Supporting Platooning Maneuvers through IVC: An Initial Protocol Analysis for the Join Maneuver

Michele Segata*†, Bastian Bloessl*, Stefan Joerer*, Falko Dressler*, Renato Lo Cigno†
*Computer and Communication Systems, Institute of Computer Science, University of Innsbruck, Austria
†Department of Information Engineering and Computer Science, University of Trento, Italy
{segata,bloessl,joerer,dressler}@ccs-labs.org
locigno@disi.unitn.it

***Abstract*—Driving vehicles in platoons has the potential to improve traffic efficiency, increase safety, reduce fuel consumption, and make driving experience more enjoyable. A lot of effort is being spent in the development of technologies, like radars, enabling automated cruise control following and ensuring emergency braking if the driver does not react in time; but these technologies alone do not empower real platooning. The initial idea of building dedicated infrastructures for platoons, has been set aside favouring the philosophy that foresees scenarios, where automated vehicles share the road with human-driven ones. This arises interesting new questions regarding the interactions between the two categories of vehicles. In this paper we focus on the analysis of interferences caused by non-automated vehicles during a JOIN maneuver. We define the application layer protocol to support the maneuver, together with situations that can prevent successful termination, and describe how they can be detected. The validity of the approach is proven by means of simulations, showing either that the maneuver can successfully be performed, or safely be aborted. Finally, we analyze the impact of the Packet Error Rate (PER) on the failure rate of the maneuver, showing that packet losses mainly affect the maneuver from a coordination point of view, rather than stability of the system, i.e., even at high loss rates, cars never violated a minimum safety distance.**

I. INTRODUCTION

Better road usage and increased safety will pass through the capability of vehicles to implement cooperative driving, *platooning* for short. Albeit recently there has been a strong focus on autonomous or semi-autonomous driving [1], where Inter-Vehicular Communication (IVC) is not needed, only platooning, which requires fully developed IVC, can guarantee improved road safety, while increasing the infrastructure usage and reducing fuel consumption [2], [3].

Platooning is much more than simple car following. Platoons must be built and split, vehicles must be able to join and leave platoons, the platoon leader must be changed, e.g., because the driver is tired or has reached his destination [4], [5]. All the possible maneuvers must be supported by a proper application level protocol, providing the communication primitives or Application Programming Interface (API) needed to implement them. Indeed, this is only the starting point, as the API must provide also the means to cope with impairments, unexpected situations, partially failing communications, interfering vehicles, and finally also the emergency maneuver to relinquish the vehicles' control to all the drivers safely in case there are no more the conditions to operate the platoon. The resulting IVC strategies need to be designed to support all these requirements. For all communications, we assume standard Dedicated Short Range Communications / Wireless Access in Vehicular Environment (DSRC/WAVE) radios using IEEE 802.11p [6] and beacon-based radio communication as defined for cooperative awareness applications by ETSI and IEEE [7].

The definition and possibly the standardization of an application level protocol for platooning is a formidable task. This paper presents an initial analysis and classification of elementary maneuvers needed to manage a platoon and explores the feasibility and performance of some simple test-cases implemented in the Veins simulation toolkit [8] including the analysis of some common impairments that leads to the necessity of aborting the maneuver, e.g., because a slow truck interferes with the platoon while a car is joining in the middle.

The key contribution of this work is threefold. First, we analyze the communication exchange needed between a platoon and a car that wants to join the platoon and define the state machines of an application level protocol to support the car to join the platoon in an arbitrary position, i.e., not only in the last position, but anywhere within the platoon. Next, we implement the protocol in Veins and show the feasibility using standard beaconing techniques, implementing the required reliable communications via application level acknowledgements; our claim is that this implementation is safer than using 802.11p unicast communications. Finally, the simulation results show that the definition and implementation of the protocol are correct and that it can effectively handle situations where human driven vehicles interfere with the maneuver, as well as bad channel conditions that lead to high packet loss rates.

II. RELATED WORK

The scientific community investigated different ways to perform maneuvers in an Automated Highway System (AHS), both with and without the infrastructure cooperation. One approach assuming infrastructure cooperation tackles the problem from a control theoretic point of view, defining the laws to control the vehicles during the maneuvers, together with higher layer mechanisms to the cars involved [9], [10]. The authors of [9] consider a join maneuver of two consecutive platoons driving on the same lane. They define the state machine to support the operation, the data that should be shared among the vehicles,

978-1-4799-4937-3/14/$31.00 ©2014 IEEE

and how to use such data to control the following platoon and join the leading one. In [10], instead, the authors focus on the join/leave maneuvers for a single car.

Other works focus on a high level approach, in particular by considering an agent-based system to perform inter-vehicle coordination [11]. The paper stresses the importance of teamwork, and the proposed approach assigns roles to the vehicles involved in the maneuver. For example, when performing a maneuver to separate from a platoon, the vehicle that wants to leave is defined as the *Splitter*, the one behind *Gap creator*, and other vehicles monitoring the situation *Safety observers*.

The authors of [12] focus more on networking by defining a round robin based protocol for information dissemination within a platoon. The protocol is tested by computing its theoretical maximum throughput rate, as well as using it for merging two platoons which come together at an in-ramp.

Another high level approach is presented in [13]. The authors describe a set of different communication patterns that can be used in order to exchange data while performing a maneuver. In a nutshell, the authors consider a *decentralized* strategy, where no communication is employed at all, a set of *distributed* communication strategies among the vehicles involved, and a *centralized* communication where the leader of the platoon is responsible for coordinating the entire maneuver. Furthermore, the authors define a set of controllers responsible for sending commands to the actuators of the vehicles. Each controller is dedicated to a particular situation, e.g., one to vehicle following, one to maneuvers, and so on. Additionally, one controller takes care of detecting and reacting to situations that prevent the successful execution of the maneuver, but no further details on how a particular fault should be detected and communicated to other parties are given.

