

Stability Challenges and Enhancements for Vehicular Channel Congestion Control Approaches

Ali Rostami, *Student Member, IEEE*, Bin Cheng, *Student Member, IEEE*, Gaurav Bansal, *Member, IEEE*,
Katrin Sjöberg, Marco Gruteser, *Member, IEEE*, and John B. Kenney, *Member, IEEE*,

Abstract—Channel congestion is one of the major challenges for IEEE 802.11p-based vehicular networks. Unless controlled, congestion increases with vehicle density, leading to high packet loss and degraded safety application performance. We study two classes of congestion control algorithms: reactive state-based and linear adaptive. In this paper, the reactive state-based approach is represented by the Decentralized Congestion Control (DCC) framework defined in European Telecommunications Standards Institute (ETSI). The linear adaptive approach is represented by LInear MESSage Rate Integrated Control (LIMERIC) algorithm. Both approaches control safety message transmissions as a function of channel load (i.e. Channel Busy Percentage, CBP). A reactive state-based approach uses CBP directly, defining an appropriate transmission behavior for each CBP value, e.g., via a table look-up. By contrast, a linear adaptive approach identifies the transmission behavior that drives CBP towards a target channel load. Little is known about the relative performance of these approaches and existing comparison are limited by incomplete implementations or stability anomalies. To address this, the paper makes three main contributions. First, we study and compare the two aforementioned approaches in terms of channel stability, and show that the reactive state-based approach can be subject to major oscillation. Second, we identify the root causes and introduce stable reactive algorithms. Finally, we compare the performance of the stable reactive approach with the linear adaptive approach and the legacy IEEE 802.11p. It is shown that the linear adaptive approach still achieves a higher message throughput for any given vehicle density for the defined performance metrics.

I. INTRODUCTION

Cooperative Intelligent Transport System (C-ITS) technology enables a wide variety of vehicular ad hoc networking applications, including collision avoidance, road hazard awareness, and route guidance. Based on the Medium Access Control (MAC) and Physical Layer (PHY) protocols specified in the IEEE 802.11p standard [1], C-ITS is moving rapidly towards deployment in Europe and other regions. Twelve members of the Car-2-Car Communications Consortium (C2C-CC) have mutually pledged to begin equipping their vehicles with C-ITS by the end of 2015 [2]. In the US, where the technology is known as Dedicated Short Range Communication (DSRC), the Department of Transportation has published

an Advance Notice of Proposed Rulemaking with an intention to require this equipment in new cars within a few years [3].

While most aspects of the communication system have been finalized and standardized (i.e. in IEEE 1609 WG[4][5]), one remaining aspect in need of further study is channel congestion control [6]. With a typical communication range of hundreds of meters, a C-ITS device may share a 10 MHz channel with hundreds or even a few thousand other devices. The Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) MAC protocol used in C-ITS is optimized for low-to-moderate channel loads. [7] illustrates that for higher density of vehicles, IEEE 802.11p shows a behavior similar to ALOHA. With increasing channel load due to high density of vehicles, the channel becomes saturated, the probability of overlapping transmissions (i.e. packet collisions) increases considerably, and the aggregate channel throughput falls off after reaching a plateau [8][9].

While in general a C-ITS channel may support a variety of applications, congestion in the 5.9 GHz spectrum is likely to be associated with a high volume of vehicle safety messages. These are Cooperative Awareness Messages (CAMs) [10] in Europe and Basic Safety Messages (BSMs) [11] in the US. Congestion reduces the rate at which these safety messages are successfully communicated to neighbors, and the resulting reduced awareness harms the C-ITS safety mission.

Broadcast channel congestion has previously been investigated in the context of Mobile Ad Hoc Networks (MANET) [12], but the car-2-car communication settings differs. Previous studies on MANETs, such as [13], focus on techniques to control congestion arising due to re-broadcasting in multi-hop protocols. These techniques do not apply when congestion arises due to frequent broadcast in a single-hop communication setting, which we discuss in this paper.

In addition, known congestion control techniques such as Internet flow control do not adequately address this issue due to the unique characteristics of the vehicular networking environment. These include broadcast transmissions, one hop communication, and a shared wireless channel. Therefore, researchers have proposed several algorithms [14], [15], [16] for the vehicular network environment that are considered in the ETSI standardization process. The effectiveness of these algorithms have largely been evaluated individually and there are few comparative studies available that evaluate the algorithms under common assumptions and scenarios [17], [18]. To the best of our knowledge, however, no prior work has considered a complete implementation of DCC with mandatory CAM generation rate control in the facilities layer or proposed

Manuscript received on ...

A. Rostami, B. Cheng and M. Gruteser are with the WINLAB at Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854 USA (e-mail: rostami@winlab.rutgers.edu; cb3974@winlab.rutgers.edu; gruteser@winlab.rutgers.edu).

G. Bansal and J. B. Kenney are with the Toyota InfoTechnology Center, Mountain View, CA 94043 USA (e-mail: gbansal@us.toyota-itc.com; jkenney@us.toyota-itc.com).

K. Sjöberg is with the Volvo Group Trucks Technology, Gothenburg, Sweden (e-mail: katrin.sjoberg@volvo.com).

DCC versions that do not suffer from stability issues. Since these protocols are serious contenders for standardization, a thorough understanding of their performance and stability is particularly important.

This paper revises and builds on our previous performance comparison of the congestion control techniques [19]. The new contributions can be summarized as follows:

- Analyzing the stability of two important VANET channel congestion control mechanisms in dense scenarios on two different road topologies.
- Demonstrating DCC instability in a simulation scenario inspired by a real-world road configuration.
- Identifying root causes for instability in such scenarios.
- Proposing a stable DCC congestion control algorithm.
- Demonstrating the stability of stable DCC congestion control and comparing it with linear-adaptive congestion control.

The remainder of the paper is organized as follows. Section II, following this introduction, reviews the previous works in the literature. Section III, explains the essential background of the selected congestion control algorithms. Section IV studies the stability of LIMERIC and DCC. Then, we investigate the problems associated with the DCC, and propose some solutions for the parts of the algorithm which are responsible for the identified problems in Section V. Simulation results showing achieved improvements and comparing the algorithms are presented in Section VI. Discussion about the results and rationale behind important parameter selections are found in Section VII. Finally, the conclusions are presented in Section VIII.

II. RELATED WORKS

With increasing demands for a shared resource, such as a wireless channel, control mechanisms become a requirement to prevent poor service. Perhaps best known in this domain is the extensive work on Internet congestion control algorithms (e.g. [20], [21] and [22]). While there is some overlap between Internet congestion control and vehicular network channel congestion control issues, existing congestion control algorithms are not suitable for delay sensitive, reliable single hop communications over wireless networks and rely on acknowledgment feedback which is unavailable in vehicular network broadcast messaging. Instead, vehicular network congestion control algorithms can exploit richer direct measurements of the congestion level than a TCP agent in an Internet environment. With this precise feedback, it becomes beneficial to use more fine-grained control algorithms, as shown for example in a comparison [23] of a binary adaptive control algorithms (e.g. AIMD algorithm as used in TCP) with more fine-grained linear adaptive control algorithms such as LIMERIC [15].

There are some other efforts to solve the congestion control problem for MANETs by focusing on rate-based flow control and broadcast application's characteristics [24][25], but still the main assumption of these works is the wireless networks with re-broadcast requirement, mostly for the routing phase. The current vision for vehicular safety messages, however,

assumes an environment with only single-hop broadcast communication [26]. the safety applications considered here do not require messages to be re-broadcast or flooded through the network.

Existing MAC standards, such as IEEE 802.11p, cannot maintain optimal throughput while the number of wireless devices increases, unless they rely on a higher layer control mechanism. [27] shows how adaptive congestion control can outperform legacy IEEE 802.11p and motivates the use of a channel congestion control mechanism on top of the legacy IEEE 802.11p MAC layer.

To date, several proposals have been presented to conquer the wireless channel congestion problem. In [28], the authors use both power and rate control to reach asymptotically optimal performance. [29] also proposes another adaptive scheme to solve the channel congestion issue. The authors use both rate and power control to overcome this issue, but manipulate the transmission power only once the message rate is already reduced to the minimum defined in the protocol. [30] introduces a new adaptive approach that controls channel congestion while it tries to meet minimum application requirements for multihop information dissemination. This paper's focus, however, is on transmission rate control (TRC) approaches, since some previous works, such as [31], concluded that message rate is the most effective control parameter in terms of reachability. Hence, we focus on TRC technique, which we will detail in the next section.

Few comparative evaluations of congestion control algorithms exist. [32] compares the Linear Memoryless Range Control (LMRC) and the Gradient Descent Range Control (GDRC) congestion control algorithms. The authors observed that when local channel load measurement is used, LMRC suffers instability. They concluded that a global CBP measurement can improve stability of adaptive congestion control. The focus, however, is on a different approach, where the control parameter is the transmission power with a fixed message rate. Another work, [33] compares European DCC with Self-organizing Time Division Multiple Access (SoTDMA) in terms of awareness and emergency coverage range, focusing on the effect of simultaneous transmissions. The bottom line of the work is that DCC provides slightly better performance, but the work does not provide the resource management analysis to explain why the results are such as they are. These studies do not compare algorithms that are serious candidates for standardization.

