

From PLATO to Platoons

Christian Quadri¹, Vincenzo Mancuso², Valerio Cislaghi¹, Marco Ajmone Marsan^{2,3}, and Gian Paolo Rossi¹

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

²IMDEA Networks Institute, Madrid, Spain

³Electronics and Telecommunications Department, Politecnico di Torino, Turin, Italy

Abstract—Grouping vehicles into platoons promises to improve road capacity, driver safety, and fuel consumption. However, when platoons have to allow for cross traffic maneuvers, the ability to control single large platoons is not sufficient, and chaining smaller platoons becomes necessary. To this aim we define PLATO, an edge-assisted multi-platoon control architecture and, by delving into the dichotomy of unity and plurality of platooning, we analyze costs and benefits of multi-platooning. We investigate on the feasibility and formulate the utility of multi-platoons by analyzing the underlying edge computing and broadband cellular connectivity requirements. Using a detailed simulator, we show that, in realistic environments, multi-platoons can be effectively controlled with PLATO, as long as the latency between individual platoon managers and the multi-platoon manager is kept below a few tens of milliseconds. Surprisingly, the latency between vehicles and managers is one order of magnitude less critical.

Index Terms—Multi-platooning; MEC; Simulation

I. INTRODUCTION

The dichotomy between the one and the many is the subject of several speculations of early philosophers, and in particular of Plato’s dialogue *Parmenides*, in which the author portrays Parmenides and Zeno debating with Socrates about unity and plurality of beings. At a much more modest level, we discuss the same dichotomy in the context of vehicles’ platooning.

Platooning of commercial vehicles along highways is likely to be one of the first domains of application of automated driving, since it brings the potential for better road efficiency, lower fuel consumption and reduced drivers’ fatigue. The idea of grouping vehicles into platoons dates back to the late 80s (see for instance [1]), but only recently has become viable, thanks to the combined development of advanced traffic management systems (ATMS) and autonomous driving. Controlling platoons to maintain safe inter-vehicle distancing and string stability, i.e., to ensure that the inter-vehicle distance error does not amplify downstream along the string of vehicles, proved to be a challenging task, heavily relying on wireless communication technologies able to guarantee effective, low latency vehicle-to-vehicle (V2V) and vehicle-to-anything (V2X) communications. However, the ability to control a single platoon may soon become insufficient. Indeed, the success of platooning may lead to very long strings of vehicles travelling at short distances, that can hamper the flow of vehicles that need to cross the platoon to reach a highway exit or a service station. This calls for spacing after some number of vehicles, thus producing a multi-platoon setting, i.e., a chain of platoons that follow one another. Multi-platoon control is very challenging because it requires enforcing stability among platoons, and is more demanding than the control of single platoons in terms of network resources. Two standard approaches to support communications within and among platoons are possible: the distributed approach, based on dedicated short-range communications (DSRC), and the centralized approach based on cellular radio access networks

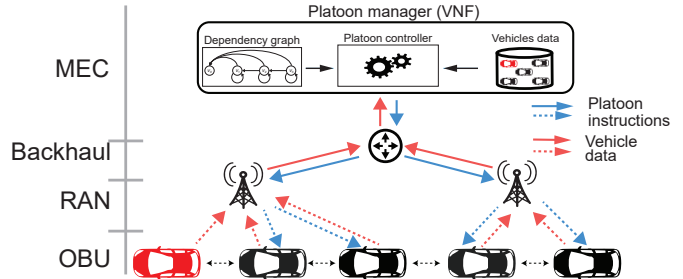


Fig. 1: Edge-based platoon controller architecture, using MEC and cellular base stations

(RANs). However, while DSRC would be put under stress by high vehicle density and number of vehicles in multi-platooning [2], cellular infrastructures offer high capacity and low latency, larger coverage range, and flexible multi-access edge computing (MEC) resources.

In this paper, we look at such MEC-based cellular scenario, investigating the feasibility of edge-assisted multi-platooning application, and leveraging our previous work for single platoons [3]. As shown in Fig. 1, multi-platoon-specific data, together with vehicle safety information, namely speed and acceleration, are transmitted over the RAN to platoon managers in the MEC. Platoon managers are coordinated by a multi-platoon manager, to generate actuation commands coherently across platoons. These commands are transmitted from MEC to vehicles, again through the RAN, so as to effectively control the inter-platoon distance. We name the resulting control architecture PLATO (Platoon Layout Architecture for Traffic Optimization). To test PLATO, we develop a detailed simulator based on SUMO [4] and derive a platoon utility function that accounts for road utilization, aerodynamic drag, as well as costs of connectivity, computation, fuel consumption and reconfiguration.

The main contributions of this paper are the following: (i) we define PLATO as the edge-assisted cruise control architecture for multi-platoons; (ii) we define a utility function that accounts for the different benefits and costs of multi-platooning; (iii) we design a tool for the simulation analysis of multi-platoons; (iv) we discuss numerical results showing the achievable multi-platoon stability under several configurations, and the reachable utility values. Our results unveil the importance of placing platoon managers close to the multi-platoon manager to avoid inconsistencies in control instructions.

