

Edge-Based Control of Multi-Platoons

Christian Quadri¹, Vincenzo Mancuso², Marco Ajmone Marsan², and Gian Paolo Rossi¹

¹*Computer Science Department, Università degli Studi di Milano, Milan, Italy*

²*IMDEA Networks Institute, Madrid, Spain*

Abstract— Vehicle platooning is expected to significantly increase road utilization while reducing transport cost and driver fatigue. Of course, the larger the platoon size the higher the efficiency. However, long platoons generate challenging road conditions when, for instance, platoons must allow for cross traffic at intersections, roundabouts and highway junctions. In this paper, we study the performance of platoons that split and merge in response to the traffic and road context. We propose PLATO, an edge-based control system to efficiently manage the resulting set of coordinated sub-platoons, which acts as a platoon of platoons, i.e., a *multi-platoon*. We analyze costs and benefits of a multi-platoon by means of analysis and realistic simulations. We show that what is most critical for good performance is the coherence of the instructions received by individual platoons, and the optimal adaptation of the number of sub-platoons in response to the changing road and traffic conditions.

Index Terms—Adaptive multi-platooning; Cellular network; MEC; Optimization; Simulation

I. INTRODUCTION

Coordinating the maneuvers of vehicles to form platoons is becoming a reality, with a growing market fostered by freshly sprouting commercial applications [1]. The reason why platooning is attracting the attention of researchers and logistic companies is that it is an application of automated driving that promises to lower transport costs and driver stress, while improving road utilization, hence reducing congestion.

While the idea of platooning first appeared in the 70s [1], [2], it has become viable only in the last few years, thanks to the rise of advanced traffic management systems (ATMS) and autonomous driving. Making a platoon stable in terms of inter-vehicle distance proved to be a challenging task requiring low-latency communications between vehicles. For this reason, platooning control is normally seen as a vehicle-to-vehicle (V2V) application, although vehicle-to-anything (V2X) communications have been considered as well, for instance with road-side units (RSUs) deployed along the road to provide connectivity and computing [3].

What is challenging and needs to be addressed urgently is how to support platooning at scale. One aspect of scaling consists in managing long platoons, for which V2V communications might soon become insufficient due to limited bandwidth resources and latency when multi-hop wireless paths must be used [4]. A second aspect is cross-platoon control, which becomes necessary when many platoons share the same road. For instance, platoons need to coordinate at highway junctions, in order to allow merging, or leave space for vehicles to reach the exit.

We address these challenges by proposing a two-tier platooning control system assisted by the cellular infrastructure.

Such a control system, which we call PLATO (Platoon Layout Architecture for Traffic Optimization), handles multiple platoons and instructs them to merge or split with carefully designed smooth and safe transitions and depending on road traffic conditions. We focus on the simple case of a group of vehicles that travel together, although they can split in several subgroups, each of them organized as a legacy platoon, and the ensemble moving as a platoon of platoons. We call this structure a *multi-platoon* formed by *sub-platoons*, and we remark that sub-platoons are not fixed in shape, number of vehicles and distances maintained with other sub-platoons, but flexibly adapt to the dynamics of road demand.

PLATO is the first proposal for multi-platoon control, and its purpose is to harvest the potentiality of edge computing in cellular networks to build a system in which multiple back-to-back platoons can interact and coordinate under ideal operational conditions. However, building a full-fledged intelligent transportation system for the control of vehicles over multiple lanes, as well as its interaction with manned vehicles, is out of the scope of this work and will require further investigation.

As discussed in a preliminary conference paper [5] and depicted in Fig. 1, our PLATO architecture defines new virtual network functions (VNFs) to handle multi-platoon and vehicle status information and to issue driving instructions. Each sub-platoon has a dedicated VNF controller (the platoon controller) and all platoon controllers are coordinated by one multi-platoon controller VNF, so as to coherently instruct sub-platoons and keep the platoon of platoons stable.

While an initial assessment of PLATO's performance was presented in our preliminary conference paper [5] using a detailed home-grown simulator based on SUMO [6], here we focus on cost and benefits of dynamically reconfiguring the multi-platoon into sub-platoons as a consequence of road traffic conditions and constraints. For that purpose, we define a refined platoon utility function that weights road utilization, fuel costs, and connectivity and computing overheads.

Besides pointing at the fact that sub-platoon controller VNFs must be instantiated close to the multi-platoon VNF, our results show that optimizing the dynamics of multi-platoons is a non-linear problem that involves the evaluation of multiple intertwined configuration parameters, as well as the mechanical constraints of the vehicles involved in the multi-platoon, and the presence of other road users.

The main contributions of this paper are:

- the design of a control system, named PLATO, to efficiently manage and coordinate multi-platoons as a function of variable road traffic conditions, target cruising

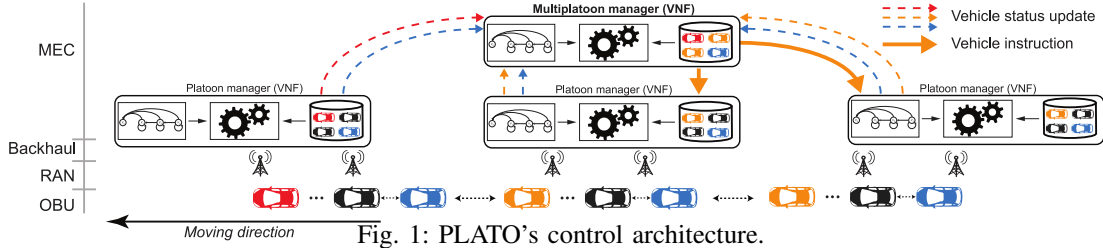


Fig. 1: PLATO's control architecture.

- speeds and junction crossing, relying on 3GPP-defined 5G-edge communication and computation mechanisms;
- a utility function accounting for costs and benefits of multi-platoon dynamic reconfiguration, which improves the static solution of our previous conference paper [5];
 - the formulation and solution of the Multi-platoon Re-configuration Problem (MRP) that maximizes the utility function and provides the optimal multi-platoon configurations and harmonized maneuvering coordination instructions to both preserve the platoon benefits and allow for cross traffic at highway junctions;
 - an extensive performance evaluation using a full-fledged simulator based on OMNeT++ and SUMO, showing the effectiveness of PLATO and validating the optimality of the solution of MRP's solution.

Through the analysis, we show that it is possible to augment classical platoon controllers to smoothly and safely manage any multi-platoon reconfiguration with timing guarantees in a wide variety of road traffic and communication conditions.

The remainder of this article is organized as follows. Section II discusses related work. Section III introduces PLATO and discusses its overheads, whereas we define the general structure of costs and benefits of multi-platooning in Section IV. Section V analyses the dynamic behavior of multi-platoons traveling along a highway junction, and Section VI discusses the corresponding optimization. We evaluate PLATO in Section VII and conclude in Section VIII.

II. RELATED WORK

Platooning is meant to improve road utilization and reduce fuel consumption thanks to the drag reduction effect, which has been evaluated in wind tunnels [7], [8] and on the road [9], showing gas saving of as much as 12%. However, the exact fuel saving of an individual vehicle changes with its position in the platoon [8], [9]. In addition, platooning enhances traveler's comfort and increases safety [10], [11]. As such, it was the subject of several research efforts, including the European Commission Horizon 2020 ENSEMBLE research project (ENabling Safe Multi-Brand pLatooning for Europe) [12]. A recent comprehensive overview of platoon coordination approaches is provided by [13].

Platoon control architecture and communication – Most existing proposals rely on V2V (typically through IEEE 802.11p) or V2X (typically with dedicated short-range communications as defined by 3GPP) for inter-vehicle communications [10], [14], [15], [16], [17], [18]. In those papers, the authors unveil the need for integrating control and communication functions, otherwise maintaining short inter-vehicle distances becomes impossible, especially as the platoon size

grows. In particular, in [17] a theoretical model of platoon message delays has been presented, providing the community with reference values for performance evaluation of 802.11p technology. What was also observed in those works is that, under some circumstances, and due to communication delays, the error on inter-vehicle distance can become larger and larger towards the tail of the platoon. When instead such error remains stable, the platoon is said to be *string stable*, and the most challenging problem becomes to control the distance between the leader and its first follower, the other vehicles observing smoother variations and smaller control errors.

The above works highlight that radio interference, shadowing and multi-hop transmissions with V2V cause string instability in the case of long platoons. In contrast, broadband cellular networks like 5G provide better support and can even be used to enable the virtualization of the platoon controller outside the platoon, e.g., at the edge of the cellular network [19], [20], [21], [22]. In this work, we follow this research stream and design PLATO to work as an ensemble of chained VNFs in a vehicular edge computing (VEC) framework, e.g., enabled by the Radio Access Network (RAN), V2X and Multi-access Edge Computing (MEC) entities as defined by 3GPP [23].

Multi-platooning – The concept of multi-platooning is not new. It has been proposed and studied in the context of V2V or by using cellular resources plus RSUs deployed along the vehicle path [4], [5], [24], [25], [26]. In addition to raising communication issues, multi-platooning poses challenges in controlling vehicle maneuvers [25], [27], [28], [29]. In particular, [4] compares the 3GPP evolved multimedia broadcast multicast service (eMBMS) to device-to-device (D2D) technologies for multi-platooning, and claims that the desirable solution is a form of multicast D2D. [26] proposes to use legacy V2V within each platoon, while the RAN should provide support to coordinate all platoons centrally, via an ATMS controller. Similarly, Rubin *et al.* [24] develop a traffic management multi-platoon solution based on V2V communication and show that there exists a fundamental trade-off between the maximization of road utilization and networking latency. Our proposal is different because PLATO has a two-tier control architecture relying on two layers of VNFs that run on edge computing resources. PLATO takes full advantage not only of the negligible interference in the RAN and in the backhaul connecting VNFs, but also of the geographic scalability of edge resources. This makes PLATO ready for integration within the vehicular networking vertical of 5G and with any edge/cloud-based intelligent transportation systems.

Multi-platooning maneuvering – Maneuver planning and

platoon reconfiguration were studied to handle different traffic situations, such as vehicles joining/leaving platoons, obstacle avoidance, and traffic optimization [30], [31], [32], [33]. In particular, [30] provides an overview of the main solutions for managing platoon merging and splitting. Firoozi *et al.* [31] propose a hierarchical architecture for managing multi-lane platoons composed of three layers: a centralized traffic operation system that collects traffic status information, an offline decision-maker that produces maneuvering instructions based on configurable motion primitives, and an on-board path-follower. Moreover, in [32] a multi-lane platooning algorithm coordination is proposed. The decision-making is performed at both strategic mission level and tactical motion level, resulting in precise trajectories that vehicles have to follow to meet a specific goal. Similarly, the strategic goal we address is to determine the best multi-platoon configuration by taking into account different traffic and road conditions in a highway scenario. Ibanez *et al.* [33] proposes a methodology for synchronously managing roundabouts to maximize the traffic and avoid stops by grouping vehicles into platoons when they reach the roundabouts. Unlike [31] and [32], our approach is to design a centralized controller that imposes a common acceleration pattern to achieve a smooth splitting and merging of platoons by relying on MEC and 5G for timely delivering instructions. Compared to [33], our multi-platoon solution preserves the platoon formation across highway junctions and provides optimal multi-platoon configurations to satisfy road traffic constraints and maximize the benefits of traveling in a platoon.

To resume, PLATO represents an important innovation leap in the research on platooning. It is the first to formulate platoon control in terms of VNFs, with an innovative hierarchical edge computing architecture, and shows how the control of platoons is possible with the tools offered by cellular networks like 5G. Notably, we show that it is possible and convenient to adapt the configuration of multi-platoons not only in terms of composition and distances, but also in terms of harmonized phase transitions when a change of configuration is needed. Indeed, we are the first to show how that translates into an optimization problem, which we solve exactly.

