Reducing Pollutant Emissions through Virtual Traffic Lights

Ahmadreza Jame, Marco Rapelli and Claudio Casetti[*]

Department of Control and Computer Engineering, Politecnico di Torino, Torino, 10129, Italy

ARTICLE INFO

Keywords:
Urban Mobility
Vehicle-to-Vehicle communication
Vehicular Application
Virtual Traffic Light

ABSTRACT

Among the key goals of urban policies, the reduction of traffic congestion and pollutant emissions surely tops the list. Although solutions leveraging vehicular communication, such as GLOSA, have been proposed to smooth traffic at regulated intersections, cities normally have a large number of unregulated intersections where queues can build up, worsening emissions caused by the stop-and-go motion of vehicles. The problem is further compounded by the future presence of self-driving cars, where the need to coordinate with other autonomous cars at intersections will be even greater. In this paper, we propose V^3TL, a Vehicle-to-Vehicle (V2V) Virtual Traffic Light system for infrastructure-less unregulated crossroads. We aim at providing a low-complexity, yet effective, algorithm and communication protocol to let vehicles at an unregulated intersection decide if a virtual traffic light is needed and, in that case, self-organize to establish one. Our results highlight significant improvements compared to both unregulated and traffic light-based intersections. We tested the performance of V^3TL in different, realistic scenarios (isolated or consecutive three- and four-way junctions, in single- and multi-lane configuration) and different vehicle generation rates, obtaining improvements in terms of number of passing vehicles per minute, number of stop-and-go maneuvers and scheduling fairness, when compared to unregulated or traffic-light regulated junctions.

1. Introduction

Air pollution is a serious environmental health risk. In 2016, the World Health Organization (WHO) estimated that about 7 million deaths were caused by air pollution, largely as a result of heart or lung diseases [1]. Among different kinds of pollutants, Particulate Matters (PM10 and PM2.5) are classified as Group 1 carcinogens by IARC (International Agency for Research on Cancer) and WHO itself.

PMs are generated by fossil fuel combustion and their largest density is measured in urban environments. The main causes of PM, as well as of other pollutants, were found to be the emissions of industrial plants, ageing heating systems and vehicular emissions. Regarding the latter, the manufacturing and use of hybrid and electric cars is going to mitigate the problem. However, the high cost of electric vehicles on the market prevents this from being a short-term solution. For this reason, alternative answers have to be considered.

It is a well-known fact that vehicle emissions depend on the way the driver uses the car itself. Driving at variable speeds with abrupt accelerations and frequent stop-and-go maneuvers is known to increase pollutant emissions [2], [3]. It is common to find this kind of behavior in high traffic congestion areas, where long queues delay drivers and increase their level of impatience. Apart from car accidents, roadworks or other unpredictable events, the greatest part of daily congestion is due to regulated and unregulated intersections. This is why urban planning is becoming paramount for a good mobility environment. In particular, Intelligent Transport Systems (ITS) have recently helped significantly by the introduction of vehicular applications to improve road safety as well as mobility efficiency and eco-friendliness. An example of an infrastructure-based ITS solution is GLOSA, which has been shown to reduce both CO_2 emissions and fuel consumption by regulating the flow of vehicles at intersections [4].

In this paper, we present the V2V-communication-based Virtual Traffic Light (V^3TL) system, a dynamic, distributed solution for infrastructure-less management of unregulated intersections. The system is technology-agnostic, meaning that it can be implemented both using legacy WAVE/ITS-G5 protocols, as well as the novel C-V2X technology proposed by 3GPP. We make no assumption on who is driving the vehicle, therefore the application of our system to intersection management of self-driven vehicles is also possible.

This work presents several improvements with respect to the existing literature, namely: 1) we introduce a simplified, yet effective, intersection representation system, encoding all legal movements of vehicles in few bytes; 2) we introduce a distributed protocol, which allows selected vehicles to have a full view of the intersection before running the scheduling algorithm; 3) we describe a heuristic scheduler based on decision trees whose aim is to maximize the number of cars crossing the intersection at a given time and to minimize the number of stop-and-gos per vehicle, hence polluting emissions; 4) we evaluate our system in different intersection environments, i.e., isolated or consecutive three- and four-way junctions, in single- and multi-lane configurations.

Also, with respect to our previous work [5], the present paper: i) describes a more flexible, streamlined leader election protocol, introducing the possibility of resigning from leadership and joining an incomplete group of vehicles; ii) presents a complete complexity analysis of our heuristic scheduler and a comparison with an exhaustive-search algorithm; iii) takes into consideration more realistic mobility scenarios, like the unbalanced traffic scenario, a scenario with two consecutive intersections with multiple lanes per

[*] Corresponding authors:

ahmadreza.jame@polito.it (A. Jame); marco.rapelli@polito.it (M. Rapelli); claudio.casetti@polito.it (C. Casetti)

ORCID(s): 0000-0001-9259-5387 (M. Rapelli); 0000-0002-9507-8526 (C. Casetti)

direction and different types of common intersection (i.e., three-way and four-way crossroads); iv) features an entirely new results section, accounting for the changes introduced into the model and showing several additional metrics.

Section 2 of this paper is dedicated to related works, while in Section 3 technologies and use cases considered are described. Section 4 describes the four-steps procedure followed, the scheduling algorithm and the algorithm complexity. Section 5 reports the parameters setup used in our simulations, while Section 6 is dedicated to results. Conclusions and future works are presented in Section 7.

2. Related Work

In the last decades, several projects aimed to reduce the waiting time experienced by drivers at intersections and, consequently, to reduce emissions due to stop-and-go maneuvers. Back in 2009, the Travolution project [6] developed the Green Light Optimal Speed Advisory (GLOSA) system. GLOSA has the aim of reducing vehicle fuel consumption by predicting the next green traffic light phase and informing drivers whether they can pass in the current green phase or not. Thanks to GLOSA a vehicle can reduce its emissions by up to 22% in a single-car scenario or up to 8% for a congested scenario [7]. In 2012, MIZAR Automazione S.p.a. developed an urban traffic control system architecture called UTOPIA [8]. UTOPIA aims at synchronizing phases of different traffic lights on the same path according to live traffic patterns. Thanks to the UTOPIA phase-adaptation algorithm, travel times can be reduced by more than 15%.

Both systems cited above are currently used worldwide in many cities as working solutions for reducing traffic in regulated intersections. However, addressing the traffic in infrastructure-less intersections still remains an open problem. The definition of a Local Dynamic Map (LDM) [9], [10] by the European Telecommunication Standard Institute (ETSI) and by the International Organization for Standardization (ISO) and consensus finding studies [11] for decentralized information exchange between vehicles have shown a significant improvement in this research field. Leveraging those concepts, many studies in the literature have proposed car-to-car communication systems to schedule traffic at unregulated junctions. Ferreira et al. [12], [13] first introduced the idea of a Virtual Traffic Light (VTL) system. VTL exploits beacon communication between cars and leader-based message exchange in a completely infrastructure-less scenario. This system was designed to inform all the vehicles approaching the crossroads of the current VTL phase, although no optimization of the phases themselves was attempted.

From this idea, Hagenauer, Sommer et al. [14], [15] introduced a novel leader election algorithm and Bazzi et al. [16] performed a field test using vehicles equipped with low-cost IEEE 802.11p devices. Cruz-Piris et al. [17] suggested a grid schema of the intersection to assist the crossroads representation for VTL applications. Neither, however, introduced a scheduling algorithm to optimize the number of cars crossing the intersection in each phase. Additional studies [18, 19, 20, 21, 22, 23] present priority-based scheduling algorithms for autonomous vehicles, although not relying on infrastructure-less, vehicle-to-vehicle communication procedures that are essential in our solution.