The rest of this paper is organized as follows. Section II presents the related work. Section III describes our edge-assisted control architecture, PLATO. Section IV discusses overheads of PLATO, while we define the general structure of costs and benefits of multi-platooning in Section V, where we propose a utility function. The evaluation of PLATO is

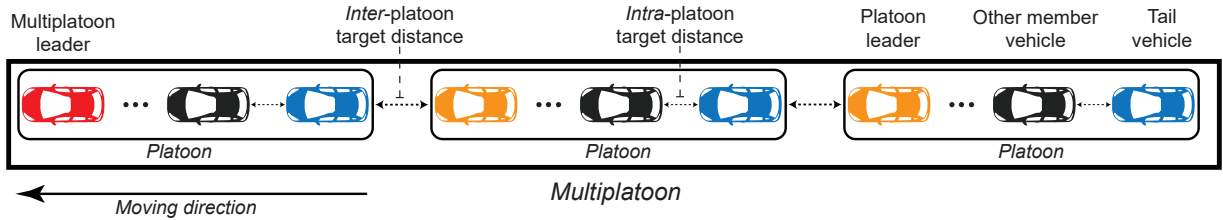


Fig. 2: Multi-platoon composed of three platoons

presented in Section VI. Eventually, Section VII summarizes the paper and comments on future work.

II. RELATED WORK

Platooning was introduced to improve road capacity and traffic flow efficiency, reduce fuel consumption, and enhance passengers comfort [5], [6]. Fuel saving results from the reduction of air drag resistance. The problem has been studied through experiments in wind tunnels [7], [8], simulations [9], [10], and field experiments [11]. From these studies, the overall platoon reduction of fuel consumption can be estimated in 4 to 12% depending on inter-vehicle distance, while the fuel saving for single vehicles depends on their position within the platoon [8], [11].

Regarding road utilization, a few works describe the optimal organization of platoons. Rubin *et al.* [12], e.g., consider admission ramps and transit delays in multi-lane highways. The associated solution was experimented in a real highway, relying on V2V and showing the ineradicable trade-off between the maximization of vehicular traffic flow and networking delays. Part of our formulation in Section VI is based on numerical data coming from these experiments.

Traditional policies to control a platoon rely on a distributed approach based on either IEEE 802.11p or 3GPP V2X dedicated short range communication standards [5], [13], [14], [15]. These studies show that a tight integration between control and communication systems is necessary to maintain short inter-vehicle distances and preserve string stability. However, V2V struggles against radio interference and shadowing, and multi-hop transmissions in the case of long platoons. Recent studies show that 5G networks do not suffer those problems and are suitable to run edge-assisted platooning control [3], [16], [17], [18], with network delays bounded to a few tens of milliseconds. So, in PLATO, we rely on 5G and MEC.

When platoons encompass several tens of vehicles, platoon control becomes challenging from both communication and vehicle maneuvers perspectives [19]. Hence, multi-platooning has been proposed [2], [12], [19], [20], relying on fully distributed V2V approaches or hybrid V2I/V2N paradigms. In particular, [2] focuses on radio resources allocation, and compares the LTE evolved multimedia broadcast multicast service (eMBMS) with device-to-device (D2D) multicast communication. Results show that the latter requires less network resources than eMBMS and outperforms unicast D2D. [20] proposes a hybrid solution with intra-platoon communications on V2V, and cooperative management of multiple platoons delegated to a central ATMS entity. Instead, by expanding the edge-assisted platooning we adopted in [3], we design a multi-platoon manager on top of a set of platoon managers, all of them running in the MEC as virtual network functions (VNFs), so as to take full advantage of negligible

interference, high scalability, and simplicity of integration with other intelligent transportation systems by operating in a cellular network.

III. EDGE-ASSISTED PLATOONING

We consider platooning control as a network service that runs at the edge of the cellular network, as in [3]. This means that we rely on V2N (vehicle to network) for what concerns communication aspects between vehicles and platoon manager, whilst we rely on edge servers (e.g., ETSI MEC hosts) for what concerns the virtualization of the platooning control algorithm, i.e., CACC [21]. The algorithm maintains a practically constant inter-vehicular distance δ , based on the received position/speed/acceleration updates as far as the network delay is below about 200 ms, as shown in [3].

A. A platoon in isolation

The single platoon controller can be implemented as depicted in Fig. 1. The architecture relies on vehicle's equipment, (broadband) cellular base stations and a MEC host. In the vehicles, the on-board unit (OBU) acquires data from the on-board sensors and sends it over the cellular RAN, from which it also receives control instructions on how to adapt the acceleration of the vehicle. Platoon's vehicles do not need to be under the coverage of a single base station and require limited bandwidth, in the order of a few kbps per vehicle [3].

A dependency graph owned by a platoon manager VNF determines which updates are needed in order to re-compute and issue new instructions for each of the platoon's members separately. In the graph, nodes represent vehicles, and an arrow from node a to b tells that the instructions computed for vehicle a depends on data updates sent by vehicle b .

The edge-assisted approach eases the management of platoons, and lowers the requirements on the OBU that is in charge of interacting with an external manager but does not need to run any control logic on its own. This is extremely useful when the platoon manager needs to account not only for its platoon, but also for the presence of other platoons and, in general, other intelligent transportation systems.