Several studies have reported instability for the DCC algorithm. [34] conducts a simulation experiment to show that fewer number of control parameters could lead to a better performance of DCC. It has chosen PHY data rate as the control parameter of a simpler DCC algorithm. While the results show that DCC with just PHY data rate as the control parameter works better than the DCC, the authors did not explain why playing with one control parameter leads to such a better performance or why the resulting loss of range due to PHY rate increases is tolerable. [18] identifies an oscillation problem in the DCC approach. The authors of this work conclude that this oscillatory behavior is due to frequent state changes in DCC's Finite State Machine (FSM), however, the

study does not appear to implement the recently approved CAM generation rules required by ETSI in [10]. Similar results have also been presented by [31], albeit also without the CAM generation rules. Additionally, the authors also compare the impact of different DCC control parameters in terms of reachability and stability. They emphasize the transmission rate control as the most important control parameter in terms of reachability. [17] compares the awareness level of WAVE with European DCC approach. One of the observations is again channel load oscillation due to frequent state changes. This study also does not implement the CAM algorithm.

None of these studies have examined the root causes of such oscillations, whether such oscillations persist with a complete implementation that uses the CAM algorithm, or how to design stable reactive-state based congestion control algorithms. These aspects are the focus of this paper.

III. BACKGROUND

In Europe, the ITS G5 architecture is slightly different than Wireless Access in Vehicular Environment (WAVE) protocol stack, which is used in the US. The major differences between the two approaches are: (i) in Europe, DCC is required by regulation (EN 302 571 [35]) and it must be situated at the access layer, whereas there is not yet a DCC regulation in the US; (ii) at the networking & transport layer, Europe has support for multihop communication through GeoNetworking (GeoNet), whereas no such capability is specified in the US; and (iii) in the US events such as hard braking are indicated within the BSM, while in Europe such events are communicated not by the CAM but rather in the distinct message Decentralized Environmental Notification Message (DENM). The common elements between US and Europe are IEEE 802.11p and Logical Link Layer (LLC) at the lower layers. In addition, a high degree of harmonization has been achieved between the BSM and CAM.

A. BSM and CAM generation

The position messages, BSM and CAM, will be the basis for increased road traffic safety. They contain more or less the same information except for some minor differences. The BSM structure is outlined in SAE J2735 [11] and CAM in EN 302 637-2 [10], and they contain position information, time stamp, heading, speed, driving direction, path history, vehicle type etc. The BSM generation rules have not yet been specified in the US. Most testing and trials have used a fixed 10 BSMs/second rate. Specific generation rules will most likely be standardized as part of a congestion control algorithm for precise channel access control. The generation of CAMs, on the other hand, is outlined in EN 302 637-2 [10] and in short the generation is based on vehicle dynamics and can be restricted by DCC.

CAMs are generated at intervals of no less than 100 msec and no more than 1000 msec. This time boundary is checked whenever an updated T_GenCam_Dcc is available. The parameter $T_CheckCamGen$ is set to 10 msec, which decides how often the algorithm should be executed to check if a new CAM should be generated. A new CAM shall be generated when

TABLE 1
CAM PARAMETERS AND VARIABLES

Parameter	Description
$T_CheckCamGen$	Time Interval to check possible CAM generation
T_GenCam_Dcc	Inter message interval if the vehicle's dynamic is high
$Cam_elapsed_time$	Time passed since the last CAM generation
T_GenCam	Inter message interval if the vehicle's dynamic is not high
N_Cam	Number of consecutive CAMs generated due to low vehicle's dynamic

both of the following conditions, measured relative to the prior CAM message, are met: the interval provided by DCC, via the T_GenCam_Dcc parameter, expires, and one of the following dynamics criteria are met: (Cond.1) heading changed $>4^\circ$, (Cond.2) position changed >4 meters, or (Cond.3) magnitude of speed changed >0.5 m/sec. When a CAM is triggered by one of the dynamics conditions, a second and third CAM will also be generated at the same intervals through T_GenCam , unless subsequent dynamics lead to an even shorter interval. N_Cam is used to keep track of number of consecutive CAM generations based on the last CAM generation by dynamic rules. A CAM is also generated after one second even if the two conditions are not met. T_GenCam_Dcc is set via the management plane by DCC residing in the access layer [10]. Detailed steps for CAM generation is shown in Algorithm 1. Table 1 shows the most important CAM parameters as well.

Algorithm 1 CAM Generator

```

every  $T\_CheckCamGen$  do:
   $T\_GenCam\_Dcc \leftarrow$  look-up result from DCC
  Check  $T\_GenCam\_Dcc$  boundaries
  if  $Cam\_elapsed\_time \geq T\_GenCam\_Dcc$  then
    if Cond.1 OR Cond.2 OR Cond.3 then
      Generate a CAM
       $T\_GenCam \leftarrow Cam\_elapsed\_time$ 
       $N\_Cam \leftarrow 0$ 
    else if  $Cam\_elapsed\_time \geq T\_GenCam$  then
      Generate a CAM
      Increment  $N\_Cam$  by 1
      if  $N\_Cam > 3$  then
         $T\_GenCam \leftarrow$  upper bound time boundary
      end if
    end if
  end if

```

To visualize the impact of CAM generation rules on the number of generated CAMs, Figure 1 shows the number of generated CAM's on different parts of the winding highway for 20 seconds of simulation with 1000 nodes. Each bar represents the number of generated CAM's (at the facilities layer) in all vehicles for a 100 meter interval along the x-axis of the road. The reason for generating a higher number of CAMs at the edges is because of lower channel load as well as the fact that vehicles are changing their speeds significantly to

make a turn. Further, the spikes in the winding part are mostly because of the longer length due to the curves, in addition to the continuous change in the vehicle's headings.

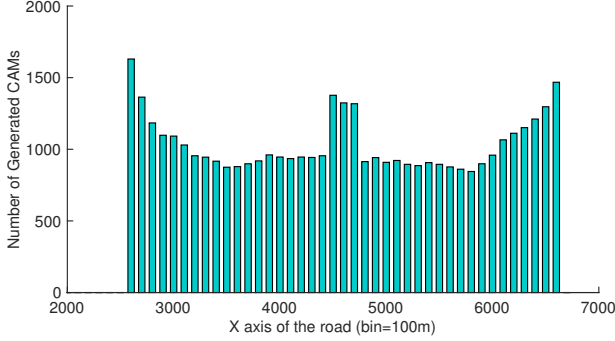


Fig. 1. Generated CAM's for 20 seconds in all vehicles vs X position of the cars in winding highway scenario, introduced in section VI-A

CAMs and BSMs are broadcasted in ad hoc networks, rendering traditional automatic repeat request (ARQ) feedback infeasible. The best feedback in IEEE 802.11p networks is CBP. If congestion control is not present, the channel can be overloaded as the vehicle density increases. Congestion control improves predictability, reliability and efficient use of channel resources, and is considered a necessary function in vehicular networks. Common methods for congestion control are: (i) transmit message rate control (TRC), (ii) transmit power control (TPC), and (iii) transmit data rate control (TDC). As we mentioned in previous section, the focus in this paper is on TRC.

B. Reactive control: European DCC

TS 102 687 [14] outlines a DCC framework for Europe. Conformance to TS 102 687 is a requirement in the harmonized EN 302 571 [35], regulating the European C-ITS frequency bands. TS 102 687 is a toolbox, with several optional methods. The most prominent method is a table look-up using TRC. see Table 2, where the message rate is specified as a function of measured CBP. The specific values in Table 2 are consistent with those under consideration for trials and deployment.

TABLE 2
LOOK-UP TABLE FOR DCC RATE CONTROL

State	Index	CBP	Message Transmission Interval	Message Rate
RELAXED	1	<30%	100 msec	10 Hz
ACTIVE	2	30-39%	200 msec	5 Hz
	3	40-49%	300 msec	3.33 Hz
	4	50-59%	400 msec	2.5 Hz
RESTRICTIVE	5	>60%	500 msec	2 Hz

Each entry of the DCC look-up table represents a state in a Finite State Machine (FSM). The DCC framework proposes a high-level FSM with three logical states: RELAXED, ACTIVE, and RESTRICTIVE. In this high level

FSM, RELAXED represents a state where the CBP is below $minChannelLoad$ and the channel is considered to be relatively idle. ACTIVE is within the channel load range that DCC desires to stay in. In the RESTRICTIVE state, CBP is beyond $maxChannelLoad$ and in this state DCC is no longer able to control the channel load [14] (i.e., once in the RESTRICTIVE state, the transmission parameters cannot be controlled in response to the increased channel load). The ACTIVE state is further divided into several sub-states (ACTIVE1, ACTIVE2, etc) for more fine-grained control. The final FSM used herein, reflecting Table 2, is depicted in Figure 2. DCC dedicates a unique index number between $1 \dots N$ to each state, where N is the maximum number of states in the FSM, starting from RELAXED in ascending order (i.e., in Figure 2 $N = 5$).

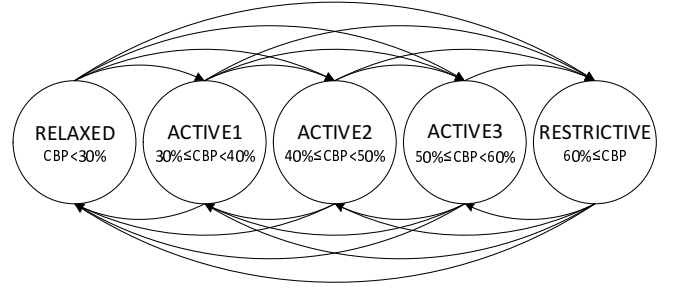


Fig. 2. The DCC finite state machine

In the DCC approach, the channel loads are measured locally by each vehicle during a sampling interval T_{CBP_update} . The main difference between the DCC implementation in [19] and in this paper is that DCC is no longer using immediate CBP values to determine message rates. Instead, it uses a more complete implementation where a function over a window of CBP values is used to determine the state transition.