Features	Literature Works
Infrastructure-based	[18], [19], [20], [21], [22], [23]
V2V-based	[5], [12], [13], [14], [15], [16]
Optimized Scheduler	[18], [19], [20], [21], [22], [23]
Emission Reduction	[5], [7], [8], [12], [13]

Table 1
Literature works comparison.

Table 1 reports all the main literature works about VTL studies, subdividing them according to their main characteristics.

3. Scenario assumptions

The purpose of V^{3}TL is to act as a virtual traffic light in unregulated intersections. The goal is to reduce the waiting times of drivers attempting to cross an intersection and to reduce the environmental impact by minimizing the number of stop-and-go maneuvers. In the following, we detail the assumptions we make on vehicle communication capabilities and intersections.

3.1. Communication settings

In the scenarios we are considering, depicting different intersections, vehicles are supposed to be able to communicate with each other at a close range, via V2V direct communication in a completely distributed environment, can create an LDM and are aware of the V^{3}TL protocol. In the following, we always assume a 100% technology penetration rate and we discuss lower penetration rates in Section 7.

The messages exchanged are of BSM (Basic Safety Message) type, as defined by the SAE J2735 standard [24]. They are broadcast by every vehicle at a frequency of 10 Hz (i.e., the standard frequency mandated by IEEE).

The main fields that were used to populate the BSM for our goals are:

- Transmission timestamp
- Vehicle anonymous random identification number
- Vehicle motion information, such as: position, speed, acceleration, heading, yaw angle, signaling lights (turns and breaks) and the identification number of the current road, lane and next junction
- Vehicle relative position with respect to the other vehicles grouped in the same direction
- Leader Dataset (explained below)
- Junction Dataset (explained below)
- Solution Dataset (explained below)

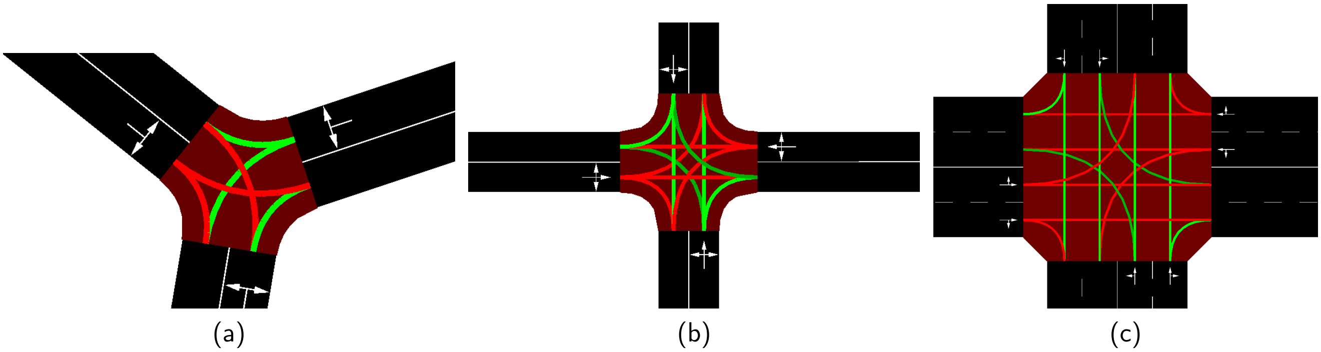
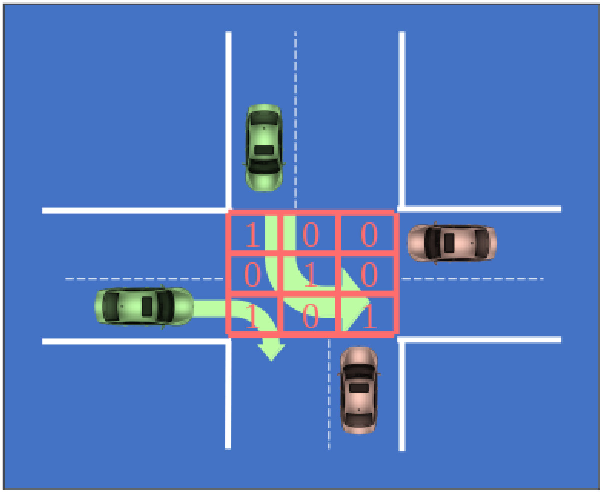


Figure 1: Right-to-pass priorities (green lines) in use case intersections. (a) Three-way intersection. (b) Four-way intersection with single lane. (c) Four-way intersection with double lane.

- Intersection Flag (to notify if the intersection has been crossed)
- Scheduled Flag (to notify if a vehicle has already been considered in the scheduling process)
- Leader Elected Flag (explained below)

Among these fields, transmission timestamp, vehicle identification number, vehicle motion information and relative position are part of the standardized BSM fields. All other fields can be mapped in the optional VehicleStatus field defined by the SAE J2735 standard [24] for the part II of a BSM, dedicated to non-critical safety applications.
The channel was modeled by taking into consideration simple path loss, obstacle shadowing and Nakagami fading attenuation models. To achieve a more realistic mobility environment, buildings alongside the roads merging into the intersection(s) were introduced so as to reduce the line of sight between vehicles. From a navigational perspective, all vehicles are assumed to carry a GNSS-capable device and can place themselves on a map that allows them to identify upcoming junctions.



1 0 0
0 1 0
1 0 1

Figure 2: Bitmap representation of a legal movement.

3.2. Intersection model

We focus on crossroads with no traffic lights and no roadside communication infrastructure, making it a full-fledged V2V scenario. We considered both single- and multiple-lane intersections, focusing on the former since they are frequently unregulated in an urban context. Among these we analyzed three-way and four-way junctions as depicted in Figure 1. A scenario with two consecutive intersections is also modeled in order to mimic a sort of "green wave" effect, i.e., the ability to sustain a long flow of vehicles across consecutive intersections by coordinating their green phases. For every intersection considered, the right-to-pass priorities are shown in Figure 1. Green lines represent the trajectories on which vehicles are allowed to cross simultaneously. In particular, bright green lines depict the move with right-to-pass, while dark green lines are used for left turns at intersections and represent priority-based moves.

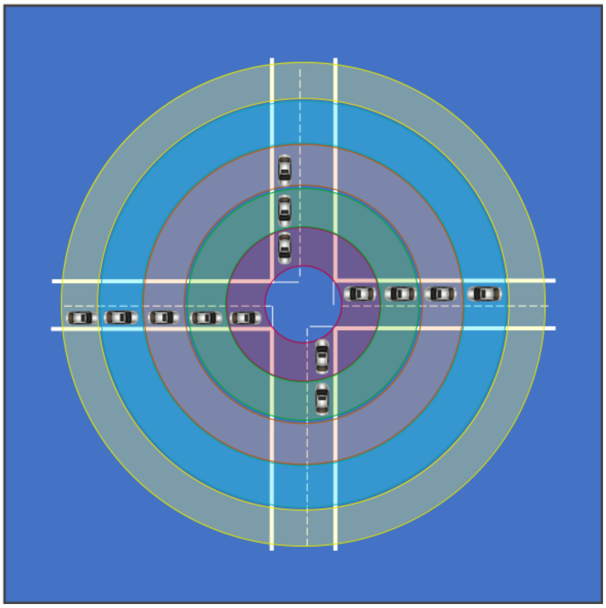


Figure 3: Tiers representation of the intersection.