B. Multi-platooning: a chain of platoons

We define a multi-platoon as a set of coordinated platoons that form a chain of vehicles on a single road-lane. Fig. 2 shows a generic multi-platoon composed of three platoons. Platoons are managed to maintain a target inter-platoon distance Δ , and vehicles assume one of the following four roles: (i) *multi-platoon leader* (red vehicle in Fig. 2), which is to the leader of the first platoon; (ii) *platoon leader* (orange), which is the first vehicle of any other platoon; (iii) *tail* (blue), the last vehicle of each platoon, or simply (iv) *member* (black) in all other cases. The multi-platoon leader is the only vehicle that does not need control instructions, while all other

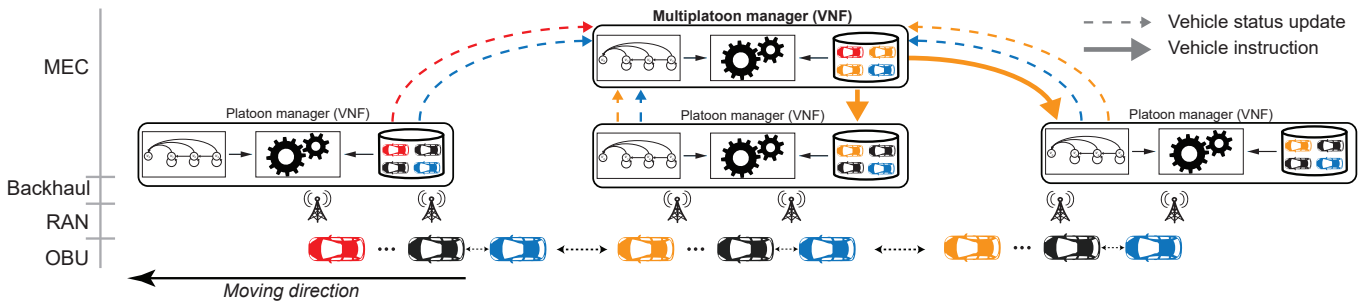


Fig. 3: PLATO's control architecture

vehicles are subject to the instructions received from their platoon managers, computed with the CACC control law.

For a specific vehicle, CACC computes the acceleration by using the data of three vehicles: the vehicle itself, the vehicle in front, and the platoon leader. Unlike in a single platoon scenario where each vehicle looks at its platoon leader, the computation of CACC for each multi-platoon vehicle is based on the data of the multi-platoon leader. Thus, multi-platoon leader data need to be forwarded to all platoon managers. Besides, the computation of CACC for a platoon leader requires data from the tail vehicle of the preceding platoon. Hence, only the first and the last vehicle of each platoon are involved in the multi-platoon CACC computation.

C. PLATO

Each platoon in the chain has a platoon manager VNF on the MEC which is in charge of controlling the platoon. In analogy with that, we propose multi-platoon control by means of a dedicated multi-platoon manager VNF, which enforces control and stability along the string of platoons by simply exchanging control data and instructions on the backhaul network with each platoon manager. The resulting architecture, which we call PLATO, is depicted in Fig. 3. Each platoon manager forwards the data updates of leading and tail vehicles to the multi-platoon manager (coloured and dashed arrows in the figure). Upon receiving data, the multi-platoon manager identifies, by means of a dependency graph (see Section III-D) the vehicles requiring new instructions. For each of them, the multi-platoon manager computes the control law and sends the instructions back to the platoon manager in charge of controlling that vehicle (solid arrows in the figure). As both the platoon and the multi-platoon managers lie in the MEC layer, these communications occur on the backhaul.

PLATO requires a proper assignment of computational and networking resources to the multi-platoon manager VNF; in addition, to avoid multi-platoon instability, the communication delay between multi-platoon and platoon managers should not exceed critical values. The first point is partially mitigated by observing that the multi-platoon manager is not in charge of controlling all the vehicles of the multi-platoon. In fact, the multi-platoon manager sees each platoon as a single vehicle spanning from leader to tail. Therefore, for each platoon, the multi-platoon manager receives at most two updates from leader and tail, and sends one instruction within an update period. As regards the impact of network delay on multi-platoon control, it is critical that the interaction between platoon managers and multi-platoon manager be delayed as little as possible, since in PLATO the cross-

platoon info needed by CACC flows only through the multi-platoon manager. In Section VI-B we will provide extensive simulation results to evaluate the impact of latency.

D. PLATO's dependency graph

With PLATO, vehicle dependencies are set according to their role. Fig. 4 shows the dependency graph in a multi-platoon. The multi-platoon manager monitors only leaders and tails (*orange circles*) and controls only platoon leaders (*orange circles with a dot inside*). There are three types of edges in the graph, to distinguish CACC computations: (i) a *gray* edge indicates that a platoon manager computes CACC for the destination node as in a single platoon scenario; (ii) a *blue* edge indicates that CACC is computed by the platoon manager by using the data of the multi-platoon leader; (iii) a *orange* edge indicates that the control law is directly computed by the multi-platoon manager. Within each platoon, this graph topology maintains the same dependency structure as for independent platoons, and relies on the multi-platoon manager to connect platoon leaders to the multi-platoon leader and their respective preceding tails.

IV. PLATO'S MANAGEMENT OVERHEAD

Controlling a multi-platoon with PLATO entails computation and communication overhead, both affected by the multi-platoon composition.

The computation overhead is the sum of the overheads of platoon and multi-platoon managers. Since the CACC control law is computed each time a vehicle's update reaches a manager, the overhead per update cycle is the number of edges in the dependency graph. Two edges depart from each member immediately following a leader, and three from any other vehicle except the multi-platoon leader (see Fig. 4), hence there are $(2N_p + 3(N_v - N_p - 1)) = (3N_v - N_p - 3)$ edges, where N_v is the number of vehicles distributed over N_p chained platoons, and the overhead is the number of edges multiplied by ϕ_u , which is the control update frequency. Note that a single platoon of N_v vehicles without PLATO would incur an overhead of $(3N_v - 4)\phi_u$ [3]. Thus, PLATO's computational overhead with $N_p = 1$ is the same as for a single platoon without PLATO. However, PLATO's overhead *decreases* with the number of chained platoons for a fixed number of vehicles. Therefore, multi-platooning is computationally convenient with respect to normal platooning.

