A Real-Hardware Multi-Instance Evaluation of ETSI DCC

Diego Gasco*, Giuseppe Perrone*, Francesco Raviglione†, Marco Rapelli*, Claudio Casetti*, Alessandro Bazzi‡

* Department of Control and Computer Engineering, Politecnico di Torino, Italy
† Department of Electronics and Telecommunications, Politecnico di Torino, Italy
‡ DEI, Università di Bologna, 40136 Bologna, Italy
{diego.gasco, giuseppe.perrone, francesco.raviglione, marco.rapelli, claudio.casetti}@polito.it
{alessandro.bazzi}@unibo.it

***Abstract*—Decentralized Congestion Control (DCC) is a key mechanism for guaranteeing scalable and reliable Vehicle-to-Everything (V2X) communications, especially in dense traffic scenarios. Despite several analytical and simulation-based studies, extensive experimental evaluations with real V2X hardware of complete standard-compliant DCC implementations remain limited. This paper presents an experimental assessment of DCC based on a fully standard-compliant, open-source implementation integrated into the OScar V2X software stack. The proposed system implements the complete DCC architecture, including Access-Layer (DCC_ACC), Network-Layer (DCC_NET), and Cross-Layer (DCC_CROSS) components, and is evaluated using a controlled laboratory testbed composed of nine On-Board Units (OBUs), each carrying three instances of the protocol stack, emulating a realistic vehicular scenario with vehicles traveling as a platoon. Mobility and perception dynamics are faithfully reproduced through the replay of traces collected during real-world highway driving experiments. Experimental results under varying network load highlight the effectiveness of adaptive and reactive congestion control algorithms in maintaining channel stability and reliable message dissemination, while also revealing practical performance trade-offs in dense vehicular environments.**

***Index Terms*—V2X, Vehicular networks, Congestion Control, DCC, Field tests**

I. Introduction

Vehicle-to-Everything (V2X) communications enable cooperative driving and road safety applications by supporting periodic and event-driven information sharing among vehicles and infrastructure. As traffic density increases, the resulting communication load creates channel congestion, reliability, and end-to-end latency. European Telecommunications Standards Institute (ETSI) standardized Decentralized Congestion Control (DCC) [1] to address these issues by using the Channel Busy Ratio (CBR) to dynamically adapt transmission parameters to locally and remotely observed channel conditions, ensuring scalable and stable channel access in dense ITS-G5 scenarios. Although widely studied through simulations and analytical models, experimental evaluations of complete standard-compliant implementations remain limited, especially across different parameter sets standardized by ETSI and profiled by the CAR 2 CAR Communication Consortium (C2C-CC); most existing work relies on simplified assumptions, a single parameter set, or partial implementations.

In this paper, we present an experimental evaluation of the ETSI DCC framework implemented in *Open Stack for car (OScar)* [2], a fully open-source[1] and ETSI-compliant V2X software stack. Our contribution includes the implementation of the complete DCC architecture, encompassing the Access Layer, Network Layer, and Cross-Layer components [1], [3], [4]. The system is evaluated using a controlled laboratory testbed composed of nine customizable On-Board Units (OBUs), emulating up to 24 vehicles. Realistic mobility and perception dynamics are reproduced through replay of traces collected from real-world driving sessions, while configurable background traffic is injected to emulate varying congestion levels. Experimental results are obtained with different parameter sets and provide insights into the effectiveness and limitations of the congestion control mechanisms under realistic traffic loads. This contributes to a deeper understanding of the behavior of DCC in dense vehicular environments.

The main contributions of this paper can be summarized as follows:

- Implementation of a fully ETSI-compliant DCC architecture integrated into the open-source OScar V2X stack;
- Design of a scalable and reproducible experimental testbed, capable of emulating dense vehicular scenarios using real-world mobility and perception traces;
- Comprehensive experimental evaluation of DCC performance under controlled congestion conditions, providing insights into system behavior and practical deployment considerations.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III describes the implemented DCC architecture. Section IV details the experimental setup and methodology. Section V presents the experimental results, and, finally, Section VI concludes the paper.