In order to provide a compact representation of the intersection and the maneuvers to be input to the scheduling algorithm, we resorted to a bitmap, where every cell corresponds to a position in a lane that can be occupied by a vehicle. If a position is occupied, the cell in the bitmap model is marked with a 1, while a 0 indicates that no vehicle is present in that slot, as shown in Figure 2. Using this bitmap permits to distinguish between legal and hazardous/forbidden movements of vehicles. There are indeed some configurations of turning intentions at an intersection that either lead to a hazard situation (e.g., a vehicle turning left while an other is crossing straight on) or are not allowed by the road signage at that junction. Those configurations, of course, should be discarded by the scheduling algorithm, leaving only the legal moves, namely M, to be considered. The goal of such a bitmap is merely to differentiate between the M legal moves and the hazardous/forbidden ones of cars at the junction and it constitutes one out of many possible representations for this purpose. The comprehensive representation of the intersection configuration is indeed the input of the scheduling algorithm and it is instead represented by the *Junction Dataset*, detailed in the following. Specifically, in Figure 2 the southbound vehicle in green wishes to turn left, resulting in '1's in the main bitmap diagonal. The green vehicle from the eastbound street is turning right, as shown by the '1' in the bottom left cell of the bitmap. The other two gold vehicles are stopped and they do not appear in the bitmap representation. Indeed, assuming that the gold vehicle on the westbound street intends to continue straight and the gold vehicle on the northbound street intends to turn right, they would both generate overlapping '1's in the current bitmap representation, highlighting an intersection configuration that would almost certainly result in a hazard. A similar bitmap, with '1's is used to indicate every possible legal movement of the junction for the vehicles in the first position of the queue at the intersection, namely the first "tier". An example of a tier configuration of the intersection is provided in Figure 3, where we can see four cars in the first tier in red, four vehicles in the second tier depicted in green, three in the orange one, and so on.

4. V^{3}TL procedure

This Section describes the methodology and steps followed in order to define the V^{3}TL procedure. First, the main idea of the whole procedure is described in a step-by-step fashion, followed by a more detailed analysis of the scheduler algorithm. Finally, we address the algorithm complexity.

4.1. Overview

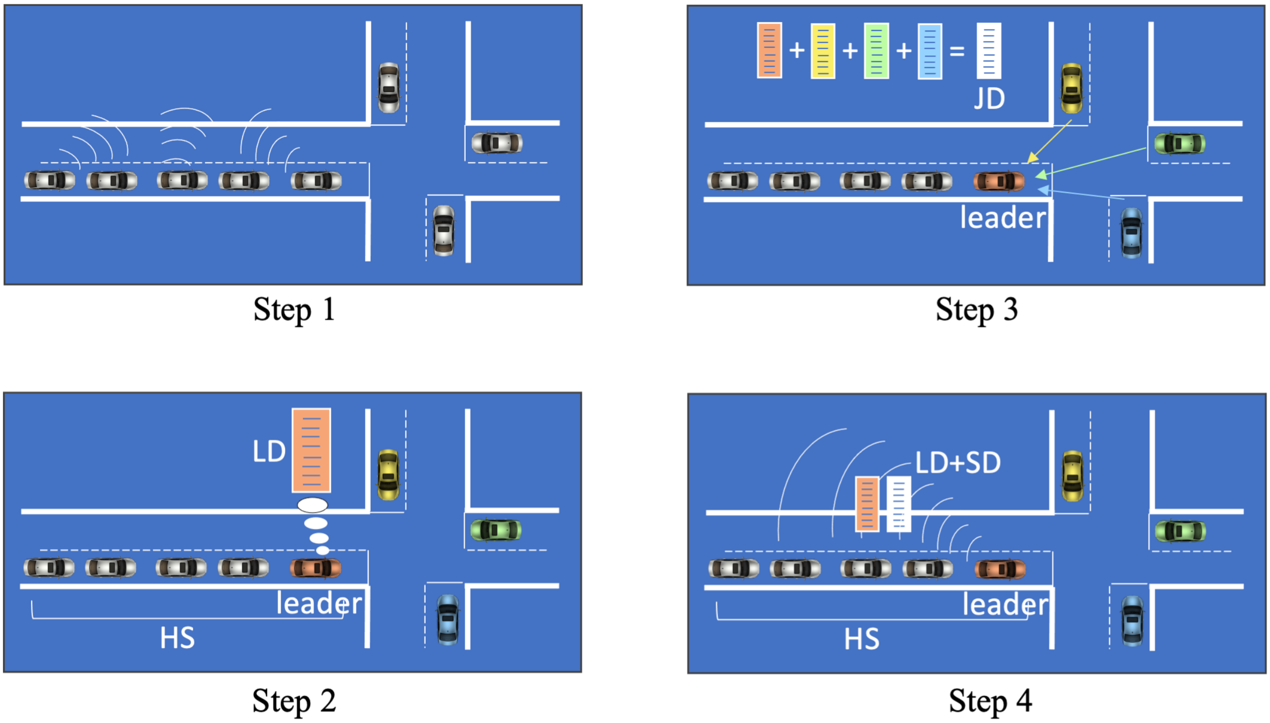
The V^{3}TL scheme works in a cyclic way, scheduling a variable number of vehicles per cycle. Every scheduling cycle is divided into four steps. A representation of the scheduling process is reported in Figure 4 and a pseudo-code of the whole V^{3}TL procedure is reported in Algorithm 1.

4.1.1. *Discovery procedure*

The first step of the cycle starts as soon as a vehicle finds itself within a *triggering distance* (discussed in Section 5.1) from the next intersection on the map. This phase aims at identifying groups of vehicles approaching the intersection from a similar direction. During this phase, vehicles exchange positional and navigational information, broadcasting it through BSMs in V2V communication and thus populating their LDMs. Since the goal of our model is not to regulate *any* intersection at *any* time, the next phase of the V^{3}TL procedure is triggered only when a congested situation is detected in one of the directions. Such a condition is verified when one of the vehicles detects $N_c - 1$ vehicles in the quadrants of the LDM superimposed to the road where it travels on, leveraging the information carried by the received BSMs. N_c is thus a crucial parameter that triggers the whole procedure. As soon as this happens, the vehicle then sets a flag called Leader Elected Flag in its BSM and starts the next phase of the V^{3}TL procedure.

Algorithm 1 V^{3}TL workflow.

```
Data: On BSM received
Update vehicle LDM
if ($N_c - 1$ other vehicles are in LDM are in my direction)
or (BSM has Leader Elected Flag set) then
    if (There is no leader elected) then
        Select a leader vehicle
        Set the Leader Elected Flag
    end if
    if (I am the leader) then
        Update LD
    end if
end if
if (BSM contains LD of other HSs) then
    Leaving current HS is no more allowed
    if (I am leader) then
        Merge received LD with others
        if (I have a complete JD) then
            Start the scheduling process
            Broadcast LS and SD
        end if
    end if
else
    if (BSM belongs to a HS closer to the intersection)
    then
        if (The HS of the transmitter allows a join) then
            if (I am leader) then
                Handover leadership to vehicle behind
            end if
            Leave the current HS and join the other
        end if
    end if
end if
```



Step 1

Step 3

Step 2

Step 4

Figure 4: Steps of information collection in a cycle. Step 1: vehicles exchange BSMs and populate their LDMs; Step 2: leader election, Heading Set creation and Leader Dataset compilation; Step 3: leaders exchange Leader Dataset, then merge them into the Junction Dataset; Step 4: Each leader broadcasts the Leader Dataset and the Solution Dataset to the rest of the vehicles.