Some fundamental studies focusing on fault detection have been carried out in the scope of the PATH project [14]. In [15], the authors develop a system able to perform fault diagnosis for a set of vehicles driving in a platoon. This work is mostly oriented on mechanical and electronic problems; for example, detecting faulty speed sensors, radars, brake actuators, and so on. Based on the type of fault, different types of actions can be performed using a dedicated network protocol for coordination [16]. Since in the PATH architecture no human driven vehicles are considered, problems related to non-automated vehicles interfering during maneuvers have not been addressed.

Recently, mixed highway scenarios have gained more attention [17]. The aim is to make platoons able to travel on public roads, avoiding the deployment of dedicated infrastructure. This poses new challenges that need to be addressed due to the presence of human drivers which might interfere with platooning operations.

The idea presented in [18] is that vehicles in a platoon should not simply detect an anomaly and react by disrupting the platoon, but also reorganize it to prevent hazards. In [19], instead, the authors study mechanisms to perform cooperative maneuvers (e.g., a lane change by an entire platoon) to avoid dangerous situation.

In this context, the challenge of defining an application level protocol that support the different maneuvers, seen as different applications, has not been tackled to the best of our knowledge. The identification of external events due to the presence of other road users, or to other impairments as communication faults, and the algorithms that the applications deploy to react to the situation are extremely important to make platooning safe and acceptable by the broad public.

A recent paper that starts considering these aspects is [20], where, if a communication fault is detected, it is signaled to the control system that switches from one particular Cooperative Adaptive Cruise Control (CACC) control law to another requiring less coordination. The authors focus only on pure following, but this work represent a necessary step towards safe deployment of this technology.

III. MANEUVERS AND SCENARIOS

To properly support platooning, a set of required maneuvers needs to be implemented. The first and most studied one is the FOLLOW maneuver, i.e., standard cruising, where interesting issues on multi-body control have to be solved and that represent the steady-state of a platoon. From a communications perspective FOLLOW can be realized with standard DSRC/WAVE beacons [7]; a working version implementing the platoon controller defined in [10] is available in an extension of Veins simulator we use for evaluation [8], [21].

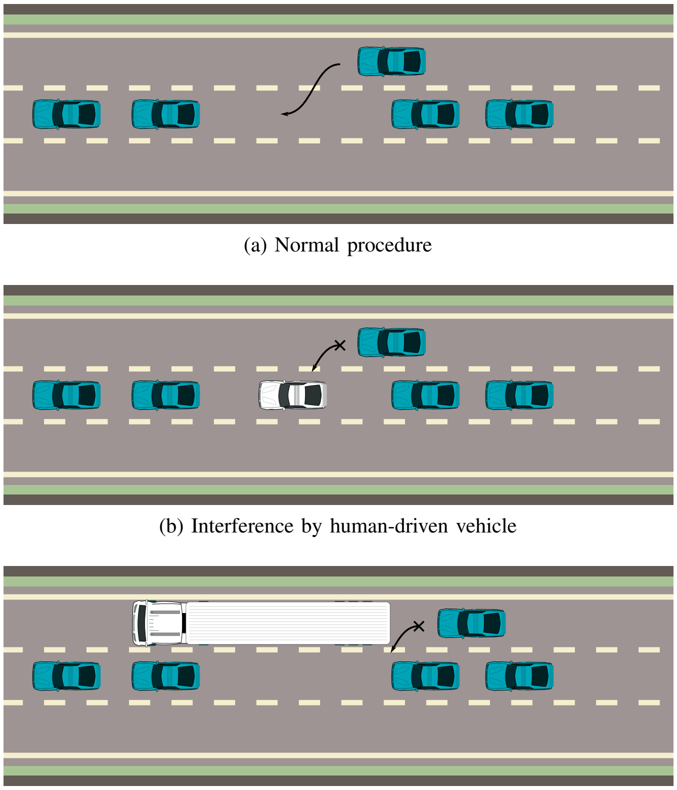
From a protocol point of view the maneuvers to form and to manage a platoon are more challenging, e.g., JOIN, LEAVE, MERGE, and SPLIT[1] require a more sophisticated coordination among cars than simply receiving beacons from the other cars in the platoon. Moreover, they have additional parameters, e.g., the position in the platoon where a car wants to JOIN, or can be combinations of simpler maneuvers. For instance, a SPLIT maneuver that wants to separate a large platoon of N cars in two smaller platoons of N_1 and N_2 cars, respectively ($N_1 + N_2 = N$) can be described as "opening a gap" between car N_1 and car $N_1 + 1$. This will be followed by the election of car $N_1 + 1$ as leader of the second platoon, which can also be interpreted as a sub-maneuver as it can be reused, for instance, when the leader has to leave and the second car in the platoon has to take its role.

As we already pointed out, it is beyond the scope of this paper to classify platooning maneuvers that are worth implementing, and also the analysis that splits them into elementary maneuvers that represent the primitives of an application protocol. We are instead interested in shedding some insight on platoons management and the communication challenges they pose, with a focus on "external threats", such as human driven vehicles interfering with the maneuvers and high packet loss rates "bursting" during the maneuver execution.

A. Scenarios for the Join Maneuver

As a representative of management maneuvers in this paper we focus on the JOIN procedure, assuming that one car joins the platoon in the middle, which is clearly more challenging than joining at the head or at the tail of the platoon. Besides

[1] As mentioned this work does not try to build an exhaustive and comprehensive set of maneuvers or primitives, thus this "list" is offered just by means of example.