II. Related Works

DCC has been extensively studied as a fundamental mechanism to ensure scalable and reliable V2X communications in dense traffic scenarios.

[1] https://github.com/DriveX-devs/OScar

Several works overview ETSI DCC and its operation. The survey [5] provides a taxonomy of DCC methods, contrasting standardized reactive mechanisms with adaptive and machine-learning-based approaches for high-density scenarios. Most studies are simulation-based: [6] reviews the main DCC mechanisms and evaluates them in IEEE 802.11p and LTE-V2X, discussing adaptation of transmission power, transmission frequency, and modulation to maintain scalability and reliability. Complementing these, [7] shows experimentally that different parameter sets lead to markedly different convergence and stability for the adaptive mechanism, while Lyamin et al. [8] analyze the reactive state machine and CBR thresholds, highlighting the impact of system-level application parameters. Overall, both adaptive and reactive strategies depend strongly on configuration and context.

Different control strategies have also been explored. In [9], a comparison between LIMERIC (message-rate) and PDR-DCC (data-rate) shows that data-rate control achieves higher packet delivery ratios in high-density scenarios by minimizing channel occupancy time, and suggests a hybrid approach for safety-critical communications. Similarly, [10] reports that PDR-DCC performs better at short ranges, while LIMERIC is more effective as distance increases, highlighting trade-offs among awareness range, packet delivery ratio, and channel occupancy. Sepulcre et al. [11] evaluate packet dropping–based congestion control in ETSI DCC, showing that while dropping reduces channel load and improves radio-level communication, it can degrade application-level performance because dropped packets are not delivered to nearby vehicles. Thandavarayan et al. [12] similarly show that Access Layer DCC alone can increase latency and reduce perception timeliness, whereas combining access and facilities layer control improves cooperative perception. A practical DCC implementation is presented in [13] using four Raspberry Pi devices. Two devices transmit Cooperative Awareness Messages (CAMs) at a fixed 5 ms period, instead of ETSI CAM generation rules, thus emulating about 20 vehicles; no standard stacks such as OScar [2] or Vanetza [14] are used. The other two devices run a patched `iperf` with different Access Categories to generate controlled interference, enabling a low-cost evaluation of DCC but limiting realism and interoperability.

In this work, we extend these studies with a standard-compliant experimental evaluation of ETSI DCC in a controlled laboratory testbed, providing empirical insights beyond simulation-based analyses.

III. Decentralized Congestion Control Implementation

DCC is defined by ETSI to guarantee scalable and reliable communication in vehicular networks by preventing channel congestion through dynamic adaptation of transmission parameters (e.g., maximum transmission power and minimum inter-packet time) based on locally observed channel conditions. DCC is specified as a cross-layer function operating across the Access and Network layers of the ITS-G5 protocol stack [1].[2]

Channel congestion is measured through the CBR, i.e., the fraction of time the channel is busy, due to local transmission/reception or sensed power above a threshold. In our implementation, CBR is derived from channel utilization statistics exposed by the Linux kernel via the *netlink* Application Programming Interface (API). We rely on MAC-layer busy-time measurements reported by the IEEE 802.11p subsystem, using the same mechanism as the `iw` tool to retrieve channel survey information [15].

By periodically querying *netlink*, the stack computes CBR as the ratio between total channel busy time within an observation interval and the interval itself, as foreseen by ETSI [1]. This provides a lightweight, standards-compliant estimate of channel occupancy without hardware-specific extensions or driver modifications.

Three closely integrated components are defined by the ETSI standard: DCC_ACC [1], DCC_NET [3], and DCC_CROSS [4]. While DCC_ACC and DCC_NET implement the main congestion control mechanisms at the access and networking layers, DCC_CROSS primarily acts as a coordination interface for sharing parameters across layers. Therefore, the following sections will focus on DCC_ACC and DCC_NET. Together, these components enable control of channel access behavior and network-layer message handling, while maintaining consistency across layers. This design facilitates extensibility and experimentation within open V2X stacks.

