Towards Communication Strategies for Platooning: Simulative and Experimental Evaluation

Michele Segata, Bastian Bloessl, Stefan Joerer, Christoph Sommer, Mario Gerla, Renato Lo Cigno and Falko Dressler

Abstract—Platooning, the idea of cars autonomously following their leaders to form a road train, has huge potentials to improve traffic flow efficiency and, most importantly, road traffic safety. Wireless communication is a fundamental building block – it is needed to manage and to maintain the platoons. To keep the system stable, strict constraints in terms of update frequency and communication reliability must be met. We investigate different communication strategies by explicitly taking into account the requirements of the controller, exploiting synchronized communication slots as well as transmit power adaptation. As a baseline, we compared the proposed approaches to two state of the art iterative systems driving applications, even in very crowded freeway scenarios. Our simulation models have been parameterized and validated by means of real-world experiments. Our results demonstrate that the combination of synchronized communication slots with transmit power adaptation is perfectly suited for cooperative driving applications, even in very crowded freeway scenarios.

Index Terms—Vehicles, Networks, Automated highways, Cooperative systems

I. INTRODUCTION

Since research on Vehicular Ad Hoc Networks (VANETs) started more than twenty years ago, many applications based on Inter-Vehicular Communication (IVC) have been proposed; only few have been implemented and tested in field operational tests [1]. Cooperative Adaptive Cruise Control (CACC), which has become widely known as platooning, is among these applications. Being investigated since the eighties, e.g., within the PATH project [2], it is still an active topic due to the challenging problems it raises. One of the core reasons behind such a huge interest are the benefits that platooning could provide once deployed.

Platooning can enhance the travel experience covering consumption issues, safety, and comfort: First, it has the potential to improve the traffic flow and to reduce the fuel consumption, reducing jams on freeways and decreasing pollution [3], [4]. Second, platooning can improve drivers’ safety if a system fault is less likely than a human error, which is the main cause of accidents [4]. Last but not least, a vehicle autonomously following its leaders permits the driver to relax, as shown in the recent SARTRE project [5].

From a research point of view, platooning is extremely challenging, as it involves several research fields – including control theory, communications, vehicle dynamics, and traffic engineering. From a communication perspective, the main reasons are the requirements in terms of latency and reliability. For what concerns networking, any controller designed for supporting platooning, such as the CACCs in [6], [7], needs frequent and timely information about vehicles in the platoon to avoid instabilities which might lead to crashes. For this reason, CACC is often cited as one of the most visionary applications of IVC [8], [9].

Looking at the state of the art in IVC protocol design, we see that many protocols are focusing on cooperative awareness. This is particularly the case for all protocols investigated in large scale field operational tests. In the U.S. and Europe, the IEEE DSRC/WAVE stack [10] and its ETSI counterpart, the ITS-G5 stack [11], [12], dominate all practical tests.

The biggest advantage of the mentioned protocol stacks is their ability to provide cooperative awareness while keeping the network load reasonable. Alternative approaches presented in the literature argue that more aggressive approaches such as Dynamic Beaconing (DynB) [13], [14] are needed to support traffic safety applications. A platooning system has a recommended information update frequency of 10 Hz [7]; thus, delays in the order of several hundred milliseconds due to congestion control mechanisms cannot be tolerated. Whether these communications requirements can be satisfied by the plain DSRC/WAVE or ETSI ITS-G5 stacks is still unclear, and further work is needed before platooning can become a reality.

In this article, which extends our contribution presented at [15], we study the suitability of state of the art beaconing based cooperative awareness protocols for platooning and highlight the challenges that are still open, proposing and investigating communication protocols that can help tackle them. We only consider platoons of homogeneous cars, and we do not take into account mixed platoons made of cars, trucks, buses, or vans. We investigate the reliability of wireless communications for platooning under high channel load imposed by a large number of platoons, showing how the proposed approaches support applications’ needs, and how they compare to two adaptive beaconing solutions: The current ETSI standard Decentralized Congestion Control (DCC) [11], [12], [16] and DynB [14].

Based on these insights, we explore design options for communication strategies suitable for the platooning application,
and finally derive a novel set of communication protocols that are able to build upon the standard IEEE 802.11p protocol. As a baseline for an extensive set of simulation studies, we report on an experimental validation of the simulation model, using a platoon of four cars equipped with IEEE 802.11p compliant devices. Our results clearly indicate that a combination of synchronized communication slots with transmit power adaptation performs best in our platooning scenario. This even holds for very dense network scenarios such as on a crowded freeway.

Our main contributions can be summarized as follows:

- We first describe in detail the CACC controller we consider in this work and present its main features and benefits (Section III).
- We define a set of different communication strategies, specifically taking into account CACC controller requirements (Section IV).
- In order to calibrate our simulation models, we performed a measurement campaign on the road using four cars driving in a platoon (Section V).
- We compare the proposed approaches against state of the art adaptive beaconing strategies from a network and an application layer perspective, showing the substantial benefits of our approaches for platooning (Section VI). In this analysis we assume the communication channel to be dedicated to the platooning application. We then relax this assumption and test the impact of human driven vehicles using DCC on our approach and vice versa. Finally, we test the controller in an emergency braking scenario under different message generation rates to understand the real requirements of the CACC we consider.

In the remainder of the paper, to avoid ambiguity, we use the terms collision and congestion when referring to network, and crash and traffic jam when referring to vehicles.

II. RELATED WORK

A. Cooperative Adaptive Cruise Control

The platooning research community focused initially on the problems connected to the automated control of vehicles, because the design of such a system is a non-trivial task. Indeed, the characteristic which makes a CACC different from a standard Adaptive Cruise Control (ACC) is the capability to closely follow the car in front by making use of wireless links in order to communicate with nearby vehicles. A standard ACC exploits only data provided by the radar, thus the distance and the relative speed to the vehicle in front. Such a system must keep a safety gap in the order of 1 to 2 s [7], [17], making it unsuitable for close following as required by platooning. Smaller inter-vehicle gaps would make the system unstable and might lead to crashes [17].

In the literature we can find several CACC controllers, each of those employing different communication patterns and having different characteristics. The CACC designed in [7] makes use of communication only between direct followers. In this case, the distance that can be maintained by the controller has to be speed-dependent as for ACC. The headway time, however, can range from 0.5 to 0.7 s, much smaller than for a standard ACC. Another type of controller uses data communicated from both the vehicle in front and the platoon leader [6]. The benefit is that the system can be proven to be stable under a constant-spacing policy, i.e., the inter-vehicle distance does not need to be speed-dependent. This means that the inter-vehicle gap can be fixed and chosen in meters and not in seconds as for a standard ACC or the CACC in [7]. For example, the Field Operational Tests (FOTs) in the PATH and SARTRE projects [5], [6] used this kind of controller, and the distances chosen in the experiments were between 5 and 7 m. Other solutions can configure the logical topology to adapt to network conditions [18], [19].