(a) Normal procedure

(b) Interference by human-driven vehicle

(c) Interference by slower vehicle in front

Figure 1. Graphical sketch of different situations for the JOINMIDDLE maneuver. Automated cars shown in dark color.

considering the plain procedure, we also include in the protocol "escape" procedures, to handle situations when *i)* there are interferences by human-driven vehicles, and *ii)* there are communication problems. For the sake of simplicity the escape procedure is aborting the maneuver and either returning to normal platooning or, sometimes, splitting the platoon.

An example of a JOIN maneuver is shown in Figure 1. In the standard setup (Figure 1a), a vehicle creates a gap to let another one in. A human-driven vehicle, however, might change lane and get in the platoon (Figure 1b). This situation must be detected and reported to the high layer logic which should decide what is the best action to undertake. Furthermore, a slower vehicles may be encountered while approaching the platoon which prevents the joiner to conclude the maneuver (Figure 1c).

We specifically consider five scenarios that differentiate each other as follows:

- Scenario 0 (no interference): the platoon and the joining vehicle freely travel on the motorway. No vehicle disturbs the maneuver.
- Scenario 1 (far truck interference): the joining vehicle encounters a truck on the lane where it is trying to join, but the truck does not prevent the conclusion of the maneuver as it is far enough.
- Scenario 2 (close truck interference): as for Scenario 1, a slow truck obstructs the joining vehicle, but this time it is forced to abort to avoid a collision.
- Scenario 3 (car interference): while the vehicle is opening the gap to let the joiner in, a human-driven car gets in forcing to abort the procedure.
- Scenario 4 (network impairments): we use Scenario 0, but with increasing packet loss rate, to identify the limit of the maneuvering safety when communications are disturbed.

B. Vehicle Capabilities and Network Assumptions

We assume that vehicles are controlled and travel as envisioned in the SARTRE project [4], [5], [17]: drivers instruct the vehicle, which are otherwise entirely autonomous, through a Human Machine Interface (HMI). Actions like steering or touching the brakes disengage the CACC and lead to the platoon split. How this happens in a safe way, however, is out of the scope of this paper.

All platooning-capable vehicles are equipped with an IEEE 802.11p compliant device, a GPS receiver, and a radar. The CACC, in order to safely perform automated close-following, needs acceleration and speed values of a subset of vehicles in the platoon. Such subset depends on the design of the controller itself. We adopt the controller designed during the PATH project [10], where each vehicle requires acceleration and speed of the platoon's leader and the vehicle in front. Other designs, with different characteristics, are possible [20], but do not influence the maneuvers we focus on.

All protocols are implemented on top of standard broadcast beacons transmitted at 10 Hz as commonly required [22]. "Unicast" messages are obtained by identifying the intended recipient at the application level, with a proper tagging in messages, which however can be read by any other vehicle, adding redundancy and reliability to the system.

C. Impairments and Faults

The number of events that can interfere with platoon maneuvering are humongous, but here we only consider those envisaged in the five scenarios already described: the goal of this work is verifying the feasibility of automatic maneuvering controlled via a standard DSRC/WAVE vehicular network environment in a mixed scenario, and we do not pretend to make an exhaustive study.

We think that Scenario 1 and Scenario 2 (Figure 1c) can be very common in case of platooning cars, which travel faster than trucks. Note that whether the truck is equipped with communication devices or not is irrelevant: it will in any case interfere with the maneuver. We want to explore if implementing proper reactions to this situation, i.e., completing the maneuver if the truck is far enough or abort it if the truck is too close, is feasible and if the situations are distinguishable with the on-board sensing (the radar).

Also Scenario 3 can be fairly common, at least with high traffic densities. Moreover this case is more challenging, as the sensing systems of the car opening the joining gap (again the radar, but can be helped by other devices too), need to identify that the object in front is not part of the platoon, and the vehicle has to instruct the car joining the platoon that the maneuver cannot be completed. In this case we assume that the platoon splits, but other options can be considered, including re-forming the platoon again if the "intruder" leaves.

Obviously also communication failures can hamper maneuvering, and it must be checked that the maneuver can be aborted safely when information is lost, without endangering the vehicles occupants. As most communications happen with broadcast messages at the physical layer, we consider a very simple scenario of random packet loss, leaving for future work the investigation of more realistic network impairments/failures impact on maneuvers (e.g., network congestion due to the presence of too many platoons).

These are the issues that we consider in this work to design these first prototypes of application protocols for platoon maneuvers in mixed traffic.

IV. Join Application Protocol

Consider the JOIN maneuver, in particular with a car entering in the middle of the platoon. Three vehicles are "actively" involved in the procedure. The leader L, which is coordinating the maneuver, the follower F that is creating the gap in the platoon, and the joining vehicle M.

The entire procedure can be summarized as follows:

1) M discovers the existence of the platoon lead by L and sends a join request;
2) L, based on some criteria, decides whether to deny or accept the request. In case of a deny, a negative acknowledgement is sent and the procedure terminates. Otherwise, L computes the position where M has to join (in this case in front of F), and sends a positive acknowledgement to M including the identity of the two cars between which it has to join;
3) M approaches the insertion position using the information received from the vehicle in front of F. This can be done either automatically by feeding proper data into the CACC controller, or manually by the driver with the help of the interface, but we assume that after the join is accepted all the maneuver is automatic and cooperative;
4) Once in position, M notifies L that it is able to join;
5) L tells F to open a gap to let M in. In its beacons, F activates a flag to notify the vehicles behind it to use it as a temporary leader. Basically F and the vehicles behind form a second platoon to avoid instabilities in the case that F needs to brake for safety reasons[2];
6) F slows down and leaves space to M, notifying L when done;
7) L communicates M to move in;
8) M changes lane, closes the gap and notifies L;
9) L tells F to close the gap;
10) F closes the gap and informs L when done. Moreover, F disables the temporary leader flag in its beacons, so that vehicles behind F set their leader back to L;
11) L communicates the changes in the formation to all the vehicles.