III. PLATO: VEC MULTI-PLATOONING IN PRACTICE

We consider multi-platoon control as a virtualizable network service, similar to what is done in [19] for a single indivisible platoon. Vehicles use the vehicle to network (V2N) communication paradigm instead of vehicle-to-vehicle (V2V). For virtualization, the CACC (Cooperative Adaptive Cruise Control) platooning control algorithm [34] is instantiated on edge computing resources (e.g., ETSI MEC hosts). Hence, our work lays on the ground of VEC [23]. CACC aims to maintain an almost constant inter-vehicular distance δ , based on the position/speed/acceleration updates sent by vehicles to their controllers. The edge-virtualized version of CACC was shown to be robust to round-trip network latency below 200 ms [19]. However, CACC was not meant to coordinate multiple platoons. Hence, we had to design a novel architecture that extends the virtualized CACC architecture to account for the

TABLE I: Notation and used parameter values.

Vehicle and road parameters	
Road length (L)	5 to 50 km
Junction length (S)	500 m
Vehicle length (ℓ)	6 m
Vehicle speed (v)	variable (m/s)
Vehicle acceleration (a)	variable (m/s ²)
Vehicle mass (M)	3 300 kg
Air drag coefficient (D_0)	0.4
Sectional area (A)	4 m ²
Engine transmission efficiency (η)	0.9
Gravitational constant (g)	9.81 m/s ² (standard gravity)
Rolling friction coefficient (r)	0.013 (car tyre on concrete)
Air density (α)	1.225 kg/m ³
Drag coefficient ratio ($\gamma_p(\delta, i)$) at intra-platoon distance δ	$\delta = 10$ m $\delta = 15$ m
	Leader: 0.92 0.96
	Middle: 0.73 0.76
Trailer: 0.74 0.75	
Traffic allow at the junction (β)	≥ 0 vehicles/s/m
Traffic allowed between platoons (p)	≥ 0 vehicles/s/m
Multi-platoon parameters and configurations	
Number of vehicles (N_v)	Stability: 20
	Utility function (steady): 50 Utility function (junction): 20
Number of platoons (N_p)	Stability: 1,2,4,5
	Utility function (steady): 1,2,5,10 Utility function (junction): divisors of N_v
Intra-platoon distance (δ)	Stability: 10 m
	Utility function (steady): 10, 15 m Utility function (junction): 10 m
Inter-platoon distance (Δ)	Stability: 25 m
	Utility function (steady): 25,50,75,100 m Utility function (junction): [25, 200] m
No-platoon vehicle distance (δ_0)	50 m
Reconfiguration displacement (d)	variable (m)
Time (t) and duration of a multi-platooning phase (T)	variable (s)
Update frequency (ϕ_u)	10 Hz
Status and Instruction packet size	200 Byte
Utility and costs (with units)	
Utility (U)	unitless
Road occupation without platooning (L_0) and with platooning (L_p)	m
Road utilization gain (R)	unitless (L_0/L_p)
Relative fuel cost (C_f)	unitless (normalized to non-platooning)
Relative computation cost (C_c)	unitless (normalized to non-platooning)
Relative communication cost (C_t)	unitless (normalized to non-platooning)

presence of multiple coordinated platoons that can merge and split in response to changing road conditions. We assume in our description the 5G MEC framework, although our proposal applies as well to other edge computing frameworks that support the VEC paradigm. Table I presents the notation used in this article, jointly with the values of the parameters used later in the performance evaluation section. Where not needed, the table omits indexes and superscripts in the notation; for instance since a refers to the acceleration, we also use a_i for the acceleration of vehicle i and a^* to indicate the optimal acceleration when we solve an optimization problem.

In PLATO, a set of vehicles is controlled through virtualized CACC instances, each one handling a disjoint set of adjacent vehicles (i.e., a sub-platoon), as shown in Fig. 1. The number and composition of sub-platoons can change over time, in response to road and/or traffic conditions. Each sub-platoon has a platoon controller VNF that is located in a MEC. PLATO also uses a multi-platoon control by instantiating a multi-platoon controller VNF. This VNF runs a modified CACC algorithm whose objective is to stabilize distances between sub-platoons. To do this, the VNF exchanges control messages with sub-platoon controllers. This communication process relies on the cellular backhaul network, which is typically an optical ring,

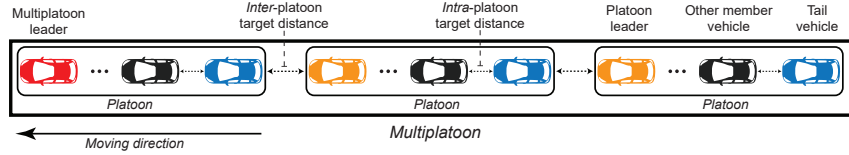


Fig. 2: Multi-platoon composed of three platoons.

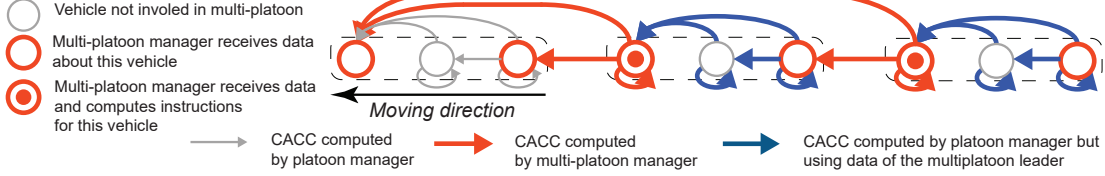


Fig. 3: Dependency graph of PLATO.

and is hence reliable and fast.

Sub-platoon controllers only forward data from leading and tail vehicles to the multi-platoon controller, and the multi-platoon controller relies on a dependency graph (see Fig. 3) to identify which vehicles to instruct, depending on the received updates. In particular, the multi-platoon controller considers a sub-platoon as a single vehicle characterized by a variable length, hence it monitors only leader and tail (*orange circles* in Fig. 3) and instructs only leaders (*orange circles with a dot inside*), since the rest of the sub-platoon will adapt as a consequence of its leader's movement.

PLATO has computation and communication overheads. It computes the CACC control law at each vehicle's update, which means that the overhead per update interval is the number of edges in the dependency graph of PLATO. Hence, with N_v vehicles and N_p platoons, and with a control update frequency ϕ_u Hz, the computational overhead is $(3N_v - N_p - 3)\phi_u$ computations/s. However, the computation overhead *decreases* with the number of chained platoons for a fixed number of vehicles. Therefore, multi-platooning is computationally convenient with respect to single platooning.

As regards communication, PLATO uses both RAN and backhaul network, while traditional ACC and CACC would only use direct communications between vehicles. PLATO generates $N_v\phi_u$ update messages per second from vehicles to controllers, whereas in downlink PLATO uses as many messages per second as the count of edges in the dependency graph (each edge represents a flow of instructions that has to reach a vehicle). So we have $(3N_v - N_p - 3)\phi_u$ downlink messages per second, which again decreases in the number of sub-platoons N_p . In the backhaul, PLATO sends $2N_p\phi_u$ messages/s from sub-platoon VNFs, and $3(N_p - 1)\phi_u$ messages/s from the multi-platoon VNF (to be forwarded to the sub-platoon leaders). The final count reaches $(5N_p - 3)\phi_u$ messages/s, which is the only significant drawback of PLATO, because the number grows linearly with the number of sub-platoons N_p . Note however that an increase in the number of messages in the backhaul is not critical, since the backhaul bandwidth is typically abundant. In addition, the fact that this is compensated by a reduction in the number of messages on the radio interface, where bandwidth is often scarce, results quite positive.

For more implementation details on PLATO, please refer to our conference paper [5].

IV. MULTI-PLATOONING UTILITY

Here we establish a general metric to evaluate the utility of a multi-platoon configuration, which is useful to optimize the composition of (multi-)platoons.

A. PLATO's utility function

The advantages of platooning stem from leveraging the drag effect to reduce air resistance, hence fuel consumption. Platooning is also beneficial because it allows vehicles to travel and maneuver in a more packed way, thus occupying the road space more efficiently.

Of course, platooning and multi-platooning also incur costs and have to satisfy constraints to be considered for optimization. Indeed, management costs include computation and communication costs incurred while controlling a group of vehicles and when the road context changes, calling for a rearrangement of platoons which has to be determined and optimized by a controller, e.g., when approaching a junction with merging traffic. Moreover, distances between vehicles have to be safe and leave a margin for network and control latency, also accounting for delay fluctuations and for the presence of other vehicles not participating in the platooning. The latter consideration indicates that platoons cannot be too long and too packed, or would otherwise impede maneuvering for other vehicles on the same road segment.

The multi-platoon utility function U that we propose takes into account the above mentioned factors with respect to the case without platooning and accounts for global indicators (road utilization efficiency, computing and communication costs) as well as individual benefits (fuel saving). At each point in time t , we use the following expression:

$$U(t) = \log \left(\frac{R(t)}{C_c(t) C_t(t)} \prod_{i=1}^{N_v} U_i(t) \right), \quad (1)$$

where $U_i(i, t)$ is the inverse of the fuel cost of the i -th vehicle, $C_f(i, t)$, i.e.:

$$U_i(t) = 1/C_f(i, t); \quad (2)$$

R is the relative road utilization, normalized to the space occupied by a same number of independently driven vehicles moving at the same speed as the multi-platoon. C_c and C_t express computing and communication costs, normalized to the costs incurred when using individual vehicle's cruise

control mechanisms at the MEC, e.g., using Adaptive Cruise Control (ACC) to maintain inter-vehicle distances [34].

We propose to use the log function of a product of benefits and costs because the resulting optimization would prevent high global utility obtained at the expense of high costs for the network or of unfair individual vehicles' fuel consumption. The derivation of the quantities that compose the utility function is detailed in what follows.

1) *Road utilization efficiency*: $R(t)$ is the ratio between the distance from the first to the last vehicle of the multi-platoon when assisted by ACC rather than being part of the multi-platoon, and the (more packed) distance between those two vehicles when they are under the control of the multi-platoon:

$$R(t) = L_0/L_p(t), \quad (3)$$

where L_0 is the distance computed with N_v vehicles of length ℓ in the case of no platooning, assuming an average spacing of δ_0 larger than the intra-platoon spacing δ :

$$L_0 = N_v \ell + (N_v - 1) \delta_0, \quad (4)$$

and $L_p(t)$ is the multi-platoon length at time t when the N_v vehicles are spread over $N_p(t)$ platoons separated by an inter-platoon gap Δ :

$$L_p(t) = N_v \ell + (N_v - N_p(t)) \delta + (N_p(t) - 1) \Delta. \quad (5)$$

2) *Computation cost*: provided that, by using the CACC/ACC control law, the runtime computation cost of coordination instructions is constant w.r.t. the multi-platoon composition, we model the computation cost as the number of computing jobs per time unit at the multi-platoon controller, which, as mentioned in Section III, is $(3N_v - N_p(t) - 3)\phi_u$. We normalized it to the corresponding rate of ACC computing jobs for the same fleet size, i.e., $2(N_v - 1)\phi_u$:

$$C_c(t) = \frac{3N_v - N_p(t) - 3}{2(N_v - 1)}. \quad (6)$$

We note that the costs for managing the life-cycle of the VNFs are negligible compared to the computational costs for computing the control law, because the two types of events occur at time scales that differ by at least three orders of magnitude. In addition, the memory footprint of PLATO's VNFs is small and deployment requires at most a few seconds [35]. Moreover, the operations of deployment/undeployment are dictated by the road configurations and can therefore be performed in advance without incurring extra overheads.