CACC controllers have been investigated since the beginning by the pioneering projects PATH [2] and Auto21 CDS [20], but they are still under continuous improvement either by academic research [7], [18], [19] or by car manufacturers, as in the SARTRE project [5]. What differentiates pioneering projects from recent studies is the “philosophy”. In the case of PATH or Auto21 CDS, platoons were designed to run on dedicated freeways, managed by a centralized system [21]. The idea in SARTRE, instead, is that platoons form autonomously and can travel on public motorways mixed with human driven vehicles. In both cases, network conditions are a major concern: 802.11-based networks can suffer high packet loss ratios even in moderate channel load conditions, and given the frequent updates needed by the CACC to ensure the platoon stability [17], the impact of the network performance on the safety of the overall system is non-marginal.

B. IVC Protocols for Cooperative Awareness

Cooperative awareness is among the most prominent VANET applications. Broadcast-based IVC, or beaconing, has been identified as the underlying protocol primitive, and has been investigated in detail by the vehicular networking community. The consensus is to periodically send beacons to all vehicles in communication range in order to improve cooperative awareness in general. ETSI defined Cooperative Awareness Messages (CAMs) for this purpose, supported by Decentralized Environmental Notification Messages (DENMs) for event triggered safety warnings. This line of research is still featuring very diverse proposals.

Most recently, ETSI ITS-G5 announced a new standard taking into consideration the network dynamics and the need for congestion control, DCC [11], [12], [16], which features a variety of protocol variants. In the scope of this paper, we consider the complete ETSI CAM generation algorithm which includes Transmit Rate Control (TRC), Transmit Power Control (TPC), Transmit Datarate Control (TDC), and DCC Sensitivity Control (DSC) mechanisms. Each algorithm component controls beacon generation rate, transmit power, physical layer datarate, and CCA-threshold, respectively. Furthermore, the latest release includes a set of vehicle dynamics-based rules for CAM triggering.

A state machine drives each component of the algorithm. The active state depends on the currently observed channel busy ratio, which is the amount of time the channel was sensed as busy by the physical layer. In this work we consider the three-state state machine designed for the Control
Channel (CCH), i.e., the channel designated to CAMs [12]. The state change decision is taken by monitoring the busy changes in the state of the vehicle – these can of course for the R

The minimum interval is restricted to 95 s

that regulates packet interval rules for each DCC state. DPs are numbered from 32

streams in the access, the network, and the transport layers.

respectively. Moreover, the minimum CAM generation interval and maximum CAM generation intervals, i.e.,

is computed as

be immediately sent. Moreover, if the CAM is triggered due to the dynamics-dependent conditions, the protocol must schedule three consecutive CAMs with an interval equal to the time elapsed since the last CAM generation. The packet interval must be reset to the maximum (i.e., 1 s) when all the repetitions are sent.

Another approach, DynB [13], [14], tries to maintain network load at a fixed, predefined value. Similar to DCC, DynB monitors the channel busy ratio and computes the interval to be used for sending the next beacon accordingly, but adapts more aggressively to the current channel conditions. More formally, let N be the number of neighbors computed using frames received from nearby vehicles, I_{des} the desired (i.e., the minimum) beacon interval, and let b_t and b_{des} be the measured and the desired bus ratio, respectively. The beacon interval I is computed as

If any of the aforementioned conditions is met and the minimum packet interval T_{off} has elapsed, a CAM should be immediately sent. Moreover, if the CAM is triggered due to the dynamics-dependent conditions, the protocol must schedule three consecutive CAMs with an interval equal to the time elapsed since the last CAM generation. The packet interval must be reset to the maximum (i.e., 1 s) when all the repetitions are sent.

Another approach, DynB [13], [14], tries to maintain network load at a fixed, predefined value. Similar to DCC, DynB monitors the channel busy ratio and computes the interval to be used for sending the next beacon accordingly, but adapts more aggressively to the current channel conditions. More formally, let N be the number of neighbors computed using frames received from nearby vehicles, I_{des} the desired (i.e., the minimum) beacon interval, and let b_t and b_{des} be the measured and the desired bus ratio, respectively. The beacon interval I is computed as

be extended according to new applications’ requirements. In particular the conditions are the following:

• The absolute difference between last sent heading and current heading exceeds 4°;
• The distance between last sent position and current position exceeds 4 m;
• The absolute difference between last sent speed and current speed exceeds 0.5 m/s.

If any of the aforementioned conditions is met and the minimum packet interval T_{off} has elapsed, a CAM should be immediately sent. Moreover, if the CAM is triggered due to the dynamics-dependent conditions, the protocol must schedule three consecutive CAMs with an interval equal to the time elapsed since the last CAM generation. The packet interval must be reset to the maximum (i.e., 1 s) when all the repetitions are sent.

Another approach, DynB [13], [14], tries to maintain network load at a fixed, predefined value. Similar to DCC, DynB monitors the channel busy ratio and computes the interval to be used for sending the next beacon accordingly, but adapts more aggressively to the current channel conditions. More formally, let N be the number of neighbors computed using frames received from nearby vehicles, I_{des} the desired (i.e., the minimum) beacon interval, and let b_t and b_{des} be the measured and the desired bus ratio, respectively. The beacon interval I is computed as

be extended according to new applications’ requirements. In particular the conditions are the following:

• The absolute difference between last sent heading and current heading exceeds 4°;
• The distance between last sent position and current position exceeds 4 m;
• The absolute difference between last sent speed and current speed exceeds 0.5 m/s.

If any of the aforementioned conditions is met and the minimum packet interval T_{off} has elapsed, a CAM should be immediately sent. Moreover, if the CAM is triggered due to the dynamics-dependent conditions, the protocol must schedule three consecutive CAMs with an interval equal to the time elapsed since the last CAM generation. The packet interval must be reset to the maximum (i.e., 1 s) when all the repetitions are sent.

Another approach, DynB [13], [14], tries to maintain network load at a fixed, predefined value. Similar to DCC, DynB monitors the channel busy ratio and computes the interval to be used for sending the next beacon accordingly, but adapts more aggressively to the current channel conditions. More formally, let N be the number of neighbors computed using frames received from nearby vehicles, I_{des} the desired (i.e., the minimum) beacon interval, and let b_t and b_{des} be the measured and the desired bus ratio, respectively. The beacon interval I is computed as

be extended according to new applications’ requirements. In particular the conditions are the following:

• The absolute difference between last sent heading and current heading exceeds 4°;
• The distance between last sent position and current position exceeds 4 m;
• The absolute difference between last sent speed and current speed exceeds 0.5 m/s.

If any of the aforementioned conditions is met and the minimum packet interval T_{off} has elapsed, a CAM should be immediately sent. Moreover, if the CAM is triggered due to the dynamics-dependent conditions, the protocol must schedule three consecutive CAMs with an interval equal to the time elapsed since the last CAM generation. The packet interval must be reset to the maximum (i.e., 1 s) when all the repetitions are sent.