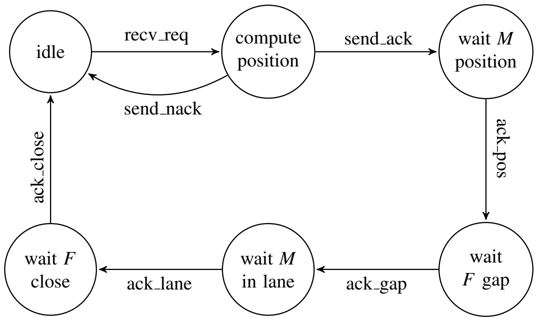
All packets sent for notification, i.e., to perform state changes, must be reliably transmitted at least to the vehicle that has the active role in the maneuver. This is obtained including in the broadcast beacons the identity of the intended recipient, which will return an application layer acknowledgement enabling the detection of lost packets and possibly triggering retransmissions. It is conceivable to achieve the same goal by using IEEE 802.11p unicast frames. This possibility, however, has the drawback that the other cars do not receive this message, thus, they are missing part of the information about the maneuver status. One can argue that IEEE 802.11p unicast packets can be overheard, thus other cars are aware too. However this would introduce many more complications and it is a non-standard operational mechanism which might not be accessible to applications. Furthermore, the usage of standard broadcast beacons leaves room for the implementation application layer reliability which is more flexible than the stiff retransmission policy of IEEE 802.11p unicast.

The state machines at the different vehicles that define this JOIN protocol are shown in Figure 2. We only represent the maneuver itself for the sake of clarity, without including all the details to detect faults and impairments and the actions taken to counter them: considering every possible fault or impairment is more a task for a standard specification than for a proof-of-concept prototype. In our implementation when the maneuvers cannot be completed as intended, it is simply aborted, i.e., M does not join the platoon. In some cases more sophisticated actions can be conceived to complete the maneuver in a different way, e.g., M joins the platoon at the tail instead of in the middle, but we think these are details that do not change the contribution of this work. The state machines in Figure 2 are almost self-explaining, as they are the straightforward representation of the 11 steps we have identified and explained before. The 'idle' state corresponds indeed to the steady state platooning for all the cars but M, which remains human driven until it has received the acknowledge from L. At the end of the procedures all car return to the steady state 'idle' platooning, thus for the joiner M entering the 'follow' state of this procedure means becoming a normal follower car.

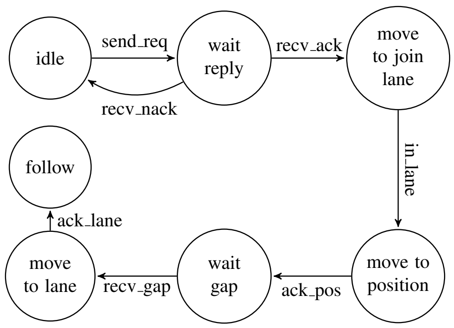
We now consider the case when a car interferes with the maneuver by entering in the platoon gap that has been opened by F. The state machine for the detection is shown in Figure 3a; the state 'status' can be any of the normal states of the protocol for car F (Figure 2c). 'Detect' and 'detect_danger' are events issued by the radar that detects a sudden change in distance, which is not coherent with the maneuver. Moreover, this distance will not be coherent with the GPS position broadcast in beacons by the car in front. Detecting these events will normally take 200 ms to 300 ms, which is negligible compared to cars' dynamics. 'Detect' means that an interfering car has been identified, but it is not posing any safety threats. The protocol enters a 'warning' state for some time; if the interferer leaves the platoon, then the procedure can be retrieved, otherwise after a timeout the maneuver is aborted. If a dangerous situation, e.g., the vehicle slows down posing an immediate threat, the maneuver is immediately aborted, in this case by forming two separate platoons (one lead by L and one by F), while the control of M is gently returned to the human driver.

The last case we consider is when M detects a vehicle in front while trying to get in the correct position to join the platoon. This can happen during the 'move to position' and the 'wait gap' states of the state machine in Figure 2b. To handle

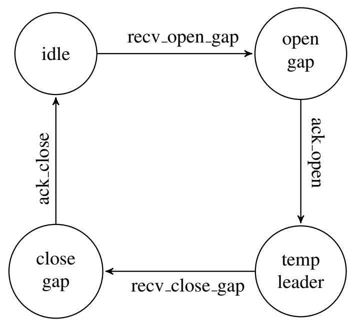
[2]The technical reason for this choice lies in the multi-body control technique adopted, for additional details we refer the reader to the original paper [10].