Specifically, the following procedure and equations are used for determining a possible state change.

$$\begin{aligned} minCL(T_{up}) &\geq ChanLoadThrd(stateUp - 1) \\ minCL(T_{up}) &< ChanLoadThrd(stateUp) \end{aligned} \quad (1)$$

$$\begin{aligned} maxCL(T_{down}) &< ChanLoadThrd(stateDown + 1) \\ maxCL(T_{down}) &\geq ChanLoadThrd(stateDown) \end{aligned} \quad (2)$$

Here, T_{up} and T_{down} define the CBP window lengths and are set to 1 and 5 seconds, respectively. $minCL(T_{up})$ is then the minimum channel load (CBP) value among all the CBP samples measured over the last T_{up} second(s). Similarly, $maxCL(T_{down})$ is the maximum channel load (CBP) value among all the CBP samples measured over the last T_{down} second(s). $ChanLoadThrd(.)$ is a function that returns the upper channel load threshold for a state, as defined in Table 2. For example, for state ACTIVE1 the threshold would be 40%.

The state selection procedure then proceeds in three steps.

- Find a state index $stateUp$ that satisfies Eq. (1). This step will evaluate the equation for all possible states and

find the one unique state that satisfies both constraints in the equation.

- Find a state index *stateDown* that satisfies Eq. (2). Again, this step will evaluate the equation for all possible states and find the one unique state that satisfies both constraints in the equation.
- Change the current state to the larger of the two state indices, i.e. to $\max(StateUp, StateDown)$.

Once the new state has been decided, DCC updates the message interval (T_GenCam_Dcc), which limits the CAM generation in the facilities layer, and also shapes the traffic into the MAC layer. In this paper, if a CAM is generated before the prior CAM is passed to the MAC, the prior CAM is replaced by the new one, such that no more than one CAM is in the gatekeeping queue at a time (while not common, this situation occurs occasionally). As an early observation, it is evident from Table 2 and Figure 2 that there is no control above a channel load of 60%, leaving only the MAC protocol to manage channel usage as the CBP increases.

C. Adaptive control: LIMERIC

LIMERIC [15] is a distributed and adaptive linear rate-control algorithm where each vehicle adapts its message rate in such a way that the total channel load converges to a specified target. As the goal of LIMERIC is to share the channel equally among all the nodes in terms of rate, like every other adaptive mechanisms, LIMERIC tracks the changes of dynamic parameters of the environment, i.e. the measured total rate, and minimizes the error between the goal and current total rates in each step. This makes the congestion control mechanism to actively response to the environment changes and takes the control over the channel.

Having the ability to measure the channel congestion level by measuring CBP, the only other parameter a station needs to know, in order to fairly choose a message rate, is K , the number of nodes sharing the channel together. Not all the nodes are in carrier sense range, however, which makes it hard to estimate number of nodes directly. This is why LIMERIC determines the message rate for the j^{th} vehicle (denoted as r_j) adaptively according to the following equation:

$$r_j(t) = (1 - \alpha)r_j(t - 1) + \beta(r_g - r_C(t - 1)) \quad (3)$$

$$r_C(t) = \sum_{j=1}^K r_j(t) \quad (4)$$

where r_C is the total rate of all K vehicles in a given area, r_g is the target for total rate, and α and β are adaptation parameters that control stability, fairness, and steady state convergence. Starting with α , the main role of this parameter is to promote fair convergence by acting as an exponential forgetting factor. The adaptive gain factor, the β , confine the message rate offset to linear order. Therefore, these two parameters jointly trade off the algorithm convergence with the speed of the convergence. It is shown using linear systems theory that in steady state LIMERIC converges to a unique and fair rate for

all vehicles, and the total rate r_C converges to r_f which is proven as [15]:

$$r_f = \frac{K\beta r_g}{\alpha + K\beta} \quad (5)$$

Stability conditions and convergence speed are also derived, and it is shown that LIMERIC adapts quickly to changing network conditions. For a practical implementation of LIMERIC, the CBP created from all K vehicles is used to estimate the total rate $r_C(t)$, and the target channel load r_g is then mapped to an equivalent CBP. CBP is measured every δ time and the rate is adapted according to Eq. (4). More details are provided in [15].

In order to improve global fairness all vehicles contributing to congestion at a given location should participate in congestion control in a fair manner. For this purpose, LIMERIC uses the PULSAR[16] information dissemination functionality. PULSAR requires vehicles to piggyback a high precision CBP measurement on their safety messages. Thus, the CBP values used in the LIMERIC rate update equation is defined to be the maximum CBP reported by its 2-hop neighbors. With this approach, all vehicles running LIMERIC are contributing to congestion control more fairly.

IV. STABILITY CHALLENGE

One of the critical features for a channel congestion control protocol is scalability. In this paper, we examine the scalability of both approaches in terms of channel load stability through three different vehicle densities. Most of the figures include results from legacy IEEE 802.11p simulations without any congestion control present for benchmarking. We used simulation results for the stability analysis carried out in this section. A detailed simulation configuration is presented in Section VI.

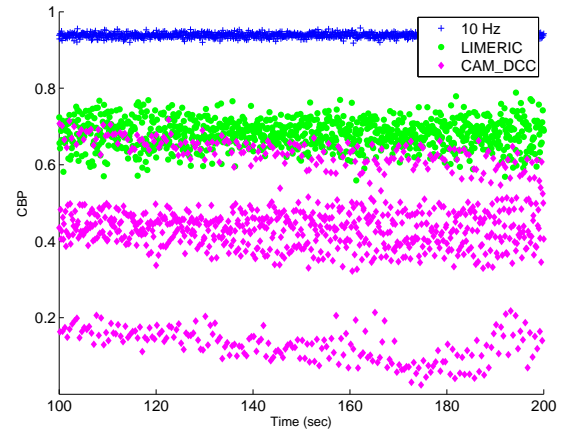


Fig. 3. CBP sampled at the winding part of the road versus time for various approaches.

Using Winding HW. Figure 3 illustrates the CBP values of the three studied schemes, where the figure shows CBP measured from when the simulation has run for 100 seconds and 100 sec onward (i.e., transients from the initialization phase are removed). The number of vehicles in these simulations is 1000. Each colored dot represents a CBP value sampled

every 100 msec. It is observed that the fixed 10 Hz scheme (no congestion control present) does not control the channel load and thus results in a constantly high CBP around 92%. LIMERIC converges to a predictable CBP value which is close to the defined target and is governed by Eq. (5) in Section III-C. It is also clearly seen that CBP for the DCC scheme does not converge; instead it oscillates in three levels within the range from 2% up to 70%.

Using Multi-Bridge HW. To confirm the CBP oscillation phenomenon from Figure 3, in a more realistic road-way topology, we created a scenario where a highway passes over two bridges with 3km separation between them. We call this scenario the multi-bridge highway. We started the simulation for this scenario in the ideal state at the beginning of the simulation. This helps to remove the transient phase in the beginning of the simulation and observe the emergence of the CBP oscillation. Figure 4 illustrates the CBP samples for a car moving from the left edge of the horizontal highway to the right side. The simulation ends when the aforementioned car reaches the second bridge. In this figure, the first three-level CBP oscillation is started when the group of cars contribute more message traffic to the interference region of the cars on the bridge. After some time, once the number of additional cars contributing to interference of the cars on the bridge starts to decrease (and the car passes the bridge), the CBP start to converge. Before a perfect level of convergence, the group of cars reaches the second bridge and the same CBP oscillation phenomenon is observed. This confirms our observation in Figure 3.

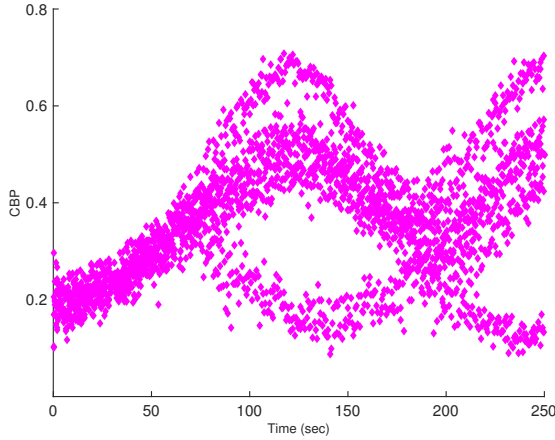


Fig. 4. CBP sampled at the winding part of the road versus time for various approaches.

Are these high CBP oscillations in the DCC scheme caused by frequent message rate changes? Continuing with the primary scenario (the winding highway scenario), Figure 5 shows that the DCC approach yields a constant message interval, which means that DCC never changes the message rate despite these large CBP oscillations. State changes are prevented by the windowing function of the CBP value in DCC (see section III-B). Specifically, a message interval of 0.5s indicates that DCC is in RESTRICTIVE state, the most restrictive state. It can only change to a lower state if the CBP remains below 60% for a period of T_{down} , 5 secs in our simulation. Due to the

CBP oscillations, CBP values over 60% occur within these 5 secs periods preventing possible state changes. Therefore, state changes are not the cause of the observed CBP oscillation.