4.1.2. Leader election and Heading Set identification

When a vehicle receives a BSM with the Leader Elected Flag set, it sets the flag on its own BSMs to propagate the information that the leader election should start everywhere around the intersection. Then, it initiates the procedure for the identification of the leader vehicle and thus the formation of the *Heading Set (HS)*, i.e., the group of vehicles that will collectively be scheduled by V^{3}TL. Ideally, consecutive vehicles find themselves in a HS if they are travelling in the same direction while approaching an intersection and if they are within radio visibility of each other. The vehicle closer to the intersection is elected[1] as leader and it takes it upon itself: i) to form the HS by selecting *no more than* the $N_c - 1$ vehicles behind, ii) to gather data from its HS and iii) to share such information among other leaders in neighboring roads stemming from the junction. It is to be noted that, thanks to the propagation of BSMs with the Leader Elected Flag set, vehicles approaching the crossroad from other directions will immediately initiate the election procedure after a complete HS formation in a direction. This process may lead to the formation of HSs of size less than N_c on one or more roads if there are not enough vehicles.
Joining and leaving a HS is still permitted in this second step of the procedure. Indeed, in a real-life situation, it is frequent that distances between vehicles are reduced in proximity of a crossroad due to traffic conditions. This kind of situation presents high variability in the mobility environment, so we decided to focus on this by creating a more flexible procedure in the leader election. In particular, we envisioned the possibility for a leader to resign its leadership and to join another HS as a normal member. This may happen when a leader of a HS recognizes another HS, closer to the junction, on the very same street, with fewer than N_c members. In this case, the leader of the group far from the intersection can join the HS approaching the crossroad, resigning its leadership (and handing over all the data thus far collected as a leader) in favor of the car immediately behind. The new leader can now start the same procedure until N_c cars are grouped in the same HS and the group is thus considered complete. In such a way, we can maximize the number of vehicles we can schedule in a single scheduling cycle, despite the traffic fluctuations in proximity of intersections.

This paradigm works even in scenarios where traffic is unbalanced between directions, with many vehicles queued up in a street and fewer cars approaching the intersection from other roads. Additionally, it can be easily extended to a multi-lane scenario. Vehicles in the HS are thus selected, progressively filling each tier with vehicles from different lanes that are at the same distance from the intersection, until up to N_c are selected. Then, the leader is selected as the rightmost vehicle in the first tier.

The leader is in charge of gathering information from its group members. Those data are saved in a *Leader Dataset (LD)*, which includes all the information of every vehicle

[1] Procedures for distributed consensus finding such as those in [11] can be used.

in the HS. In particular, for every vehicle, the following are saved: vehicle anonymous identifier, position in the queue, current lane, intersection identifier, signaling light (or, in its absence, the intention of going straight on) and current traveling direction (e.g., eastbound, southbound, westbound or northbound in a four-way intersection). The LD is very important for collecting common knowledge of the scenario in order to schedule all the vehicles synchronously. Therefore, in a real implementation, this exchange should be implemented using a robust high-layer protocol to ensure that all leaders have the same information. The definition of such a protocol is however outside the scope of this work.

4.1.3. *Leaders information exchange*

When leaders approach the intersection, they start sharing the LD of their HS, thus initiating step three of the procedure. In such a phase, joining or leaving a HS is no longer allowed and leaders start to merge LDs from different directions into a single *Junction Dataset (JD)*.
With the JD, leaders have a full view of the intersection and they can compute the scheduling solution, called *Solution Dataset (SD)*, as explained in the following subsection. Since leaders use the same JD as input, they will inevitably output the same SD.

4.1.4. *Scheduling procedure*

In the fourth and final step, leaders compute and broadcast the SD together with their LDs. In such a way, upon a SD reception, a vehicle can search in the LD if its identification number is present for the current schedule cycle. If this is the case, it sets the Scheduled Flag in its BSM and follows the schedule according to the SD. Every vehicle that was not part of the scheduled HS at the time of the JS creation will not find its identification number in the broadcast LD. This can happen if the vehicle is part of the HS that follows or if it has just reached the intersection and is not yet part of a HS. Those vehicles will not be scheduled and will have to wait for a new leader election procedure to start in the following cycle.

Algorithm 2 Scheduling algorithm.

```
Data: Complete JD
 1: while (There are still vehicles to schedule in the JD) do
 2:     j=0
 3:     while ($j < L$) do
 4:         Select all legal movements of first available tier
 5:         for all (Legal movements $A_{i,j}$) do
 6:             Compute $c(A_{i,j})$, $e(A_{i,j})$, $C_{i,j}$ and $E_{i,j}$
 7:         end for
 8:         j++
 9:     end while
10:     Select the best branch of the tree according to selection explained in text
11:     Update SD with current tier solution
12: end while
```

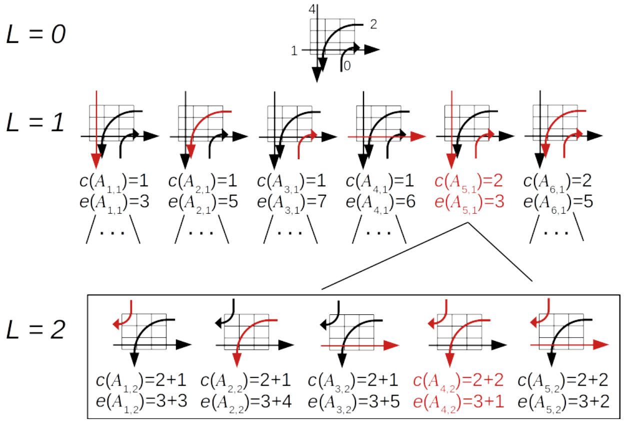


Figure 5: Example of an iteration of the scheduling algorithm. Level 0 shows the initial configuration with the turn intentions for the vehicles in the first tier. The small numbers near the arrows show how many vehicles in the HS are queued up behind the leaders and are useful for the computation of stop-and-gos. All the possible legal movements are inspected level by level and eventually the best configuration among all is chosen, according to the sum of the inspected parameters, in the very last level of the scheduler.

4.2. Scheduling solution computation

Given a configuration defined in the *Junction Dataset*, at most $H \cdot N_c$ cars can be scheduled at a time, where N_c is the maximum number of vehicles in a *Heading Set* and H is the number of headings in the intersection. Note that this is independent from the number of lanes P, since up to N_c vehicles are grouped per direction and not per lane. Among all the possible turning configurations of those $H \cdot N_c$ vehicles, only the M legal moves involving the first tier are selected, nominally $A_{i,1}$. For each $A_{i,1}$ with $i = 1, \dots, M$ the number of passing vehicles for that move is computed, $c(A_{i,1}) \in [1, H \cdot P]$, and likewise the number of generated stop-and-gos, $e(A_{i,1}) \in [0, H \cdot N_c - 1]$, identified as the number of halts issued per vehicle in the *Solution Dataset*.
The scheduling algorithm now proceeds as a decision tree, where every node $A_{i,j}$ is a legal movement characterized by its own $c(A_{i,j})$ and $e(A_{i,j})$ for $i = 1, \dots, M$ legal movements and for $j = 1, \dots, L$, where L represents the number of levels of the tree in depth for every iteration of the scheduling algorithm, see line 3 in Algorithm 2. L is thus a parameter that allows us to curb complexity. Furthermore, every node $A_{i,j}$ has associated cumulative values of passing vehicles $C_{i,j}$ and stop-and-gos $E_{i,j}$ resulting from the sum of individual $c(A_{i,j})$ and $e(A_{i,j})$ along its branch.
As can be seen in the example shown in Figure 5, for every node in the tree, multiple branches are generated, one per legal turning configuration. Notice that at level L, up to M^L values of $C_{i,j}$ are computed. When all the branches of the tree down to level L are computed, the scheduling solution $s(L)$ is chosen as the branch that maximizes $C_{i,j}$ at level L. In case of a lingering tie, the branch that both maximizes $C_{i,j}$ and minimizes $E_{i,j}$ at level L is taken as a solution. If there are two or more such branches, the solution is chosen randomly

among them. It is to be noted that, depending on the choice of L, less than $H \cdot N_c$ cars can be scheduled in this way. Therefore, this procedure is repeated in an iterative way until all the vehicles present in the *Junction Dataset* have been scheduled, as described in the pseudo-code of the scheduling procedure in Algorithm 2 and depicted in Figure 5, where an example of an iteration of the scheduling algorithm is shown.