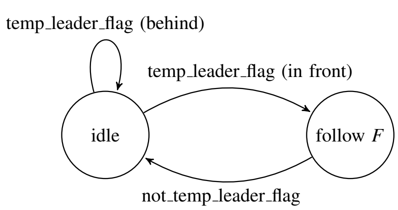
(a) Leader L



(b) Joining vehicle M



(c) Follower F

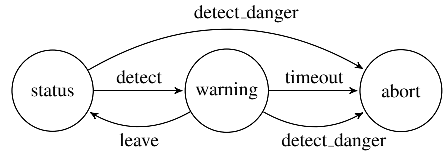


(d) Other platoon vehicles

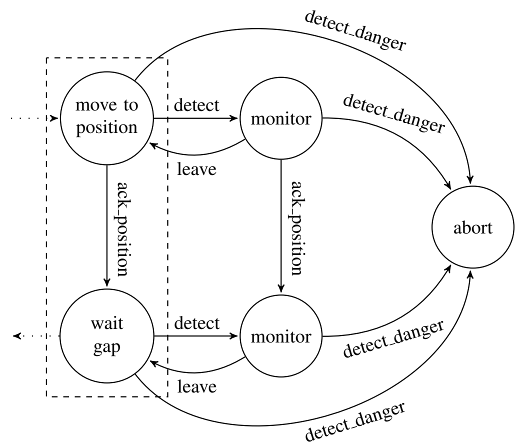
Figure 2. State machines for the vehicles involved in a JOINMIDDLE maneuver. No fault/misbehavior detection included.

this case we can extend the state machine of M as shown in Figure 3b, where the two states enclosed in the dotted box are the same states of Figure 2b. When M detects a vehicle in front, it first switches to the 'monitor' state. The radar can indeed detect objects which are up to $250\,\mathrm{m}$ distant [23] and hence also detects cars, which do not immediately interfere with the maneuver. Whenever a dangerous situation is detected, e.g., the Adaptive Cruise Control (ACC) is mandating to decelerate to avoid a collision, then the maneuver is aborted.

Please note that being in the 'monitor' state does not prevent to continue the maneuver. If M is able to move to the join



(a) Interfering vehicle in platoon



(b) Slow vehicle in front

Figure 3. State machines for the detection of interfering vehicles.

position, and the vehicle in front does not endanger maneuver's safety, it can continue waiting for F to open the gap and, in case, successfully complete the maneuver.

V. Evaluation

A. Simulation Setup

To evaluate the protocol, we implemented it into the platooning enabled extension of Veins [21] and tested it in the aforementioned five scenarios. To show the validity of our approach, we implement anomaly detection mechanism connected to basic countermeasure procedures. The parameters we used to configure the network simulator are shown in Table I. Notice that in this study we are interested in the analysis of the state machine and the detection of human-driven vehicles, rather than network faults. For this reason, we disregard complex channel models which consider multi-path fading, shadowing, and we do not generate network interferences by other vehicles. A complete analysis would clearly require to consider all possible problems together, but in this preliminary study we want to isolate the effects, rather than mixing them. Therefore, we perform a separated initial network analysis aimed to understand the effects of potential network impairments on maneuver's execution.

Figure 4 shows five snapshots taken from the simulator to represent the evolution of a maneuver in Scenario 1. The platoon leader is taken as reference point, so that the positions of the other vehicles, including the slow truck, give an idea of the maneuver evolution and its "safety margins".

In particular, to detect the presence of a slow vehicle in front, we exploit data obtained from the radar, and compute

Parameter	Value
Path loss model	Free space ($\alpha = 2.0$)
PHY/MAC model	IEEE 802.11p/1609.4 single channel (CCH)
Frequency	5.89 GHz
Bitrate	6 Mbit/s (QPSK R = 1/2)
Transmit power	20 dBm

Table I
NETWORK SIMULATION PARAMETERS.

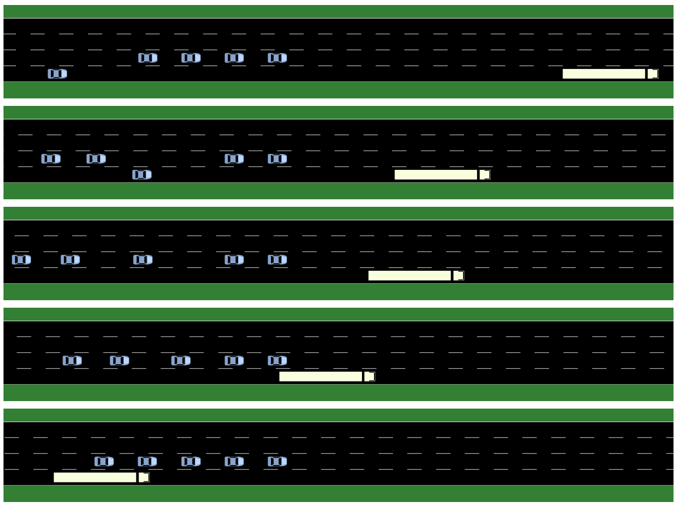


Figure 4. Temporal screenshots of Scenario 1 taken from the simulator. The screenshots cover a time lapse of roughly 30 s, and they are not uniformly spaced in time.

the acceleration that the ACC would apply. If the deceleration becomes greater than 3 m/s^2, then the system issues a warning. The countermeasure connected to this warning is to make the joiner M send an abort message to the leader, disabling CACC and switching back to ACC.

Regarding human-driven car intrusion, F continuously cross-checks radar distance with GPS distance from the vehicle which should be in front of it. If the discrepancy is greater than 2 m, then it is assumed that an unauthorized vehicle has entered the platoon. If such vehicle does not leave within 5 s, a warning is triggered. In such a case, F communicates to abort the procedure and splits the platoon in two.

In any case, no recovery procedure is attempted, e.g., if the human-driven vehicle leaves, platoons will remain split.

B. Maneuver Performance

We analyze the maneuver in the different scenarios from a vehicle dynamics point of view. Plots in Figure 5 show the dynamics of the vehicles in the platoon, plus the dynamics of the joiner M. The left column (Figures 5a, 5c, 5e and 5g), displays the GPS distances of the cars from the platoon leader, while the right column (Figures 5b, 5d, 5f and 5h), plots the distance from the vehicle in front as perceived by the radar. If no vehicle in front is detected by the radar no line is plotted.