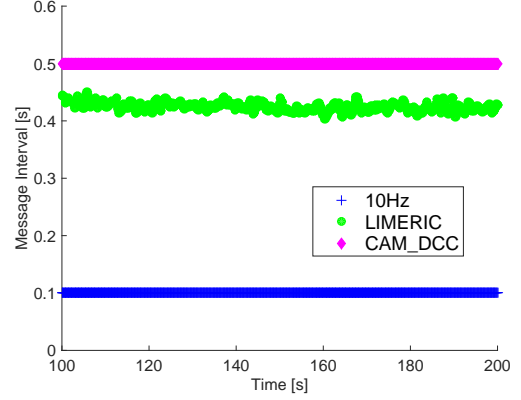


Fig. 5. Message interval for one static node located at the middle region

In Figure 6, the number of transmissions occurring in a randomly chosen one-second interval is plotted. The plot shows the results for LIMERIC and DCC for the 1000 vehicle case. The size of the time bins is 10 msec. For LIMERIC (Figure 6 (a)), the message transmissions are more or less uniformly distributed over time. However, the transmissions for the DCC approach appear in clusters (Figure 6 (b)). This clustering leads to many nodes transmitting at the same time, which results in a higher PER.

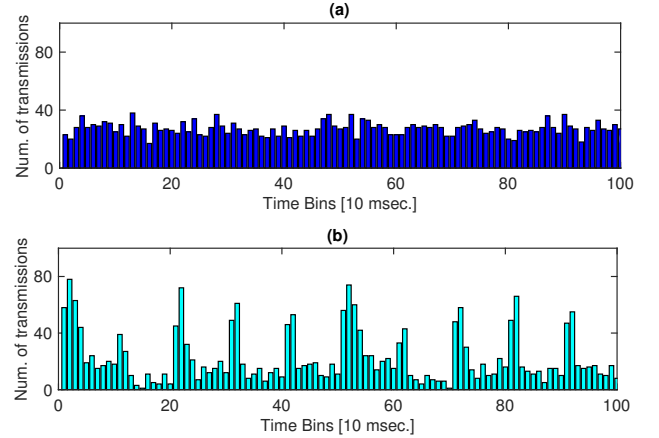


Fig. 6. The distribution of the number of transmissions in a randomly chosen one-second interval for 1000 nodes for (a) LIMERIC, (b) DCC

For DCC, Figure 7 shows a more detailed view of the channel load assessment for a single node located at the winding part of the road and the distribution of transmissions for all nodes for a 2 second snapshot. The left plot shows the CBP values perceived during these 2 seconds for a single node. In the right plot, all transmissions taking place during these 2 seconds are depicted. It is clearly seen that between two consecutive CBP measurement periods, the number of transmissions can change drastically (e.g., from 100 transmissions up to 350 transmissions), which is reflected in the CBP plot to the left.

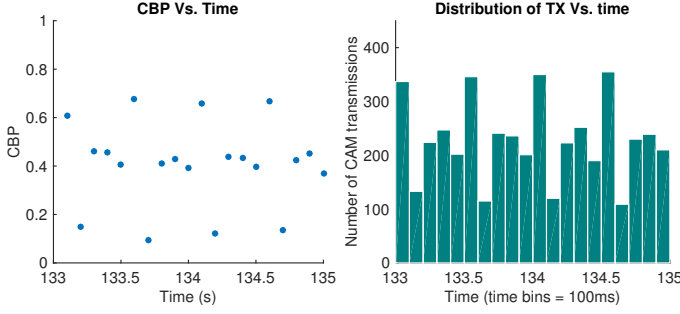


Fig. 7. Detailed view of CBP sampled at the winding part of the road versus time for DCC

These results suggest that DCC is not able to spread transmissions from all vehicles uniformly over time, but tends to create clusters of messages from vehicles that transmit simultaneously. This is highly inefficient because many packets collide during these simultaneous transmissions leading to the overall worse performance of DCC as shown in Section IV.

V. PROPOSED STABLE REACTIVE CONTROL

In this section, we investigate the reasons for detected channel load instability in the previous section for the reactive state-based DCC. Then, based on what the causes are, three alternative designs for the basic protocol is suggested.

A. Instability Analysis

DCC is susceptible to clustering of transmissions and CBP oscillations because of two main reasons:

1) *Synchronized CBP measurements with deterministic scheduling of transmissions*: When CBP measurement intervals are synchronized across vehicles, they will evaluate if a state change should take place at the same time (i.e., every 100 msec period). If many of these vehicles will choose to switch to the same rate at that time, the first transmissions of these vehicles will be scheduled at virtually the same time. While the $T_CheckCamGen$ parameter introduces a small amount of randomness in transmission scheduling, for large numbers of nodes the scheduling is still too deterministic.

2) *Limited choices for message rate*: Once a first cluster of simultaneous CAM transmission occurs, it is likely to recur on every subsequent transmission because of the few number of rates or message intervals (τ values are governed by the discrete values in DCC in Table 2). Nearby vehicles measure similar CBP and are therefore likely to choose exactly the same rate. Operating at the same rate means that future transmissions will remain clustered until a rate change occurs. The smaller the number of rate choices the higher the probability for choosing the same rate and maintaining the same rate over a long period of time.

To understand how synchronized CBP measurements and deterministic scheduling can create simultaneous transmissions in DCC an example is outlined in Figure 8 and it is explained subsequently. Assume vehicles A, B, and C are neighbors and generate a CAM each at times t_{0a} , t_{0b} , and t_{0c} . Further their current message rates are $rate_A$, $rate_B$, and

$rate_C$, thus the next CAMs are expected to be generated at $t_{0a} + \tau_A$, $t_{0b} + \tau_B$, and $t_{0c} + \tau_C$, respectively, where

$$\tau_A = \frac{1}{rate_A} \quad (6)$$

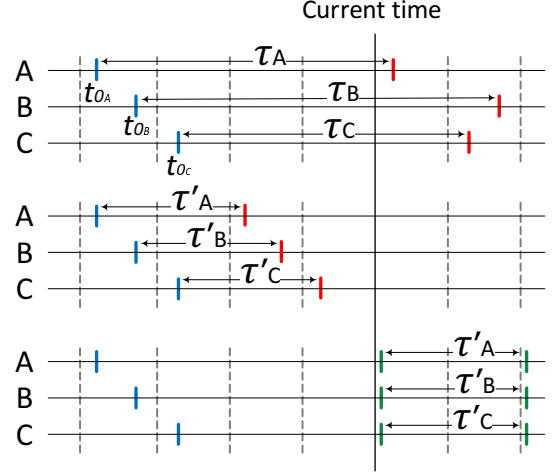


Fig. 8. Next CAM transmission schedule (up) before T_GenCam_Dcc update (middle) table look-up procedure (bottom) after T_GenCam_Dcc update

This is shown at the top portion of Figure 8. Note how the first and planned transmissions are spread out in time as expected. The dashed lines indicate the synchronized CBP measurement intervals. Let us assume that all three value of τ are larger than 200 msec, and hence a new CBP measurement becomes available before the planned CAM transmissions. This point in time is marked as current time in the figure. At this time all three vehicles reevaluate their message rate. If the CBP measurement is low, they will choose shorter message intervals τ'_A , τ'_B , and τ'_C , which changes the planned time for the next CAM generation. This new planned time is in the past, as shown in the middle part of the figure. Assuming that vehicles experience high dynamics, for all values of j that satisfy the Eq. (7), a CAM will be generated immediately. This is an example of deterministic scheduling, which leads to a simultaneous transmission.

Eq. (7) is as following:

$$\tau'_A \leq (j \times T_CBP_update) < \tau_A \quad (7)$$

Where j is the number of collected CBP measures after the last CAM generation until current time.

Further, after DCC enters the RESTRICTIVE state, DCC has no defined control in response to the increased channel load. This passive channel load handling in RESTRICTIVE state makes DCC unable to react to higher channel load until the congestion decreases because of the other parameters that DCC does not control, e.g., lower vehicle density. Figure 9 shows measured CBP values over time by a vehicle moving from the middle part of the road (most congested area) towards the right edge (less congested area). It can be seen that when the vehicle is approaching the edge (after 180 seconds of simulation), the channel congestion is decreasing, and DCC can finally change its state in FSM. Although, this passive

channel load handling cannot be considered as a cause for CBP oscillation, yet it acts as a magnifier for an already started CBP oscillation by hindering message rate changes.

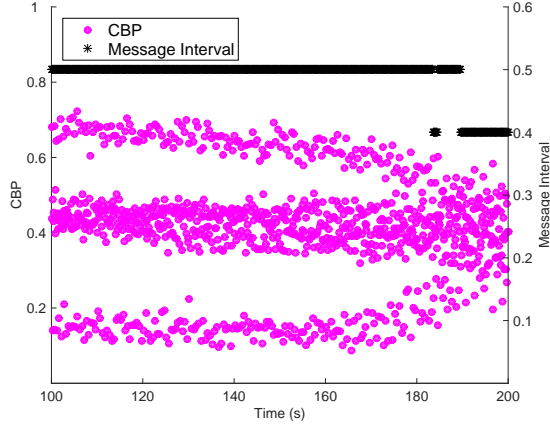


Fig. 9. CBP values and corresponding message rates for a DCC vehicle moving towards the edge of road

B. Stable Alternatives to Basic Reactive State-based Control

Based on the earlier observations, we now introduce three alternative designs for DCC based on two solutions for DCC channel instability causes that can eliminate the synchronized CBP measurements and the limited choice in message rates. First, we create asynchronous CBP measurements across all nodes by having each node select a random start time for CBP measurement intervals after the simulation begins. This modification should remove the synchronization in the first CAM transmission (as shown in Figure 8). Second, we modify the message rate selection, a continuous function is introduced for the ACTIVE state. This can be interpreted as increasing the number of ACTIVE sub-states N in the FSM to $N \rightarrow \infty$. This is implemented by replacing the look-up table with Eq. (8). Here, CBP_m is the CBP value that DCC obtains from its CBP history (see Eq. (1) and Eq. (2)). Note that the boundaries for the ACTIVE states are still preserved. The received message interval, τ , is communicated via the T_GenCam_Dcc parameter to the facilities layer.

$$\tau = \begin{cases} 0.1, & \text{if } CBP_m < 0.3. \\ (CBP_m \times \frac{0.5-0.1}{0.6-0.3}) - 0.3, & \text{if } 0.3 \leq CBP_m < 0.6 \\ 0.5, & \text{if } CBP_m \geq 0.6. \end{cases} \quad (8)$$

These two modifications lead to three variants of the DCC algorithm, which are outlined in Table 3. The DCC with synchronized CBP measurements with the table look-up is kept as a benchmark and it is called *Synch-Step*, i.e., same as used in Section VI.