As regards communication, PLATO requires the exchange of unicast messages through both RAN and backhaul networks. The number of messages sent by vehicles, in the time unit is $N_v\phi_u$, regardless of PLATO. The number of messages received in downlink corresponds to the number of edges in the dependency graph, because each edge also represents

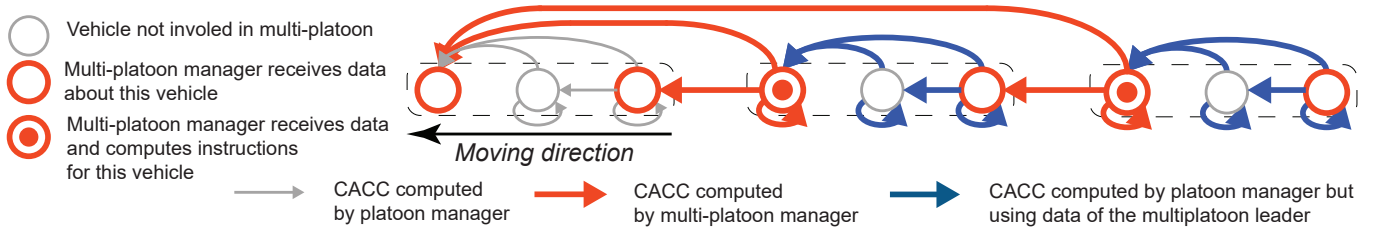


Fig. 4: Dependency graph of PLATO

TABLE I: Example of communication overhead breakdown

#msg	RAN uplink		Backhaul		RAN downlink	
	$N_v \phi_u$		$(5N_p - 3)\phi_u$		$(3N_v - N_p - 3)\phi_u$	
Composition	#msg	Mb/s	#msg	Mb/s	#msg	Mb/s
$N_v=100, N_p=1$	1000	1.6	-	-	2960	2.37
$N_v=100, N_p=2$	1000	1.6	70	0.14	2950	2.36
$N_v=100, N_p=5$	1000	1.6	220	0.45	2920	2.34
$N_v=100, N_p=10$	1000	1.6	470	0.97	2870	2.30
$N_v=100, N_p=20$	1000	1.6	970	2.01	2770	2.22

the flow of instructions that reaches the vehicle from which the edge departs. Therefore, the expression computed for the computational complexity also holds for RAN downlink overhead, which means that PLATO with $N_p > 1$ decreases the load of the RAN in downlink, without affecting the uplink.

PLATO also uses the backhaul network for communication between platoon managers and the multi-platoon manager. It generates $(5N_p - 3)\phi_u$ backhaul messages per time unit: $2N_p\phi_u$ sent by platoon managers to forward leader and tail updates to the multi-platoon manager, plus $3(N_p - 1)\phi_u$ instructions sent by the multi-platoon manager to platoon leaders (solid orange edges in Fig. 4). Although very limited, the backhaul overhead is a drawback of PLATO, which makes the overall communication overhead (in backhaul plus RAN) linearly increase with N_p .

Table I conveys an example of communication overhead for a multi-platoon of 100 vehicles, and $\phi_u = 10$ Hz (value used in [3] and other works), under some possible homogeneous configurations. The table also reports bandwidth requirements, assuming that vehicle update messages are 200-byte long, instructions take 100 bytes, while multi-platoon instruction message size is 300 bytes (100 bytes of instruction and 200 bytes of data from the multi-platoon leader). Bandwidth requirements are extremely low, with at most a fistful of Mb/s for the RAN uplink and downlink, and much less for backhaul.

V. PERFORMANCE METRICS

In the (multi-)platooning scenario, a subset of the vehicles traveling over a road must be arranged in a multi-platoon configuration in the best possible way. To determine the quality of the possible arrangements, we rely on a utility function whose maximization shows how to optimally form platoons.

A. PLATO's utility function

To design a utility function for PLATO, we consider that a platooning approach is normally justified by the following objectives: (i) reduce fuel consumption, thanks to the drafting effect generated by preceding and following vehicles on all platoon members, and (ii) increase road utilization efficiently by packing a high number of vehicles per lane and per space unit.

However, in addition to the overhead discussed in Section IV, (multi-)platooning calls for attention to a number of issues. First, the inter-platoon distance has to be chosen as a function of safety, network requirements and latency in the control architecture. Second, platoons might need to be rearranged and re-optimized due to context modifications (e.g., vehicles joining or leaving, or changing road conditions). Third, the presence of platoons impacts on other vehicles' maneuvers, considering that vehicles may need to cross a (multi-)platoon to exit a highway or reach a gas station, or stop for an emergency, and long platoons constitute an obstacle for such maneuvers (also consider the possible existence of cross flows, e.g., at a roundabout or an intersection/junction).

We structure the PLATO utility function $U(t)$ at time t in a way that accounts for the goals and issues mentioned above. We also consider that the utility function must account for individual and global benefits, and must provide a fair share of benefit to each vehicle, hence we use the function:

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

where $U(i, t)$ is the individual utility for vehicle i at time t , expressed as the inverse of the fuel cost $C_f(i, t)$, i.e.:

$$U(i, t) = 1/C_f(i, t). \quad (2)$$

The functions in (1)-(2) describe the following quantities.

