Paper number EU-TP0529

Decentralized Congestion Control for Multi-hop Vehicular Communication

Andreas Festag[1]*, Sebastian Kühlmorgen[2], and Nitin Maslekar[3]

[1]Fraunhofer Institute for Transportation and Infrastructure Systems (IVI), Germany
andreas.festag@ivi.fraunhofer.de

[2]Vodafone Chair Mobile Communication Systems, Technical University Dresden, Germany,
sebastian.kuehlmorgen@tu-dresden

[3]NEC Laboratories Europe, Germany
nitin.maslekar@neclab.eu

Abstract

WiFi-based vehicular communication systems, such as DSRC in the US and C-ITS in Europe, suffer from large delays and high packet losses in scenarios with a dense distribution of nodes and high data load, which can result in system instability and degraded safety application performance. To ensure a stable system and fair share of resources among vehicles, a decentralized congestion control (DCC) function limits the data that is generated by a network node depending on the measured channel load. In a vehicular ad hoc network with multi-hop communication, DCC need to control the data packets that are locally generated by the node as well as forwarded data packets. This paper gives an overview of the cross-layer DCC framework for the European C-ITS, presents results of a simulation-based performance evaluation of DCC and analyzes the impact of DCC on forwarding algorithms of the ad hoc networking protocol.

Keywords:

Vehicle-to-X (V2X) communication, ETSI ITS-G5, Decentralized Congestion Control (DCC), GeoNetworking

I. Introduction

Wireless communication enables vehicles to share information in order to improve safety, traffic efficiency and driving comfort. For the today's generation of vehicles, communication can greatly support drivers by providing information and warnings beyond the driver's vision. For future vehicles, it has also the potential to support their automation by sharing on-board sensor data and coordinate the maneuvering of automated vehicles. Research, standardization and development efforts in the last years have led to a vehicular communication system that is based on a variant of the wide-spread WiFi. Such a system – known as DSRC in the US [1] and C-ITS in Europe [2] – will likely be deployed in the next few years.

The WiFi-based vehicular communication system is composed of several components, including radio (PHY and MAC), networking, facilities and applications, security and management. PHY and MAC are based on a specific mode of the IEEE 802.11-2012 standard [3], called Outside the Context of a BSS (OCB), and does not make use of specific control procedures that are common to other WiFi modes, such as channel scanning, association, etc. This mode has also been adopted for the European C-ITS variant of IEEE 802.11 OCB, called ITS-G5, covering PHY and MAC. The PHY operates in the 5.9 GHz frequency band allocated for safety and traffic efficiency, uses Orthogonal Frequency Division Multiplexing (OFDM) as transmission scheme and allows for a communication range of several 100 meters over 10 MHz channels. The system has inherited the MAC scheme defined in the IEEE 802.11 standard, i.e., Enhanced Distributed Channel Access (EDCA) with Carrier Sense Multiple Access and Collision Avoidance (CSMA/CA) and Quality-of-Service (QoS) support. CSMA/CA allows for immediate and direct communication among vehicles with prior exchange of control messages, such as for registration or managing transmission resources. This approach works well for low and medium data load on the wireless channel. With growing load in the network, the wireless channel becomes saturated, which results in an unstable system with low overall data throughput and very high packet loss.

The strategy to ensure a stable system is to keep the load in the network below a pre-defined threshold. Contrary, a network node, i.e. a vehicle or roadside unit, should send data as often as possible in order to maximize the benefit of the safety and traffic efficiency application, sometimes referred to as "awareness". In order to restrict the overall load generated by a node, a function called "decentralized congestion control" (DCC) has been introduced into the protocol stack. The design of DCC is challenging because of the distributed nature of the system without the possibility of a centralized management of transmission resources. Also, the specific requirements of safety and traffic efficiency applications for fair access to the channel for every vehicle as well as the cross-layer interworking of DCC with other components of the protocol stack pose unique challenges.