3) *Transmission cost*: Transmission of control messages and mobility data messages occupy resources in the RAN and the backhaul (see Section III). The associated cost $C_t(t) > 1$ is the number of messages exchanged with multi-platooning normalized to the number experienced with ACC for the same total number of vehicles:

$$C_t(t) = \frac{4(N_v + N_p(t)) - 6}{3N_v - 2}, \quad (7)$$

where ACC only needs $3N_v - 2$ messages per control cycle, i.e., N_v messages with data from the vehicles and 2 control commands to each of the considered vehicles excluding the first one that is driven without network control (there is one update for a vehicle every time the control of the leading and following vehicles are updated).

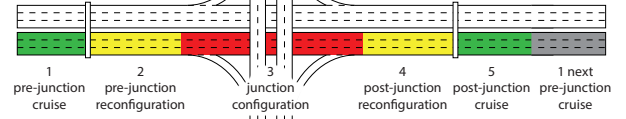


Fig. 4: Junction reconfiguration segments.

4) *Fuel consumption*: $C_f(i, t) \in (0, 1)$ indicates the relative variation of fuel consumption for vehicle i , obtained by comparing the consumption when driving without and with platooning, respectively. This coefficient accounts for the position of the vehicle, since being a leader or a trailer or standing in the middle benefits from air dragging of different magnitudes, as reported in [7].

Fuel consumption is proportional to the total resistance to advancement R , which is a quadratic function of the speed v , accounting for air and rolling resistances [7]:

$$R = \frac{1}{2} D_0 \alpha A v^2 + r M g, \quad (8)$$

the above expression being valid for a solo vehicle at constant speed on a level road. D_0 is the drag coefficient, α the air density, A the cross-section area of the vehicle in the direction of motion, r the rolling resistance, M the vehicle's mass, and g is the gravity acceleration. With platooning, the drag coefficient changes by a factor $\gamma_p(\delta, i) \leq 1$, depending on the position of the vehicle i within the platoon and the inter-vehicle distance δ [7].

Being gas consumption proportional to resistance, the ratio of resistances without and with platooning expresses the fuel reduction due to platooning, i.e., the individual vehicle utility:

$$U_i(t) = \frac{\frac{1}{2} D_0 \alpha A v^2 + r M g}{\frac{1}{2} D_0 \alpha A v^2 \gamma_p(\delta, i) + r M g}. \quad (9)$$

Remark: Thanks to its log-product form, the utility function (1) accounts for all the above described rewards and costs without prioritizing any of them. However, a generalized utility function with prioritization coefficients could be readily formulated by assigning potentially different exponents to each reward and cost term in (1). Such a straightforward generalization could be of use in those cases in which one or more parameters have higher importance for the service provider (e.g., a delivery company or a network operator). However, in what follows, for simplicity, we use the plain utility formula (1).

We also note that the provided definitions of computation cost (6) and transmission cost (7) do not consider complex parameters, such as input data size, computational workload and specific radio channel conditions, because they are independent of the multi-platoon configuration, which is entirely determined by N_v , N_p , δ and Δ .

B. Reconfigurations and accrued utility

1) *Reconfiguration requirements*: We consider the case of a multi-platoon that travels over a road segment where it must allow a traffic with density β vehicles per second per meter to cross the multi-platoon. For simplicity of presentation we consider the case of a highway junction where traffic needs to go through the multi-platoon. Therefore, the multi-platoon might have to reconfigure the values of N_p and Δ so as to

permit enough traffic to pass through the inter-platoon gaps. If the junction area has length S m, the multi-platoon must yield to traffic with intensity βL_p vehicles/s during S/v seconds. Note that the actual time is $(S + L_p)/v$ s, but we have to consider that the obstruction caused while entering and leaving the junction area changes linearly with the portion of the multi-platoon in the junction, so we have to discount an obstruction equivalent to what blocked in $\frac{1}{2} \frac{L_p}{v}$ s at both the beginning and the end of the traversing interval. In total, the traffic obstructed by the passage of the entire multi-platoon is $\beta L_p S/v$ vehicles.

The above quantity needs to be balanced by the permeability provided by the gaps between platoons. Each of the $N_p - 1$ inter-platoon gaps of size Δ m can allow cross traffic at intensity $p \Delta$ vehicles/s during S/v seconds (again, the total time from entrance to departure is $(S + \Delta)/v$, but we have to discount half of the time needed to enter and leave, both of which are equal to Δ/v). The total traffic allowed by $N_p - 1$ interplatoon gaps while crossing the junction area is therefore $(N_p - 1) p \Delta S/v$ vehicles.

The configuration (N_p, Δ) has to be such that the allowed traffic be at least equal to the obstructed traffic. Thus, using the expressions above we obtain the following constraint (which holds for $p > \beta$, otherwise the problem is unfeasible):

$$\Delta \geq ((N_v - N_p)\delta + N_v \ell) \left/ \left(\left(\frac{p}{\beta} - 1 \right) (N_p - 1) \right) \right. \quad (10)$$

2) *Inter-junction segment and multi-platoon phases:* The multi-platoon has to reconfigure itself before entering the junction and re-compact after leaving it. We can split the road length in between two consecutive junctions into *segments*, as shown in Fig. 4. Traveling through these segments, the multi-platoon goes through five phases: (i) at the beginning, the multi-platoon cruises for T_1 time units using configuration $(N_p^{(1)}, \Delta^{(1)})$, (ii) then it starts a transition to state $(N_p^{(2)}, \Delta^{(2)})$, which lasts T_2 time units and ideally completes right before the multi-platoon leader enters the junction; (iii) the multi-platoon travels through the junction until the last vehicle leaves it, for T_3 time units, after which (iv) the multi-platoon starts a transition back to state $(N_p^{(1)}, \Delta^{(1)})$, which takes T_4 time units. Eventually, the multi-platoon keeps moving for T_5 time units until the leader enters the next junction. The total time is $\sum_{i=1}^5 T_i = I/v$, where I is the distance between consecutive highway junctions.

3) *Average accrued utility function:* During the above phases, the multi-platoon utility changes over time. In particular, the utility changes continuously during the transitions between configurations, while the values of the instantaneous utility is constant in the other three phases, although not necessarily equal to the same value.

To account for the variability of instantaneous utility, and considering that the log function penalizes high instantaneous utilities (or, which is the same, small costs), we approximate the computation of the utility function by accounting for average costs and road efficiency over the entire highway length between two consecutive junctions. This is important because a simple average of the utility computed over instantaneous costs and road efficiency would be highly influenced by periods in which one or more vehicles slow down while

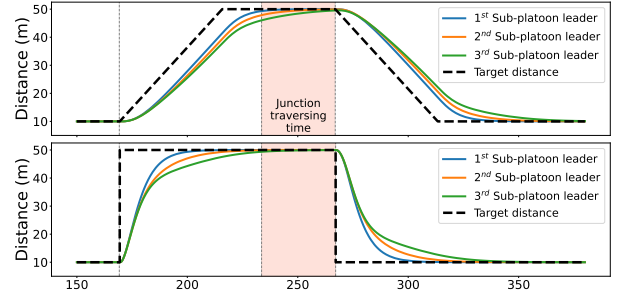


Fig. 5: Example of inter-sub-platoons distance while changing their target distance gradually (top) or suddenly (bottom).

changing configuration, and therefore their fuel cost goes very low, and can ideally reach zero (and therefore yield infinite instantaneous utility). We therefore use the the following expression:

$$U_I = \log \left(\frac{\bar{R}}{C_c C_t \prod_{i=1}^{N_v} C_{fi}} \right). \quad (11)$$

V. DYNAMIC ANALYSIS WITH HARMONIZED MULTI-PLATOON RECONFIGURATION

In this section, we analyze the utility function of a multi-platoon that goes through a change of configuration when it travels along a junction, so as to adapt to the cross flow of vehicular traffic.

A. Timing for reconfiguration

We start from the observation that when sub-platoons close the gap with their predecessors to join platoons, or when a platoon splits, gaps between sub-platoons change smoothly. Fig. 5 shows that there is some inertia while adjusting to a requested change, with a slow approach to the new regime value. The curves have been obtained by simulating a CACC-controlled multi-platoon with a target inter-platoon distance varying gradually (upper part of the figure) or suddenly (lower part). In all cases, the resulting change of distance is different for each sub-platoon and has a long tail. This requires to start a reconfiguration well before a target point (e.g., a junction), early enough to allow for the last sub-platoon to settle. Moreover, a fully CACC-controlled change of distance can require rough accelerations. However, to obviate all these problems while achieving a smooth gap change and the same speed before and after the change is complete, it is enough to impose a smooth acceleration pattern with zero mean, as we propose next.

Let's consider the case of a platoon that splits into two parts and assume that the first sub-platoon keeps a constant speed while the vehicles of the second apply a deceleration with the following pattern during an interval of length T :¹

$$a(t) = -a_x \sin(2\pi t/T), \quad (12)$$

where a_x is a positive constant. This pattern expresses the fact that the second group of vehicles must first decelerate,

¹As discussed in this section, the constant speed assumption for the leading sub-platoon is not important and is used here only to simplify notation.

and then re-gain speed to keep the pace of the leading group after having created a gap. Indeed, it is trivial to show that the speed of the second group starts from the initial value v m/s and sinusoidally oscillates back to v , decelerating first. Then it is also straightforward to derive that the gap between the two sub-platoons opens starting from the value of the inter-vehicle distance δ , according to the following sigmoidal expression:

$$d(t) = \delta + \frac{a_x T}{2\pi} \left(t - \frac{T}{2\pi} \sin \left(\frac{2\pi t}{T} \right) \right), \quad (13)$$

in which we have used $t = 0$ to indicate the time at which the maneuver starts. At time T , the distance is $d(T) = \delta + \frac{a_x T^2}{2\pi}$. So, if we impose that $d(T) = \Delta$, we have an expression that relates the inter-platoon distance Δ after the split and the time T needed to open the gap:

$$T = \sqrt{2\pi(\Delta - \delta)/a_x}. \quad (14)$$

The above expression also applies to the case in which the leading platoon accelerates with any pattern $a_1(t)$ m/s² while the following one uses an acceleration pattern $a_2(t) = a_1(t) + a(t)$. This holds because the relative acceleration of the second platoon would be again $a(t)$, and the initial speed is v for all vehicles. It is therefore possible to adjust the relative position of multiple vehicles at the same time by applying the proper relative acceleration pattern with respect to the predecessor.

It is also possible to operate the reconfiguration of multiple sub-platoons in the same time interval, imposing the same value of T for all sub-platoons drifting apart. For example, assume the multi-platoon splits into three parts and the first sub-platoon moves with acceleration pattern $a_1(t)$. Then the second sub-platoon needs to use a pattern $a_2(t) = a_1(t) + a(t)$ and the third $a_3(t) = a_2(t) + a(t) = a_1(t) + 2a(t)$ to further decelerate *in phase* with the previous sub-platoons, and to complete the maneuver in the same time as the other sub-platoons. Since the third sub-platoon needs to move backward (in relative terms) by a distance $2(\Delta - \delta)$ in a time T , the third sub-platoon has to brake twice as hard as the second. This is actually intuitive and easy to generalize: when a platoon splits into n sub-platoons, the i -th sub-platoon will use acceleration amplitude equal to $(i - 1)a_x$, relative to the acceleration of the leading sub-platoon, which is free to change speed as road conditions demand.

B. Change of configuration

The above reasoning can be further generalized to any change of configuration from $(N_p^{(1)}, \Delta^{(1)})$ to $(N_p^{(2)}, \Delta^{(2)})$. It is enough to compute how much each vehicle j has to move relatively to the multi-platoon leader and impose that the relative displacement d_j be reached in T time units, which requires the acceleration amplitude (relative to the acceleration of the leader) computed as follows:

$$a_j = 2\pi d_j / T^2, \quad \forall j = \{2, \dots, N_v\}. \quad (15)$$

This expression holds also for negative relative displacements, i.e., when the vehicle that follows has to get closer to the multi-platoon leader, and hence use a sinusoidal acceleration pattern with acceleration first (i.e., $-a_j > 0$).