A. DCC_ACC: Access Layer Congestion Control

The DCC at the Access Layer serves as a gatekeeper for the upcoming messages sent by the upper layers. It implements a combination of prioritization, queuing, and flow control mechanisms to regulate channel access and reduce channel load under dense scenarios. Incoming packets are first classified according to four DCC Profiles (DP0–DP3), with DP0 having the highest priority. Each profile maps to a separate queue, which follows a First-In-First-Out (FIFO) policy. Packets exceeding their lifetime or arriving when the queue is full are dropped, preventing excessive congestion. Flow control schedules packet transmissions on the wireless medium by prioritizing higher-priority queues, which may cause starvation of lower-priority traffic. To regulate the overall transmission rate, DCC_ACC applies one of two ETSI-defined approaches, reactive or adaptive:

- The reactive approach uses a state machine where states correspond to CBR ranges. Each state defines transmission parameters, including the inter-packet interval T_{off} and the maximum transmission power. Higher CBR drives the system toward more restrictive states, reducing the transmission rate and dropping excess packets; however, reactive DCC can suffer from stability issues and coarse adaptation under highly dynamic conditions.

[2] In the ETSI documents, the DCC architecture includes also facilities-layer functionalities, which are, however, not specified.

- The adaptive approach applies a linear control process. It periodically updates an internal transmission fraction parameter δ based on the expected transmission time T_{on}, the CBR, and the desired channel occupancy (*CBR Target*); T_{off} is then updated from δ. This provides smoother adaptation, better channel stability, and higher throughput than the reactive approach [16].

Further details on the underlying equations, parameter definitions, and configuration guidelines are provided in [1].

B. DCC_NET: Network Layer Congestion Control

In the latest standardized versions of GeoNetworking [3], the sharing of CBR is mandatory through the GeoNetworking header. Accordingly, ETSI defines the DCC Network component, which provides additional congestion control functionalities for ITS-G5 by leveraging the CBR sharing mechanism, whose use is recommended in [1]. DCC_NET extends local congestion sensing by maintaining and exchanging multiple CBR state variables, including locally measured values as well as CBR information received from one-hop and two-hop neighbors. These CBR values are stored in dedicated tables containing congestion-related information, transmit power, and received Received Signal Strength Indicator (RSSI).

Based on CBR locally measured and CBR values received from neighbors, each station periodically computes a global CBR (i.e., CBR_{G}), combining information from up to two hops to capture congestion beyond the local transmission range. Specifically, for each station, the mean and maximum CBR values from one-hop and two-hop neighbors are calculated. For each hop, if the mean of the received CBRs exceeds the *CBR Target*, the maximum observed CBR is selected as the representative value (e.g., CBR_{L1} for one-hop, CBR_{L2} for two-hop); otherwise, the second-highest values are chosen. The CBR_{G} is then obtained as the maximum among the previously measured local CBR, CBR_{L1}, and CBR_{L2}. By doing so, each station can capture congestion beyond its local transmission range and adapt its transmission behavior to ensure timely and reliable delivery even under high network load. This computation is performed every 100 ms and is intended to mitigate issues such as hidden nodes and asymmetric sensing. When available, the CBR_{G} can be used by the DCC_ACC adaptive mechanism in place of the locally sensed CBR. Although ETSI does not explicitly state whether the reactive approach may use CBR_{G} instead of local measurements, OScar implements this capability through a configurable flag, allowing either local or global CBR to be selected as input to the reactive algorithm logic.

The *CBR Target* used by DCC_NET is aligned with the value used at the access layer to ensure cross-layer consistency. While the global CBR mechanism offers a wider view of congestion, it has not yet been widely adopted, with most current system profiles still depending on local CBR measurements.