We start with the analysis of Scenario 0 (Figures 5a and 5b). The plots show how the maneuver is correctly performed. The joiner M approaches the platoon from the side, and when in position, F and car 4 slow down to open a gap. M then gets in (as shown by the radar traces), the gap is slowly closed, and the procedure terminates.

The maneuver succeeds also in Scenario 1 (Figures 5c and 5d). This time, the joiner M detects a truck in front, as shown by the radar trace, but it is far enough to let M terminate the procedure.

In Scenario 2 (Figures 5e and 5f) instead, M reaches the join position, F starts to open the gap, but M has to abort to avoid a collision. At this time, M switches to ACC and remains behind the truck, while F closes the gap and the platoon continues to drive as before.

Finally, in Scenario 3 (Figures 5g and 5h), when F opens the gap, a human-driven vehicle moves to the platoon's lane. F communicates to abort the procedure, splitting the platoon in two sub-platoons. M switches to ACC and slows down, remaining in the side lane.

These results show how the protocol can be easily extended to detect and react to anomalies in the procedure. Using the same approach, we can develop the state machines for other maneuvers, analyze possible weaknesses, and study how to tackle them.

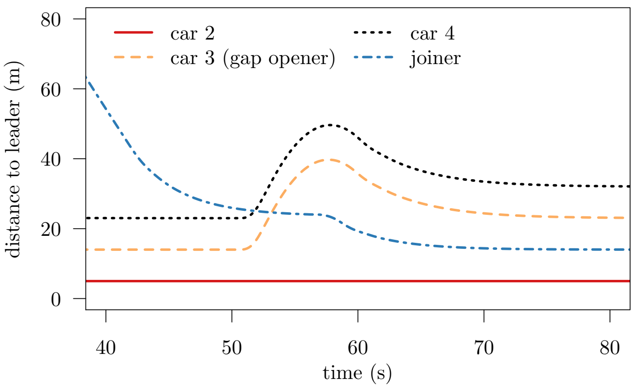
C. Impact of Packet Loss

As second contribution, we study the impact of packet loss on maneuver's success. We introduce independent Bernoullian random losses and observe the procedure failure rate. When investigating wireless networks, specially in the case of Vehicular Ad Hoc Networks (VANETs), considering independent packet losses is an unrealistic assumption, as losses are correlated in time and space due to interferences and MAC layer mechanisms. As previously stated, however, in this preliminary analysis we want to isolate the effects, before moving to an in depth study considering all possible faults and impairments together. In order to reach the Packet Error Rates (PERs) we are interested to (i.e., up to 50 %), we would need a traffic density that would cause the maneuver to fail because of other vehicles, thus making it difficult to analyze it from a network perspective.

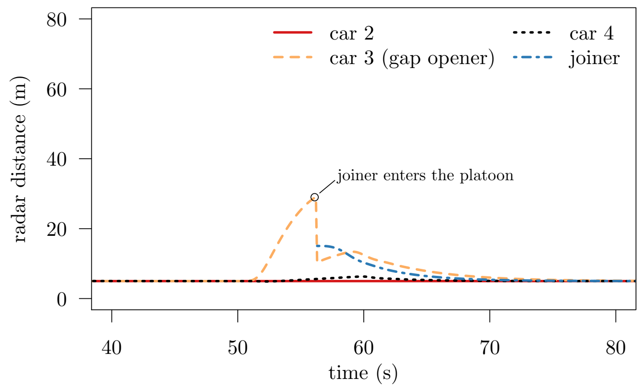
Network problems are treated as follows: the maneuver is aborted whenever a unicast packet is not acknowledged for three times in a row, or the measured radar distance is smaller than 2.5 m, i.e., half of the target intra-platoon distance. Here we consider only Scenario 0, to understand when the maneuver needs to be aborted due to network problems. We start from a PER of 1 %, and we increase it up to 50 %. To obtain confidence intervals, the experiments are repeated between 30 to 100 times.

The results are shown in Figure 6, where maneuver's failure rate is plotted as function of PER with 95 % confidence intervals. For PERs smaller than 15 %, the failure rate is negligible, if not null. As expected, failure rate increases with PER, being around 50 % for a PER of 35 %, and up almost 100 % for a PER of 50 %.

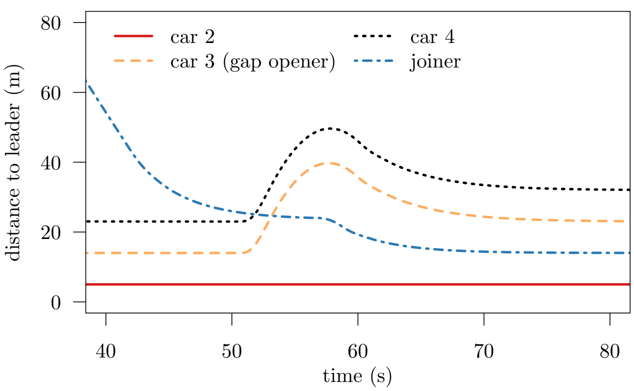
Analyzing the reason leading to the failures we observed that the maneuver is always aborted because of missing acknowledgements for protocol messages, and not because of reduced distance that hampers safety. Even at packet error rates higher than 30 %, in the cases when no acknowledgements are lost, M and F are able to safely complete the maneuver. This suggests that the CACC controller described in [10] is robust to packet losses, which is a further indication that platoons