In practice, there is a maximum for the absolute value of a_j , which is imposed by the mechanical characteristics of vehicles. Therefore, by indicating with $a_{\max} > 0$ the mechanical acceleration limit of vehicles, we can select the shortest time needed to complete a transition as the time needed by the vehicle that has to be relatively displaced the most, using the maximum possible acceleration amplitude:

$$T_{\min} = \sqrt{\frac{2\pi}{a_{\max} - a_1^*} \max_j |d_j|}, \quad (16)$$

where the term a_1^* expresses the maximum absolute acceleration of the leader during the reconfiguration interval. Note that any vehicle that needs to be displaced less than $\max_j |d_j|$ will need to use a smaller acceleration amplitude, so imposing T_{\min} for the vehicles that need to maneuver the most, it is always possible to impose that all vehicles execute and conclude their maneuvers in phase.

C. Duration of phases

Having derived (16), we obtained an expression to compute the minimum possible transition times T_2 and T_4 . Considering that Phase 2 and Phase 4 consist of symmetric displacements, and due to the fact that the expression for T_{\min} is insensitive to the sign of the displacement, we can also conclude that

$$T_2 = T_4 = \sqrt{\frac{2\pi}{a_{\max} - a_1^*} \max_j |d_j^{(1) \rightarrow (2)}|},$$

where $d_j^{(1) \rightarrow (2)}$ is the relative displacement of vehicle j when changing from configuration 1 to configuration 2.

The time needed to cross the junction is simply given by

$$T_3 = (S + L_p^{(2)}) / \bar{v},$$

where $L_p^{(2)}$ is the length in meters of the multi-platoon in the configuration used while crossing the junction, and \bar{v} is the average speed of the multi-platoon leader.

Moreover, Phases 1 and 5 are also symmetric by construction and therefore we can compute them as half of what is left after subtracting T_2 , T_3 , and T_4 from the total time I/\bar{v} , i.e.:

$$T_1 = T_5 = \frac{I - S - L_p^{(2)}}{2\bar{v}} - \sqrt{\frac{2\pi}{a_{\max} - a_1^*} \max_j |d_j^{(1) \leftrightarrow (2)}|}.$$

We finally need to compute the average road utility and costs through the 5 phases whose duration we have just derived.

D. Average road utilization

The value of the road utilization R during Phases 1, 3, and 5 is a constant equal to the ratio between the road occupation without platooning and the length of the multi-platoon:

$$R(t) = \begin{cases} L_0/L_p^{(1)} & t \in I_i, i = \{1, 5\}; \\ L_0/L_p^{(2)} & t \in I_3, \end{cases} \quad (17)$$

where I_i indicates the time interval corresponding to Phase i .

During Phases 2 and 4 the function R evolves symmetrically. We can therefore just focus on Phase 2, and in particular

on the distance between the first and the last vehicles of the multi-platoon. The initial distance is $L_p^{(1)} - 2\ell$, while the final distance is $L_p^{(2)} - 2\ell$, therefore the relative displacement of the tail vehicle is $L_p^{(2)} - L_p^{(1)} = (N_p^{(2)} - 1)\Delta^{(2)} - (N_p^{(1)} - 1)\Delta^{(1)}$, and the maneuver lasts $T_{\min} = T_2$. The acceleration amplitude (with sign) of the tail vehicle is

$$a_{N_v} = \frac{(N_p^{(2)} - 1)\Delta^{(2)} - (N_p^{(1)} - 1)\Delta^{(1)}}{\max_j |d_j^{(1) \rightarrow (2)}| / a_{\max}} = \frac{2\pi}{T_2^2} (L_p^{(2)} - L_p^{(1)}). \quad (18)$$

So we can use an expression similar to (13) to describe the evolution of the distance between the leader and the tail vehicles, and add the length of the two vehicles to express the total multi-platoon length. The same holds for Phase 4, with a change of sign in the acceleration and in the displacement distance, and $L_p^{(2)}$ instead of $L_p^{(1)}$ as initial distance. The road utilization during reconfiguration is:

$$R(t) = \begin{cases} \frac{L_0}{L_p^{(1)} + \frac{a_{N_v} T_2}{2\pi} \left((t - T_1) - \frac{T_2}{2\pi} \sin\left(\frac{2\pi(t - T_1)}{T_2}\right) \right)}, & t \in I_2; \\ \frac{L_0}{L_p^{(2)} - \frac{a_{N_v} T_4}{2\pi} \left((t - T_3) - \frac{T_4}{2\pi} \sin\left(\frac{2\pi(t - T_3)}{T_4}\right) \right)}, & t \in I_4. \end{cases} \quad (19)$$

E. Average costs for computation and transmission

Computation and transmission costs are only influenced by the logical multi-platoon configuration, not by the state of a transition. So, as soon as a transition starts in Phase 2 or Phase 4, both computation and transmission costs switch to the values associated with the final configuration after the transition completes. Therefore, using the definition of these two types of costs and averaging over time, we obtain the following expressions:

$$\overline{C_c} = \frac{3(N_v - 1) - \frac{(T_1 + T_4 + T_5)N_p^{(1)} + (T_2 + T_3)N_p^{(2)}}{I/\bar{v}}}{2(N_v - 1)}; \quad (20)$$

$$\overline{C_t} = \frac{4N_v - 6 + 4 \frac{(T_1 + T_4 + T_5)N_p^{(1)} + (T_2 + T_3)N_p^{(2)}}{I/\bar{v}}}{3N_v - 2}. \quad (21)$$

F. Average fuel cost

We compute the average fuel cost of each vehicle considering that the drag coefficient changes linearly over transition times (this entails an approximation) for those vehicles whose role within the multi-platoon changes with a configuration change. For simplicity, we will only consider three possible roles and their associated drag coefficients, i.e., leader, middle and trailer, but the (numerical) computation can be carried out for any number of roles and per-vehicle drag coefficients.

In Phases 1, 3 and 5 the fuel cost, relative to the case without platooning, is nothing but the ratio of resistances in (8). More in general, we have the following per-user instantaneous fuel cost depending on the instantaneous acceleration:

$$C_{fi}(t) = \frac{D_0 \alpha A \left(v + \int_{t_0}^t a_i(\tau) d\tau \right)^2 \gamma_p(d_i(t), i) + 2(rg + a_i(t))M}{D_0 \alpha A v^2 + 2r M g} \quad (22)$$

where $a_i(t)$ is the acceleration of vehicle i . The expression is a simple modification of (8) obtained by accounting for (i) speed modifications due to acceleration, expressed with an integral of the acceleration, (ii) changes over time in the drag coefficient γ_p , expressed through the change of distance from the predecessor vehicle, (iii) the force applied by the engine (hence the energy) needed to change the inertia of the vehicle, which is proportional to the vehicle's mass M , and (iv) the ratio of resistances, that cannot go negative when braking (we assume that the system cannot obtain energy from braking, although some electric vehicles might, at least partially, recharge batteries while braking). Note that v is the speed at time t_0 , i.e., when the reconfiguration begins.

Although we have a sinusoidal expression for the acceleration, hence also for its integral, we do not have an expression for the way the drag coefficient changes over time. This led us to introduce the assumption that the drag coefficient changes linearly with transition time. This assumption is justified by the fact that drag coefficients are numbers close to 1, typically not below 0.7 [9] and are determined by the inter-platoon distance on the one side, and by the role of the vehicle within the platoon on the other side. In any case, decreasing the distance from a predecessor is always beneficial in terms of drag coefficient reduction. For instance, a sub-platoon leader has typically a drag coefficient higher than the rest of the vehicles because it does not have a close-by predecessor, and any trailer has the second highest value, while all other vehicles see similar values close to the minimum within the sub-platoon [9]. This also explains why we can simplify the numerical analysis and consider only three types of roles. Moreover, although it is possible to derive an analytical expression for the average fuel consumption under our approximations, it would not be a handy function because of its complexity, so we omit its expression here.

VI. A TRAFFIC-YIELDING RECONFIGURATION PROBLEM

We consider the optimization of the multi-platoon reconfiguration dynamics, which becomes important when the platoon crosses a junction and needs to yield to cross traffic.

A. Problem formulation

A multi-platoon with N_v vehicles travels through a sequence of homogeneous road segments of length I . For simplicity we assume that the leader's speed v is constant (i.e., $a_1^* = 0$), although extending the formulation to a more general case is straightforward. Each segment includes a junction of length S . At the junction the platoon must not obstruct the flow of traffic $\beta > 0$ vehicles per second per meter flowing through the junction. Aside from the junction, the multi-platoon travels unconstrained over the rest of the road (i.e., β goes to zero outside the junction). We want to find the multi-platoon configuration pair $\{(N_p^{(1)}, \Delta^{(1)}), (N_p^{(2)}, \Delta^{(2)})\}$ that maximizes the average multi-platoon utility (11) where $(N_p^{(1)}, \Delta^{(1)})$ is the configuration used in Phases 1 and 5 while $(N_p^{(2)}, \Delta^{(2)})$ is used in Phase 3 in each road segment, assuming that the maneuvers associated to configuration changes start and complete at the same time instant for all vehicles.

Our Multi-platoon Reconfiguration Problem (MRP) can then be formulated as follows:

$$\left\{ \begin{array}{l} \text{maximize} \quad \log \left(\frac{\bar{R}}{\bar{C}_c \bar{C}_t \prod_{i=1}^{N_v} \bar{C}_{fi}} \right) \\ \text{s.t.:} \quad N_p^{(1)}, N_p^{(2)} \text{ are whole divisors of } N_v; \\ \quad \Delta^{(i)} \in [\Delta_{\min}, \Delta_{\max}], \quad \text{for } i \in \{1, 2\}; \\ \quad 0 \leq a^* \leq a_{\max}; \\ \quad \Delta^{(2)} \geq \frac{(N_v - N_p^{(2)})\delta + N_v \ell}{(p/\beta - 1)(N_p^{(2)} - 1)}; \\ \quad I - S - L_p^{(2)} \geq 2v\sqrt{2\pi d^*/a^*}, \end{array} \right. \quad (23)$$

where there are five optimization variables ($N_p^{(1)}$, $N_p^{(2)}$, $\Delta^{(1)}$, $\Delta^{(2)}$ and a^*); the first three sets of constraints define the range in which the optimization variables must be searched, the fourth constraint is the necessary condition on the intra-platoon spacing that guarantees the traffic to be yielded, whereas the last constraint is necessary to guarantee that the total transition time $T = \sqrt{2\pi d^*/a^*}$ is enough to go through the configuration changes. Indeed, in the above formulation we have assumed that configuration transitions occur with a maximum acceleration a^* , the maximum relative displacement of a vehicle is d^* and the vehicle acceleration pattern $a_i(t)$ in Phases 2 or 4 is a sinusoidal function, i.e.:

$$\left\{ \begin{array}{l} T = \sqrt{2\pi d^*/a^*}; \\ d^* = \max_j |d_j^{(1) \rightarrow (2)}|; \\ a_j(t) = \mp \frac{d_j^{(1) \rightarrow (2)}}{d^*} a^* \sin\left(\frac{2\pi t}{T}\right), \quad \forall i \in \{1, 2, \dots, N_v\}; \end{array} \right.$$

in the expression of $a_j(t)$, the sign “−” applies to Phase 2, while the sign “+” applies to Phase 4, and $t \in [0, T]$. Notice that T and d^* are not decision variables because the latter is fully determined by the values of $N_p^{(1)}$, $N_p^{(2)}$, $\Delta^{(1)}$ and $\Delta^{(2)}$.