Another approach, DynB [13], [14], tries to maintain network load at a fixed, predefined value. Similar to DCC, DynB monitors the channel busy ratio and computes the interval to be used for sending the next beacon accordingly, but adapts more aggressively to the current channel conditions. More formally, let N be the number of neighbors computed using frames received from nearby vehicles, I_{des} the desired (i.e., the minimum) beacon interval, and let b_t and b_{des} be the measured and the desired bus ratio, respectively. The beacon interval I is computed as

be extended according to new applications’ requirements. In particular the conditions are the following:

• The absolute difference between last sent heading and current heading exceeds 4°;
• The distance between last sent position and current position exceeds 4 m;
• The absolute difference between last sent speed and current speed exceeds 0.5 m/s.

If any of the aforementioned conditions is met and the minimum packet interval T_{off} has elapsed, a CAM should be immediately sent. Moreover, if the CAM is triggered due to the dynamics-dependent conditions, the protocol must schedule three consecutive CAMs with an interval equal to the time elapsed since the last CAM generation. The packet interval must be reset to the maximum (i.e., 1 s) when all the repetitions are sent.

Another approach, DynB [13], [14], tries to maintain network load at a fixed, predefined value. Similar to DCC, DynB monitors the channel busy ratio and computes the interval to be used for sending the next beacon accordingly, but adapts more aggressively to the current channel conditions. More formally, let N be the number of neighbors computed using frames received from nearby vehicles, I_{des} the desired (i.e., the minimum) beacon interval, and let b_t and b_{des} be the measured and the desired bus ratio, respectively. The beacon interval I is computed as

be extended according to new applications’ requirements. In particular the conditions are the following:

• The absolute difference between last sent heading and current heading exceeds 4°;
• The distance between last sent position and current position exceeds 4 m;
• The absolute difference between last sent speed and current speed exceeds 0.5 m/s.
As an example, Lei et al. [26] showed the impact of different packet loss rates on the string stability of CACC, considering a controller with constant time headway policy. Fernandes and Nunes [9] analyzed strategies to improve communications reliability considering five different protocols, all based on Time Division Multiple Access (TDMA). Furthermore, they proposed a dynamic adaptation of CACC parameters to cope with different situations. Böhm et al. instead analyzed the co-existence of CAMs (used for platooning) with DENMs, showing how the choice of different MAC layer priority classes for the two categories heavily affects the effectiveness of data dissemination [27]. From a communication point of view, some infrastructure might be used to make transmissions more efficient [28], or additional channels can be incorporated for better scalability [29]. Other communication technologies have also been analyzed. For example, Abualhoul et al. [30] proposed the use of visible light communication in order to communicate efficiently [28], or additional channels can be incorporated for data dissemination [27]. From a communication point of view, all these papers point out that the integration with the standardized cooperative awareness applications is very challenging and needs further investigations. In this paper, we propose a set of communication strategies explicitly taking the requirements of the CACC controller into account. This opens up new opportunities for the integration of cooperative awareness oriented beaconing with application tailored protocols.

III. CONTROLLER MODEL

In this study we consider the discretized version of the CACC controller detailed in [17]. The model assumes that each vehicle i (with i being its position in the platoon) knows the position $x_{i-1}$, the speed $\dot{x}_{i-1}$, and the acceleration $\ddot{x}_{i-1}$ of the preceding vehicle, as well as the speed $\dot{x}_0$ and the acceleration $\ddot{x}_0$ of its platoon leader. The distance to the preceding vehicle can be obtained using a radar (and to some extent also variations of the distance), while all other parameters must be conveyed through wireless communications.

Let $\Delta_t$ be the time interval of the sampled system, $n$ the current sampling step (or discrete time). Processing time is negligible compared to $\Delta_t$. For each vehicle i

$$\dot{x}_i[n] = (x_i[n] - x_i[n-1]) / \Delta_t$$

$$\ddot{x}_i[n] = (\dot{x}_i[n] - \dot{x}_i[n-1]) / \Delta_t$$

$$\Delta_t (x_i[n] - 2x_i[n-1] + x_i[n-2]) / \Delta_t^2$$

and the CACC control law computes the control input (i.e., the desired acceleration) to maintain the platoon as:

$$u_i[n] = \alpha_1 u_{i-1}[n] + \alpha_2 u_0[n] + \alpha_3 \ddot{x}_i[n] + \alpha_4 (\dot{x}_i[n] - \dot{x}_0[n]) + \alpha_5 \varepsilon_i[n]$$

where

$$\varepsilon_i[n] = x_i[n] - x_i[n-1] + l_{i-1} + \text{gap}_{\text{des}}$$

$$\ddot{x}_i[n] = \dot{x}_i[n] - \ddot{x}_i[n-1].$$

In Equation (7), $l_{i-1}$ is the length of the preceding vehicle, while $\text{gap}_{\text{des}}$ is the desired inter-vehicle gap, in meters. The distance to the front vehicle is given by the radar, which always provides up-to-date information with negligible error, so the term $x_i[n] - x_i[n-1] + l_{i-1}$ is considered to be exact. The terms $u_{i-1}[n]$, $u_0[n]$, $\dot{x}_{i-1}[n]$, and $\ddot{x}_0[n]$ are instead obtained via wireless communication, thus their value is affected by errors and can be outdated, as the beaconing process is slower than $\Delta_t$ and it is not fully reliable. Their value will be the one received with the last beacon message from the leader or from the front vehicle.

The $\alpha_i$ parameters in Equation (6) are defined as

$$\alpha_1 = 1 - C_1; \quad \alpha_2 = C_1; \quad \alpha_5 = -\omega_n^2$$

$$\alpha_3 = -\left(2\xi - C_1 (\xi + \sqrt{\xi^2 - 1})\right) \omega_n$$

$$\alpha_4 = C_1 (\xi + \sqrt{\xi^2 - 1}) \omega_n.$$ (11)

$C_1$ is a weighting factor between the accelerations of the leader and the preceding vehicle, which we set to $0.5$, $\xi$ is the damping ratio, set to $1$, and $\omega_n$ is the bandwidth of the controller, set to $0.2$ Hz as in [31].

As stated in [17], the desired acceleration $u_i$ computed by the controller is not applied instantaneously because of the actuation lag introduced by the mechanical components. Such lag can be modeled by a first order low-pass filter [7], [17]. In [17] the lag is assumed to be $\tau = 0.5$ s. We compute the actual acceleration as

$$\ddot{x}_i[n] = \beta \cdot u_i[n] + (1 - \beta) \cdot \ddot{x}_i[n-1]; \quad \beta = \frac{\Delta_t}{\tau + \Delta_t}.$$ (12)

The acceleration of the platoon leader is governed by a standard ACC. We implemented a discretized version of the ACC detailed in [17], computing the desired acceleration as

$$u_i[n] = -\frac{1}{T} (\ddot{x}_i[n] + \lambda \delta_i[n])$$

$$\delta_i[n] = x_i[n] - x_{i-1}[n] + l_{i-1} + T \dot{x}_i[n]$$

In Equations (13) and (14), $T$ is the headway time in seconds, and $\lambda$ a design parameter. If no other car is in front of the platoon leader, the speed converges to a desired value.