IV. Experimental Setup

For the experiments, a controlled laboratory setup was realized, consisting of nine OBUs located in the same environment, as illustrated in Figure 1. This configuration enables repeatable, fully observable, and realistic communication experiments. Each OBU is based on a modular vehicular communication platform originally introduced in [17] and later extended in [18]. While preserving the original system architecture built around the PC Engines APU2 embedded platform, the hardware has been upgraded to improve communication and adopt a more up-to-date Operating System (OS). As a matter of fact, a MikroTik wireless module was integrated to improve the communication range and maximum transmission power with respect to the Unex/Atheros modules used in [17], [18].

The V2X software stack running on each OBU is OScar [2], which we extend in this work. OScar is a fully open-source vehicular connectivity stack compliant with ETSI standards. The main enhancement with respect to [2] is the implementation of the full DCC architecture according to the standards. The adaptive algorithm follows the parameter specification in [1], with *CBR target* set to 0.62 as specified in [3], [4]; in accordance with the standard, its input is CBR_{G} propagated from DCC_NET through DCC_CROSS, rather than the locally sensed CBR. The reactive approach follows [1], where each state is associated with a distinct T_{off} value determined by the T_{on} estimation, and relies on locally sensed CBR. Besides the strictly ETSI-compliant configuration, we also evaluate two CBR variants: the C2C-CC profiling [19] average of the two most recent CBR measurements, and the average of the last two CBR_{G} values, consistent with the adaptive algorithm.

Regarding the queue management policy, the DCC mechanism operates on priority queues. In our implementation, two separate queues are used: one dedicated to CAMs and one to Collective Perception Messages (CPMs), with the CAM queue assigned higher priority. Each queue has a capacity of five packets and enforces a packet lifetime of 600 ms. This design choice reflects the application requirements, as retaining information beyond this interval would result in outdated data that is no longer meaningful for the considered scenario. Due to the close proximity of OBUs in our setup, full mutual radio visibility is ensured, allowing each station to transmit and receive messages to and from all others without obstruction or hidden-node effects.

A. Realistic Traces Reproduction

To reproduce a realistic vehicular environment in the laboratory setup, we use the `TRACEN-X` framework [20] to collect Global Navigation Satellite System (GNSS) and Controller Area Network (CAN) traces from real field tests and reproduce them in the testbed. We collected GNSS navigation data and on-board sensor data via CAN during a driving session on the A4 highway near Turin, Italy, including high-accuracy Real-Time Kinematic (RTK)-GNSS and perception information for detection and tracking of a leading vehicle obtained through sensor fusion of front camera and front radar. Using the lightweight replay mode, traces are reproduced on the OBUs via virtual serial interfaces (GNSS) and virtual CAN bus interfaces (perceived objects), allowing OScar to operate as

Fig. 1: Laboratory setup with nine OBUs. Each device hosts three TRACEN-X instances [20] and three OScar instances [2], enabling the replication of a realistic vehicular environment without requiring the deployment of physical vehicles in field experiments.

if connected to real positioning hardware and an in-vehicle network.

B. Traffic Load Configuration

A set of tests was designed to evaluate DCC under different network loads, parameter sets, and channel utilization conditions. Controlled background traffic was generated on the shared wireless medium using an enhanced `iperf` to progressively increase the offered User Datagram Protocol (UDP) load up to a maximum value. The OBUs were configured with a nominal bitrate of 3 Mbit/s; the `iperf` load was increased in 2 kbit/s steps until saturation, reaching about 2.8 Mbit/s, with measured CBR around 0.95/0.96. By dynamically changing the injected load, we assess the impact of channel saturation on DCC and on CAMs/CPMs transmission and reception, and the resulting key performance metrics (Section V).

To emulate a dense and realistic scenario, each OBU (except the one running `iperf`) executes three independent instances of the OScar stack, emulating multiple nearby vehicles without requiring additional hardware. These virtual vehicles transmit CAMs and CPMs according to ETSI generation rules, reproducing typical V2X traffic patterns. The resulting setup provides a scalable and controllable framework for evaluating V2X performance under dense conditions and different DCC strategies.