B. Solving the problem

The MRP problem is a non-linear mixed integer program with integrals to be solved in the objective functions, some of which involve a max function. This kind of problem can be hardly programmed with standard tools. For instance, Matlab and many commercial solvers do not accept the max function and the integrals in the objective function, nor max, min and absolute values in the constraints. To implement the MRP problem in Matlab with the `fmincon` function, d^* cannot be defined as the maximum absolute displacement of a vehicle upon a change of configuration, and instead it can be set as an auxiliary optimization variable with extra constraints. I.e., it has to be an optimization variable larger than or equal to any displacement $d_j^{(1) \rightarrow (2)}$ and its opposite value $-d_j^{(1) \rightarrow (2)}$, and the product $\prod_{j=1}^{N_v} (d^* - d_j^{(1) \rightarrow (2)}) (d^* + d_j^{(1) \rightarrow (2)}) = \prod_{j=1}^{N_v} ((d^*)^2 - (d_j^{(1) \rightarrow (2)})^2)$ must be constrained to zero to ensure that d^* takes exactly one of the (absolute) values of the displacements. Moreover, the integrals needed to compute the fuel costs in the objective function need to be replaced with finite sums, within which the non-negativity of (22) is expressed through $[X(t)]^+ = (X(t) + \sqrt{X(t)})/2$, where

$X(t)$ is the resistance ratio that appears in (22). The result is that, even for fixed values of $N_p^{(1)}$ and $N_p^{(2)}$ and the integrals approximated with a sum of as few as 6 points, solving the problem with $N_v = 20$ requires days and tens of GB of RAM, and does not necessarily converge to the optimum due to the presence of local minima.

For a fixed value of β and specific road topology—specified in terms of the length of the junction and the distance between junctions—and once the characteristics of the homogeneous vehicles used in the multi-platoon are given, the utility function to optimize still depends on 5 parameters: the number of sub-platoons outside and inside the junction ($N_p^{(1)}$ and $N_p^{(2)}$), the spacing between sub-platoons, which can be different inside and outside the junction ($\Delta^{(1)}$ and $\Delta^{(2)}$), and the maximum feasible peak acceleration (a^*). It is not easy to spot a monotonicity or a convexity in the relation between the utility and any of the 5 mentioned parameters. Hence, the search space where to find the optimal solution needs to be explored cautiously, which can take long because it is quite vast even if considering a discretized version of the continuous decision variables $\Delta^{(1)}$, $\Delta^{(2)}$ and a^* .

However, the complexity of the search can be quite low, on average. To show why, let’s analyze the impact of the continuous decision variables.

First of all, the optimal value of $\Delta^{(2)}$ must be the minimum value that satisfies the fourth constraint of our problem for a given value of $N_p^{(2)}$. This is due to the fact that increasing the inter-platoon distance without changing the number of sub-platoons can only require more maneuvering and costs.

Furthermore, once all other parameters are fixed, $\Delta^{(1)}$ determines the maximum relative distance d_x to cover during transitions. Such a distance is the maximum absolute value in a set of position differences, each of which can be positive or negative. Moreover, position distances must be monotonic (linear) functions of $\Delta^{(1)}$. Therefore, d_x must be a continuous function of $\Delta^{(1)}$ which initially decreases and at some point starts to increase. This means that the impact of $\Delta^{(1)}$ on the cost of maneuvers has a single minimum. We can therefore stop searching for a better $\Delta^{(1)}$ once we observe that the cost reaches a local minimum.

Finally, once all other parameters are fixed, the acceleration must be larger than the value that allows to complete the required maneuvers, which is the smallest value of a^* that satisfies the last constraint in the formulation above. Starting from that value, increasing a^* is always feasible as it will introduce faster maneuvers, requiring monotonically less time and yielding monotonically increasing gains, except for fuel costs. In fact, higher values of a^* incur higher fuel costs that increase monotonically and faster than linearly. Therefore, the curve of utility vs a^* is the result of a tradeoff between a function that decreases and one that increases, which must result in a curve with a single maximum.

In conclusion, although the problem is not solvable optimally in polynomial time because it requires the evaluation of all combinations of $N_p^{(1)}$ and $N_p^{(2)}$, the complexity is tractable because the continuous decision variables can be optimized in polynomial time. In particular, the optimal value of $\Delta^{(2)}$ does

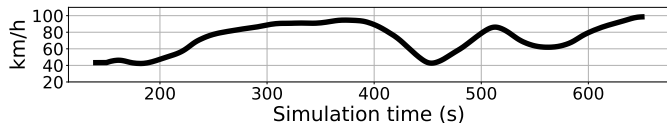


Fig. 6: Multi-platoon leader speed profile.

not need to be searched at all, while $\Delta^{(1)}$ and a^* can be found, for each pair $(N_p^{(1)}, N_p^{(2)})$, by exploring increasingly higher values until there is no observed utility gain by increasing any of the two parameters. In turn, this means that the search can be done in linear time by using discrete steps. The precision of the search can be improved by iteratively zooming into the search area around the optimal point found at each iteration, which requires a logarithmic number of steps. This method allowed us to compute all the analytical numerical results presented in this paper with an unnecessarily high resolution of 10^{-6} on $\Delta^{(1)}$ and a^* in a few tens of seconds using a personal computer (just a couple of seconds or less for each value of β).

VII. SIMULATION-BASED PERFORMANCE EVALUATION

To evaluate PLATO-controlled multi-platoons, we simulate a highway scenario with a fleet of light duty commercial vehicles. In this work we implement all PLATO inter-VNFs communications and the management of the multi-platoon dependency graph (see Fig. 1 and Fig. 3) using OMNeT++ on top of the SUMO simulator [6]. In particular, we rely on the Veins [36] and Simu5G [37] frameworks to model vehicles and 5G mobile network, respectively. Compared to the work presented in [5] this new simulator greatly increases the model realism, especially for what concerns the 5G RAN.

In our simulations, the multi-platoon leader proceeds according to the speed profile of Fig. 6.² This speed profile combines a wide range of speeds, from 45 to 100 km/h together with acceleration/deceleration maneuvers. Realistic dynamics were obtained by accounting for mechanical vehicle constraints and aerodynamic factors, and basic principles like

²From the Floating Car Dataset of TIM Big Data Challenge 2015

TABLE II: Simulation parameters.

General parameters	
Simulated road	50 km (straight 3-lane highway)
Simulation duration	650 s (140 s of warm-up time)
Network and RAN configuration	
RTT UPF to platoon mng. via link A (RTT_{U-PM})	Stability: 10, 20, 50, 100, 200 ms Utility function: 10 ms
RTT platoon mng. to multi-platoon mng. link C (RTT_{PM-MM})	Stability: 10, 20, 40, 60, 120 ms Utility function: 10 ms
Inter-Base station distance	1500 m
Base station bandwidth (numerology μ)	5 MHz ($\mu = 0$) 25 RBs per TTI (1 ms)
UE Tx power (antenna gain)	26 dBm (+0dBi)
Base station Tx power (antenna gain)	46 dBm (+18dBi)
Pathloss model	Free Space with $\alpha = 2.5$
Base station scheduler	Proportional Fair
Background network traffic	
Number of vehicles (N_v^{bg})	20, 30, 40 vehicles
Application type	Symmetric CBR
Packet size (UL/DL)	3000 Byte
Packet frequency (UL/DL)	25 pkt/s

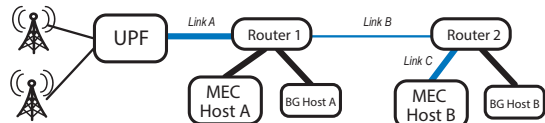


Fig. 7: Network topology.

inertia, like in [38]. Table I includes the main parameters of the specific vehicle we have simulated.³

We simulate a simple access and backhaul network as depicted in Fig. 7. We deploy all platoon managers on *MEC Host A*, while the multi-platoon manager is on *MEC Host B*. Links A and C are associated with configurable propagation delays to simulate different MEC host deployments. Moreover, we simulate mild congestion conditions by generating bursty background traffic between *BG Host A* and *BG Host B* traversing link B⁴. The base stations are deployed along the highway at 1.5 km distance from one another, and each BS offers a 5 MHz bandwidth, thus simulating the case of a dedicated RAN slice. We note that we have chosen 5MHz for simulation purposes only and this value does not represent a system requirement. In particular, this value allows us to generate background traffic that saturates the BS avoiding simulation time overhead. In Table II, we report the main parameters used for the RAN configuration. These values allow us to simulate multiple deployment settings. The lowest values of uplink/downlink delay correspond to the deployment of platoon manager VNFs at the edge, whereas the highest delays indicate unrealistically high edge distances, which could be interpreted as potential cloud deployments. As for the VNF-to-VNF communication latency, short values mean that both VNFs are deployed in proximity of each other, e.g., on the same MEC host, while high values indicate that VNFs are instantiated on far MEC hosts or that the multi-platoon manager runs in the cloud.

We run two sets of simulations, whose main parameters are reported in Tables I and II. The first set, which is marked with label *stability* in the tables, is for evaluating the string stability⁵ obtained with PLATO, and measuring the intra-vehicle distance error by varying propagation delays on link A and C (see Fig. 7) and multi-platoon compositions. We simulate 20 vehicles, considering only multi-platoon configurations with the same number of vehicles per platoon. The second set of simulations (labeled as *utility function*) is meant to analyze the values of our proposed utility function. For this case, we consider 50 vehicles, and we vary, in addition to the multi-platoon composition, also the values of δ and Δ . For each simulation scenario, we perform at least 20 runs. In the figures presenting simulation results, we report averages of the values observed in each run for the considered metrics. In the experiments, we have observed small 95%-confidence intervals (below 1.5% of the values reported in the figures), which we omit in the plots to avoid cluttering effects.

³We used the specifications of Fiat Ducato from <https://www.cars-data.com/>

⁴For simulation efficiency we limit the bandwidth of Link B rather than generate large amounts of traffic.

⁵It is customary to evaluate a platoon's stability based on string stability, which is achieved when the error with respect to the target inter-vehicular distance decreases with the vehicle position index, counting from the leader.

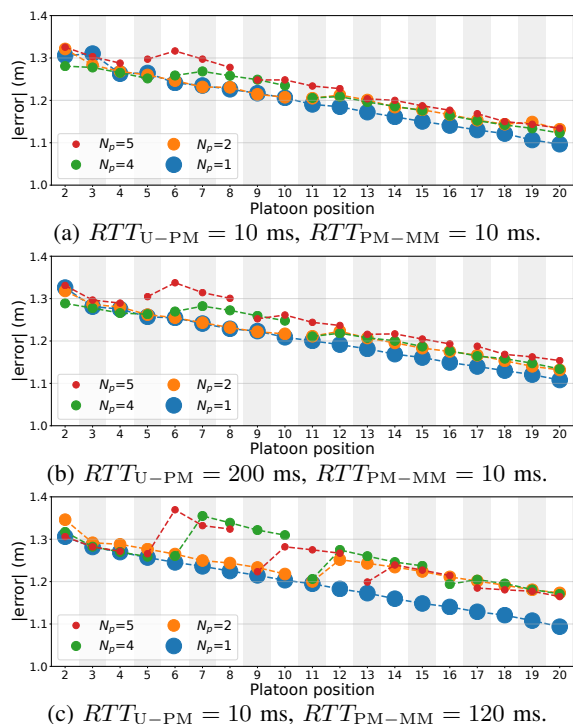


Fig. 8: CACC error in PLATO under different latency conditions. Latency increases in the RTT_{U-PM} (UL plus DL) are much less critical than in the backhaul (VNF to VNF).

The value of δ_0 reported in Table I is 50 m, which is in line with the 2-second safety rule at the average speed of about 90 km/h, which is not far from the average speed in our simulations.⁶ For platooning, we use $\delta \in \{10, 15\}$ m, which means that our platooning simulations adopt a safety distance 3 to 5 times lower than what can be achieved by relying on human control only.