The real acceleration of the car is computed as in Equation (12). To ensure the stability of the system under the presence of a first order lag, the following must hold [17],

$$T \geq 2\tau; \quad \lambda > 0.$$ (15)

We set $T = 1.5$ s and $\lambda = 0.1$.

IV. COMMUNICATION PROTOCOLS

The set of communication protocols for platooning we propose are based on the IEEE 802.11p/IEEE 1609.4 PHY/MAC, hence scheduled messages contend for the channel in a CSMA/CA fashion.

We adopt the platooning controller employed in [6], [17] (Equation (6)) where the inputs to the system are the leader’s and the front vehicle’s speed and acceleration. For the design of the algorithms we exploit the specific requirements of the controller. In particular, we assume that each vehicle is aware of its position in the platoon, and uses this information to decide how and when to send a beacon.

To decide how, we can exploit the fact that, besides the leader, each vehicle needs to communicate its speed and acceleration
Algorithm 1: SLB protocol.

\[
\text{ON\textsc{STARTUP}}():
\begin{align*}
\text{if } \text{myRole} = \text{leader} & \text{ then} \\
& \text{schedule(SEND\textsc{BEACON}, beaconInterval)};
\end{align*}
\]

\[
\text{SEND\textsc{BEACON}}():
\begin{align*}
& \text{sendBroadcast(getVehicleData())}; \\
& \text{schedule(SEND\textsc{BEACON}, beaconInterval)};
\end{align*}
\]

\[
\text{ON\textsc{BEACON}(beacon)}:
\begin{align*}
& \text{updateCACC(beacon)}; \\
& \text{if beacon.sender = leader then} \\
& \text{ON\textsc{LEADER\textsc{BEACON}(beacon)}};
\end{align*}
\]

\[
\text{ON\textsc{LEADER\textsc{BEACON}(beacon)}}:
\begin{align*}
& \text{unschedule(SEND\textsc{BEACON})}; \\
& \text{schedule(SEND\textsc{BEACON}, myPosition - offset)};
\end{align*}
\]

Table II

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beacon frequency</td>
<td>10, 20, and 25 Hz</td>
</tr>
<tr>
<td>Tx power (leader)</td>
<td>20 dBm</td>
</tr>
<tr>
<td>Tx power (followers)</td>
<td>20, 10, and 0 dBm</td>
</tr>
<tr>
<td>Modulation</td>
<td>QPSK R=1/2</td>
</tr>
</tbody>
</table>

only to the vehicle immediately behind. The transmit power can thus be reduced in order to increase spatial reuse and avoid interfering with cars which are “not interested” in receiving such data. Leaders can instead use high transmit power in order to reach all vehicles within the platoon.

In general TPC is complex because it must cope with highly dynamic networks [32], but, for the application we consider, the setup is simplified due to the linear topology of the platoon. One issue to consider in the effect of different types of vehicles. For example, a truck with an antenna placed on its front might not be able to communicate with a car behind [33], requiring an ad-hoc power calibration in order to overcome the problem. This is, however, out of the scope of this paper, so we set a fixed transmit power of 0 dBm for the followers.

To decide when to send, we exploit the vehicle’s position within the platoon. The leader can send its beacon first, then the others can follow in a cascading fashion, i.e., the second vehicle, the third, and so on. Notice that this is different from a standard TDMA approach, as with TDMA every node participating in the communication obeys the same rules. In our approach, only nodes within a platoon cooperate in a TDMA-fashion in order to reduce intra-platoon channel contention.

The pseudo code of this slotted approach is listed in Algorithm 1. The idea is to divide the time after a beacon from the leader into slots, and have each vehicle send its beacon in the slot corresponding to its position in the platoon. As shown in Algorithm 1, only the leader starts to send beacons at protocol startup. The followers use the beacon received from the leader for synchronization, computing the time at which they should send the beacon depending on their position and a time offset. To avoid that a lost beacon from the leader blocks the protocol, each vehicle, upon sending a beacon, always schedules another send event after one beacon interval. Upon reception of leader’s beacon, this event is updated to synchronize with the leader.

The rationale behind this protocol is to reduce random channel contention by adding synchronization among nodes. Moreover, even if there is no inter-platoon collaboration, the leaders can end up roughly synchronizing with other platoons when performing CSMA/CA at the MAC layer. In the remainder of the paper, we refer to this slotted beaconing protocol as SLB and SLBP (without and with TPC, respectively).

In this work we consider a platoon size of 20 cars. Thus, we define a slot time offset between consecutive vehicles equal to 5 ms, i.e., the beacon interval (0.1 s) divided by the platoon size. The leader always broadcasts messages with the maximum transmit power (20 dBm). The followers, when TPC is disabled, use 20 dBm as well, and 0 dBm otherwise.

In order to obtain a deep understanding of the benefits of each of the two proposals, we compare them with a baseline approach which uses standard static beaconing, i.e., periodic broadcasting. This protocol uses only CSMA/CA, so nodes randomly contend for channel access. We refer to this approach, with and without TPC, as STB and STBP, respectively. Transmit power values for leaders and followers are the same as in SLB and SLBP.

V. EXPERIMENTAL VALIDATION

In the first step, we performed a set of experiments with real cars (Figure 1). The goal is to validate and calibrate the network model we employ in simulations against real world measurements. In the experiment, we used four cars and drove on a private road to safely maintain a distance of 5 m when driving at 20 km/h. The vehicles were driven by humans in respect of the Austrian legislation while the system was automatically recording network statistics without requiring any action from the driver.

For the communication, we used two Cohda Wireless MK2\(^1\) and two Unex DCMA-86P\(^2\) devices, both IEEE 802.11p compliant. We connected each device to a Mobile Mark ECOM9-5500 dipole antenna with a gain of 9 dBi, magnetically mounted on the rooftop of the vehicles.

We implemented STB and SLB, and tested them while repeatedly driving on a 2 km stretch of road, using the parameters shown in Table II. Each experiment lasted roughly 30 s and was repeated three times in order to collect results.

\(^1\)http://www.cohdawireless.com/product/mk2.html
\(^2\)http://unex.com.tw/product/dcma-86p2
in different environmental conditions. The same conditions (number of cars, protocols, parameters, etc.) were reproduced in a simulation scenario to calibrate the simulation model.

In our experimental setup, we always received at least 99\% of frames sent, making the frame error rate not valid for comparison. For this reason, we only compare the received power distribution.

We model fading at the receiver with a Rice distribution with a strong Line Of Sight (LOS) component. We assume this as, for this paper, we only take into account cars: As stated in [34], if the first Fresnel zone is less than 40\% obstructed, then shadowing has no major impact on signal strength, and we experimentally verified the statement in another measurement campaign [35]. With a strong LOS component in a Rician channel, we can approximate the amplitude with a Lognormal distribution [36], thus we assume Lognormal fading.