1) $R(t)$: is the metric that accounts for the road utilization efficiency of the multi-platoon configuration. It can be expressed as the ratio between the space occupied on the road with no platooning and the one with platooning, including the spacing between platoons that allows other vehicles to, e.g., reach an exit or a gas station. We can thus write:

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

where L_0 is the road length occupied by N_v vehicles of length ℓ in the case of no platooning, with average spacing $\delta_0 > \delta$, i.e.:

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

and $L_p(t)$ is the multi-platoon length at epoch t , i.e.:

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

and where $N_p(t)$ indicates the total number of platoons in the multi-platoon at time t , and, as previously stated, δ and Δ indicate intra- and inter-platoon distances, respectively.

2) $C_c(t) \geq 1$: is the computation cost of the multi-platoon control, i.e., $(3N_v - N_p(t) - 3)\phi_u$ (see Section IV) normalized to $2(N_v - 1)\phi_u$, which is the computational overhead of the Adaptive Cruise Control (ACC) algorithm [21] that could be used at the MEC in absence of platooning, to control inter-vehicle distances:

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

3) $C_t(t) > 1$: is the transmission cost for RAN and backhaul as described in Section IV, normalized to the transmission cost in the case without platoons, i.e.:

$$\begin{aligned} C_t(t) &= \frac{N_v + (3N_v - N_p(t) - 3) + (5N_p(t) - 3)}{3N_v - 2} \\ &= \frac{4(N_v + N_p(t)) - 6}{3N_v - 2}, \end{aligned} \quad (7)$$

where $3N_v - 2$ is the number of transmissions per update cycle needed with ACC, resulting from the sum of N_v transmissions in uplink and $2(N_v - 1)$ transmissions in downlink.

4) $C_r(t) \geq 1$: is the relative reconfiguration cost incurred to switch to the single platoon configuration, taken as reference. This cost is due to the time T_{join} needed to close the gap $\Delta - \delta$ at leader's speed v at time t . We normalize T_{join} to the time needed to travel the average distance S between two consecutive re-configurations (e.g., the average distance between successive highway exits), scale the result with the number of platoons that have to act, i.e., $N_p(t) - 1$, and we add a term 1 to express the fact that the relative cost of a single platoon, not requiring any reconfiguration, is 1, while with $N_p(t) > 1$ the relative cost must be higher:

$$C_r(t) = \frac{T_{\text{join}}(\Delta - \delta, v(t))}{S/v(t)} (N_p(t) - 1) + 1. \quad (8)$$

The value of T_{join} must be estimated, e.g., via simulation.

5) $C_f(i, t) \in (0, 1)$: is the relative fuel consumption of vehicle i (considering its position within the multi-platoon and the platoon), normalized to the fuel consumption that would be incurred without the drag reduction brought in by platooning.

According to [7] the fuel consumption of a vehicle at given speed v can be expressed as

$$F = P c_f / v = R c_f / \eta, \quad (9)$$

where P is the engine power that is required to contrast the resistance encountered by a vehicle moving forward, c_f represents the specific fuel consumption of the vehicle, R is the total resistance, and η is the transmission efficiency of the engine. The two expressions in (9) hold because of the relation $P = R v / \eta$. When a vehicle is travelling alone at constant speed on a level road, the total resistance R has two components, i.e., aerodynamic drag and rolling resistance:

$$R = \frac{1}{2} D_0 \alpha A v^2 + r M g, \quad (10)$$

where D_0 is the drag coefficient, α is the air density, A is the cross-sectional area of the vehicle in the plane orthogonal to the direction of motion, r is the rolling resistance coefficient, M is the mass of the vehicle, and g is the gravity acceleration. If a vehicle is travelling in a platoon, the drag coefficient

changes to $D_p \leq D_0$. The actual value of D_p depends on the position of the vehicle within the platoon and the inter-vehicle distance. In [7], the ratio D_p/D_0 is reported as a function of inter-vehicle distance and vehicle position. We use those values to derive a function that provides the ratio between the drag coefficient as a function of distance δ and position within the platoon i , namely $\gamma_p(\delta, i) < 1$. Therefore, the ratio between the total resistance encountered by a vehicle in a platoon and the resistance it would face if it were travelling alone is

$$\rho_p(\delta, i) = \frac{\frac{1}{2} D_0 \alpha A v^2 \gamma_p(\delta, i) + r M g}{\frac{1}{2} D_0 \alpha A v^2 + r M g} < 1. \quad (11)$$

With platooning, fuel consumption drops proportionally to total resistance, with $\rho_p(\delta, i)$. Thus, the fuel consumption normalized to the case of no platoon is simply given by $C_f(i, t) = \rho_p(\delta, i)$. In conclusion, we can express the individual vehicle's utility (2) as

$$U(i, t) = \frac{1}{\rho_p(\delta, i)} = \frac{\frac{1}{2} D_0 \alpha A v^2 + r M g}{\frac{1}{2} D_0 \alpha A v^2 \gamma_p(\delta, i) + r M g}, \quad (12)$$

which shows how the utility of vehicle i vanishes at slow speeds and is upper-bounded by the quantity $1/\gamma_p(\delta, i)$.

B. Constraints

The utility function (1) can be used to evaluate alternative multi-platooning configurations under specific constraints. Although optimization is out of the scope of this work, we remark that constraints must account for safety and comfort aspects, and for the presence of other vehicles. In what follows we express the mathematical form of the corresponding constraints.