Using such a paradigm, leader-vehicles are able to compute a scheduling solution in real time and broadcast it before the vehicles in HS reach the intersection. In order to minimize the size of the *Solution Dataset (SD)* on BSMs, a compact representation can be used. It is a matrix representation, composed by $H \cdot P$ columns and $r \in [1, H \cdot N_c]$ rows, where every row represents a legal movement selected by the scheduling algorithm. Therefore, every element refers to a single vehicle and contains the instructions for a specific vehicle finding itself at the head of the lane after the previous legal movement. In particular, every element of the matrix $SD_{i,j}$ is a tuple composed by the anonymous random identification number of the vehicle to which the instruction refers and the identification number of the road and the lane to which access has been granted. If the latter refers to the street on which the vehicle is currently traveling, then it means that the vehicle is forced to brake and stop before the intersection. If there is no vehicle in a given direction, the special element $(-1; -1)$ is used for a void position.

Figure 6 depicts a possible scenario after a *Solution Dataset* has been delivered to vehicles in a single-lane, four-way intersection. In particular, we can see the green vehicles from southbound and eastbound streets that are allowed to pass together, while cars from northbound and westbound are forced to stop. On the next move, the orange cars from eastbound and westbound can cross together. Finally, another vehicle from eastbound, in blue, is allowed to pass together with a vehicle from northbound. The corresponding SD for the scenario of Figure 6 is reported below.

N1	E1	S1	O1
(v11, E2)	(v8, E1)	(v37, S1)	(v6, S2)
(-1, -1)	(v8, O2)	(v37, S1)	(v2, E2)
(-1, -1)	(-1, -1)	(v37, N2)	(v13, S2)

4.3. Algorithm complexity

It is conceivable that, for small values of N_c, a brute-force approach can be computed with all the possible combinations of M legal moves for all the vehicles in the scheduling cycle. The output of the brute-force approach could be then stored a priori on vehicles for different values of N_c and for the most common values of $H \in \{3, 4\}$ and of $P \in \{1, 2\}$, i.e., three-way or four-way intersections with one or two lanes per direction. However, the complexity of the problem grows exponentially with respect to the number of L levels considered.

If we consider a H-way P-lanes intersection, we can count up to $H \cdot P$ vehicles in the first tier. Each of those vehicles can choose among $P \cdot (H - 1)$ possible direction (since the u-turn is not allowed) or to brake and stop before the junction, leading to $P \cdot (H - 1) + 1$ different choices. Considering only the first move, the total number of possible combinations (legal or not) are at most $[P \cdot (H - 1) + 1]^{H \cdot P}$, since there are $P \cdot (H - 1) + 1$ decisions for $H \cdot P$ vehicles. We remark that the possible movements are $[P \cdot (H - 1) + 1]^{H \cdot P} - 1$, since we discount the case of none of the cars making any move. Considering now a decision tree with L levels in depth, we can count $\{[P \cdot (H - 1) + 1]^{H \cdot P} - 1\}^L$ possible permutations.

However, the solution space of our problem is defined by a subset of the possible permutations above, defined by the number of M legal movements for each level L. The value of M must be manually computed for each combination of H and P, for example:

$$M = 13 \quad for \quad H = 3, \quad P = 1$$

$$M = 49 \quad for \quad H = 4, \quad P = 1$$

$$M = 311 \quad for \quad H = 4, \quad P = 2$$

Consequently, the complexity of a brute-force approach will be exponential with $\mathcal{O}(M^L)$. Using a decision tree paradigm, we exploit a low-complexity heuristic approach with an average complexity of $\mathcal{O}(L \log M)$.

It is important to notice that the L parameter refers to the number of levels in depth per iteration of the decision tree, not the number of vehicle tiers scheduled. The only chance of having L equal to the number of tiers is to schedule for each level the maximum number of vehicles, so that $c(A_{i,j}) = H \cdot P, \forall i = 1, \dots, M$ and $j = 1, \dots, L$, i.e., all cars in the first tier turning right. Since this probability is negligible in a realistic scenario, the computed complexity for scheduling a complete tier of vehicles is usually higher than for scheduling a level in depth of the decision tree.

5. Performance evaluation setup

Results shown in this section are obtained from simulations in which every vehicle is modeled with an On-Board

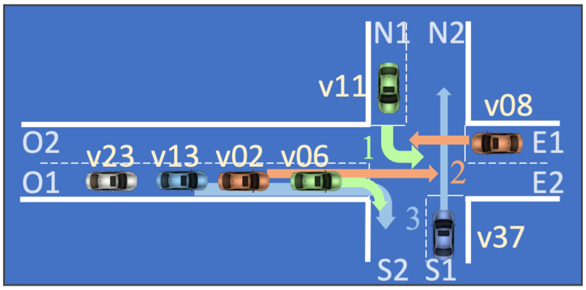


Figure 6: Example of three legal actions: action 1 allows the vehicle v11 from the southbound street and the vehicle v06 from the eastbound street to move simultaneously, followed by action 2 which dictates that v02 crosses, together with the westbound one v08; eventually, action 3 allows the northbound vehicle v37 to move together with v13 turning right.

Unit (OBU) equipment capable of transmitting and receiving BSMs.
SUMO (Simulator of Urban Mobility) [25] was used as a mobility simulator, while OMNeT++ [26] is the tool used in order to model the transmission between vehicles in our network. Finally, the Veins simulator [27] was used to interconnect SUMO and OMNeT++.

5.1. Environment

In order to evaluate the performance of the V^3TL procedure, three use cases were analyzed. We focused on a single-lane three-way intersection ($H = 3$ and $P = 1$), a single-lane four-way intersection ($H = 4$ and $P = 1$) and a scenario with two consecutive double-lane four-way intersections ($H = 4$ and $P = 2$). We believe that these use cases represent the vast majority of unregulated intersection types in an urban scenario.

In all these use cases we set the *triggering distance* to 300 m, a value that nominally results in a preemptive formation of up to 10 HSs scheduled per direction on average. Ultimately, this parameter does not have a significant impact as long as it allows to have at least one scheduled HS every time the previous HS has cleared the intersection. In perspective, it could also be a parameter inherently associated to the map depending on the distance between consecutive intersections.

Regarding generation flows, vehicles are created according to Poisson distributions with different values of inter-arrival time λ. In particular, for every use case, we developed a generation process where traffic is balanced between directions and λ is constant for every incoming road. In addition, for the single-lane three-way scenario and the single-lane four-way scenario, we also analyzed an unbalanced traffic model, where Poisson generation processes with different λ were used for every direction. Regarding the consecutive intersections scenario, the traffic distribution is unbalanced even if we use the same λ value for every input road. This is due to the particular mobility environment, which creates a main flow between intersections due to the topological characteristic of this scenario.
As a result, our system is dependent on the amount of traffic we inject into the model: we are interested in high traffic densities so as to highlight the performance of our solution. In particular, we have observed that values of inter-arrival time with $\lambda > 8$ s/veh per direction results in too shallow densities which fail to trigger the V^3TL procedure.

The evaluation of the performance of our model was conducted through a comparison with other two scenarios.
Firstly, we inspected a simple scenario, referred to as *Unregulated*, where vehicles cross the intersection without being scheduled, therefore, in a completely unregulated manner. In this case vehicles follow the “right-before-left” priority model used by SUMO.
A second scenario with a fixed traffic light was then developed, referred to as *Traffic Light*. Since, by default, SUMO emulates non-realistic control phases, the Webster method [28] was used. The Webster method is an algorithm used by civil engineers to set traffic light phases according to the current incoming traffic. The goal of the Webster method is to define an optimum cycle length C and optimum green light phases G_i using the following formulas:

$$C = \frac{1.5 \cdot T + 5}{1 - \sum_k y_k} \tag{1}$$

$$G_i = \frac{(C - T) \cdot y_k}{\sum_k y_k} \tag{2}$$

where T is the total lost time for every cycle due to human reaction times. In the formulas, y_k is the critical flow rate, computed as the ratio between the number of vehicles per direction and the saturation flow, i.e., the maximum number of vehicles that can pass the intersection in an hour. For every single-lane use case considered, we will set a saturation flow of 3600 veh/h for both balanced and unbalanced models.