Figure 2 shows the comparison between the simulation and the experimental results for 20 dBm and 0 dBm transmit power for the leader and the followers, respectively. Before running the experiments, we tested the equipment by pairing the NICs using a cable with a 90 dB attenuator, and found that one device is transmitting with lower transmit power than selected, as well as reporting incorrect received signal strength. These tests were used to calibrate and equalize the received power values prior to analyzing the data.

The first aspect we focus on is the shape of the distribution. Different real world experiments show slightly different standard deviations; The one we choose for the simulation (\(\sigma = 2 \text{ dBm}\)) is a good compromise between all of them, matching also the LOS measurements we reported in [35].

The second aspect is the average received power. In the simulation we employed a Free Space path loss model with \(\alpha = 2.0\). As can be seen, the average received power is slightly higher in the experiment. This is due to the antennas we used, as they provide high transmission gain. In the simulation we instead considered theoretical isotropic antennas with no gain, as frequently used in vehicular simulations. For the time being, we are mainly interested in the shape of the resulting curves rather than the exact quantities, thus ensuring better comparability with other simulation studies.

VI. Simulations and Results

To compare the different approaches we use PLEXE, a dedicated Open Source\(^3\) platooning simulator [37]. The simulator is based on the well known Veins [38] framework and provides a high level of detail and realism, featuring mixed scenarios with ACC and CACC controlled vehicles [6] and human behavioral car-following models. A fully fledged IEEE 802.11p/IEEE 1609.4 network stack [39], [40] permits us to develop and to evaluate arbitrary freeway scenarios, high level applications, and communication protocols. In these aspects, our model differs from the one in [41], which assumes completely automated and dedicated freeways. Our vision is the deployment in a more flexible scenario containing both fully automated vehicles and vehicles controlled by traditional car following models, like a generalization of the SARTRE philosophy. Simulating such a mixed scenario is easily possible in our modeling approach.

In this work we want to understand the characteristics and the behavior of the protocols and network conditions in a “stressful” configuration. In the main part of our evaluation, we focus on performance on a global scale, both from a networking and, most importantly, from an application perspective. Finally, to give an idea of the real delay requirements, we analyze the performance of CACC in light of the control model presented in Section III using an emergency braking scenario.

A. Simulation Model and Setup

Table III summarizes all simulation parameters. To model channel phenomena, we employ a Free Space propagation loss model with \(\alpha = 2.0\) plus a log-normally distributed fading with \(\sigma = 2.0\), as obtained from the model calibration in Section V. As physical and MAC layer models we use the IEEE 802.11p and IEEE 1609.4 models presented in [40]. We disable, however, the switching between CCH and Service Channel (SCH), using only the CCH. The bitrate for STB, SLB, and DynB is 6 Mbit/s, which has been reported to be optimal for highly demanding vehicular applications [42].

Regarding the application layer, packets have an MSDU size of 200 B. They use the AC_VI access category, and are sent with a beacon frequency of 10 Hz, the minimum required by CACC [7]. The fixed beacon interval value of 10 Hz only holds for STB and SLB. DynB and DCC compute their own beacon intervals; Further, DynB uses a static transmit power of 20 dBm. The implementation and the parameters for DynB are taken from [14] and listed in Table III; The DCC parameters are set to their default values, as given in Table I. All protocols send standard CAMs which, as defined by ETSI, are broadcast frames [12]; Thus, they are not acknowledged.

The last physical layer parameter we change for all protocols but DCC is the CCA-threshold. The CCA-threshold is used to assess channel busy status when a station misses the preamble portion of a frame, for example, when multiple frames are

\(^3\)http://plexe.car2x.org
Figure 3. Busy time ratio and collisions for the 160 and the 640 cars scenarios, CCA-threshold set to $-95 \text{ dBm}$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path loss model</td>
<td>Free space ($\alpha = 2.0$)</td>
</tr>
<tr>
<td>Fading model</td>
<td>Log-normal ($\sigma = 2.0$)</td>
</tr>
<tr>
<td>PHY/MAC model</td>
<td>802.11p/1609.4 single channel</td>
</tr>
<tr>
<td>Bitrate</td>
<td>$6 \text{ Mbit/s (CCH)}$</td>
</tr>
<tr>
<td>Access category</td>
<td>AC, VI</td>
</tr>
<tr>
<td>MSDU size</td>
<td>$200 \text{B}$</td>
</tr>
<tr>
<td>Transmit power</td>
<td>$20 \text{ and 0 dBm}$</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>$-95 \text{ dBm}$</td>
</tr>
<tr>
<td>Noise floor</td>
<td>$-95 \text{ dBm}$</td>
</tr>
<tr>
<td>CCA-threshold</td>
<td>$-95, -85, \text{ and } -65 \text{ dBm}$</td>
</tr>
<tr>
<td>Number of cars</td>
<td>160, 320, and 640</td>
</tr>
<tr>
<td>Number of lanes</td>
<td>4</td>
</tr>
<tr>
<td>Platoon size</td>
<td>20 cars</td>
</tr>
<tr>
<td>Car length</td>
<td>4m</td>
</tr>
<tr>
<td>Intra-platoon distance</td>
<td>5m</td>
</tr>
<tr>
<td>Inter-platoon distance</td>
<td>$\approx 41 \text{ m}$</td>
</tr>
<tr>
<td>Speed</td>
<td>100 km/h</td>
</tr>
<tr>
<td>$C_1$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\omega_n$</td>
<td>0.2 Hz</td>
</tr>
<tr>
<td>$\xi$</td>
<td>1</td>
</tr>
<tr>
<td>$T$</td>
<td>1.5 s</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\tau$</td>
<td>0.5 s</td>
</tr>
<tr>
<td>$I_{\text{des}}$</td>
<td>0.1 s</td>
</tr>
<tr>
<td>$b_{\text{des}}$</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table III

**NETWORK AND ROAD TRAFFIC SIMULATION PARAMETERS**

Simultaneously received (cumulative interference). For the CCA-threshold we use values of $-65 \text{ dBm}$ (as defined in the standard [43, 18.3.10.6]), $-85 \text{ dBm}$ (the minimum required sensitivity for the lowest modulation and coding scheme), and $-95 \text{ dBm}$ (equal to model’s minimum sensitivity). In the IEEE 802.11 standard, sensitivity is different from CCA-threshold. For an IEEE 802.11-compliant device, sensitivity is defined as the power threshold above which 90% of the preambles are correctly detected [43, 18.3.10.6]. In our model, instead, the sensitivity is the frame detection threshold and it is set to $-95 \text{ dBm}$. Any frame received with a power lower than the sensitivity is simply ignored.