Safety and comfort affect the choice of the intra-platoon spacing δ , which is the target of CACC. The actual constraint for vehicle i , except the multi-platoon leader, is expressed as

$$d_i(t) \geq (1 - \epsilon) \delta, \quad \forall t, \forall i \geq 2,$$

where $d_i(t)$ is the distance between vehicles $i - 1$ and i at time t , and $\epsilon \ll 1$ is the margin of error for CACC.

In the case of long multi-platoons, the optimization also requires imposing a constraint on the volume of traffic that can maneuver in the presence of a multi-platoon. Maneuvers are made possible by the $N_p - 1$ inter-platoon spaces of length Δ ; thus, if $\beta(t)$ is the density of traffic maneuvers that the multi-platoon must yield, a simple constraint is formulated as

$$k (N_p(t) - 1) \Delta > \beta(t),$$

k being the traffic-maneuver density per unit of inter-platoon spacing. This imposes a non-trivial minimum on $N_p(t)$.

VI. PERFORMANCE EVALUATION

To evaluate PLATO, we simulate a highway scenario with a fleet of light duty vehicles. The multi-platoon leader follows the predefined speed profile shown in Fig. 5 extracted from the Floating Car Dataset¹ and pre-processed by applying a moving average filter. The speed profile combines a wide range of speed, from 65 to 125 km/h together with acceleration/deceleration maneuvers (see, e.g., Part A in Fig. 5).

¹TIM Big Data Challenge 2015: Floating-Car-Data-Milano

TABLE II: Simulation parameters

Vehicle and road parameters	
Length (ℓ)	6 m
Mass (M)	3 300 kg
Air drag coefficient (D_0)	0.4
Sectional area (A)	4 m ²
Engine transmission efficiency (η)	0.9
Rolling friction coefficient (r)	0.013 (car tyre on concrete)
Air density (α)	1.225 kg/m ³
Drag coefficient ratio ($\gamma_p(\delta, i)$)	$\delta = 10$ m $\delta = 15$ m
	Leader: 0.92 0.96
	Middle: 0.73 0.76
	Trailer: 0.74 0.75
Multi-platoon configurations	
Number of vehicles (N_v)	Stability: 20 Utility function: 50
Number of platoons (N_p)	Stability: 1,2,4,5 Utility function: 1, 2, 5, 10
Intra-platoon distance (δ)	Stability: 10 m Utility function: 10, 15 m
Inter-platoon distance (Δ)	Stability: 25 m Utility function: 25, 50, 75, 100 m
No-platoon vehicle distance (δ_0)	50 m
Average distance between consecutive re-configurations (S)	10 km
Update frequency (ϕ_u)	10 Hz
Network latency	
Distribution family for uplink/downlink RTT	Lognormal (with unit standard deviation)
$\mathbb{E}[RTT_{\text{RAN}}]$ (vehicle to VNF of its platoon mng.)	Stability: 30, 40, 70, 120, 220 ms Utility function: 70 ms
Distribution family for VNF-to-VNF RTT	Uniform
$\mathbb{E}[RTT_{\text{BH}}]$ (platoon mng. VNF to multi-platoon mng. VNF)	Stability: 10, 20, 40, 60, 120 ms Utility function: 10 ms

Besides, we simulate realistic vehicle dynamics using the approach presented in [22] which takes into account mechanical and aerodynamic characteristics, achievable acceleration at given speed, and inertia of the vehicle. In the upper part of Table II, we report the main parameters of the Fiat Ducato² vehicle we used for our simulations.

We model the RAN delay as a random variable, and assume symmetric channels. In particular, we model the uplink and the downlink delays as lognormal random variables with unit standard deviation (of the natural logarithm of the variable) and expected value varying from 15 to 110 ms, which results in an average round-trip time $\mathbb{E}[RTT_{\text{RAN}}]$ from 30 to 220 ms in the RAN. In the case of VNF-to-VNF communications over the backhaul between platoon and multi-platoon managers, we use a uniform distribution of latency, and we consider $\mathbb{E}[RTT_{\text{BH}}]$ values from 10 up to 120 ms. These values allow us to simulate multiple deployment settings. The lowest values of uplink/downlink delay correspond to the deployment of platoon manager VNFs on 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, 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 present results for two sets of simulations, whose parameters are reported in Table II. The first set is labeled as *stability* in the table, and is meant to investigate the string stability with PLATO, and the intra-vehicle distance error according to different RTT values and multi-platoon compositions. In there, we simulate 20 vehicles considering only multi-platoon configurations with the same number of vehicles per platoon. The second set of simulations (labeled

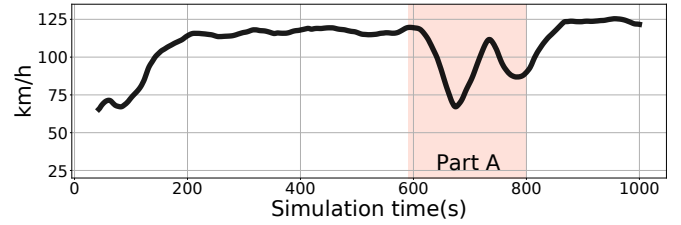


Fig. 5: Multi-platoon leader speed profile

as *utility function* in Table II) is meant to analyze the values of our proposed utility function. For this case, we consider 50 vehicles, and we vary, beside the multi-platoon composition, also the values of δ and Δ . For each simulation scenario, we perform at least 20 runs.

