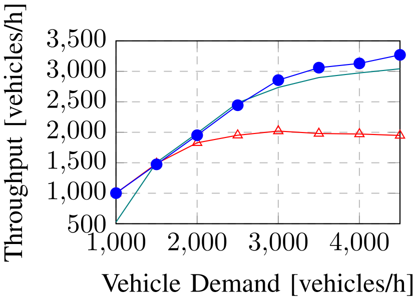
Federate Communications. The sequence diagram in Fig. 8 depicts the process of managing interactions between the vehicle application and the developed federate to calculate bids. In the study, the Jadex Federate subscribes to four different interactions and publishes one other; it subscribes to *NotifyMaximumArrival* and *NotifyEstimatedArrival* to receive the MTA and ETA respectively for computing a bid, the *RequestBidCalculation*, and the *PayBid* interaction for starting the bidding process and updates its "wallet" respectively. The federate publishes forwards a *ResponseBidCalculation* interaction containing the estimated bid value.

V. RESULTS

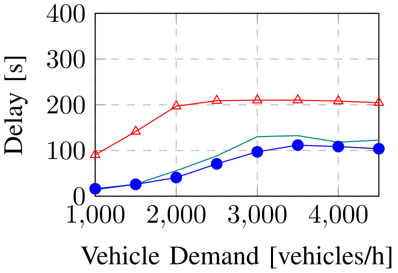
To evaluate the simulation framework we perform a preliminary analysis on how platoons can impact the performance of vehicle coordination at intersections. We consider a single intersection that both individual vehicles and platoon formations transverse.

A. Scenario

The scenario consists of a four-way intersection (Fig. 9a) with the following characteristics: *i)* it is composed of eight incoming and eight outgoing lanes and *ii)* the lane length is 150 m to allow insertion of all vehicles before the simulation starts. The vehicle's maximum speed is 10 m/s and the minimum safety gap is 2 m. The vehicle trajectories are also depicted in Fig. 9a. Two different trajectories were defined for the two different existing lanes of every incoming edge. Different vehicle demands were generated to emulate low (1000 $vehicles/h$) to high (4500 $vehicles/h$) traffic volume environments, with increments of 500 $vehicles/h$. All experiments lasted 30 min of simulation time.



a) Throughput



b) Delay

Fig. 10: Intersection management (TL —△—, FCFS ——, RB —●—)

B. Benchmark.

We compare our reservation-based coordination strategy with the following approaches:

- *conventional traffic lights*: The access to the intersection is exclusive to the edge that has been given a green light. At most one edge can have a green phase at any given time. Fig. 9b depicts the round-robin scheme of how traffic light phases change over time.
- *First Come First Served* The vehicles can provide information about their anticipated arrival time, speed, and acceleration/deceleration. The intersection manager reserves the needed space-time slots (tiles) for navigating the intersection.

C. Illustrative Results

Fig. 10a depicts the intersection throughput as a function of vehicle demand in a platooning scenario. As expected, the throughput increases for higher vehicle demands until the intersection capacity is reached. This *plateau* is reached in the worst case for 2500 $vehicles/h$. Our reservation-based method has improved performance with respect to the conventional traffic lights and the FCFS methods; note that the gain of our method is much larger when compared with the conventional traffic lights benchmark. However, in the case of FCFS the gain is marginal.

For low traffic demands, the performance difference is not very significant between the autonomous approaches and the traditional approach as the traffic light's 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 cannot effectively handle vehicle demands higher than 2500 $vehicles/h$, i.e. 20 s of green phase are not enough to flush all waiting vehicles in a given intersection edge due to vehicle build-up. On the other hand, the designed reservation-based approach shows a better ability to handle higher vehicle demands by allowing vehicles from multiple lanes to cross at the same time, which results in a much smaller queue of vehicles. A throughput improvement of up to 75.8% when compared to the traffic light management system is obtained. The comparative analysis between the negotiation-based approach and the FCFS approach shows very similar throughput for increasing vehicle demands. The same behavior is present when observing the average delay (see fig. 10b). The improvements provided by the negotiation-based approach are the result of a lower number of conflicts.

VI. CONCLUSIONS

This paper presents a simulation framework for investigating MAS-based coordination mechanisms for vehicle networks. The framework provides a vehicular traffic simulator framework that allows for close to real-world modeling of constraints within vehicular networks (both at the kinematic and network levels), as well as to equip each vehicle with advanced cognitive decision-making abilities through the use of BDI-based agent modeling.

A preliminary analysis of the framework was performed considering a case of the management of a single intersection in the co-existence of individual vehicles and platoon formation in the context of connected and autonomous vehicles. We consider the case of an agent-based reservation mechanism. We show that the framework can indeed simulate the scenario and the complexity of the agent-based approach. Future work will focus on an extensive scalability analysis of the simulation framework in more complex scenarios and coordination mechanisms.