We simulate a stretch of a 4-lane freeway filled by platoons of 20 cars each, for a total number of cars of 160, 320, and 640, respectively. Such a high number of vehicles might seem unreasonable, but we choose it for two reasons. First, we want to understand if there is any upper limit, meaning that we want to know if the protocols are always behaving as expected, i.e., if they stop working properly above a certain vehicle density. Second, such high densities are well possible on big freeways during rush hours, and platooning might exactly be the application we want to run in such situations. Therefore, understanding whether it can be supported or not is crucial. We choose 640 as the upper bound value because in this case the interference domain of each vehicle does not cover the entire scenario so adding further cars does not affect the ones in the middle. Other relevant parameters are the intra-platoon vehicle distance ($\text{gap}_{\text{des}}$), set to 5 m, and the speed of the platoons, set to 100 km/h. In this work we assume a constant speed because we focus on the analysis of the network. Each simulation experiment (each combination of density and protocol) has been repeated 10 times in order to improve the confidence in the results. As a further processing step, we partially removed data collected at simulation borders to get rid of border effects which biased the evaluation. More details about this procedure are given during the discussion of the single metrics.
B. General Networking Performance

The simulations we run for the analyses in Sections VI-B to VI-D assume the wireless channel to be dedicated to the platooning application. This eases the interpretation of the results and permits to obtain some fundamental understanding of the network’s behavior. In Section VI-E we relax this assumption and simulate a highway with automated vehicles (running our beaconing protocols) and human driven vehicles (running DCC).

We begin our analysis by looking at two generic network metrics: Channel busy ratio and collisions. Channel busy ratio is the amount of time the physical layer declares the channel as busy over a certain time window. Each vehicle samples and records the busy ratio once a second throughout the entire simulation. Collisions instead count the number of frames that each vehicle was not able to decode due to interferences. This metric is sampled and recorded once a second as well. Sampled values for both metrics are grouped in boxplots, which display the first and third quartiles as a box and the median as center line, as well as the minimum and maximum value with whiskers. The boxplots do not include data collected during simulation warm-up. Moreover, for the 640 vehicles scenario, we discard data of cars at the border of the simulation, to avoid border effects. In particular, we removed the data of 15% of the vehicles (7.5% at the front and the tail) after verifying that this amount was enough to get rid of such effects. Vehicles at the head and at the tail of the stream experience lower channel congestion because the interference domain of each car is smaller than the size of the scenario. For this reason, border vehicles falsify the results by showing higher performance than the actual one.

Figure 3 shows busy ratio and collisions for the 160 and 640 cars experiments, for a CCA-threshold of $-95$ dBm. From a network perspective, 160 cars do not cause network overload. This is shown by a maximum channel load in the order of 50% and a very limited amount of collisions. DynB keeps the channel load at the desired level (25%), while for DCC the load spans between 5 and 20% due to its dynamic behavior. For what concerns STB, STBP, SLB, and SLBP, instead, we see that TPC in low-density scenarios is not helpful because the network is not saturated. Moreover, the slotted approach shows a slight improvement in network utilization and collision reduction, but the difference is statistically irrelevant.

For the high density scenario (640 cars), results are totally different. The dynamic approaches are still capable of keeping the load and the collisions under control by adapting their behavior to the high number of nodes simultaneously contending for the channel. STB and SLB, in contrast, completely saturate the channel reaching about 80% channel load and a large amount of collisions per second. Using TPC in STBP and SLBP improves the performance in terms of collisions. Even if channel load is close to complete saturation, the number of collisions is drastically reduced compared to STB and SLB. Results for 320 cars scenarios show similar behavior. For the sake of brevity, however, we omit the graphs for such scenarios.

C. Application Layer Perspective

According to busy ratio and collisions, the dynamic approaches definitely show better performance. DynB and DCC...
are indeed designed to improve the overall network conditions without considering specific application requirements. In platooning systems, missed (or omitted) packets can harm the application and thus passengers’ safety. If a CACC controller misses data packets it is forced to perform a “blind” control action, i.e., it computes the desired acceleration based on outdated information, which might result in instabilities or crashes. To measure protocols’ effectiveness from the application layer perspective, however, we would need to have precise information about the theoretical requirements of the controller, which are out of scope of this paper. We thus define an application layer metric which is parameterized on a maximum tolerable delay.

In particular, let \( \delta_{\text{req}} \) be the maximum allowable inter-message delay, and let \( D \) be the set of all inter-message delays collected by a vehicle. We define the set of all delays satisfying the requirement \( \delta_{\text{req}} \) as

\[
D_{\text{safe}} = \{ d : d \in D \land d \leq \delta_{\text{req}} + \Delta \},
\]

where \( \Delta \) is a small grace period in which the information is still useful, which accounts for uncertainties such as MAC layer backoffs. In our computation, we set \( \Delta = 10 \text{ ms} \), which is assumed to be the CACC controller sampling time [7], [37]. The safe time ratio metric \( r_{\text{safe}} \) is defined as

\[
r_{\text{safe}} = \frac{\sum_{d \in D_{\text{safe}}} d_s}{\sum_{d \in D} d}.
\]

For example, if all delays in \( D \) equal 200 ms, by setting \( \delta_{\text{req}} = 100 \text{ ms} \) we obtain \( r_{\text{safe}} = 0 \), meaning that the vehicle was never in safe conditions. Conversely, by setting \( \delta_{\text{req}} = 300 \text{ ms} \) we obtain \( r_{\text{safe}} = 1 \), thus the requirement was always satisfied.

Figure 4 shows the results for the \( r_{\text{safe}} \) metric. Each point in the plot represents the average with 95% confidence intervals among all cars and all simulations for that particular configuration. We removed border vehicles for the 640 cars scenario. Moreover we discarded all the inter-arrival times of leader messages for the first platoon in each lane for the 640 cars scenario, as for DynB the first leaders are beaconing way faster than other cars because of their border position. Because each vehicle expects to receive data from the leader and the vehicle immediately in front, we plot the metric for both kind of packets and we refer to them as leader and front messages.

In Figure 4 DynB shows the consequences of forcing the busy ratio to a fixed value. To keep channel utilization at the desired level, DynB needs to increase the beacon interval indefinitely. For the 160 cars scenario the performance is still reasonable, but in the 640 cars case, even for a delay requirement of 1 s, the vehicles are in a safe state for less than 30% of the time. DynB, however, shows a fair behavior, as the metric is similar for both leader and front messages.

DCC, instead, behaves “orthogonally” with respect to DynB. DCC shows similar performance independently of the number of vehicles but behaves unfairly, as \( r_{\text{safe}} \) for front messages is higher than for leader messages. This is due to the low transmit power employed in the RESTRICTIVE state (\(-10 \text{ dBm}\)), making communication with the leader almost impossible.