In line with the experiments presented in [9], the drag coefficient ratio $\gamma_p(\delta, i)$ was computed for three vehicle positions in a platoon, *leader* ($i = 1$), *middle* ($1 < i < N_v$), and *trailer* ($i = N_v$), as reported in Table I. The value of the middle vehicle is then applied to all internal platoon positions.

A. Multi-platoon stability

To evaluate the string stability of different multi-platoon configurations we plot the 95th percentile of the absolute distance error experienced by each vehicle, with respect to the target distance δ , versus the position of the vehicle in the string. We simulated several platoon configurations, obtaining similar qualitative results. Here we just report an example with 20 vehicles, and the parameters shown in Table I and Table II under the label *Stability*.

Fig. 8a and Fig. 8b depict two extreme scenarios for the RTT_{U-PM} , i.e., 10 and 200 ms, respectively, when the backhaul RTT_{PM-MM} is low (10 ms, on average). We can observe a generally stable behavior with error values monotonically decreasing, except for a few cases, and by small amounts. We also see that the number of platoons has barely an impact. The

⁶The 2-second rule, typically taught at driving schools, suggests to take as safety margin the distance covered by the vehicle in 2 seconds.

negligible difference between the two figures is due to the high inertia of the simulated vehicles, whose actuation delays are larger than those of ideal vehicles: so, if the communication delays from/to the platoon manager and from/to the multi-platoon manager are of the same order of magnitude, the multi-platoon remains practically stable (in line with our previous results on isolated platoons [19]).

Fig. 8c tells a different story. Indeed, when the backhaul RTT becomes comparable to the one seen with a VNF running in the cloud (120 ms), even if the RTT_{U-PM} is low (15 ms), the multi-platoon visibly exhibits lower stability. In particular, the second vehicle in each platoon, except for the leading platoon, experiences a larger error than its platoon leader. This is clearly visible for $N_p = 4$ (for vehicles 7 and 12) and $N_p = 5$ (for vehicles 6 and 10). The reason for this string instability lays in the different levels of freshness of the data used for CACC. The additional 120 ms latency of the backhaul is suffered only by data flowing through the multi-platoon manager, and is longer than the update cycle length $1/\phi_u = 100$ ms. This causes incoherence in the virtual representation of the multi-platoon state built by each platoon manager upon receiving stale data from the multi-platoon manager. This result suggests that combining cloud and edge deployments, or using MEC hosts with highly different RTT, is not a wise solution to manage multi-platoons.

Nevertheless, it is worth noting that in all cases distance errors are below 1.5 m, which means that PLATO makes a good job, although it is not strictly string stable. Indeed, all the considered configurations satisfy the safety constraint with $\epsilon \leq 0.1$ when $\delta = 15$ m.

B. RAN Congestion

To evaluate the impact of temporary RAN congestion on the platoon stability, we simulate N_v^{bg} additional vehicles that do not participate in the multi-platoon, but generate background traffic that uses the same radio resources as the multi-platoon vehicles. At the bottom of Table II we report the main parameters regarding background traffic. We simulate a symmetric constant bit rate application sending and receiving data at 0.6 Mb/s (25 packets of 3000 bytes per second, i.e., one every 40 ms). In Fig. 9 we report the results considering two different speed profiles of the background vehicles. On the left, all background vehicles travel at constant speed on another lane and come up beside the multi-platoon for almost 150 s (between 400s and 550s). On the right, all vehicles follow the speed pattern of the multi-platoon leader and travel alongside the platoon for the entire duration of the simulation. In the figure, we report three metrics. On top, the distance error w.r.t. the target distance of the first follower behind the multi-platoon leader. In the middle, the uplink delay, i.e. the time for sending on-board sensor data to the MEC platoon controller. On the bottom is the percentage of the RAN slice physical resource blocks (RBs) used, considering the base station that is serving the first follower behind the multi-platoon leader.

The results show that PLATO is robust against temporary saturation of RAN resources due to background traffic, tolerating a delay above 100 ms for a short amount of time,

as we can observe on the left side of Fig. 9. In fact, we see that this kind of RAN congestion does not significantly impact platoon stability (errors are at most of 1.5 m) and uplink delays rarely exceed 100 ms. On the contrary, the plots on the right side show that a prolonged RAN overload may affect platoon behavior. In particular, in the case of $N_v^{bg} = 40$ additional vehicles generating RAN background traffic, the combination of high traffic load and poor signal quality causes a significant increase in the uplink delay, which reaches peaks around 500 ms⁷. This causes strong inconsistency in the virtual representation of the platoon, leading to untimely instructions. Interestingly, in spite of this, the distance error observed in the top right plot of Fig. 9 remains at an acceptable level. This is due to the large difference between the time constants of vehicle dynamics and data transfer.

The simulations have shown that the sharing of RAN resources might compromise platoon stability. In particular, when the network delays increase and approximate the vehicle actuation lag, the negative contribution of the network delays becomes evident. As shown on the right side of Fig. 9, the curve reporting the distance error in the case of $N_v^{bg} = 40$ deviates from the others as a result of a non-negligible contribution of the network delays. Even though the observed deviation is contained, in those cases the RAN is unable to guarantee a suitable level of QoS and longer periods of RAN saturation could lead to more critical situations. On the other hand, the reservation of the required bandwidth for platooning may lead to a significant waste of resources given the low data-rate requirements of PLATO. This indicates the need to foster a traffic differentiation policy suitable to efficiently mix different network traffic, while prioritizing safety-critical services.

C. Evaluation of the utility function in steady state

For starters, we look at the utility function components described in Section IV and we report their log value in Fig. 10. Positive values are gains (road efficiency increase and fuel consumption reduction, summed over all vehicles), while negative values are costs (due to computation and transmission). The figure reports averages over time for each component and for the resulting utility, with different combinations of N_p , δ , and Δ .

Fig. 10 clearly shows that fuel saving benefits from small values of δ , and is practically insensitive to other parameters. In contrast, the road utilization component is sensitive to the multi-platoon configuration, and its contribution is highly reduced when the multi-platoon is fragmented in a large number of platoons (see, e.g., $N_p = 10$). In all cases both road utilization and fuel consumption are largely better than without platooning.

The computation cost slightly decreases for an increasing number of platoons, while the transmission cost increases. In particular, the transmission cost with 10 platoons is twice as much as for a single large platoon, as a result of the backhaul communication overhead. These results are in line with the

PLATO management overheads discussed in Section III and showcase the scalability properties of PLATO.

As clear from Fig. 10, the benefits of platooning are much higher than its costs, no matter which configuration is used. Indeed, the utility is always positive, and the smaller δ the better, as also shown in the heatmaps of Fig. 11 for a wide range of configurations. This suggests that the multi-platoon should be compacted as much as possible, which is mostly due to the relevance of the road efficiency component in the utility. Thus, in the presence of extra-platoon vehicular traffic, the rule of thumb is to keep the values of N_p and δ as small as possible, while meeting the constraints on the volume of traffic that has to be allowed to cross the multi-platoon at a junction, notwithstanding the presence of the multi-platoon.

The last aspect we analyze is the change of contribution to the utility function of the fuel component over time. Fig. 12 shows the fuel cost, since nothing else changes by only varying the speed of the multi-platoon leader. In the figure, we only refer to the simulation time interval [500 s, 750 s] of Fig. 6, where speed changes are more pronounced. We can observe that fuel saving goes with speed variations and the differences among configurations are more pronounced at high speed due to the larger contribution of the drag reduction effect.

The evaluation of the utility function in steady state shows that with no interference of extra-platoon vehicles, the best platoon configuration is to organize vehicles in a single platoon with the lowest possible value of δ . This is because the single platoon configuration maximizes the benefits of platooning, thanks to large fuel savings and minimum road occupation.

D. Junction crossing

Finally, we consider the case of a multi-platoon that travels along a highway junction of length $S = 500$ m within a road segment of $I = 12.5$ km, assuming that the junction lays at the center of the road segment. For a multi-platoon of $N_v = 20$ vehicles with the characteristics of Table I, we solve the MPR optimization problem formulated in (23), for various values of the ratio β/p , with $p = 0.05$ vehicles/s/m, $v = 100$ km/h (i.e., a vehicle can cross the platoon in 1 second using a space equal to 20 m) and a realistic delivery van's maximum peak acceleration of 0.4 m/s². The intra-platoon distance is set to 10 m, while inter-platoon distances, which need to be optimized, can take any value in between 25 and 200 m.

Fig. 13 (in the right y-axis) shows the maximum utility of the multi-platoon as the cross traffic in the junction grows. Being the utility logarithmic and computed with quantities normalized to the case without platooning, any value greater than 0 represents a gain introduced by the multi-platoon. The curve shows that the utility decreases with the cross traffic intensity at the junction β , which is expected since β causes changes in the multi-platoon configuration (hence increases fuel costs) and imposes using progressively larger inter-platoon gaps, which reduce the efficiency in the use of the road. Moreover, increasing the cross traffic results in using more sub-platoons while the multi-platoon travels along the junction, as shown in the same figure (left y-axis). The two curves marked as "In" and "Out" report the optimal

⁷With $N_v^{bg} = 40$, the background traffic uses $\sim 100\%$ of RAN UL resources with the best channel conditions and $\sim 300\%$ in case of low channel quality.

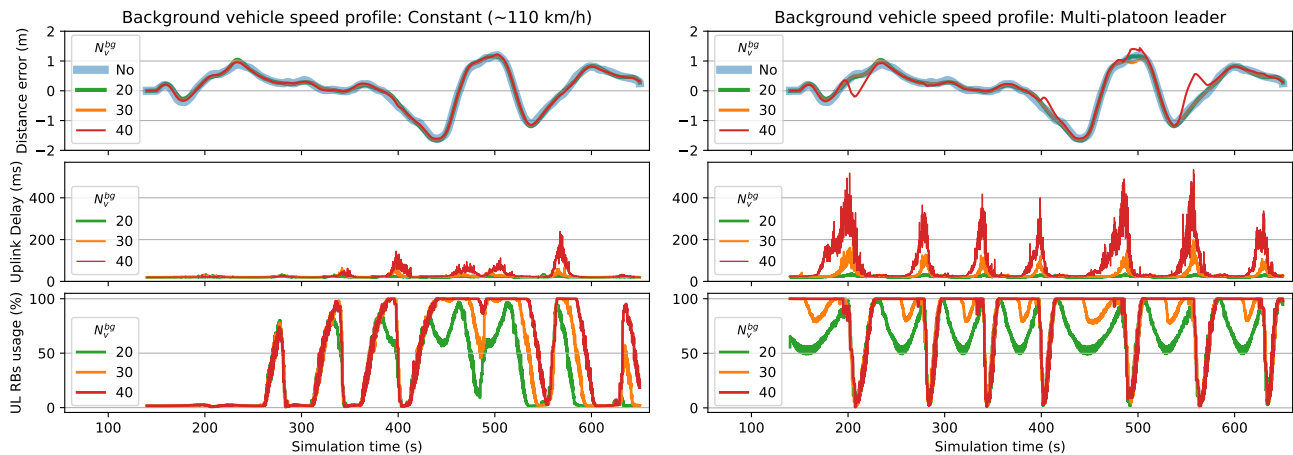


Fig. 9: Loaded RAN scenarios. $RTT_{U-PM} = 10$ ms, $RTT_{PM-MM} = 10$ ms.