V. Experimental Results

This section presents the results obtained from the experiments conducted in the laboratory setup described above. The analysis focuses on the behavior of the DCC mechanism under increasing channel load conditions, evaluating its impact on the transmission of CAMs and CPMs through a set of selected key performance metrics.

Figure 2 shows the channel load over time in terms of CBR, distinguishing three groups corresponding to the different DCC strategies. "No DCC" refers to a scenario in which the OBUs operate without any DCC mechanism; a baseline congestion level is introduced and maintained by `iperf` throughout the experiment. Up to about 600 s, the sensed CBR is nearly unchanged across strategies, with the reactive approach showing a slight decrease, likely due to rough state transitions and sudden variations in transmission rate. In the second half, all DCC strategies report lower CBR than "No DCC", despite the continuous `iperf` traffic, indicating that DCC becomes noticeable once channel load reaches roughly 50%. In our setup, this suggests activating DCC mainly beyond this load level.

Figure 3 presents the evolution of T_{off} throughout the experiment, reporting both its mean value (shown as the trend line) and its standard deviation (represented by the shaded area). T_{off} represents the duration for which the DCC gate remains closed between two consecutive transmitted packets, effectively imposing a minimum inter-transmission gap enforced by the congestion control mechanism.

The DCC mechanisms exhibit distinct behaviors as channel load increases. The reactive approaches show an unstable trend in regions where T_{off} transitions occur (between 300 and 600 s, and before 900 s), with large standard deviations around the mean, whereas the adaptive algorithm remains stable with negligible standard deviation. This indicates that the more mathematical-driven method used by the adaptive algorithm to compute T_{off} is more effective in ensuring clear and stable inter-transmission intervals under high channel load; across the reactive curves, using CBR_{G} appears ineffective to mitigate this effect. Moreover, the reactive approach is more conservative than the adaptive one at lower channel loads (roughly within the first 750 s) and more aggressive at higher loads.

To further analyze selection policies, we introduce the packet drop probability P_{drop}, i.e., the likelihood that a scheduled packet is prevented from accessing the wireless medium by the DCC gating mechanism. Drops occur due to queue saturation (discarded upon arrival when the queue is full) or queue lifetime expiration (removed after exceeding the maximum waiting time). Formally, this metric can be expressed as:

$$P_{\text{drop}} = \frac{N_{\text{drop}}^{\text{sat}} + N_{\text{drop}}^{\text{timeout}}}{N_{\text{intended}}} , \tag{1}$$

where $N_{\text{drop}}^{\text{sat}}$, $N_{\text{drop}}^{\text{timeout}}$, and N_{intended} are the number of packets

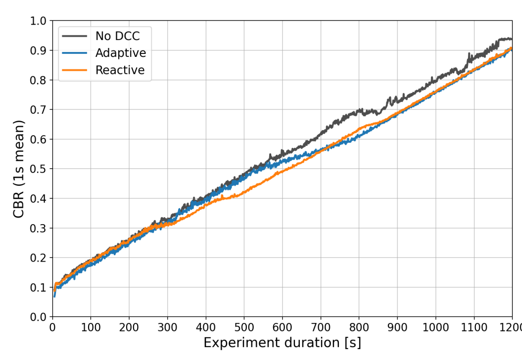

Fig. 2: Sensed CBR over time for three different DCC strategies: No DCC, Adaptive, Reactive.

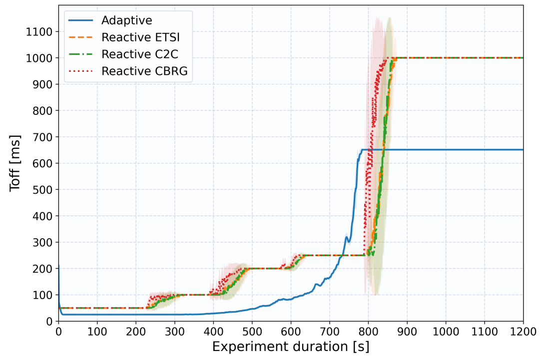

Fig. 3: T_{off} mean and standard deviation evolution, over the test for the different DCC strategies.