Concerning STBP and SLBP, instead, it is clear that taking into account specific application requirements can bring enormous benefits. Not using TPC indeed results in poor performance in a high density scenario. STB and SLB perform better than dynamic approaches, but worse than their TPC counterparts. In the 640 cars scenario, for a delay requirement of 100 ms using TPC results in a performance gain of roughly 40% and 20% for leader and front messages respectively. In the worst case (\( r_{\text{safe}} \) for 640 cars, leader messages) the protocols ensure that vehicles are in safe conditions roughly 70% of the time for the most demanding delay requirement (0.1 s). Overall the slotted approach provides slightly, but not significantly better results. Front vehicle messages are easier to receive due to the small distance between consecutive vehicles. The benefits of the slotted approach are, however, influenced by the size of the platoon: The bigger the platoon, the smaller the channel contention among vehicles in the same platoon.

The results show that, even if STB and SLB cause in general a higher number of collisions (Figure 3), such collisions interest data frames that are not needed by the CACC application. STB and SLB manage to deliver leader and front vehicle messages in 90% of the time within 200 ms in the most demanding scenario. To keep the collisions under control, instead, the dynamic approaches need to lower the beacon rate, causing extremely large delays. In summary, an increase in the collisions count does not necessarily worsen application layer performance.

D. Impact of CCA-threshold

In this section we briefly analyze the impact of the CCA-threshold on the performance of SLB and SLBP. The CCA-
threshold defines the amount of energy required to declare the channel busy when the preamble portion has been missed. For example, during a transmission, a station will not hear preambles of frames sent by other stations. Thus, when the transmission is completed, the station is required to measure the amount of energy in the channel to understand if there are ongoing communications. The IEEE 802.11 standard mandates a CCA-threshold of $-65\,\text{dBm}$, but in this study we consider $-85$ and $-95\,\text{dBm}$ as well.

Figure 5 shows the safe time ratio of leader messages for SLB and SLBP for the 160 cars scenario. When the number of cars is limited and all nodes use full transmit power, the CCA-threshold has no impact at all (for the values we consider). Conversely, when using TPC, a lower CCA-threshold helps increasing the awareness about other vehicles’ transmissions, thus decreasing the number of collisions. In particular, for a delay requirement of $0.1\,\text{s}$, $r_{\text{safe}}$ increases of roughly $5\%$ when using a CCA-threshold of $-95\,\text{dBm}$.

For a larger number of vehicles (Figure 6, 640 cars scenario), a low CCA-threshold is too conservative and reduces spatial reutilization. Indeed, for a delay requirement of $0.1\,\text{s}$, the safe time ratio for $-95\,\text{dBm}$ performs roughly $10\%$ worse than higher CCA-thresholds.

By considering all the cases, the best approach would be to adapt the threshold based on the network load, as DCC mandates or as considered in other works [44]. In absence of such a mechanism, however, a threshold of $-65\,\text{dBm}$ as mandated by the IEEE 802.11 standard provides, on average, the best performance.

### E. Coexistence with DCC

In the analysis considered so far we assumed the channel to be dedicated to the platooning application, together with a $100\%$ market penetration. Especially during the introduction of this technology, the road will be shared among human and autonomously driven vehicles, and human driven vehicles might use different applications concurrently accessing the channel. We thus modified the 640 vehicles scenario by filling two lanes with 320 platooning vehicles using STBP and SLBP, while the remaining two lanes with human driven vehicles using DCC.

Figure 7 shows safe time ratios of leader and front messages, for both STBP and SLBP. We apply the same border removal procedure used in previous plots. The performance is comparable to the one in Figure 4: All approaches have a safe time ratio greater than $85\%$ for a delay requirement of $100\,\text{ms}$. Due to the load caused by STBP and SLBP, DCC goes in RESTRICTIVE state, thus using the minimum transmit power of $-10\,\text{dBm}$. As a results, STBP and SLBP are unaffected by human driven vehicles, but DCC suffers a large amount of packet losses. DCC is indeed not designed to coexist with other protocols, thus its parameters should be tuned differently. This is, however, out of scope for this paper.

### F. Impact of Communication on CACC Performance

In this section we make a base study of the performance of the CACC described in Section III. The idea here is to provide a generic understanding of what the requirements of the controller might be. To this purpose we simulated a single platoon of 20 cars running at $130\,\text{km/h}$ using SLB to avoid any kind of channel contention among vehicles. We perform a full-stop braking with different deceleration rates and different beacon frequencies, and measure the minimum distance between any pair of cars as a measure of the reliability of the system. This will give a basic idea of the value for the $\delta_{\text{req}}$ previously defined. Providing a definite value is out of the scope of the paper, as this requires a dedicated study.

We used a beacon rate from $1\,\text{Hz}$ to $10\,\text{Hz}$ in steps of $1\,\text{Hz}$, and from $10\,\text{Hz}$ to $20\,\text{Hz}$ in steps of $5\,\text{Hz}$. The leader decelerations we used are $2$, $4$, $6$, and $8\,\text{m/s}^2$. The maximum deceleration for the follower vehicles were set to $9\,\text{m/s}^2$, so vehicles have a higher braking capability than what is performed by the leader. This is a different setup from the real world experiments in [45], where the driving conditions of the vehicles were different. In a first experiment, maximum deceleration was inhomogeneous among the different trucks and they were driving on different lanes to avoid a real crash, while in a second experiment following trucks had a stronger braking capability than front ones. Even if the maximum braking capabilities are the same, in our experiments the system can become unsafe because of delayed message reception.

We repeat each simulation 10 times, and we take the minimum distance over all pairs of cars and simulation runs for each specific configuration of beacon frequency and deceleration rate. We thus show a worst case analysis.

Figure 8 shows the resulting minimum distances. The abscissa is plotted in logarithmic scale in order to highlight CACC behavior for beacon frequencies up to $10\,\text{Hz}$. For the sake of comparison with the $\delta_{\text{req}}$ values used in Figure 4, we added a second scale showing the inter-message interval corresponding to a particular beacon frequency. When the minimum distance is $0\,\text{m}$ it means that two cars crashed into each other.

The first noticeable fact is that the allowable inter-message interval for such emergency braking scenario highly depends on the deceleration rate. The higher the deceleration, the lower the tolerable delay. For example, for a $2\,\text{m/s}^2$ deceleration rate, $0.5\,\text{s}$ of inter-message interval seems enough to avoid a crash, as the worst-case minimum distance is $2.5\,\text{m}$, whereas when decelerating at $8\,\text{m/s}^2$ a delay of $0.33\,\text{s}$ can result in a crash.

A non-crash, however, does not necessarily translate into a “safe” situation, as drivers might feel uncomfortable when
coming too close to the vehicle in front. Anyhow, the graph suggests that an inter-message interval greater than 0.3 s might harm system’s safety and result in crashes. To obtain a definite value, however, it is required to study the controller more in depth, for example by considering different controller parameters values, different actuation lags, and different inter-message intervals for leader and front messages.

To conclude, by considering the results in Figure 4c and by comparing them with the performance in Figure 8, STBP and SLBP by being able of providing data to the application with a minimum rate of 5 Hz in 90% of the time, even in the occurrence of an emergency situation the vehicles would avoid a crash. Indeed, Figure 8 shows that the CACC is robust to a “blind control” action when few packets are lost. In contrast, the dynamic approaches show unacceptable performance for a CACC application, having an update time larger than 1 s 80% of the time.