The value of δ_0 shown in Table II 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.³ 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.

A. Simulations setup

The computation of our utility function requires to estimate $T_{\text{join}}(\Delta - \delta, v(t))$, which is the time needed by a platoon leader to catch up with the immediately preceding platoon. T_{join} depends on how CACC operates and on the vehicle characteristics. As the analytical modelling of CACC operations during a join event is outside the scope of this paper, we opted for ad-hoc platooning simulations to empirically characterize T_{join} , with the speed profile of Fig. 5, the vehicle characteristics of Table II, and different combinations of δ and Δ . Simulations show that T_{join} slightly depends on speed, which remains under the top speed of the vehicle, while it is strongly affected by the gap to close up. We empirically obtained values between 25 and 35 seconds, which have been used to compute the reconfiguration cost C_r .

In line with the experiments presented in [11], 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 II. The value of the middle vehicle is then applied to all internal platoon positions.

B. Multiplatoon stability

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

Fig. 6a and Fig. 6b depict two extreme scenarios for the RAN RTT, i.e., 30 and 220 ms, respectively, when the backhaul RTT is low (10 ms, on average). We can observe a general stable behavior with error values monotonically decreasing, except in a few cases, and by small amounts, and the number of platoons has barely an impact. The 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 all messages suffer similar

²Vehicle specifications taken from <https://www.cars-data.com/>

³The 2-second rule, typically taught at driving schools, suggests to take as safety margin the distance covered by the vehicle in 2 seconds.

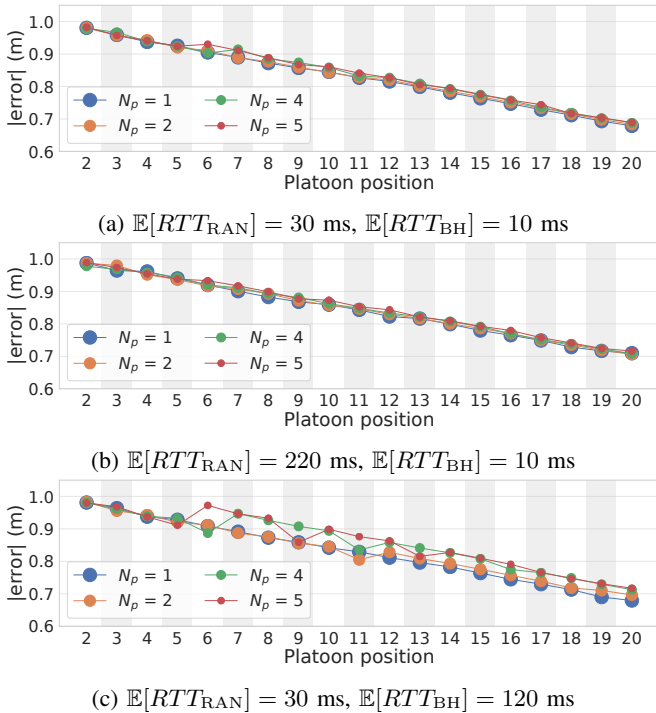


Fig. 6: CACC error in PLATO under different latency conditions. Latency increases in the RAN (uplink plus downlink) are much less critical than in the backhaul (VNF to VNF).

delays, the multi-platoon remains practically stable (in line with our previous results on isolated platoons [3]).

Fig. 6c tells a different story. In fact, when the backhaul RTT becomes comparable to the one seen with a VNF running in the cloud (120 ms), even if the RAN RTT is low (30 ms), the multi-platoon visibly exhibits a lower stability. In particular, the second vehicle in each platoon, except in the leading platoon, experiences more error than its platoon leader. This is clearly visible for $N_p = 4$ (e.g., on vehicles 7 and 12) and $N_p = 5$ (e.g., on vehicles 6 and 10). The reason for instability lays in the different levels of freshness of the data used for CACC. In fact, 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 high 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.

In all cases, the error is below 1 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 $\delta = 10$ m.

C. Evaluation of utility function

We break the utility function into its components described in Section V and show their log value in Fig. 7. Positive values are gains (road efficiency and fuel consumption reduction, summed over all vehicles), while negative values are costs (for computation, transmission and reconfiguration). The figure reports averages over time for each component and for the resulting utility, with different combinations of N_p , δ , and Δ .

Fig. 7 makes clear 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., $N_p = 10$).

The computation cost slightly decreases as a function of the 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 IV. The reconfiguration cost strongly penalizes configurations with more platoons, because the more the platoons the higher the number of vehicles that need to close the gap $\Delta - \delta$ (see (8)).

As clear from Fig. 7, 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. 8 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 safety and maneuvers of other vehicles described in Section V-B.

The last aspect we analyze is the change of contribution to the utility function of the different components over time, when no reconfiguration occurs. Fig. 9 shows fuel and reconfiguration cost components, since nothing else changes by only varying the speed of the multi-platoon leader. In the figure, we only refer to Part A of Fig. 5, where speed changes are more pronounced. We can observe that fuel saving goes with speed variations, being it very small while decelerating and following the acceleration curve otherwise. The reconfiguration cost is less variable and follows the shape of the speed, because so does T_{join} due to non-ideal engine limits. The dependency on speed is more evident in case of large numbers of platoons, because more vehicles are involved in reconfiguration maneuvers.