ACKNOWLEDGMENTS

This work is a result of project DynamiCITY, Ref. NORTE-01-0145-FEDER-000073, supported by NORTE 2020 Operational Programme, under PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), and the project Route 25, Ref. TRB/2022/00061 - C645463824-00000063, funded by the EU/Next Generation, within call n.º 02/C05-i01/2022 of the Recovery and Resilience Plan (RRP). This work was also supported by national funds through the FCT/MCTES, within the Research Units: LIACC, University of Porto [FCT/UID/CEC/0027/2020], CISTER, ISEP/IPP [UIDP/UIDB/04234/2020], Center for Systems and Technologies (SYSTEC) [UIDB/00147/2020 & UIDP/00147/2020], and Associate Laboratory Advanced Production and Intelligent Systems (ARISE) [LA/P/0112/2020].

REFERENCES

[1] M. Boban and P. M. d'Orey, "Exploring the practical limits of cooperative awareness in vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3904–3916, 2016.

[2] R. J. F. Rossetti, P. A. F. Ferreira, R. A. M. Braga, and E. C. Oliveira, "Towards an artificial traffic control system," in *11th IEEE Conf. on Intelligent Transportation Systems*, 2008, pp. 14–19.

[3] J. L. F. Pereira and R. J. F. Rossetti, "An integrated architecture for autonomous vehicles simulation," in *Proceedings of the 27th annual ACM symposium on applied computing*, 2012, pp. 286–292.

[4] M. C. Figueiredo, R. J. F. Rossetti, R. A. M. Braga, and L. P. Reis, "An approach to simulate autonomous vehicles in urban traffic scenarios," in *12th IEEE Conf. Intelligent Transportation Systems*, 2009, pp. 1–6.

[5] R. Fernandes, P. M. d'Orey, and M. Ferreira, "Divert for realistic simulation of heterogeneous vehicular networks," in *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*. IEEE, 2010, pp. 721–726.

[6] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.

[7] P. K. Singh, S. Sharma, S. K. Nandi, R. Singh, and S. Nandi, "Leader election in cooperative adaptive cruise control based platooning," in *Proc. 1st International Workshop on Communication and Computing in Connected Vehicles and Platooning*. ACM, 2018, pp. 8–14.

[8] Z. Kailong, W. Min, S. Hang, Y. Ansheng, A. de La Fortelle, and M. Kejian, "Qos-cits: A simulator for service-oriented cooperative its of intelligent vehicles," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. IEEE, 2017, pp. 751–756.

[9] K. Zhang, C. Xie, Y. Wang, S. M. Ángel, T. M. T. Nguyen, Q. Zhao, and Q. Li, "Hybrid short-term traffic forecasting architecture and mechanisms for reservation-based cooperative its," *Journal of Systems Architecture*, vol. 117, p. 102101, 2021.

[10] J. Zhang, C. Chang, Z. He, W. Zhong, D. Yao, S. Li, and L. Li, "Cavsim: A microscopic traffic simulator for evaluation of connected and automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[11] M. Teixeira, P. M. d'Orey, and Z. Kokkinogenis, "Simulating collective decision-making for autonomous vehicles coordination enabled by vehicular networks: A computational social choice perspective," *Simulation Modelling Practice and Theory*, vol. 98, p. 101983, 2020.

[12] Z. Kokkinogenis, M. Teixeira, P. M. d'Orey, and R. J. F. Rossetti, "Tactical level decision-making for platoons of autonomous vehicles using auction mechanisms," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1632–1638.

[13] G. Soares, Z. Kokkinogenis, J. L. Macedo, and R. J. F. Rossetti, "Agent-based traffic simulation using sumo and jade: an integrated platform for artificial transportation systems," in *Simulation of Urban MObility User Conference*. Springer, 2013, pp. 44–61.

[14] J. Görmer and J. P. Müller, "Multiagent system architecture and method for group-oriented traffic coordination," in *2012 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*. IEEE, 2012, pp. 1–6.

[15] A. Batista and L. R. Coutinho, "A multiagent system for combining green wave and adaptive control in a dynamic way," in *IEEE Conf. on Intelligent Transportation Systems*. IEEE, 2013, pp. 2439–2444.

[16] K. Bhargava, K. W. Choy, P. A. Jennings, S. A. Birrell, and M. D. Higgins, "Traffic simulation of connected and autonomous freight vehicles (cav-f) using a data-driven traffic model of a real-world road tunnel," *Transportation Engineering*, vol. 2, p. 100011, 2020.

[17] T. Tettamanti, M. Szalai, S. Vass, and V. Tihanyi, "Vehicle-in-the-loop test environment for autonomous driving with microscopic traffic simulation," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2018, pp. 1–6.

[18] C. Sommer, J. Härri, F. Hrizi, B. Schünemann, and F. Dressler, "Simulation tools and techniques for vehicular communications and applications," *Vehicular ad hoc Networks: standards, solutions, and research*, pp. 365–392, 2015.

[19] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 2575–2582.

[20] A. Pokahr, L. Braubach, and K. Jander, "Unifying agent and component concepts: Jadex active components," in *Multiagent System Technologies: 8th German Conference, MATES 2010, Leipzig, Germany, September 27-29, 2010. Proceedings 8*. Springer, 2010, pp. 100–112.

[21] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management." *J. Artif. Intell. Res. (JAIR)*, vol. 31, pp. 591–656, 01 2008.