5.2. V^3TL parameters

First of all, the maximum number of vehicles that can be grouped into a *Heading Set*, namely N_c, was chosen to be 6, so as to limit the size of the queue to approximately 30 m from the intersection[2].
Recall that we defined as L the number of levels in depth for every iteration of the scheduler. On every iteration of the scheduler, the local optimum solution across L rows is written in the *Solution Dataset* matrix made of $r \in [1, H \cdot N_c]$ rows in total. However, building a complex tree for every iteration dramatically affects the algorithm processing time. Since leader vehicles have to compute the scheduling solution on the fly, the decision tree procedure has to abide to tight time limits. The presence of buildings at the intersection can hinder communication between a leader and vehicles coming from perpendicular directions. Therefore, leaders may only receive transmissions from nearby crossing roads approximately 5 m before stopping at the intersection (simulated values in this case can be considered a worst-case scenario for the attenuation in real systems). Considering a BSM transmission frequency of 10 Hz and negligible propagation and processing times, the scheduler procedure has to output a solution within 260 ms for a scenario with vehicles moving at 50 Km/h[3]. In a single-lane four-way scenario, measuring the time it takes for the scheduling algorithm to be executed on the hardware used for our simulations (2.5-GHz QuadCore i7), we observed 123.9 ms for $L = 2$ and 434.49 ms for $L = 3$. In order not to exceed the processing time limit of 260 ms, a scheduling algorithm with $L = 2$ levels in depth was used.

6. Results

We evaluated the performance of the system by inspecting two main metrics: the average number of cars per minute

[2] the vehicle length of a standard passenger car in SUMO is 4.3 m.

[3] 360 ms to drive 5 m at 50 Km/h minus the worst case waiting time for the next BSM transmission, 100 ms.

passing the intersection and the average number of stop-and-go maneuvers per vehicle. The first metric measures congestion, i.e., the clearance rate for the intersection. The number of stop-and-go maneuvers, instead, is considered as a secondary goal. A vehicle that performs fewer stop-and-gos on average will pollute less. Additionally, for the consecutive intersections scenario, a fairness metric was introduced to evaluate how much the flow of vehicles crossing both intersections is favored with respect to other flows. Eventually, we analyzed the dependence of our system with respect to two crucial parameters: the average vehicle interarrival time λ and the maximum number of cars in a *Heading Set*, i.e., N_c.
All results are averaged over ten simulation runs, each corresponding to a simulated time of one hour.

6.1. Three-way intersection

The first use case we analyzed is a three-way intersection with a single lane per direction. For the balanced scenario, we generated vehicle flows of $v_1 = v_2 = v_3 = 600$ vehicles, with critical flow rates of the Webster method $y_1 = y_2 = y_3 = \frac{1}{6}$ and inter-arrival time $\lambda = 6$ s/veh per direction. Assuming the lost time per phase $T = 6$ s (i.e., 1 s of amber phase and 1 s of clearance interval per phase), we obtain an optimal cycle length $C = 28$ s and green phases $G_1 = G_2 = G_3 = 7.33$ s. For the unbalanced scenario, vehicle flows of $v_1 = 900$ vehicles, $v_2 = 720$ vehicles and $v_3 = 180$ vehicles were modeled with inter-arrival times of $\lambda_1 = 4$ s/veh, $\lambda_2 = 5$ s/veh and $\lambda_3 = 20$ s/veh. The corresponding critical flow rates are $y_1 = 0.25$, $y_2 = 0.2$ and $y_3 = 0.05$, resulting in $C = 28$ s cycle length and $G_1 = 11$ s, $G_2 = 8.8$ s and $G_3 = 2.2$ s green phases (using $T = 6$ s). Traffic light phases of balanced and unbalanced scenarios are reported in Figure 7.

In Table 2 the performances of V^{3}TL scheduler are shown for the three-way intersection use case. In particular, we can see the average number of cars per minute passing the intersection and the average number of stop-and-gos per vehicle per minute. Metrics are reported for all comparison scenarios and for every traffic distribution. From the table, it is possible to see how the scheduler procedure outperforms both unregulated and traffic light scenarios. In particular, the scheduler procedure improves the unregulated scenario by 2.68% of passing cars per minute and by 17.07% of stop-and-go maneuvers for the balanced scenario (4.61% and 15.43% for the unbalanced traffic).

Distribution	Scenario	Cars/min	Stop-and-go/min
Balanced	V^{3}TL	48.24	1.02
	Unreg.	46.98	1.23
	TL	32.14	4.43
Unbalanced	V^{3}TL	39.46	1.37
	Unreg.	37.72	1.62
	TL	26	6.29

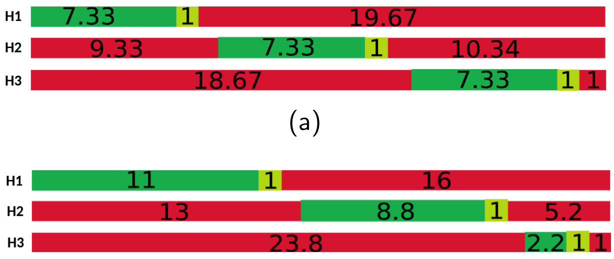
Table 2
Three-way intersection results for the V^{3}TL system, the unregulated case and the traffic light case.

It is possible to notice how the choice of a traffic light in this intersection type is penalizing since a dedicated phase is needed for every heading (a four-way intersection only needs two phases). The V^{3}TL scheduler significantly improves the traffic light scenario, as reported in Table 2.

6.2. Four-way intersection

The second use case focuses on a single-lane, four-way intersection. In this model, a balanced scenario was created using $v_1 = v_2 = v_3 = v_4 = 900$ vehicles in every direction with inter-arrival times of $\lambda = 4$ s/veh. The corresponding critical flows are $y_{1\wedge 3} = y_{2\wedge 4} = 0.25$, resulting in $C = 22$ s optimum cycle length and two green phases of $G_{1\wedge 3} = G_{2\wedge 4} = 9$ s (we used a total lost time of $T = 4$ s, i.e., 2 s per phase divided into amber phase and clearance interval). The unbalanced scenario uses vehicle flows $v_1 = v_3 = 1440$ vehicles and $v_2 = v_4 = 360$ vehicles with corresponding inter-arrival times $\lambda_1 = 2.5$ s/veh and $\lambda_2 = 10$ s/veh. Following the Webster method, we computed critical flow rates $y_{1\wedge 3} = 0.4$ and $y_{2\wedge 4} = 0.1$, optimum cycle length $C = 22$ s and green phases $G_{1\wedge 3} = 14.4$ s and $G_{2\wedge 4} = 3.6$ s. Both the traffic light phases of balanced and unbalanced traffic are reported in Figure 8.

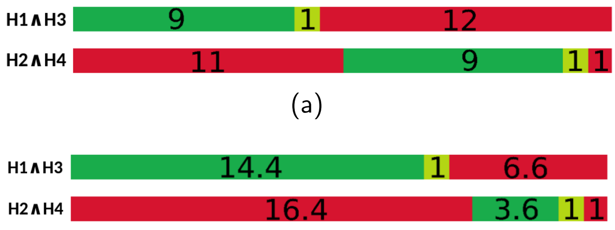
Table 3 reports the average number of passing cars per minute and the average number of stop-and-go maneuvers per vehicle per minute for both balanced and unbalanced traffic distributions of a four-way intersection. In this use



(a)

(b)

Figure 7: Traffic light phases of a three-way intersection (with headings H1, H2 and H3) computed with the Webster method. The phases are expressed in seconds. (a) Balanced traffic. (b) Unbalanced traffic.