Despite the fact that the vehicular communication system is close to commercial deployment, DCC is still an active research topic. In a deployed system, DCC functions are expected to become relevant when many vehicles are equipped and generate data in dense road traffic situations. Nevertheless, DCC is required to be part of the first generation of equipped vehicles due to the potential long lifetime of vehicles.

The contributions of the present paper can be summarized as follows: First, the paper gives an overview of the current status of cross-layer DCC framework for the European C-ITS taking into account recent developments in standardization. Second, it presents results of a simulation-based performance evaluation of DCC. Third, we study the impact of DCC on multi-hop communication considering the network-layer forwarding algorithms of the C-ITS. In particular, we highlight the investigation of DCC with multi-hop communication since previous DCC studies considered single-hop communication only.

The remainder of this paper is structured as follows: Section II presents a framework for DCC for the European C-ITS, followed by brief overview of DCC-related protocols and algorithms at access, networking and facilities layer, i.e. EDCA, CBF and facilities layer entities in Sec. III. Section IV explains the operation of the gatekeeper, which represents the DCC core component in the C-ITS protocol stack, and its interaction with other protocols. Section V describes the scenario and metrics of the performance evaluation and Sec. VI presents the performance results. Section VII concludes the paper.

II. Framework for Decentralized Congestion Control

In order to control the data load in the ad hoc vehicular network, several mechanisms can be applied: Transmit power control (TPC) adapts the effective communication range and thereby the network topology. Other mechanisms control the data rate or the sensitivity threshold. Transmit rate control (TRC) impacts the rates at which vehicles generate periodic messages. The combination of these mechanisms create a multi-variate system with a very high degree of freedom to control the system. It is common to all mechanisms that vehicles measure the load in the wireless channel, i.e. the channel busy ratio, CBR, whereas additional information about the network, such as the number of neighbor nodes or the CBR seen by neighbor nodes can be exploited.

DCC has been studied in various research publications. Most of them have shown the effectiveness of individual mechanisms. Although studies about combined mechanisms exist, such as combined transmit rate and power control, the simultaneous application of multiple mechanisms is considered to be complex and can result in an oscillation of the network load [4, 5]. Therefore, current specifications of DSRC and C-ITS foresee TRC as the main mechanism for DCC. The DCC algorithms presented in existing publications, e.g. [6, 7], can be categorized into two main approaches. The *reactive* approach maps the measured CBR to a predefined message rate. The *adaptive* approach compares the measured CBR to a target value and adapts its message rate such that it converges to the target. Another categorization refers to the algorithm to adapt the message rate: The *state-based* approach defines discrete states with respect to the data load, e.g. relaxed, active and restrictive, with different values of the upper limit for transmit parameters, here specifically the message rate. The *linear control* approach adapts the message rate such that the channel load converges to the predefined target rate. A popular algorithm is LIMERIC (see [8] and related publications), which represents an algorithm for an adaptive linear rate control approach.

The European standardization for C-ITS has developed several specifications related to DCC [9, 10, 11]. ETSI TS 102 687 [9] defines a DCC architecture and an overall framework for the various DCC mechanisms. ETSI TS 103 175 [10] specifies a cross-layer entity with management functions (see ETSI TR101 612 [12] for more details of algorithms and their performance evaluation). A revised specification of ETSI TS 102 687 (draft version 1.2.1) restricts the set of mechanisms to TRC only, with optional use of TPC to alleviate the impact on road tolling

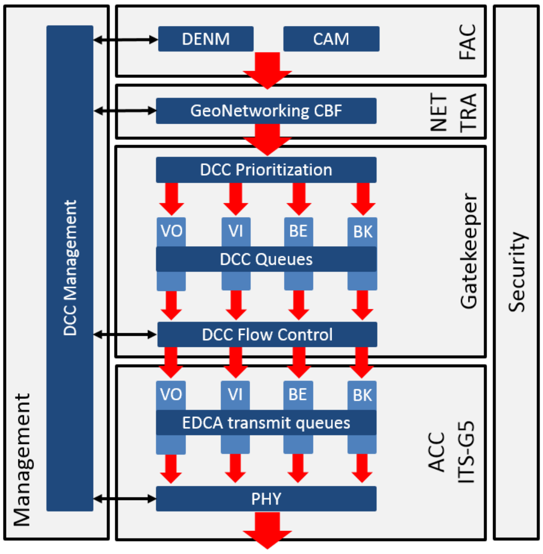

Figure 1 – Protocol stack for ITS-G5 based communication with the DCC gatekeeper between access layer (ACC) and networking & transport layer (NET &TRA), as well as DCC management and facilities (FAC) layer components