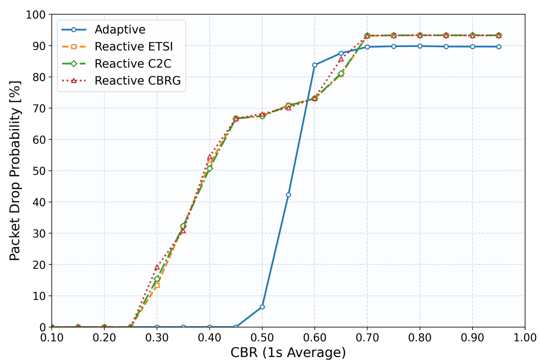

Fig. 4: Packet Drop probability as a function of the CBR evolution.

dropped due to queue saturation, dropped due to timeout, and intended for transmission, respectively.

Figure 4 illustrates the packet drop probability as a function of the CBR. Due to its state machine-based design, the reactive algorithm exhibits a significantly higher drop rate than the adaptive approach, particularly in the channel load range between 30% and 60%. Between 60% and 70% channel load, the reactive strategy shows a drop probability that is lower than the adaptive approach. Beyond 70%, however, its drop rate rises again. At the highest congestion levels, the difference in packet drop probability for the two strategies is limited, with the adaptive approach remaining slightly lower. Furthermore, no substantial differences are observed among the reactive variants.

Packet collisions in the shared wireless medium lead to packet losses and reduced reliability. Figure 5 reports the cumulative number of collisions over time. A packet is classified as collided if it passes the DCC gate at the transmitter but is not successfully received by any intended receiver. The adaptive approach shows a pronounced initial rise in collisions, even exceeding the "No DCC" case.

After 700 seconds, CBR exhibits a corresponding increase, reaching 0.55, as shown in Figure 2. However, beyond this point, both the reactive and adaptive approaches reach a steady state, showing no further rise in the number of collided

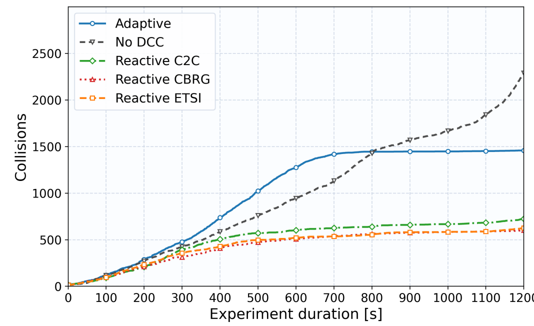

Fig. 5: Cumulative number of collisions over time.

packets. On the other hand, the "No DCC" approach shows a constant growth in packet collisions. The lower number of collided packets observed with the reactive approach is likely due to its more aggressive transmission rate control, which enforces higher T_{off} values, resulting in fewer transmitted packets and, consequently, fewer collision events.

Related to the packet collisions, another metric we consider is the collision probability P_{collided}, which assesses the likelihood that a packet that passed the gate restriction imposed by the DCC (i.e., N_{sent}) will not be received by the receivers. Considering N_{received} as the number of packets successfully received by at least one instance, it can be expressed as:

$$P_{\text{collided}} = 1 - \frac{N_{\text{received}}}{N_{\text{sent}}} \, . \tag{2}$$

Figure 6 illustrates the evolution of the collision probability under varying channel load. For clarity, only one reactive profile is shown, as all three exhibited the same behavior. Overall, collision probability remains below 2% for all approaches, except under very high channel load. The adaptive approach generally outperforms the reactive one, with a slightly higher probability only in the 0.3–0.5 CBR range (still below 2%), consistent with the increase in collided packets in Figure 5. As expected, this can be attributed to the more aggressive transmission rate control of the reactive scheme, which reduces the number of collided packets.