(a)

(b)

Figure 8: Traffic light phases of a four-way intersection (with headings H1∧H3 and H2∧H4) computed with the Webster method. Numbers reported in figure are expressed in seconds. (a) Balanced traffic. (b) Unbalanced traffic.

Distribution	Scenario	Cars/min	Stop-and-go/min
Balanced	V^{3}TL	37.96	1.56
	Unreg.	26.88	1.85
	TL	28.8	2.21
Unbalanced	V^{3}TL	35.14	0.91
	Unreg.	27.64	1.39
	TL	33.82	1.23

Table 3
Four-way intersection results for the V^{3}TL system, the unregulated case and the traffic light case.

case, we have two flows of vehicles from opposite directions of the junction, resulting in only two traffic light phases. For this reason, the traffic light scenario handles more vehicles per minute than the unregulated model. Conversely, the number of stop-and-gos increases due to the phase behavior of traffic lights. Although, V^{3}TL scheduler performs better with respect to both unregulated and traffic light for balanced and unbalanced scenarios with improvements up to 41.22% for scheduled vehicles and up to 34.53% for stop-and-go maneuvers.

6.3. Consecutive four-way, double-lane intersections

Eventually, a more realistic, complex scenario was taken into account. We modeled two consecutive four-way intersections in order to study the performance of our system for a continuous flow of vehicles in a sort of green-wave effect. For this scenario, every direction is modeled as a double-lane road, so we measured a saturation flow of 4800 veh/h per intersection. It is important to underline how the traffic distribution here differs from the other use cases: due to the turning choices of vehicles at intersections, introducing vehicles with the same inter-arrival times from every direction will not lead to a balanced traffic scenario. For this reason, a single distribution with vehicle flows $v_1 = \dots = v_6 = 1200$ vehicles was developed, where a third of vehicles cross both intersections while others reach their destinations after passing one intersection only. The inter-arrival time is $\lambda = 4$ s/veh for all directions and the critical flow rates are $y_{1\wedge 3} = y_{2\wedge 4} = 0.25$ for both intersections. This results in two optimal cycle lengths of $C = 22$ s and green phases $G_{1\wedge 3} = G_{2\wedge 4} = 9$ s for both crossing cycles ($T = 4$ s, i.e., 2 s per phase divided into amber phase and clearance interval), as reported in Figure 8a.

For this use case, we are also interested in measuring the fairness of the system. Indeed, an unfair scheduling procedure may lead to long queues on roads between intersections. In order to analyze this metric, we introduced a fairness index F using Jain's Fairness formula [29]:

$$F = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} x_i^2} \tag{3}$$

where n is the number of roads in the scenario (i.e., 8 for this use case since we have two four-way intersections) and x_i, $\forall i = 1 \dots n$ is the ratio between the measured output traffic of a direction and the ideal output traffic equally shared between directions. Values of measured x_i for every scenario are reported in Table 4. Since x_i is a ratio of output traffic, the sum of all directions coincides with the number of roads in our scenario.

Heading	V^{3}TL	Unregulated	Traffic Light
1	0.29	0.42	0.28
2	0.39	0.21	0.26
3	1.27	0.58	1.07
4	1.15	0.48	1.11
5	1.28	0.56	1.44
6	1.16	1.92	1.42
7	1.19	1.76	1.15
8	1.27	2.06	1.24
SUM	8	8	8

Table 4
Values of x_i per direction for Jain's Fairness formula.

Scenario	Cars/min	Stop-and-go/min	Fairness
V^{3}TL	78.62	0.98	87.1%
Unreg.	55.76	1.62	65.96%
TL	73.54	1.58	84.45%

Table 5
Consecutive four-way intersections results for the V^{3}TL system, the unregulated case and the traffic light case.

Jain's Fairness formula is broadly used in computer networks to measure the fairness of the system considering the different data flows. It reports an index between 0 and 1, which refers to the fairness percentage of the system.

The results of the consecutive intersections use case are reported in Table 5. From the table, we can see how the scheduler procedure brings significant improvements for both passing cars per minute and stop-and-go maneuvers per vehicle per minute. Additionally, it also increases the fairness of the considered system with a significant enhancement of 32.05% with respect to the unregulated scenario, while keeping in par with a traffic-light regulated intersection.

6.4. Analysis on different λ and N_c values

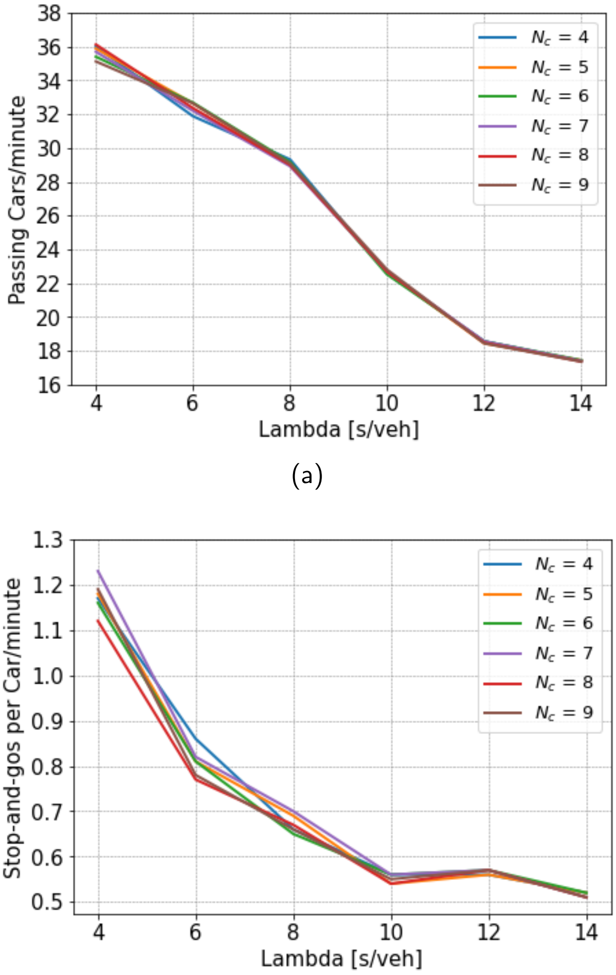
In order to study the performance of our system with different parameter settings, additional simulations were performed. Specifically, in this subsection we present results obtained from a four-way, single-lane scenario with balanced traffic generation rates. This additional study aims at inspecting the behavior of the proposed approach with different values of vehicle interarrival times λ and different instances of the maximum number of vehicles in a *Heading Set*, i.e., N_c. Each set of simulations was performed in a balanced scenario, averaging the results over five runs, each corresponding to a simulation time of ten minutes.

At first glance, Figure 9 reveals that the impact of N_c is marginal, while a larger role is played by the vehicle density at the intersection. Figure 9a highlights how high interarrival times (i.e., $\lambda > 8$ s/veh) result in shallow densities, which yield low values of passing cars per minute. The corresponding number of stop-and-gos for the same λs (reported in Figure 9b) is not to be attributed to a better performance of the scheduler, but only to fewer vehicles populating the intersection, which results in a lower number of stop-and-gos per car every minute.

7. Discussion and conclusion

This paper presents V^{3}TL, a V2V Virtual Traffic Light system for managing traffic at unregulated intersections in an entirely distributed manner. We defined a procedure made of four steps in order to gather all the data required for a complete view of the intersection status and for a proper scheduling solution.