VII. CONCLUSIONS

We have presented PLATO, the first edge-assisted control architecture for chains of platoons. Thanks to a detailed simulator and to the analysis of its costs and benefits, we have shown that: (i) PLATO scales well for what concerns communication and computation overheads; (ii) surprisingly, edge-assisted control is possible under a wide range of network conditions, but it is critical to maintain the coherency of managers' instructions by keeping short delays between platoon and multi-platoon managers (up to a few tens of milliseconds), while vehicle to manager latency plays a minor role (at least below 250 ms); (iii) compact multi-platoons are convenient in terms of global and individual costs, and (iv) compactness is limited by safety and maneuvering constraints.

Our utility function is only a first attempt to model the utility of platooning in a comprehensive way. Other utility functions might be developed starting from ours, e.g., by modifying the importance of each component with an exponent factor or expressing all costs and benefits in monetary terms or according to a metric that expresses a societal

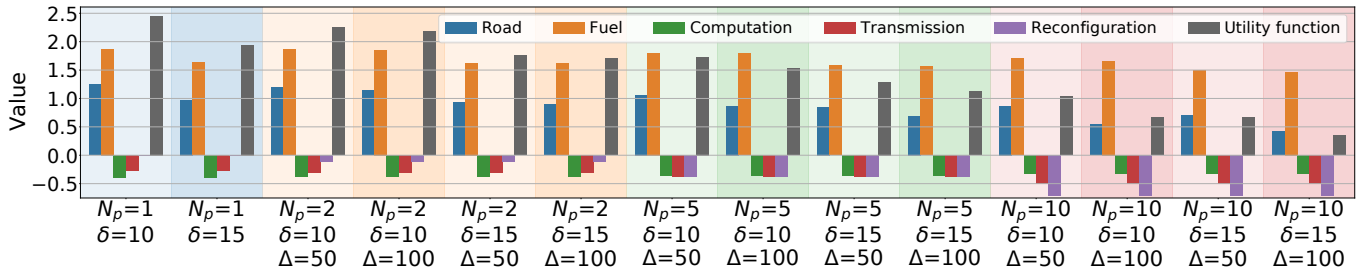


Fig. 7: Utility functions components contribution (log values)

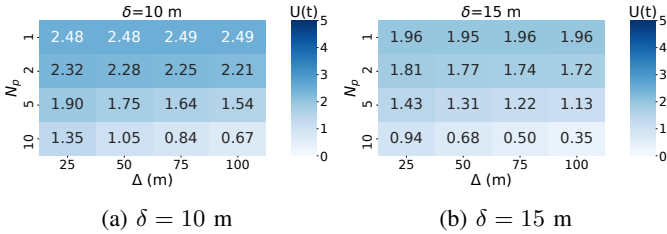


Fig. 8: Average utility function over t for different configurations of distances, δ and Δ , and number of platoons N_p

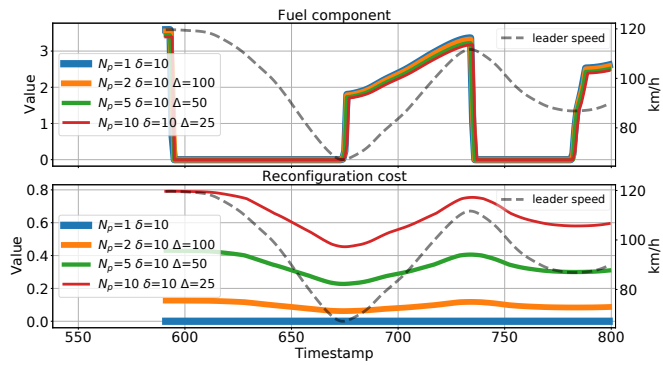


Fig. 9: Utility components over time (Part A of Fig. 5)

value (e.g., pollution, human stress level, etc.). In addition, since platooning lowers driver fatigue—because human interventions are required only to react to system anomalies—a specific metric for the evaluation of driving fatigue could be integrated in the utility function, which we leave for future work.

ACKNOWLEDGEMENTS

The work was supported by the Spanish Ministry of Science and Innovation grant PID2019-109805RB-I00 (ECID).

REFERENCES

- [1] 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.
- [2] 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.
- [3] C. Quadri, V. Mancuso, M. Ajmone Marsan, and G. P. Rossi, "Platooning on the edge," in *ACM MSWiM*, 2020.
- [4] 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.
- [5] 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.

- [6] 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.
- [7] M. Zabat, N. Stabile, S. Farascari, and F. Browand, "The Aerodynamic Performance Of Platoons: A Final Report," Institute of Transportation Studies, Research Reports, Working Papers, Proceedings qt8ph187fw, Institute of Transportation Studies, UC Berkeley, 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] S. T. Kaluva, A. Pathak, and A. Ongel, "Aerodynamic drag analysis of autonomous electric vehicle platoons," *Energies*, vol. 13, no. 15, 2020.
- [10] A. A. Alam, A. Gattami, and K. H. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning," in *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 306–311, 2010.
- [11] 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.
- [12] 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.
- [13] T. Zeng, O. Semiari, 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] H. Peng, D. Li, K. Abboud, H. Zhou, H. Zhao, W. Zhuang, and X. Shen, "Performance analysis of IEEE 802.11p dcf for multiplatooning communications with autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2485–2498, 2017.
- [20] P. Fernandes and U. Nunes, "Multiplatooning 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.
- [21] R. Rajamani, *Vehicle dynamics and control*, ch. 7. Springer, 2012.
- [22] 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.