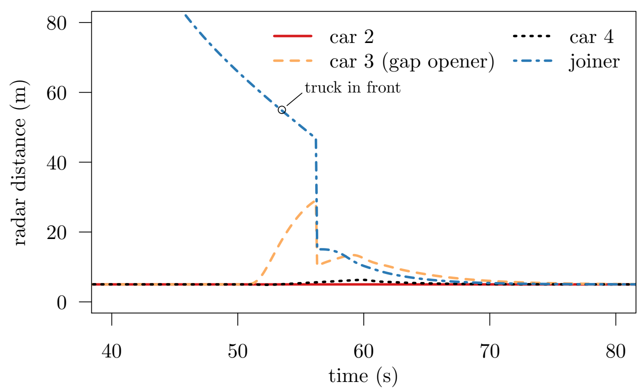
(a) Scenario 0, GPS distance from leader



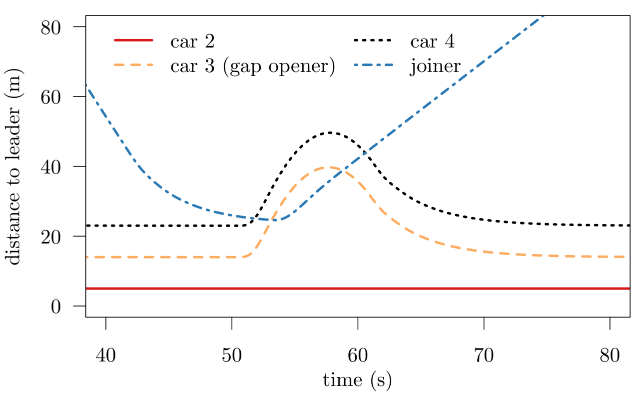
(b) Scenario 0, radar distance measured by car x



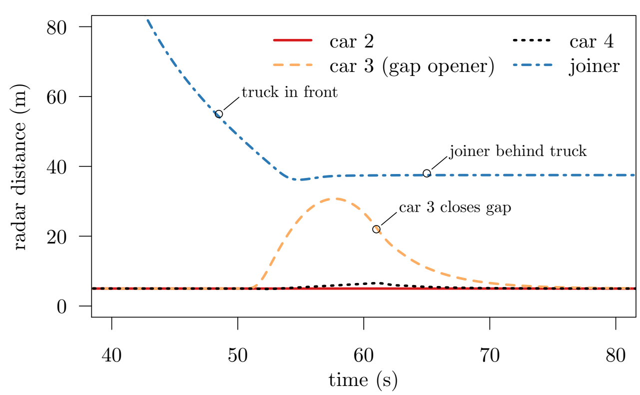
(c) Scenario 1, GPS distance from leader



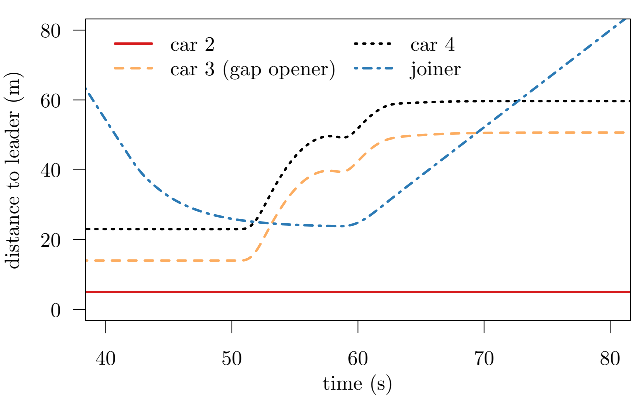
(d) Scenario 1, radar distance measured by car x



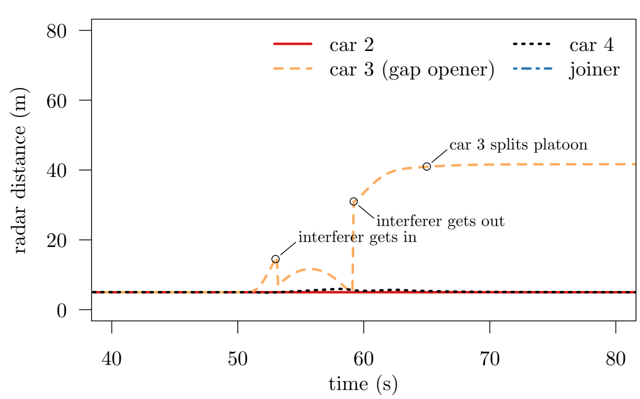
(e) Scenario 2, GPS distance from leader



(f) Scenario 2, radar distance measured by car x



(g) Scenario 3, GPS distance from leader



(h) Scenario 3, radar distance measured by car x

Figure 5. Vehicles dynamics for the different scenarios: left plots show the GPS distance to the leader; right plots show the measured radar distance.

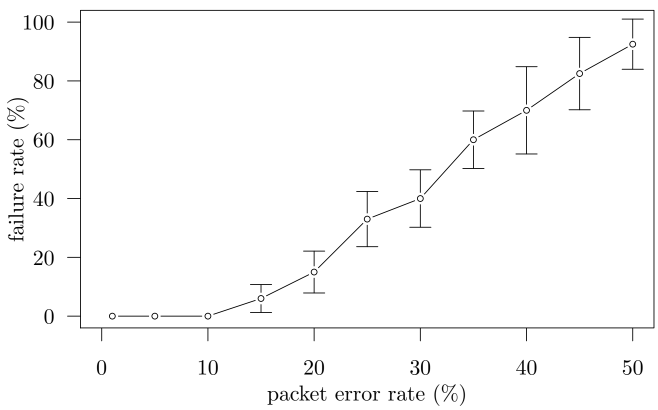


Figure 6. Scenario 4, failure rate of JOIN maneuver as function of packet error rate with 95 % confidence intervals.

can be run safely on roads. Clearly, a sudden deceleration of the leader or a bigger platoon might worsen the situation, but determining whether a CACC can still safely be operated in such conditions requires a dedicated study, and it is out of the scope of this paper.