A complete description of our scheduling algorithm is also given, describing how decision trees were built for every iteration to select local optima in a heuristic way. We focused on three use cases that resemble the majority of unregulated intersection types in an urban environment. The considered scheduling criteria are the number of vehicles crossing the intersection in a minute and the number of stop-and-go maneuvers per vehicle, identified as the main metrics for traffic congestion and pollutant emissions, respectively. We also analyzed the fairness of the system in a complex, two-intersections scenario.



(a)

(b)

Figure 9: Comparison for different λ and N_c values. (a) Number of passing cars per minute. (b) Number of stop-and-gos per vehicle per minute.

Simulations show how the proposed system leads to significant improvements concerning unregulated intersections and traffic-light-regulated ones. Enhancements have been observed for all inspected metrics in both scenarios with balanced and unbalanced traffic distributions.

As with all cooperative vehicular applications, our system guarantees the performance we have illustrated only for very high penetration rate values. Solving this issue is not the purpose of the present work. Nevertheless, a little insight of a fallback procedure to disable the scheduling process in the case of one or more unequipped cars is still given here.

Since a vehicle not equipped with V2V hardware is uncapable of communicating with others in any way, the first problem is how to identify the presence of such a car. A valid method could be using car sensors: if a neighbor of an unequipped car detects, through front or rear sensors, the presence of a vehicle unidentified via V2V communication, then it could assume its presence, in what is usually known as "collective perception" [30]. In such a situation, the complex scheduling process described in this paper may not produce a valid outcome since it assumes the full knowledge of the vehicles at the intersection. For this reason, when an unequipped car is sensed, the V^{3}TL system performs a fallback procedure consisting in a scheduling solution computed on the vehicles ahead of the unequipped one and a "special green phase", with all other cars stopped, is reserved to let the unequipped vehicle transit. We will explore such a variant in our future work, where we also plan to inject real traffic flows into our system and inspect its behavior for more complete, urban-scale scenarios.

References

[1] W. H. O. WHO, Burden of disease from household air pollution for 2016 (2018).

[2] J. Cloke, G. Harris, S. Latham, A. Quimby, L. Smith, C. Baughan, Reducing the environmental impact of driving: a review of training and in-vehicle technologies, TRL REPORT 384 (1999).

[3] M. André, U. Hammarström, Driving speeds in europe for pollutant emissions estimation, Transportation Research Part D: Transport and Environment 5 (5) (2000) 321–335.

[4] B. Bradaï, A. Garnault, V. Picron, P. Gougeon, A green light optimal speed advisor for reduced co 2 emissions, in: Energy Consumption and Autonomous Driving, Springer, 2016, pp. 141–151.

[5] M. Rapelli, C. Casetti, M. Sgarbi, A distributed v2v-based virtual traffic light system, in: 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, 2020, pp. 122–128.

[6] R. Braun, C. Kemper, C. Menig, F. Busch, R. Hildebrandt, I. Paulus, R. Pre ss lein Lehle, F. Weichenmeier, Travolution network-wide optimization of the light signal control and lsa vehicle communication, Road ss enverkehrstechnik (6) (2009) S – 365.

[7] T. Tielert, M. Killat, H. Hartenstein, R. Luz, S. Hausberger, T. Benz, The impact of traffic-light-to-vehicle communication on fuel consumption and emissions, in: 2010 Internet of Things (IOT), IEEE, 2010, pp. 1–8.

[8] M. A. SpA, Utopia–urban traffic control system architecture, MIZAR, Verona (2012).

[9] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and guidance on standardization, Standard, European Telecommunication Standard Institute (ETSI) (2011).

[10] Intelligent transport systems - co-operative its - local dynamic map, Technocal report, International Organization for Standardization (ISO) (2018).

[11] J. A. Fax, R. M. Murray, Information flow and cooperative control of vehicle formations, IEEE transactions on automatic control 49 (9) (2004) 1465–1476.

[12] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, O. K. Tonguz, Self-organized traffic control, in: Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking, 2010, pp. 85–90.

[13] M. Ferreira, P. M. d'Orey, On the impact of virtual traffic lights on carbon emissions mitigation, IEEE Transactions on Intelligent Transportation Systems 13 (1) (2011) 284–295.

[14] F. Hagenauer, P. Baldemaier, F. Dressler, C. Sommer, et al., Advanced leader election for virtual traffic lights, ZTE Communications, Special Issue on VANET 12 (1) (2014) 11–16.

[15] C. Sommer, F. Hagenauer, F. Dressler, A networking perspective on self-organizing intersection management, in: 2014 IEEE World Forum on Internet of Things (WF-IoT), IEEE, 2014, pp. 230–234.

[16] A. Bazzi, A. Zanella, B. M. Masini, A distributed virtual traffic light algorithm exploiting short range v2v communications, Ad Hoc Networks 49 (2016) 42–57.

[17] L. Cruz-Piris, M. A. Lopez-Carmona, I. Marsa-Maestre, Automated optimization of intersections using a genetic algorithm, IEEE Access 7 (2019) 15452–15468.

[18] A. Casimiro, E. Ekenstedt, E. M. Schiller, Membership-based manoeuvre negotiation in autonomous and safety-critical vehicular systems, arXiv preprint arXiv:1906.04703 (2019).

[19] J. Lee, B. Park, Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment, IEEE transactions on intelligent transportation systems 13 (1) (2012) 81–90.

[20] K. Dresner, P. Stone, A multiagent approach to autonomous intersection management, Journal of artificial intelligence research 31 (2008) 591–656.

[21] A. Giridhar, P. Kumar, Scheduling automated traffic on a network of roads, IEEE Transactions on Vehicular Technology 55 (5) (2006) 1467–1474.

[22] Y. Li, Q. Liu, Intersection management for autonomous vehicles with vehicle-to-infrastructure communication, PLoS one 15 (7) (2020) e0235644.

[23] Y. Zhang, A. A. Malikopoulos, C. G. Cassandras, Decentralized optimal control for connected automated vehicles at intersections including left and right turns, in: 2017 IEEE 56th Annual Conference on Decision and Control (CDC), IEEE, 2017, pp. 4428–4433.

[24] Dedicated Short Range Communications (DSRC) Message Set Dictionary, Standard, Society of Automotive Engineers (2015).

[25] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, Microscopic traffic simulation using sumo, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 2575–2582.

[26] A. Varga, Omnet++, in: Modeling and tools for network simulation, Springer, 2010, pp. 35–59.

[27] C. Sommer, R. German, F. Dressler, Bidirectionally coupled network and road traffic simulation for improved ivc analysis, IEEE Transactions on mobile computing 10 (1) (2010) 3–15.

[28] F. V. Webster, Traffic signal settings, Tech. rep. (1958).

[29] R. K. Jain, D.-M. W. Chiu, W. R. Hawe, et al., A quantitative measure of fairness and discrimination, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA (1984).

[30] T. ETSI, 103 562 v2.1.1, Intelligent transport systems (ITS).

Ahmadreza Jame received his B.Sc. in Computer Software Engineering in 2013 and his M.Sc. in Communications and Computer Networks Engineering at Politecnico di Torino, in 2020. He has some years of hands-on experience in computer and wireless networks. His main area of interest is IoT, Computer Networks, and Vehicular Networks.

Marco Rapelli received his B.Sc. in Telecommunications Engineering (2015) and his M.Sc. in Computer and Communication Networks Engineering (2017) both at Politecnico di Torino. He is part of FULL (Future Urban Legacy Lab), an inter-disciplinary center of Politecnico di Torino, where, in November 2018, he started his Ph.D. His main research interests focus on mobility studies and large-scale urban traffic simulators.

Claudio Casetti is a full professor with Politecnico di Torino, Turin, Italy. His research interests include vehicular networks, 5G networks, IoT. He is a senior editor of IEEE Vehicular Technology Magazine. He is the chair of the Turin Urban Digital Mobility working group within the Smart Roads project fostered by the City of Turin.