Inter-arrival time, measured at the receiver side, captures the elapsed time between two consecutive receptions from the

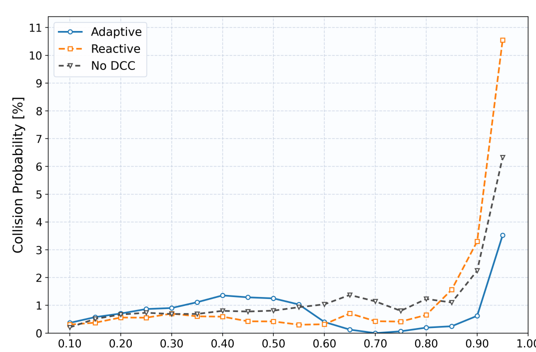

Fig. 6: Collision probability over CBR.

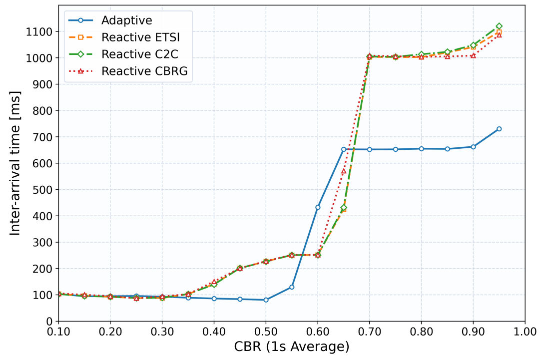

Fig. 7: Inter arrival-time over CBR.

same OScar instance. It indicates how frequently a station updates information about its surroundings and is therefore a key application-level metric. While T_{off} captures the delay introduced at the transmitter by DCC, inter-arrival time reflects its effect at the receiver, making the two metrics complementary for assessing the end-to-end impact of congestion control.

Figure 7 underlines again the differences between the adaptive and reactive approaches. The first demonstrates better performance over almost all the CBR values (except for the situation in which the CBR is around 0.6) compared to the second, demonstrating that the adaptive algorithm can guarantee more timely and reliable delivery of information under high channel congestion, as reflected in lower, more stable inter-arrival times.

VI. Conclusions

As traffic density increases, V2X load exacerbates channel congestion, reduces reliability, and increases end-to-end latency. DCC, based on CBR, mitigates these effects by adapting transmission parameters to local and remote channel conditions; however, most studies remain simulation-based, and real-world testing is needed to fully characterize its behavior. This paper experimentally evaluates ETSI DCC as implemented in the OScar stack. Using nine OBUs, each running three OScar instances, we emulate up to 24 virtual vehicles transmitting CAMs and CPMs and analyze different DCC profiles in terms of CBR, T_{off}, packet drops, and collision probability. Future work will consider different `iperf` step sizes, more challenging deployments (e.g., hidden nodes), and parameter optimization to improve stability, responsiveness, and efficiency.

Acknowledgments

This publication is part of the PNRR–NGEU project, funded by the Italian Ministry of University and Research (MUR) under Decree DM 117/2023.

References

[1] ETSI, "Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," European Telecommunications Standards Institute (ETSI), Tech. Rep. TS 102 687, 2018.

[2] M. Rapelli, F. Raviglione, and C. Casetti, "OScar: An ETSI -Compliant C- ITS Stack for Field- Testing with Embedded Hardware Devices," in *2024 22nd Mediterranean Communication and Computer Networking Conference (MedComNet)*, 2024, pp. 1–4.

[3] ETSI, "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," European Telecommunications Standards Institute (ETSI), Tech. Rep. TS 102 636-4-2, 2022.

[4] ETSI, "Intelligent Transport Systems (ITS); Cross Layer DCC Management Entity for operation in the ITS G5A and ITS G5B medium," European Telecommunications Standards Institute (ETSI), Tech. Rep. TS 103 175, 2015.