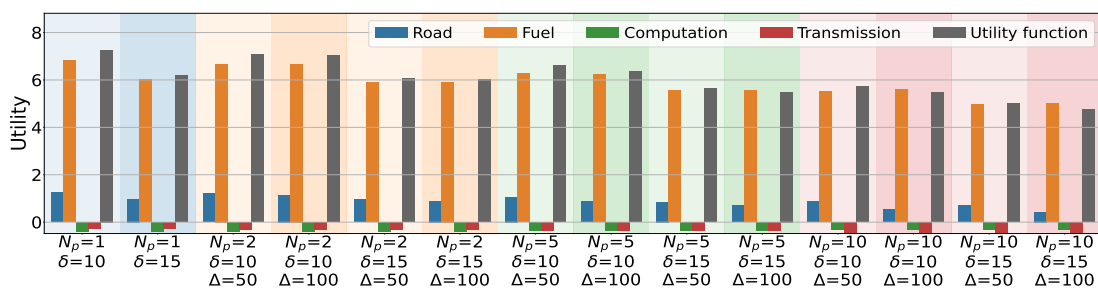
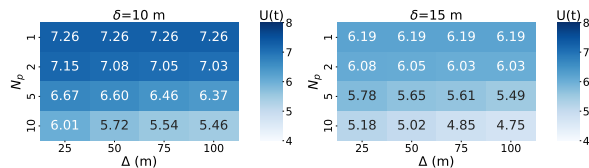


Fig. 10: Utility functions components contribution (log values).



(a) $\delta = 10$ m

(b) $\delta = 15$ m.

Fig. 11: Average utility function over t for different configurations of distances, δ and Δ , and number of platoons N_p .

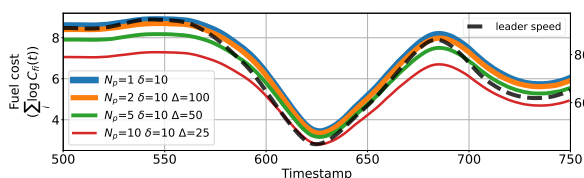


Fig. 12: Fuel component over time [500s, 750s].

number of sub-platoons inside and outside the junction area, respectively. Outside the junction, it is convenient to use a single platoon, except for extremely high loads, because in those cases maneuvering from a single platoon to many small platoons would take too long in relation to the inter-junction distance, whereas inside the junction the multi-platoon must split into a progressively larger number of sub-platoons.

The inter-platoon distance is also affected by the traffic intensity β as shown in Fig. 14, where we see that the inter-platoon gap while traveling along the junction is not a monotonic function of β . Outside the junction, and only for high traffic values, when more than one sub-platoon is needed, the value of $\Delta^{(1)}$ can be higher than while traveling the junction if this choice minimizes the cost to reconfigure

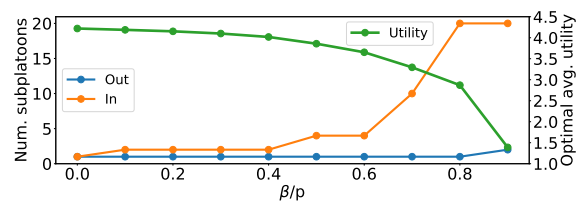


Fig. 13: Optimal number of sub-platoons (left axis) and best utility (right axis).

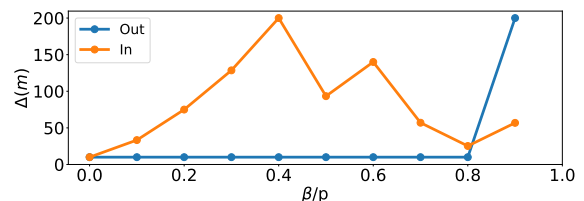


Fig. 14: Inter-platoon gaps in the optimal configuration.

the multi-platoon to be ready for the traffic in the junction. We can also see that, at the points that correspond to increases in the number of sub-platoons, the value of the gap $\Delta^{(2)}$ (inside the junction) can diminish. This behavior is expected, because an increase in the number of sub-platoons corresponds to more inter-platoon gaps through which the cross traffic can flow, and therefore it is possible that smaller yet more numerous gaps become needed at that point.

Fig. 15 shows the corresponding total length of the multi-platoon. It is interesting to note that the length inside the junction is always greater than outside. Besides, the length is not a monotonic increasing function of β either. To explain this phenomenon, we have to observe that the length of the multi-platoon can become greater than the junction length.

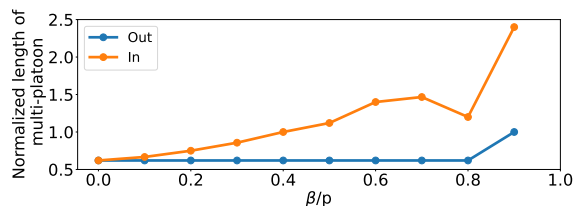


Fig. 15: Optimal multi-platoon length, normalized to S .

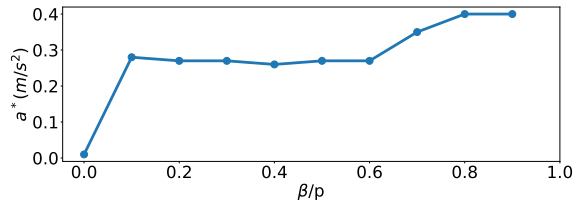


Fig. 16: Optimal peak acceleration.

This means that only a part of the multi-platoon interferes with the cross traffic in the junction, so that using frequent small gaps can be more effective than using a few large gaps, which is what we see for high values of β .

The last decision variable of the MPR optimization problem is the maximum peak acceleration used during configuration changes, a^* . Fig. 16 shows that a^* is not necessarily the maximum available acceleration. Indeed, solving MPR finds a tradeoff between fast reconfiguration (which would require high acceleration) and fuel costs (which suffer a superlinear increase when the acceleration increases).

The utility of multi-platooning depends on several parameters, among which are the number of vehicles involved in the platooning, N_v , and the characteristics of the junctions. Fig. 17 illustrates such dependency for three values of the junction traffic and for three values of the inter-junction distance.

In general, blue, red and orange bars in the figure shows that the utility grows fast with the number of vehicles, with better results obtained at lower junction traffic intensity. Comparing the three sub-figures, we can see how longer inter-junction distances allow for better utilities, especially when the number of vehicles involved becomes high. However, with short inter-junction distances, e.g., $I = 5$ km and high junction traffic (cf. Fig. 17a) long platoons are not possible at all, as their dynamic management could become counter-productive or even unfeasible. Counter-productive cases are the ones in which the utility turns negative because platooning becomes less efficient than uncontrolled road traffic, like for instance with $I = 5$ km and $\beta/p = 0.8$ with 128 vehicles. Unfeasible cases are the ones in which there is no solution to the optimization problem that can be implemented with the given number of vehicles and within the available road distance, at the leader's speed, without violating any constraint, e.g., with $I = 5$ km and 256 vehicles, at any of the considered junction traffic intensities. Results for $I = 12.5$ km and $I = 25$ km, shown in Figs. 17b and 17c, respectively, tell that the utility of very long multi-platoons are limited by the inter-junction distance when the junction traffic intensity becomes high. This effect is caused by the fact that long multi-platoons could be optimized by opting for long reconfiguration times during which the multi-platoon be split in small parts and then rebuilt, which unfortunately might not be feasible given the space in between junctions.

In contrast, short multi-platoons achieve practically the same utility under any inter-junction separation.

On top of each colored bar of Fig. 17 we added a smaller and narrower grey bar that indicates the utility achieved by imposing that each vehicle travels on its own in phase 3, within the junction. This result indicates the best achievable utility when platoons are not allowed in presence of other traffic, as suggested in [33], which is the only paper in the literature that considers the issue raised on platooning by cross traffic. The figure shows that PLATO achieves much higher utilities, and the difference grows fast with the size of the multi-platoon.

Finally, we zoom into the performance of individual vehicles, which can only be measured through their fuel cost (normalized to the case without platooning). Figs. 18 to 19 report the costs for two solutions of the MPR problem in which we have fixed the values of β/p to 50% and 70%, respectively. The figures also report simulation results, which show minor deviations from the model. This validates our modeling assumptions, and especially our approximations about costs and utility while the multi-platoon reconfigures.

Fig. 18, with $\beta/p = 50\%$ in the junction, is a case in which the multi-platoon uses one compact platoon outside the junction and splits into four sub-platoons to travel the junction, with a gap of 93.3 m and a peak acceleration of 0.27 m/s², as reported in the topmost part of the figure. Each dot in the figure represents the cost of one vehicle, while the straight line is the average cost in the multi-platoon. The first observation is that all costs are below one, which means that all vehicles have an incentive, and a gain, to participate in the multi-platoon. Second, the leader of the multi-platoon (vehicle #1) has the least benefit. Vehicles #6, #11 and #16, which are the leaders of the second to fourth sub-platoons while traveling the junction, are the only other vehicles with benefit below average. This occurs because leaders have a much higher drag coefficient ratio, hence higher fuel consumption, with respect to any other vehicle in a platoon. Similarly, since the trailer vehicle gains a little less than a middle vehicle, vehicles #5, #10, #15 and #20 gain less than the other vehicles which are never leading a sub-platoon. We also note that vehicles that form the second sub-platoon while traveling the junction, gain slightly less than the vehicles of the first sub-platoon in the corresponding positions. The gain reduces further for the third and fourth sub-platoons. This is because the multi-platoon splits and re-compacts before and after the junction thanks to the fact that the vehicles in those sub-platoons maneuver, while the ones in the first sub-platoon do not have to adjust. With high cross traffic (i.e., $\beta/p = 70\%$), Fig. 19 shows that the costs increase for all vehicles and that vehicles in the tail of the multi-platoon can experience low gain. In this case, note that the optimal configuration consists in keeping 2 sub-platoons spaced 200 m also outside the junction, and then split into single-vehicle platoons with a spacing of 56.8 m while crossing the junction (so, in practice, there is neither platooning nor drag effect). This requires long maneuvers at high acceleration equal to the maximum allowed in the scenario, i.e., 0.4 m/s²) and high costs, especially for vehicles in the tail of the multi-platoon.

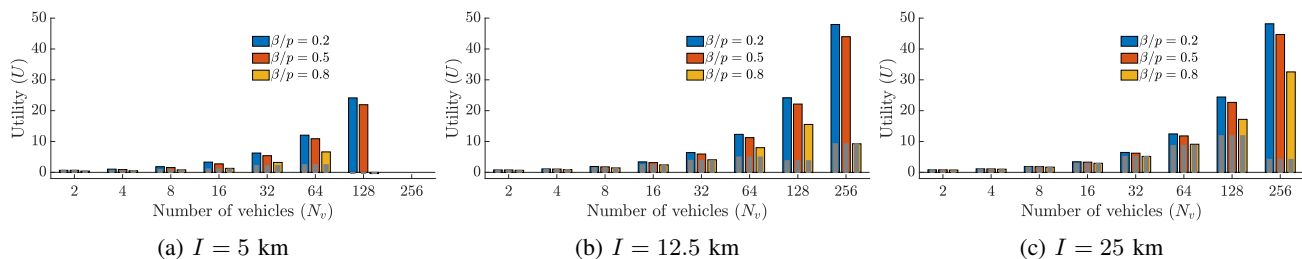


Fig. 17: Platooning utility observed in the optimal configuration as a function of the multi-platoon size for three different values of the inter-junction distance and for selected values of the junction traffic intensity. Grey bars indicate the utility achievable if platooning is not allowed within the junction.

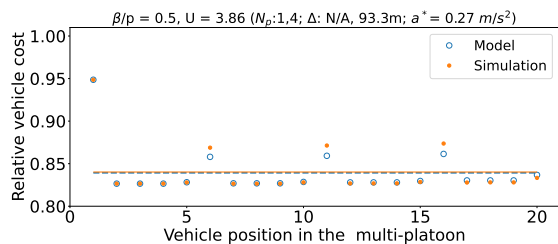


Fig. 18: Relative fuel cost of vehicles in the multi-platoon for $\beta/p = 0.5$ ($RTT_{U-PM} = 10\text{ms}$, $RTT_{PM-MM} = 10\text{ms}$).