VI. Conclusion and Future Work

Platooning in normal roads, where standard cars coexist semi-autonomous vehicles and cooperative driving ones, is beginning to appear a reachable goal. One of the key open issues is the capability of platoons to maneuver in presence of external disturbances, as vehicles moving slower than the platoon, human drivers interfering with maneuvers, communication impairments. This work has proposed and analyzed an application level protocol to support JOIN maneuver in several scenarios, showing that relatively simple logic can support complex maneuvers as letting a vehicle join a platoon in the middle of the same, while guaranteeing that in case of interference the maneuver can be aborted safely, either splitting the platoon or re-composing its original formation without the desired joining vehicle.

Detailed simulation results show that the platoon remains stable also with high packet loss rates, and only harsh networking conditions with packet loss rate larger than 10 % lead sometimes to maneuver abortion. Abortion becomes more frequent as the packet loss is increased. During maneuvers the platoon can safely be operated even with extreme cases as 50 % of packets are lost, when the maneuver almost never succeeds, but no accidents are recorded in the simulations.

Future work includes the development of a complete and fail-safe protocol, able to handle several (and possibly concurrent) problems. To ensure the feasibility of the approach, the protocol will need to be split in several sub-parts, in order to make design and verification easier thanks to a divide-et-impera approach. This will enable re-usage of such sub-parts by combining them to perform more sophisticated platooning maneuvers.

References

[1] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, "Autonomous Ground Vehicles–Concepts and a Path to the Future," *Proceedings of the IEEE*, vol. 100, no. 13, pp. 1831–1839, May 2012.

[2] A. Alam, A. Gattami, and K. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning," in *13th IEEE International Conference on Intelligent Transportation Systems (ITSC 2010)*, 2010, pp. 306–311.

[3] A. Davila, E. Aramburu, , and A. Freixas, "Making the best out of aerodynamics: Platoons," in *SAE Technical Paper 2013-01-0767*, 2013.

[4] C. Bergenhem, H. Pettersson, E. Coelingh, C. Englund, S. Shladover, and S. Tsugawa, "Overview of platooning systems," in *19th ITS World Congress*, Vienna, Austria, October 2012.

[5] E. Chan, P. Gilhead, P. Jelinek, P. Krejci, and R. T., "Cooperative control of sartre automated platoon vehicles," in *19th ITS World Congress*, Vienna, Austria, October 2012.

[6] "Wireless Access in Vehicular Environments," IEEE, Draft Standard P802.11p/D10.0, January 2010.

[7] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," European Telecommunications Standards Institute, Tech. Rep. 302 637-2 V1.3.0, August 2013.

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

[9] R. Horowitz and P. Varaiya, "Control Design of an Automated Highway System," *Proc of the IEEE*, vol. 88, no. 7, pp. 913–925, July 2000.

[10] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, July 2000.

[11] S. Hallé, J. Laumonier, and B. Chaib-Draa, "A Decentralized Approach to Collaborative Driving Coordination," in *IEEE Intelligent Transportation Systems Conference (ITSC 2004)*. Washington, DC: IEEE, October 2004, pp. 453–458.

[12] T. Sakaguchi, A. Uno, S. Kato, and S. Tsugawa, "Cooperative Driving of Automated Vehicles with Inter-Vehicle Communications," in *IEEE Intelligent Vehicles Symposium (IV 2000)*. Dearborn, Michigan: IEEE, October 2000, pp. 516–521.

[13] F. Michaud, P. Lepage, P. Frenette, D. Letourneau, and N. Gaubert, "Coordinated Maneuvering of Automated Vehicles in Platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 437–447, December 2006.

[14] S. Shladover, "PATH at 20 – History and Major Milestones," in *IEEE Intelligent Transportation Systems Conference (ITSC 2006)*, Toronto, Canada, September 2006, pp. 22–29.

[15] R. Rajamani, A. Howell, C. Chen, J. Hedrick, and M. Tomizuka, "A Complete Fault Diagnostic System for Automated Vehicles Operating in a Platoon," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 4, pp. 553–564, July 2001.

[16] D. Godbole, J. Lygeros, E. Singh, A. Deshpande, and A. Lindsey, "Communication Protocols for a Fault-tolerant Automated Highway System," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 5, pp. 787–800, September 2000.

[17] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson, "Challenges of Platooning on Public Motorways," in *World Congress on Intelligent Transport Systems*, Busan, Korea, October 2010.

[18] C. Guo, N. Wan, S. Mita, and M. Yang, "Self-defensive Coordinated Maneuvering of an Intelligent Vehicle Platoon in Mixed Traffic," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC 2012)*. Anchorage, AK: IEEE, September 2012, pp. 1726–1733.

[19] S. Lam and J. Katupitiya, "Cooperative Autonomous Platoon Maneuvers on Highways," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2013)*. Wollogong, Australia: IEEE, July 2013, pp. 1152–1157.

[20] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful Degradation of CACC Performance Subject to Unreliable Wireless Communication," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. The Hague, The Netherlands: IEEE, October 2013.

[21] M. Segata, F. Dressler, R. Lo Cigno, and M. Gerla, "A Simulation Tool for Automated Platooning in Mixed Highway Scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 4, pp. 46–49, October 2012.

[22] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and Experimental Evaluation of Cooperative Adaptive Cruise Control," in *IEEE Int. Conference on Intelligent Transportation Systems (ITSC 2011)*. Washington, DC: IEEE, October 2011, pp. 260–265.

[23] Bosch LRR3 Radar Datasheet. Last Visited: January 28, 2014. [Online]. Available: http://www.bosch-automotivetechnology.com/media/db_application/downloads/pdf/safety_1/en_4/lrr3_datenblatt_de_2009.pdf