VI. SIMULATION RESULTS

Due to the high cost of experimenting with thousands of vehicles, we evaluate our work through simulation. In

TABLE 3
FOUR VARIANTS OF THE DCC ALGORITHM

Name	Description
Synch-Step	Synchronized CBP measurements and original table look-up (Table 2).
Asynch-Step	Asynchronous CBP measurements and original table look-up (Table 2).
Synch-Continuous	Synchronized CBP measurements and the continuous function (Eq. (8)).
Asynch-Continuous	Asynchronous CBP measurements and the continuous function (Eq. (8)).

this section, first we validate our reasoning for channel load oscillations through simulation results for all the four DCC alternatives introduced in previous section (including the basic reactive state-based DCC). Then we choose the one with better results among proposed stable reactive approaches. The comparison between stable reactive, linear adaptive, and nave schemes is presented as well. To carry out the simulations, the event-based, open source network simulator ns-2 [36] is used in this paper.

As performance metrics the Packet Error Rate (PER), the 95th percentile Inter Packet Gap (IPG), and the Tracking Error (TE), are used. TE is defined as the error between the transmitter's true location and the receiver's perception of the transmitter's location. The receiver extrapolates the transmitter's location using the GPS information in the most recently received position message (i.e., CAM/BSM) and uses a constant-speed constant-heading coasting model. The IPG and the TE are related. IPG measures the time between two consecutive received packets and is an important metric since it characterizes how frequently a vehicle is able to receive information from other vehicles. The TE, on the other hand, is an application-oriented measure of how accurately a receiving vehicle can track the movements of a sending vehicle.

A. Simulation configurations

As briefly mentioned before, the SUMO mobility simulator [37] has been used to create mobility traces for the two different road typologies. We mainly used the first road topology, named winding highway shown in Figure 10, to keep the focus on a small area where the vehicle dynamics are high enough to trigger CAM generation rules at facilities layer. We also used another road topology named multi-bridge highway, illustrated in Figure 11, where two bridges cross a highway segment. This scenario introduces a more realistic road topology and initialization of the vehicles. We mainly used the latter to gain confidence that the results from the winding highway scenario will hold in practice.

Winding Highway. For the primary road topology, a highway of length 4 km, with 3 lanes in each direction. The middle part of the road is a winding section of length 375 m (with the radius of the winding part set to be 40 m), see Figure 10. This configuration permits testing of the performance of the algorithms (i.e., DCC and LIMERIC) not only on a straight road where vehicles have relatively low dynamics but also

on a winding part of the road where vehicles experience high dynamics. This is important since the CAM generation depends on vehicle dynamics. Three different vehicle densities have been used: 500, 1000, and 1500 vehicles at the same time on the highway, respectively. The average desired speed of the vehicles on the three lanes on the highway are 19 m/s in the fastest lane (left lane), 18 m/s in the middle lane, and 17 m/s in the slowest lane (right lane). However note that SUMO reduces the speed as the vehicle density increases on the road. For the highest vehicle density of 1500 vehicles, the average speed is around 10-13 m/s.

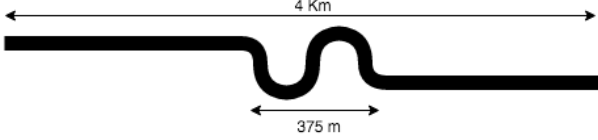


Fig. 10. Road topology for simulations as in [38]

Multi-bridge Highway. In the multi-bridge scenario, a group of vehicles are moving from the leftmost part of the horizontal highway segment towards the right edge of the highway. This scenario is inspired by a real highway setup in New Jersey, USA in terms of the distance between the two bridges. In the real world setup, the Garden State Parkway highway and the Interstate 287 highway pass above the US 1 highway. To keep both road topologies consistent, the road configuration for the multi-bridge highway scenario, such as number of lanes and maximum speed criteria for each lane, is similar to the winding highway scenario. The traffic on both bridges is moderate (33 vehicles per lane per Km). The scenario was chosen to expose vehicle to repeatedly changing channel load as they move along the highway (high load near the bridges due to additional cross traffic).

Under such circumstances, it can be shown that even starting with an ideal parameter setup where there is no clustered CAM generation among vehicles in the beginning of simulation, they eventually form and fall into clusters as the group travels through high and low CBP spots.

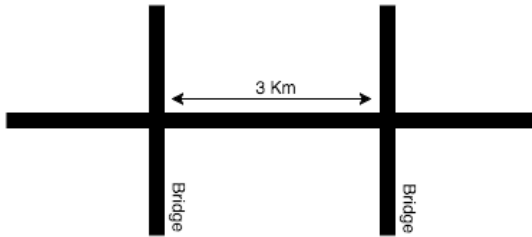


Fig. 11. A more realistic multi-bridge scenario

The wireless channel propagation is Nakagami distributed, with the same parameters as in [38]. The list of simulation parameters used in this work is given in Table 4 (also, note that the parameters for CAM generation and European DCC are used as specified in Section II, III). For the results in Section IV.B, synchronized CBP measurement periods across all vehicles have been used, i.e., each node measures the

CBP locally, but all nodes measure CBP at the same time. In vehicular networks all nodes have access to GPS clock and hence, it is possible to make synchronized measurements. This implies that in this work, both DCC and LIMERIC will update their message rate once a new synchronized CBP measurement is available.

TABLE 4
SIMULATION PARAMETERS

Parameter	Value
Noise floor	-99 dBm
Carrier sense threshold	-96 dBm
Packet Reception SINR (for 6Mbps datarate)	7 dBm
CW_{min}	15
AIFSN	2
Facilities layer payload	350 byte
Transmission Rate	6 Mbps
Transmission Power	10 dBm
GPS Update Frequency	10 Hz
CBP measurement period (T_{CBP_update})	100 msec
Simulation time	200s in Winding HW 250s in Multi-bridge HW
LIMERIC	
δ	200 msec
Goal CBP Parameter	79%
β	0.033
α	0.1
CAM Generation	
CAM generation rules checking period	10 msec

B. Evaluation of Stable Alternatives

Figures 12 - 14 show the packet error rate, 95th percentile inter-packet gap, and 95th percentile tracking error for each of these variants. The number of vehicles in the simulation is 1000. It is evident that all three alternatives achieve improved performance in all metrics compared to the *Synch-Step* approach. Although all the curves corresponding to the alternative designs are close to each other, the largest improvement is generally achieved by *Async-Continuous*, which eliminates both suspected causes for clustered transmissions.

Considering Figures 12 - 14, it also can be seen that *Async-Step* is performing slightly better than *Synch-Continuous*. Note, however, that it is not straightforward to guarantee asynchronous CBP intervals in practice. If CBP intervals are simply randomized, as in our simulations, there still remains a residual probability that accidental synchronized measurements occur. This could then lead to synchronized CAM generation and the observed performance degradation.

These simulation results support the observation that synchronized CBP measurements and a limited number of rate choices are key factors that lead to undesirable clustering of transmissions and degraded congestion control algorithm performance.

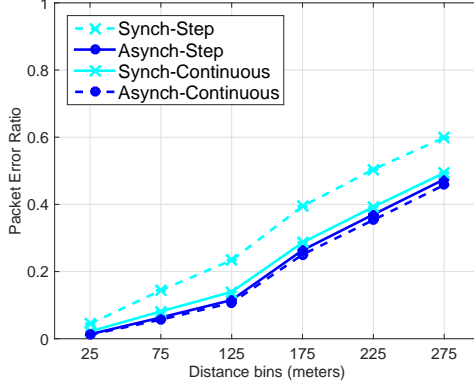


Fig. 12. PER for different alternatives, total number of vehicles = 1000

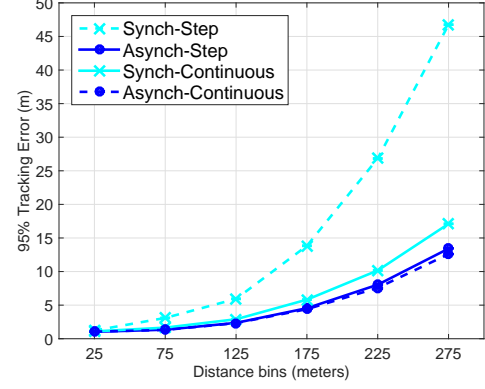


Fig. 14. 95th percentile Tracking Error for different alternatives, total number of vehicles = 1000

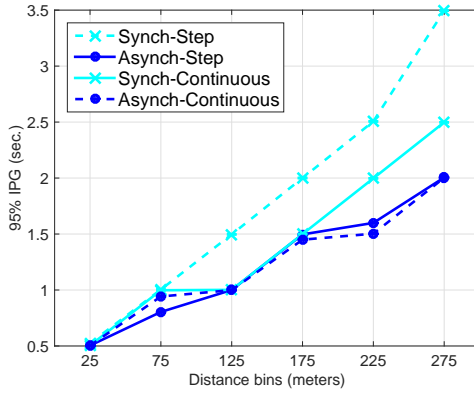


Fig. 13. 95th percentile IPG for different alternatives, total number of vehicles = 1000

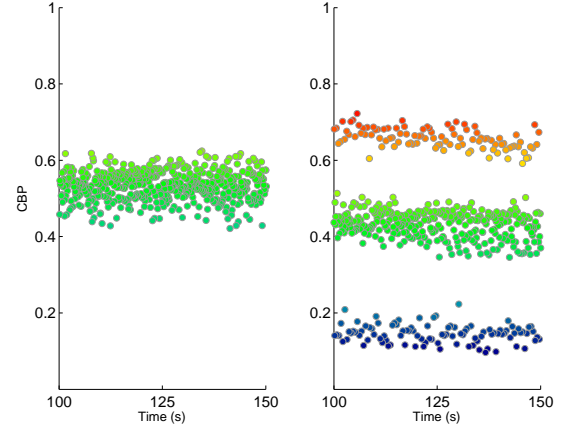


Fig. 15. CBP measures at the winding part of the road vs. time for: Asynch-Continuous (left) Synch-Step (right)

Figure 15 shows the CBP comparison between *Asynch-Continuous* (left plot) and *Synch-Step* (or default DCC in right plot). The improvement is clearly seen in the figure.