These results raise two different interesting issues: i) Can the network load further be reduced by adapting the beacon frequency to the current acceleration or in general to the “stability” of the platoon? and ii) further joint research between the networking and the vehicular control community is needed to identify controllers that are efficient, reliable and stable under dynamic beaconing, as well as under emergency situations.

VII. Conclusion

This paper has discussed the feasibility of different beaconing solutions for an automated platooning system. The goal of the study was to compare the network and application layer performance of state-of-the-art dynamic beaconing solutions, i.e., DynB and ETSI DCC, against four possible alternatives we propose. The design of these new protocols stems from the specific requirements of the application, which suggest a TDM-like approach coupled with TPC. The results prove that considering high-level requirements can greatly improve the performance from the application layer perspective while avoiding severe network congestion. Furthermore, we briefly discussed the impact of the choice of different CCA-thresholds, showing that, in the absence of a dynamically changing threshold, the value mandated by the IEEE 802.11 standard gives the best performance on average. We also considered a mixed scenario where some human driven vehicles concurrently access the channel using DCC, showing that the performance of our approaches is unaffected, but that DCC would need to be re-parameterized. Finally, we have obtained the requirement that a CACC application has in an emergency braking scenario, showing that the maximum tolerable delay depends on the dynamics of the maneuver. Maximum delay, however, should not be bigger than 0.2 to 0.3 s. We believe that these results can help the community in the development of efficient, dynamic, and application-aware beaconing protocols for platooning.

References

ARMY and IBM funding. His team is developing a Vehicular Testbed for monitoring (see www.cs.ucla.edu/NRL for recent publications). He also authored the textbook Vehicular Networking, published in 2014 by Cambridge University Press.

Stefan Joerer (joerer@ecs-labs.org) received his BSc and MSc in Computer Science from the University of Innsbruck, Austria in 2009 and 2011, respectively. In 2011, he joined the Computer and Communication Systems Group at the University of Innsbruck as a PhD Student. In 2014, he has won the prestigious Australia Award Endeavour Research Fellowship to visit the University of Sydney. He was a visiting scholar with the Wireless Networking Group (WiNG) of Prof. Abbas Jamalipour at the School of Electrical and Information Engineering, University of Sydney in 2014. His current research is focused on questions regarding Inter-Vehicle Communication (IVC) for road traffic safety applications.

Christoph Sommer (sommer@ecs-labs.org) is an Assistant Professor (AkadR a.Z.) at the University of Paderborn, joining the Distributed Embedded Systems Group in 2014. He received his Ph.D. degree in engineering (Dr.-Ing., with distinction) and his M.Sc. degree in computer science (Dipl.-Inf. Univ.) from the University of Erlangen in 2011 and 2006, respectively. In 2010, he was a visiting scholar with the research group of Ozan K. Tonguz at the Electrical and Computer Engineering Department of Cambridge University Press.

Renato Lo Cigno (locigno@disi.unitn.it) is Associate Professor at the Department of Computer Science and Telecommunications (DISI) of the University of Trento, Italy, where he leads the Advanced Network Systems research group in computer and communication networks. He received a degree in Electronic Engineering with a specialization in Telecommunications from Politecnico di Torino in 1988, the same institution where he worked until 2002. In 1998/9, he was with the CS Dep., UCLA, as a Visiting Scholar. Renato Lo Cigno has been General Chair of IEEE Int. Conf. on Peer-to-Peer Computing, and General Chair and TPC Chair of ACM WMASH and IEEE WONS in different years. He has served in many TPCs of IEEE and ACM conferences, including INFOCOM, GLOBECOM, ICC, MSWiM, VNC, and CoNext, and has been Area Editor for Computer Networks. His current research interests are in performance evaluation of wired and wireless networks, modeling and simulation techniques, congestion control, P2P networks and networked systems in general, with specific attention toward applications and sustainable solutions. Renato Lo Cigno is senior member of IEEE and ACM and has co-authored more than 150 papers in international, peer reviewed journals and conferences.

Falko Dressler (dressler@ecs-labs.org) is a Full Professor for Computer Science and head of the Distributed Embedded Systems Group at the Dept. of Computer Science, University of Paderborn. Dr. Dressler received his M.Sc. and Ph.D. degrees from the University of Erlangen in 1998 and 2003, respectively. He is an editor for journals such as IEEE Trans. on Mobile Computing, Elsevier Ad Hoc Networks, Elsevier Computer Communications, and Elsevier Nano Communication Networks. He was guest editor of special issues on self-organization, autonomic networking, vehicular networks, and bio-inspired communication for IEEE Journal on Selected Areas in Communications (JSAC), Elsevier Ad Hoc Networks, and others. Dr. Dressler was General Chair of IEEE/ACM BIONETICS 2007, IEEE/IFIP WONS 2011, IEEE VNC 2014, and ACM MobiHoc 2016. TPC Co-Chair for IEEE INFOCOM, IEEE VNC, IEEE VTC, IEEE GLOBECOM, and ACM MSWiM, and Poster/Demo Chair for ACM MobiCom. He regularly serves in the program committee of leading IEEE and ACM conferences. Dr. Dressler authored the textbooks Self-Organization in Sensor and Actor Networks published by Wiley in 2007 and Vehicular Networking published by Cambridge University Press in 2014. Dr. Dressler has been an IEEE Distinguished Lecturer as well as an ACM Distinguished Speaker in the fields of inter-vehicular communication, self-organization, and bio-inspired and nano-networking. Dr. Dressler is a Senior Member of the IEEE (COMSOC, CS, VTS) as well as a Senior Member of ACM (SIGMOBILE), and member of GI (GIBU, KuVS). His research objectives include adaptive wireless networking, self-organization techniques, and embedded system design with applications in ad hoc and sensor networks, vehicular networks, industrial wireless networks, and nano-networking.

Mario Gerla (gerla@cs.ucla.edu) is a Professor in the Computer Science at UCLA. He holds an Engineering degree from Politecnico di Milano, Italy and the Ph.D. degree from UCLA. He became IEEE Fellow in 2002. At UCLA, he was part of the team that developed the early ARPANET protocols under the guidance of Prof. Leonard Kleinrock. At Network Analysis Corporation, New York, from 1973 to 1976, he helped transfer ARPANET technology to Government and Commercial Networks. He joined the UCLA Faculty in 1976. At UCLA he has designed and implemented network protocols including ad hoc wireless clustering, multicast (ODMRP and CodeCast) and Internet transport (TCP Westwood). He has lead the $12M, 6 year ONR MINUTEMAN project, designing the next generation scalable airborne Internet for tactical and homeland defense scenarios. He is now leading two advanced wireless network projects under ARMY and IBM funding. His team is developing a Vehicular Testbed for safe navigation, urban sensing and intelligent transport. A parallel research activity explores personal communications for cooperative, networked medical monitoring (see www.cs.ucla.edu/NRL for recent publications).