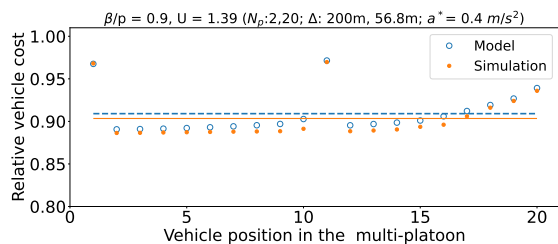


Fig. 19: Relative fuel cost of vehicles in the multi-platoon for $\beta/p = 0.9$ ($RTT_{U-PM} = 10\text{ms}$, $RTT_{PM-MM} = 10\text{ms}$).

VIII. CONCLUSIONS

In this paper we introduced PLATO, and edge-based architecture for the online optimization of multi-platoons, and we discussed how PLATO can enforce the dynamic management of vehicle maneuvers by mapping the control of a large platoon onto the coordinated control of a chain of sub-platoons. PLATO does not use V2V communications. It is instead an edge-based two-tiers control architecture, designed to run in the VEC context.

As concerns dynamic management, we modeled dynamic multi-platoon reconfigurations and leveraged our analysis to design an enhanced control of maneuvers during such reconfigurations. The analysis served as well to formulate a problem, whose solution yields the optimal multi-platoon configuration in static and dynamic conditions. This allowed us to compute the optimal multi-platoon dynamics when a large platoon travels a highway junction and must yield to cross traffic.

By computing the multi-platoon utility function in either simulated or numerically modeled scenarios, we have shown that: (i) PLATO scales very well under a wide range of network conditions, as long as the inter-VNF latency guarantees the time coherence of instructions sent to sub-platoons; (ii) the latency between vehicles and their VNF manager has

limited impact thanks to the space-time redundancy of the update messages sent by the vehicles and the fact that PLATO generates instructions asynchronously; (iii) compact multi-platoons provide high global and individual utility, although the presence of cross road traffic and the cost of maneuvering limits the degree of compactness in the optimal solution; (iv) optimizing multi-platoon dynamics is a complex task that requires to account for multiple intertwined factors and mechanical constraints; moreover, (v) leading a platoon, or maneuvering the most because of standing towards the tail of a multi-platoon, reduces the utility of individual vehicles, which may require the design of vehicle position rotation schemes within a multi-platoon if considerations of fairness in per-vehicle utility come into play.

ACKNOWLEDGEMENTS

This work has been supported by the Project AEON-CPS (TSI-063000-2021-38), funded by the Ministry of Economic Affairs and Digital Transformation and the European Union NextGeneration-EU in the framework of the Spanish Recovery, Transformation and Resilience Plan.

REFERENCES

- [1] A. Balador, A. Bazzi, U. Hernandez-Jayo, I. de la Iglesia, and H. Ahmadvand, "A survey on vehicular communication for cooperative truck platooning application," *Vehicular Communications*, p. 100460, 2022.
- [2] R. E. Fenton and R. J. Mayhan, "Automated highway studies at the ohio state university-an overview," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 100–113, 1991.
- [3] T. R. Gonçalves, V. S. Varma, and S. E. Elayoubi, "Performance of vehicle platooning under different v2x relaying methods," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1018–1023, IEEE, 2021.
- [4] H. Peng, D. Li, Q. Ye, K. Abboud, H. Zhao, W. Zhuang, and X. Shen, "Resource allocation for cellular-based inter-vehicle communications in autonomous multiplatoons," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11249–11263, 2017.
- [5] C. Quadri, V. Mancuso, V. Cislighi, M. Ajmone Marsan, and G. P. Rossi, "From plato to platoons," in *2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet)*, pp. 1–8, 2021.
- [6] 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 *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [7] M. Zabat, N. Stabile, S. Farascarioli, and F. Browand, "The Aerodynamic Performance Of Platoons: A Final Report," Institute of Transportation Studies, Research Reports, Working Papers, Proceedings, Institute of Transportation Studies, UC Berkeley, Jan. 1995.
- [8] F. H. Robertson, F. Bourriez, M. He, D. Soper, C. Baker, H. Hemida, and M. Sterling, "An experimental investigation of the aerodynamic flows created by lorries travelling in a long platoon," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 193, p. 103966, 2019.

- [9] B. McAuliffe, M. Lammert, X.-Y. Lu, S. Shladover, M.-D. Surcel, and A. Kailas, "Influences on energy savings of heavy trucks using cooperative adaptive cruise control," in *SAE Technical Paper*, SAE International, 04 2018.
- [10] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 263–284, Firstquarter 2016.
- [11] C. Diakaki, M. Papageorgiou, I. Papamichail, and I. Nikolos, "Overview and analysis of vehicle automation and communication systems from a motorway traffic management perspective," *Transportation Research Part A: Policy and Practice*, vol. 75, pp. 147–165, 2015.
- [12] ENSEMBLE Project, "The project," 2022. <https://platooningensemble.eu/project> [Online; accessed 11-Jan-2022].
- [13] V. Lesch, M. Breitbach, M. Segata, C. Becker, S. Kounev, and C. Krupitzer, "An overview on approaches for coordination of platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10049–10065, 2022.
- [14] T. Zeng, O. Semiri, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon systems," *IEEE Trans. on Communications*, vol. 67, pp. 7907–7922, Nov 2019.
- [15] G. Cecchini, A. Bazzi, B. M. Masini, and A. Zanella, "Performance comparison between IEEE 802.11p and LTE-V2V in-coverage and out-of-coverage for cooperative awareness," in *2017 IEEE Vehicular Networking Conference (VNC)*, pp. 109–114, Nov 2017.
- [16] F. Dressler, F. Klingler, M. Segata, and R. L. Cigno, "Cooperative driving and the tactile internet," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 436–446, 2019.
- [17] B. Wang, J. Zheng, Q. Ren, and C. Li, "Analysis of IEEE 802.11p-based intra-platoon message broadcast delay in a platoon of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 10, pp. 13417–13429, 2023.
- [18] B. Wang, J. Zheng, N. Mitton, and C. Li, "Inp-crs: An intra-platoon cooperative resource selection scheme for c-v2x networks," *IEEE Communications Letters*, vol. 27, no. 11, pp. 3118–3122, 2023.
- [19] C. Quadri, V. Mancuso, M. Ajmone Marsan, and G. P. Rossi, "Edge-based platoon control," *Computer Communications*, vol. 181, pp. 17–31, 2022.
- [20] A. Viridis, G. Nardini, and G. Stea, "A framework for MEC-enabled platooning," in *IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, pp. 1–6, IEEE, 2019.
- [21] X. Fan, T. Cui, C. Cao, Q. Chen, and K. S. Kwak, "Minimum-cost offloading for collaborative task execution of MEC-assisted platooning," *Sensors*, vol. 19, no. 4, p. 847, 2019.
- [22] Y. Hu, T. Cui, X. Huang, and Q. Chen, "Task offloading based on Lyapunov optimization for MEC-assisted platooning," in *International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–5, Oct 2019.
- [23] R. Meneghette, R. De Grande, J. Ueyama, G. P. R. Filho, and E. Madeira, "Vehicular edge computing: Architecture, resource management, security, and challenges," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–46, 2021.
- [24] I. Rubin, A. Baiocchi, Y. Sunyoto, and I. Turcanu, "Traffic management and networking for autonomous vehicular highway systems," *Ad Hoc Networks*, vol. 83, pp. 125–148, 2019.
- [25] H. Peng, D. Li, K. Abboud, H. Zhou, H. Zhao, W. Zhuang, and X. Shen, "Performance analysis of IEEE 802.11p dcf for multiplooning communications with autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2485–2498, 2017.
- [26] P. Fernandes and U. Nunes, "Multiplooning leaders positioning and cooperative behavior algorithms of communicant automated vehicles for high traffic capacity," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1172–1187, 2015.
- [27] M. Won, "L-platooning: A protocol for managing a long platoon with dsrc," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5777–5790, 2022.
- [28] J. Mena-Oreja and J. Gozalvez, "On the impact of platooning maneuvers on traffic," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 1–6, 2018.
- [29] D. Bischoff, F. A. Schiegg, T. Meuser, D. Schuller, and R. Steinmetz, "Adaptive heterogeneous v2x communication for cooperative vehicular maneuvering," in *Smart Cities, Green Technologies, and Intelligent Transport Systems* (C. Klein, M. Helfert, K. Berns, and O. Gusikhin, eds.), (Cham), pp. 228–254, Springer International Publishing, 2021.
- [30] Q. Li, Z. Chen, and X. Li, "A review of connected and automated vehicle platoon merging and splitting operations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22790–22806, 2022.
- [31] R. Firoozi, X. Zhang, and F. Borrelli, "Formation and reconfiguration of tight multi-lane platoons," *Control Engineering Practice*, vol. 108, p. 104714, 2021.
- [32] X. Han, R. Xu, X. Xia, A. Sathyan, Y. Guo, P. Bujanović, E. Leslie, M. Goli, and J. Ma, "Strategic and tactical decision-making for cooperative vehicle platooning with organized behavior on multi-lane highways," *Transportation Research Part C: Emerging Technologies*, vol. 145, p. 103952, 2022.
- [33] G. Ibanez, T. Meuser, M. A. Lopez-Carmona, and D. Lopez-Pajares, "Synchronous roundabouts with rotating priority sectors (syrops): High capacity and safety for conventional and autonomous vehicles," *Electronics*, vol. 9, no. 10, 2020.
- [34] R. Rajamani, *Vehicle dynamics and control*, ch. 7. Springer, 2012.
- [35] L. Stahlbock, J. Weber, and F. Köster, "An optimization approach of container startup times for time-sensitive embedded systems," in *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pp. 2019–2026, 2022.
- [36] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, pp. 3–15, Jan 2011.
- [37] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Viridis, "Simu5g—an omnet++ library for end-to-end performance evaluation of 5g networks," *IEEE Access*, vol. 8, pp. 181176–181191, 2020.
- [38] S. Santini, A. Salvi, A. S. Valente, A. Pescapé, M. Segata, and R. Lo Cigno, "A consensus-based approach for platooning with intervehicular communications and its validation in realistic scenarios," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1985–1999, 2017.

Christian Quadri is an Assistant Professor at the Computer Science Dept. of the University of Milan. He received a Ph.D. in Computer Science in 2015 at the University of Milan. His research interests focus on 5G/6G communication networks and intelligent resource management in edge computing environment. His previous research focused on the analysis of communication patterns and social interactions of mobile users.

Vincenzo Mancuso (SM'21) is Research Associate Professor at IMDEA Networks, Madrid, Spain. Previously, he was with INRIA (France), Rice University (USA) and University of Palermo (Italy), from where he obtained his Ph.D. in 2005. His research focus is on analysis, design, and experimental evaluation of wireless and intelligent edge architectures for communication and computation.

Marco Ajmone Marsan is a part-time research professor at the IMDEA Networks Institute in Spain. From 1974 to 2021 he was at the Politecnico di Torino, in the different roles of an academic career, with an interruption from 1987 to 1990, when he was a full professor at the University of Milan. He was the General Co-chair of Infocom 2013 and ICC 2023. He is IEEE Fellow, and member of the Academia Europæa and of the Academy of Sciences of Torino. He is qualified as "ISI Highly Cited researcher" in computer science. He received a honorary degree in Telecommunication Networks from the Budapest University of Technology and Economics.

Gian Paolo Rossi was full professor of the University of Milano, Department of Computer Science, where he held different institutional positions. From 1976 to 1979, he was at the Joint Research Center of the EC in Ispra (Italy) as a visiting researcher. At the University of Milano he led the computer networks Lab, whose research activities mainly focus on the design and performance evaluation of network protocols. In particular, the Lab's recent interests are 5G/6G networks, edge resource management/optimization and orchestration, Data and Network Science applied to networking, edge-based platoon control. On these topics, Gian Paolo Rossi is an author of nearly 100 papers and he coordinated and participated in many research projects. He retired at the end of 2021.