C. Comparison with 10Hz and LIMERIC

In Table 5, the three different data traffic models settings that have been used in the simulations are tabulated. They are fixed 10 Hz BSM/CAM transmissions (the legacy IEEE 802.11p with no congestion control present), the LIMERIC algorithm generating safety messages for transmission when allowed by LIMERIC, representing linear adaptive approach, and the stable reactive DCC approach from previous section (the one with label Asynch-Continuous), where CAMs are generated according to EN 302 636-2 based on vehicle dynamics adhering to the T_GenCam_Dcc parameter (T_GenCam_Dcc is determined by the DCC in the access layer).

All performance metrics are calculated based on transmissions carried out on the winding part of the road (Figure 10). That is, if the transmitter is on the winding part of the road the transmission is accounted for regardless of whether the receiver is on the winding part or not. The distance between transmitter and receiver determines in which distance bin the transmission (successful/unsuccessful) is counted. The size of the distance bins is set to 50 m.

In Figure 16, the PER is depicted for the three different algorithms (Table 5) over the three different vehicle densities of 500, 100, and 1500 nodes over 4 Km road. Figure 17 shows the 95th percentile IPG from the same set of simulations.

TABLE 5
CONGESTION CONTROL APPROACHES USED IN THE SIMULATIONS

Name	Description
10 Hz	There is no congestion control algorithm present and all vehicles transmit CAM/BSM 10 times per second. This setting is the baseline, and using the legacy IEEE 802.11p
LIMERIC	The vehicles generate and transmit CAM/BSMs when LIMERIC algorithm allows.
nDCC	The vehicles generate CAMs according to EN 302 637-2 (also described in Section II), which is based on vehicle dynamics. CAMs are generated when the T_GenCam_Dcc parameter allows. The DCC is <i>Asynch-Continuous</i> from previous section

As expected, 10 Hz transmissions without congestion control (called 10 Hz in the figures) has the highest PER for all vehicle densities. The fixed 10 Hz scheme does not control the channel load, hence its PER increases with the node density.

High PER, translates into large inter packet gaps for medium to large distances between the transmitter and the receiver. For shorter distances, the 10 Hz scheme approximates and in some cases has a better IPG performance compared to the congestion control algorithms, including LIMERIC. This can be observed particularly in the bars associated with 500 and 1000 node densities in Figure 17, and can be explained with the capture effect. At a small range, the received power tends to be high compared to the interfering signals and the transmission can often still be correctly received.

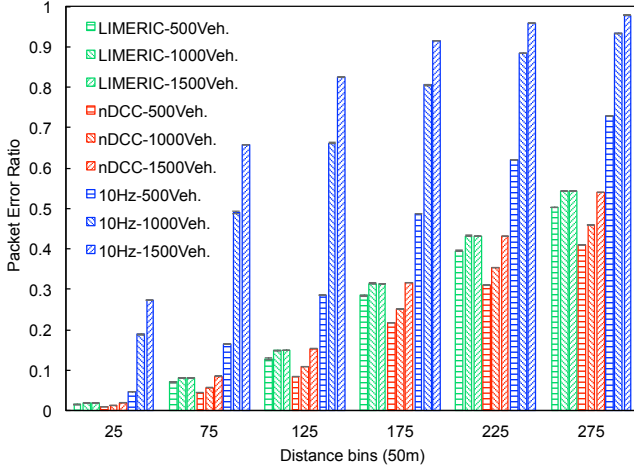


Fig. 16. Packet Error Ratio

However, for larger distances the received power is too low compared to the interfering signals and the transmission results in an error. This can be seen in the bars associated with 1000 and 1500 node densities in Figure 17, where the 95th percentile IPG of the 10 Hz transmission approach becomes quite high at medium and large distances and it has worse performance than LIMERIC, which adaptively controls the channel load. In terms of 95th percentile IPG, LIMERIC shows better performance than the reactive approach, and outperforms the legacy IEEE 802.11p across all simulations and metrics. Employing a congestion control mechanism decreased IPG, perhaps the most important metric among all three performance parameters, by a factor of 2x to 8x for larger distance bins, depending on the distance between transmitter and receiver.

The PER performance of DCC is slightly lower than the linear adaptive approach at the lower vehicle density (see Figure 16 for the bars with 500 and 1000 vehicles in comparison to the bar associated with 1500 vehicles). This is mainly because of the nature of the reactive approach. A reactive congestion control does not try to push the channel load towards a predefined, near optimum channel load. Instead, it uses predefined CBP to message rate look-up table to determine its message transmission ratio.

Interesting to note is that LIMERIC has the same PER throughout all vehicle densities, which is in line with LIMERIC's aim of converging to a CBP target (i.e., increase throughput) allowing for a higher message rate for each individual vehicle in lower vehicle densities and vice versa.

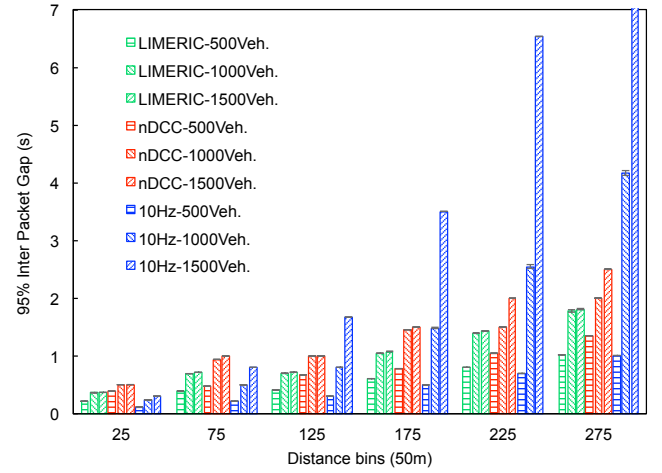


Fig. 17. 95th percentile Inter Packet Gap

This leads to higher IPG value as the vehicle density increases. Note, this is not a sign of increased errors but simply due to LIMERIC decreasing the message rate to increase the total throughput.

In Figure 18, the 95th percentile tracking error for all three schemes for 500, 1000, and 1500 node density are plotted. As with the IPG performance, LIMERIC shows a better performance. The difference between DCC and LIMERIC is particularly pronounced at lower densities and diminishes at higher densities. Recall that this is due to vehicles lower speed at higher vehicle densities. It can also be noticed that fixed 10 Hz transmission has good performance at low node density (500 node case), but as the node density increases the PER becomes high which also leads to higher tracking errors.

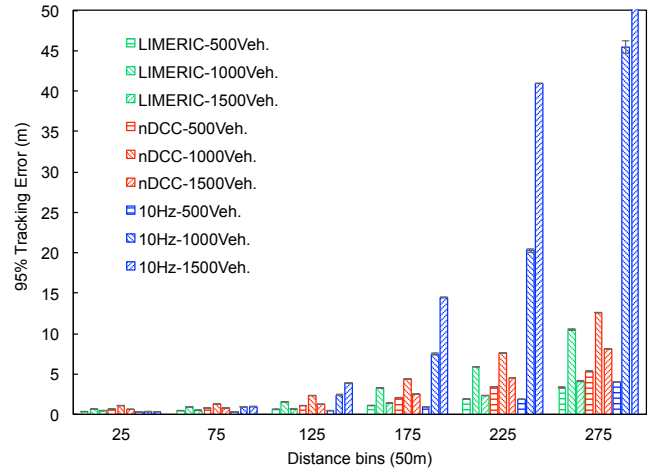


Fig. 18. 95th percentile Tracking Error

Figure 19 illustrates the 95th percentile IPG for the central part of the left bridge. Each color represents one of the three congestion control mechanisms. Since the multi-bridge scenario has a unique density of vehicles (1200 nodes), there are only three bars for each distance bin. Comparing Figure 19 with Figure 17, it can be clearly seen that the multi-bridge

scenario results are consistent with the ones from winding highway scenario.