[5] A. Balador, E. Cinque, M. Pratesi, F. Valentini, C. Bai, A. A. Gómez, and M. Mohammadi, "Survey on decentralized congestion control methods for vehicular communication," *Vehicular Communications*, vol. 33, p. 100394, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209621000632

[6] A. Bazzi, "Congestion control mechanisms in IEEE 802.11 p and sidelink C-V2X," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 1125–1130.

[7] O. Amador, I. Soto, M. Calderón, and M. Urueña, "Experimental Evaluation of the ETSI DCC Adaptive Approach and Related Algorithms," *IEEE Access*, vol. 8, pp. 49 798–49 811, 2020.

[8] N. Lyamin, A. Vinel, D. Smely, and B. Bellalta, "ETSI DCC: Decentralized Congestion Control in C-ITS," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 112–118, 2018.

[9] C. B. Math, H. Li, S. M. H. de Groot, and I. G. M. M. Niemegeers, "V2X application-reliability analysis of data-rate and message-rate congestion control algorithms," *IEEE Communications Letters*, vol. 21, no. 6, pp. 1285–1288, 2017.

[10] Y. Wei, C. B. Math, H. Li, and S. Heemstra de Groot, "SAE-DCC Evaluation and Comparison with Message Rate and Data Rate Based Congestion Control Algorithms of V2X Communication," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–7.

[11] M. Sepulcre, J. Mira, G. Thandavarayan, and J. Gozalvez, "Is Packet Dropping a Suitable Congestion Control Mechanism for Vehicular Networks?" in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[12] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Cooperative Perception for Connected and Automated Vehicles: Evaluation and Impact of Congestion Control," *IEEE Access*, vol. 8, pp. 197 665–197 683, 2020.

[13] A. Solida, C. A. Grazia, M. Klapez, and M. Casoni, "Optimizing Decentralized Congestion Control: Performance Evaluation on IEEE 802.11p Networks," in *2026 IEEE 23rd Consumer Communications & Networking Conference (CCNC)*, 2026, pp. 1–6.

[14] R. Riebl, "Vanetza: An open-source implementation of the ETSI C-ITS protocol stack," https://github.com/riebl/vanetza, 2025.

[15] cozybit Inc., "iw," https://github.com/cozybit/iw, 2012.

[16] ETSI, "Intelligent Transport Systems (ITS); Facilities Layer; Radio Resource Management Study; Release 2," ETSI, Tech. Rep. TR 104 073 V2.1.1, 2025.

[17] F. Raviglione, M. Malinverno, and C. Casetti, "Characterization and Performance Evaluation of IEEE 802.11p NICs," in *Proceedings of the 1st ACM MobiHoc Workshop on Technologies, MOdels, and Protocols for Cooperative Connected Cars*, ser. TOP-Cars '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 13–18. [Online]. Available: https://doi.org/10.1145/3331054.3331548

[18] F. Raviglione, C. Casetti, and F. Restuccia, "Edge-V : Vehicular Edge Intelligence Through Multi-Band Unlicensed Spectrum Access," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2025.

[19] CAR 2 CAR Communication Consortium, "Vehicle C-ITS Station Profile (Basic System Profile, Release 1.6.9)," CAR 2 CAR Communication Consortium, Tech. Rep. C2CCC_RS_2037_Profile_R169, 2025, public technical report, Version 1.6.9. [Online]. Available: https://www.car-2-car.org/fileadmin/documents/Basic_System_Profile/Release_1.6.9/C2CCC_RS_2037_Profile_R169.pdf

[20] D. Gasco, C. M. Risma Carletti, F. Raviglione, M. Rapelli, and C. Casetti, "TRACEN-X: Telemetry Replay and Analysis of CAN Bus and External Navigation Data," in *2025 IEEE 102nd Vehicular Technology Conference (VTC2025-Fall)*, 2025, pp. 1–5.