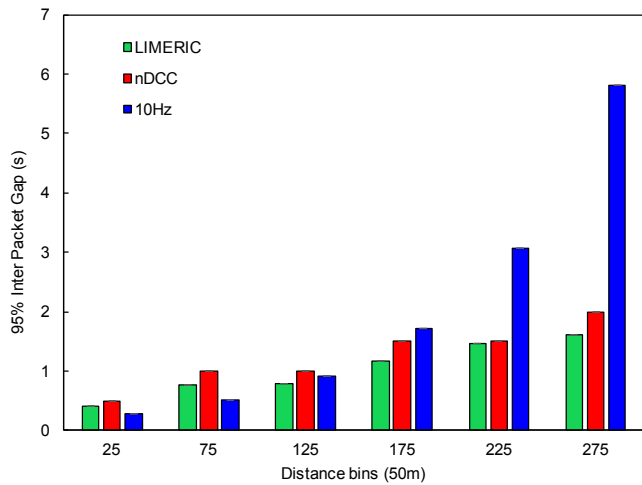


Fig. 19. 95th percentile Inter Packet Gap for the Multi-Bridge scenario

Although the CBP oscillation problem can be resolved through employing one of the proposed DCC variants outlined here, the reactive nature of the DCC will still end up in an inefficient channel usage. Given that throughput is increased when CBP is maintained at a specific level, DCC is not able to increase throughput for every vehicle density. Although for current parameter values, the performance of improved DCC is very close to LIMERIC when the vehicle density is 1000, yet results for 500 and 1500 vehicle density in Figure 17, are two examples where LIMERIC is still able to have more frequent successful receptions than all four variants of DCC.

VII. DISCUSSION

A. Reasons for Synchronized CBP Measurements

Given the observation from section V that synchronized CBP measurement periods can lead to significantly degraded DCC performance, let us discuss in more detail why DCC was implemented with such synchronized measurements.

Synchronized CBP measurements have been used in the LIMERIC algorithm primarily because it uses the PULSAR information sharing mechanism [16] to improve fairness, which in turn relies on synchronized CBP measurements. The PULSAR protocol proposes the principle that nodes which are contributing to congestion should participate in mitigating it. Since nodes can contribute to congestion outside their direct packet reception range, it uses a 2-hop CBP sharing protocol among the nodes to allow monitoring congestion in a larger area than the immediate packet reception range. This mechanism relies on the individual local CBP measurements being taken at the same time. In addition, LIMERIC has proven convergence and fairness properties when message rate updates are synchronized (i.e., synchronized CBP measurement intervals) for all nodes [15]. While there are also analytical insights to suggest that LIMERIC converges in the asynchronous CBP case as well, a complete proof does not yet

exist. Therefore, the LIMERIC simulations conducted here are implemented with synchronized CBP measurement periods.

This study chose synchronized CBP measurements for DCC for two reasons. First, a consistent CBP measurement approach for LIMERIC and DCC allows a precise and fair comparison of algorithm behavior given the same CBP inputs. The ETSI standard for the DCC algorithm [14] is ambiguous and is open to interpretation with regard to synchronized or asynchronous CBP measurement periods. Given that the LIMERIC implementation relies on synchronized measurements, it appeared reasonable to choose synchronized measurements for both algorithms. Second, ETSI has also identified the necessity of disseminating CBP measurements as in LIMERIC to strive towards a more fair system. While DCC for so-called day one C-ITS applications will not disseminate CBP measurements among neighboring nodes, ETSI standardization has prepared for the possibility to disseminate CBP values among neighboring nodes as part of the GeoNetworking protocol header in the future. The procedure is outlined in TS 102 636-4-2 [39]. It is also based on the PULSAR principle requiring synchronized CBP measurement periods. This would suggest that CBP measurements periods should be synchronized even in day one applications to maintain upward compatibility with such future deployments. For these reasons, this study primarily used DCC with synchronized measurements in Section VI.

B. Choice and Impact of the $T_CheckCamGen$ Parameter

For all DCC results in this study, the parameter $T_CheckCamGen$, in the CAM generation Algorithm 1 is set to 10 msec. This is consistent with EN 302 637-2 [10] outlining the CAM generation rules, which states that the value of $T_CheckCamGen$ shall be equal or less than 100 msec. We chose a value at the lower end of this range because a larger value would result in delayed CAM generation and could disadvantage DCC on delay metrics such as IPG.

The analysis in section V suggests that the role of this parameter may be more important than apparent from the standard documents. If the $T_CheckCamGen$ periods are not synchronized across nodes, this process adds an element of randomness in the otherwise deterministic scheduling of transmissions and reduces clustered transmissions. This parameter effectively controls the maximum time delay before all nodes react to the updated rate conveyed by the access layer through the parameter T_GenCam_Dcc . If nodes in the vicinity are experiencing a similar CBP change that leads to a clustered first transmission at the new rate, $T_CheckCamGen$ controls how dense the cluster becomes in time. For a value of 0, all nodes would transmit simultaneously, while for a value of 100 msec, the transmissions would be spread over a window of 100 msec. Larger values can therefore be expected to reduce clustering and its undesirable effects such as high collision rates. Yet, even the maximum permitted value of 100 msec is not sufficient to spread the transmissions over the entire update period, which can be as long as 500 msec at the RESTRICTIVE rate. Large values may also cause undesirable transmission delays in less congested scenarios, which suggests that using this parameter to mitigate clustering is not ideal.

VIII. CONCLUSION

This paper compares two C-ITS/DSRC congestion control approaches: a reactive approach represented by the ETSI DCC framework and an adaptive approach represented by the LIMERIC algorithm. DCC is required by regulation in Europe (EN 302 571) and is situated in the access layer on top of IEEE 802.11p. It can restrict the CAM generation in the facilities layer via the management plane when the channel load increases. In the US, the congestion control algorithm has not yet been standardized, but LIMERIC is under consideration as a candidate.

In our simulation model, LIMERIC consistently achieves lower reception intervals (inter packet gap) and tracking error than the DCC approach. This is in part due to LIMERIC's ability to target a channel load that results in high throughput and awareness, independent of vehicle density. In the DCC approach, the channel load will vary with vehicle density, which can lead to reduced performance. On the other hand, LIMERIC implementation is more complex than a lookup table approach such as European DCC, since the LIMERIC nodes are equipped with information sharing mechanism.

The superior LIMERIC performance is also due to its ability to efficiently spread messages over time, while DCC is susceptible to clustering of messages in time. We show that while LIMERIC has stable channel load convergence, DCC shows large oscillations degrading the performance, particularly when synchronized CBP measurements are used. The small number of discrete states in DCC (seemingly attractive due to its simplicity), where a range of CBP values map to one rate, inhibits DCC's ability to effectively spread transmissions in time. If the transmissions of nodes become synchronized, the small number of rates causes this synchronization to be maintained for extended periods of times. We demonstrate this through the use of a continuous function for rate adjustments for the ACTIVE states of DCC, which increases rate diversity among nodes and reduces clustering and its inefficiencies.

An implementation with asynchronous CBP measurements also decreases the probability of harmful synchronized patterns and substantially improves DCC performance. However, the numerical results show that even improved variants of DCC cannot match the performance of a true adaptive algorithm such as LIMERIC. Reactive algorithms lack the ability to converge to the optimal channel load independent of the vehicle density, which results in under utilization of the channel at lower vehicle densities or a congested channel at higher vehicle densities.

The results suggest that such factors leading to degraded performance deserve more attention in reactive approaches. For example, ETSI standardization on DCC in TS 102 687 does not mandate if the CBP measurement periods should be synchronized or asynchronous. However, the possible inclusion of information sharing in the GeoNetworking header in the future might require synchronized CBP measurement periods. The results also suggest that vehicular networks employing adaptive congestion control are more robust to CBP implementation alternatives and significantly outperform reactive congestion control.

A recent ETSI DCC standard includes "two algorithms capable of satisfying" DCC requirements [40], one adaptive algorithm and one reactive algorithm. The adaptive algorithm is LIMERIC, and its inclusion was influenced by some of the results presented in this paper, as well as our latest work in [41]. The reactive algorithm is the state-based reactive approach previously published in [14].

REFERENCES

- [1] *IEEE Standard for Information technology—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE Std. 802.11p, July 2010.
- [2] "MoU for OEMs within the CAR 2 CAR communication consortium on deployment strategy for cooperative ITS in Europe," <https://www.car-2-car.org/index.php?id=231>.
- [3] NHTSA, *Federal motor vehicle safety standards: Vehicle-to-Vehicle (V2V) communications advance notice of proposed rulemaking*, Std., August 15, 2014.
- [4] *IEEE Standard for Wireless Access in Vehicular Environments—Networking Services*, IEEE Std. 1609.3-2010, December 2010.
- [5] *IEEE Standard for Wireless Access in Vehicular Environments—Security Services*, IEEE Std. 1609.2-2013, April 2013.
- [6] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [7] S. Subramanian, M. Werner, S. Liu, J. Jose, R. Lupoie, and X. Wu, "Congestion control for vehicular safety: synchronous and asynchronous MAC algorithms," in *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*. ACM, 2012, pp. 63–72.
- [8] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed co-ordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, 2000.
- [9] T. V. Nguyen, F. Baccelli, K. Zhu, S. Subramanian, and X. Wu, "A performance analysis of CSMA based broadcast protocol in VANETs," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2805–2813.
- [10] ETSI EN 302 637-2, V1.3.2, "Intelligent Transport Systems (ITS); Vehicular communications; Basic set of applications; Part 2: Specification of cooperative awareness basic service," 2014.
- [11] *Dedicated Short Range Communications (DSRC) message set dictionary*, Society of Automotive Engineers (SAE) Std. J2735, November 2009.
- [12] C. Lochert, B. Scheuermann, and M. Mauve, "A survey on congestion control for mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 7, no. 5, pp. 655–676, 2007.
- [13] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2002, pp. 194–205.
- [14] ETSI TS 102 687, V1.1.1, "Intelligent Transport Systems (ITS): Decentralized congestion control mechanisms for intelligent transport systems operating in the 5 GHz range; Access layer part," July 2011.
- [15] G. Bansal, J. B. Kenney, and C. E. Rohrs, "LIMERIC: A linear adaptive message rate algorithm for DSRC congestion control," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 9, pp. 4182–4197, 2013.
- [16] T. Tielert, D. Jiang, Q. Chen, L. Delgrossi, and H. Hartenstein, "Design methodology and evaluation of rate adaptation based congestion control for vehicle safety communications," in *Vehicular Networking Conference (VNC), 2011 IEEE*, 2011, pp. 116–123.
- [17] D. Eckhoff, N. Sofra, and R. German, "A performance study of cooperative awareness in ETSI ITS G5 and IEEE WAVE," in *Wireless On-demand Network Systems and Services (WONS), 2013 10th Annual Conference on*. IEEE, 2013, pp. 196–200.
- [18] S. Kuk and H. Kim, "Preventing unfairness in the ETSI distributed congestion control," *Communications Letters, IEEE*, vol. 18, no. 7, pp. 1222–1225, 2014.
- [19] G. Bansal, B. Cheng, A. Rostami, K. Sjöberg, J. B. Kenney, and M. Gruteser, "Comparing LIMERIC and DCC approaches for VANET channel congestion control," in *Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on*, Sept 2014, pp. 1–7.

- [20] V. G. Cerf and R. E. Kahn, "A protocol for packet network intercommunication," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, pp. 71–82, 2005.
- [21] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *Networking, IEEE/ACM Transactions on*, vol. 1, no. 4, pp. 397–413, 1993.
- [22] M. Alizadeh, B. Atikoglu, A. Kabbani, A. Lakshmikantha, R. Pan, B. Prabhakar, and M. Seaman, "Data center transport mechanisms: Congestion control theory and IEEE standardization," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 1270–1277.
- [23] ETSI TR 101 612 V1.1.1, "Intelligent Transport Systems (ITS); Cross layer DCC management entity for operation in the ITS G5A and ITS G5B medium; Report on cross layer DCC algorithms and performance evaluation," September 2014.
- [24] K. Chen, K. Nahrstedt, and N. Vaidya, "The utility of explicit rate-based flow control in mobile ad hoc networks," in *Wireless Communications and Networking Conference (WCNC)*, vol. 3. IEEE, 2004, pp. 1921–1926.
- [25] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless networks*, vol. 8, no. 2-3, pp. 153–167, 2002.
- [26] F. Farnoud and S. Valaee, "Reliable broadcast of safety messages in vehicular ad hoc networks," in *INFOCOM*. IEEE, 2009, pp. 226–234.
- [27] C. L. Huang, Y. P. Fallah, R. Sengupta, and H. Krishnan, "Intervehicle transmission rate control for cooperative active safety system," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 645–658, 2011.
- [28] J. Jose, C. Li, X. Wu, L. Ying, and K. Zhu, "Distributed rate and power control in ds-ss," in *Information Theory (ISIT), 2015 IEEE International Symposium on*, June 2015, pp. 2822–2826.
- [29] T. Tielert, D. Jiang, H. Hartenstein, and L. Delgrossi, "Joint power/rate congestion control optimizing packet reception in vehicle safety communications," in *Proceeding of the tenth ACM international workshop on Vehicular inter-networking, systems, and applications*. ACM, 2013, pp. 51–60.
- [30] M. Sepulcre, J. Gozalvez, O. Altintas, and H. Krems, "Adaptive beaconing for congestion and awareness control in vehicular networks," in *Vehicular Networking Conference (VNC), 2014 IEEE*. IEEE, 2014, pp. 81–88.
- [31] A. Autolitano, C. Campolo, A. Molinaro, R. M. Scopigno, and A. Vesco, "An insight into decentralized congestion control techniques for VANETs from ETSI TS 102 687 V1. 1.1," in *Wireless Days (WD), 2013 IFIP*. IEEE, 2013, pp. 1–6.
- [32] N. Nasiriani, Y. P. Fallah, and H. Krishnan, "Stability analysis of congestion control schemes in vehicular ad-hoc networks," in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*. IEEE, 2013, pp. 358–363.
- [33] A. Alonso and C. Mecklenbräuer, "Effects of co-located transmissions in the performance of DCC IEEE802.11p MAC and self-organizing TDMA for VANETs," in *Proceeding 5th MC and Scientific Meeting COST IC1004*, Bristol, September 2012.
- [34] S. Yang, H. Kim, and S. Kuk, "Less is more: need to simplify ETSI distributed congestion control algorithm," *Electronics Letters*, vol. 50, no. 4, pp. 279–281, 2014.
- [35] ETSI EN 302 571, V1.2.1, "Intelligent Transport Systems (ITS); Radio communications equipment operating in the 5 855 MHz to 5 925 MHz frequency band; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive," September 2013.
- [36] "Network Simulator ns-2," <http://www.isi.edu/nsnam/ns/>, 2013.
- [37] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of Urban MObility," in *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, 2011.
- [38] G. Bansal, H. Lu, J. B. Kenney, and C. Poellabauer, "EMBARC: Error Model Based Adaptive Rate Control for vehicle-to-vehicle communications," in *Proceeding of the tenth ACM international workshop on Vehicular inter-networking, systems, and applications*, 2013, pp. 41–50.
- [39] ETSI TS 102 636-4-2, V1.1.1, "Intelligent Transport Systems (ITS); Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 2: Media-dependent functionalities for ITS-G5," October 2013.
- [40] ETSI TS 103 175, V1.1.1, "Intelligent Transport Systems (ITS); Cross layer DCC management entity for operation in the ITS G5A and ITS G5B medium," June 2015.
- [41] B. Cheng, A. Rostami, M. Gruteser, J. B. Kenney, G. Bansal, and K. Sjöberg, "Performance evaluation of a mixed vehicular network with CAM-DCC and LIMERIC vehicles," in *Proceedings of Workshop of Smart Vehicles at IEEE WoWMoM*. IEEE, 2015, pp. 1–6.



the time, his research focuses on channel congestion control in vehicular networks.



Bin Cheng received the B.E. degree (high honors) from East China University of Science and Technology, Shanghai, China; the M.S. degree in control theory and control engineering from Shanghai Jiao Tong University, Shanghai, China. He is currently working toward the Ph.D. Degree in computer engineering at Electrical and Computer Engineering Department, Rutgers University, New Jersey, USA. His research interests include vehicular wireless communication, DSRC channel congestion control. He is a student member of the IEEE.



Gaurav Bansal received a B.Tech. degree from Indian Institute of Technology (IIT) Kanpur, India and a Ph.D. degree from the University of British Columbia (UBC), Canada. From August 2007 to July 2008, he worked as a Research Intern with Mercedes Benz Research and Development North America Inc., Palo Alto, CA. He joined Toyota InfoTechnology Center, USA, in July 2010 where he currently works as a Senior Researcher in the Network Group. He is a recipient of awards including a Best Paper Award at IEEE WiVEC 2013,

Best Internal project award at Toyota InfoTechnology Center USA and an Alexander Graham Bell Scholarship by Natural Science Engineering and Research Council of Canada. He also currently serves as an Editor for IEEE Communication Surveys and Tutorials journal and has served on the Technical Program Committee of various international conferences.



Katrin Sjöberg works as connected vehicle technology specialist at Volvo Group Trucks Technology in Gteborg, Sweden. She is working with wireless access to the vehicle both short-range (e.g., WiFi, 802.11p, ZigBee) as well as long-range wireless technologies (i.e., 3G/4G) for connecting the vehicle. Her research interest ranges from channel modeling to applications within connected automation (e.g., platooning and CACC). She is actively contributing to V2V standardization in Europe within ETSI Technical Committee on ITS (where she is holding a vice chairmanship of Working Group 4) and to the CAR-2-CAR Communication Consortium. In April 2013, she defended her PhD thesis Medium Access Control for Vehicular Ad Hoc Networks at Chalmers University of Technology, Sweden.



Marco Gruteser received his MS and PhD degrees from the University of Colorado in 2000 and 2004, respectively, and has held research and visiting positions at the IBM T. J. Watson Research Center and Carnegie Mellon University. His recognitions include an NSF CAREER award, a Rutgers Board of Trustees Research Fellowship for Scholarly Excellence, a Rutgers Outstanding Engineering Faculty Award, as well as best paper awards at ACM MobiCom 2012, ACM MobiCom 2011 and ACM MobiSys 2010. His work has been regularly featured

in the media, including NPR, the New York Times, Fox News TV, and CNN TV. He is an ACM Distinguished Scientist. He is currently a Professor of Electrical and Computer Engineering as well as Computer Science (by courtesy) at Rutgers University's Wireless Information Network Laboratory (WINLAB). He directs research in mobile computing, is a pioneer in the area of location privacy and recognized for his work on connected vehicles.



John B. Kenney received the B.S. (high honors) and Ph.D. degrees in electrical engineering from the University of Notre Dame, Notre Dame IN, and the M.S. degree (NSF Graduate Fellow) from Stanford University, Stanford CA. He directs the Networking Research Division at the Toyota InfoTechnology Center in Mountain View CA. His research interests include DSRC congestion control, communication performance, and spectrum sharing. He represents Toyota in industry forums, government hearings, and international standards organizations, including IEEE, SAE, and ETSI. He is Associated Editor of IEEE Vehicular Technology Magazine, and has served as general co-chair of ACM VANET and IEEE SmartVehicles workshops.