systems operating in adjacent frequency band at 5.8 GHz. ETSI EN 302 571 [11] – a harmonized standard that specifies requirements for compliance with the European Radio Equipment Directive (RED) – defines corresponding parameters and metrics for the DCC architecture and functions. For the DCC design, it is important to consider that the European C-ITS system defines several message types, including the Cooperative Awareness Message (CAM), the Decentralized Environmental Notification Message (DENM), as well as infrastructure-related messages for static and dynamic intersection data (MAP and Signal Phase and Timing, SPAT, messages, respectively). Furthermore, in C-ITS packets can be forwarded and hence contribute to the data load. In order to enforce an aggregated rate for all message types as well as for forwarded data packets, the C-ITS protocol stack foresees a gatekeeper component in the protocol stack. The gatekeeper realizes a set of priority queues on sender side, allows differentiating packets according to their traffic class and executes traffic shaping following the DCC requirements and status. It resides on top of the EDCA queues and beneath the networking & transport layer. Overall, TRC and gatekeeper can be regarded as the core concepts and components for DCC in the C-ITS protocol stack.

While the main objective of TRC and gatekeeper is to stabilize the system, the components cause a larger data age at the receiver: First, the gatekeeper adds a delay in the message processing on the sender side caused by the queuing delay in the gatekeeper queues. Secondly, when TRC reduces the rate at which messages, i.e. CAMs, are generated, the larger inter-message time translates increases the delay at which data are updated at the receiver. The resulting data age depends on the parametrization of the gatekeeper algorithm (including the capacities of the queues and queues' service discipline, i.e. the queues' on/off-times) and on the parametrization of the TRC algorithm, which is different for reactive and proactive as well as state-based and linear control approaches. The latency performance of the gatekeeper can be enhanced by supporting traffic bursts, i.e. short duration of time during which the average rate can be exceeded for short time periods while keeping the average rate over longer periods.

The DCC gatekeeper not only restricts the message rate, but also impacts the forwarding of packets at the networking layer. The base forwarding algorithm in the C-ITS protocol stack, contention-based forwarding (CBF), relies on packet buffering, whereas the duration of time until the packet is forwarded depends on its distance from the sender. Then, the node with a largest spatial progress wins the competition among the potential forwarders. A long link layer delay, as potentially introduced by a gatekeeper component, can refrain potential forwarders to overhear transmissions on time. As a result, the CBF scheme turns into a simple, low performance flooding scheme.

Table I – EDCA parameters for ITS-G5[1]. AC_VO has the highest priority.

AC	CW_{min}	CW_{max}	AIFS (Slots | Time)	Messages
AC_VO	3	7	2 | 58 μs	High-priority DENM
AC_VI	7	15	3 | 71 μs	DENM
AC_BE	15	1023	6 | 110 μs	CAM
AC_BK	15	1023	9 | 149 μs	Others

III. Overview of DCC-related Protocols and Algorithms

This section introduces the relevant components for medium access and explains the multi-hop packet forwarding algorithm CBF.

A. IEEE 802.11 DCF and EDCA

The Distributed Coordination Function (DCF) is the basic medium access scheme of IEEE 802.11 [3]. It relies on carrier sense multiple access with collision avoidance (CSMA/CA) [13] and priority handling based on inter-frame spaces (IFS). CSMA/CA handles a single packet at a time, i.e. uses a single queue, in which all packets are queued and a FIFO (First-In-First-Out) queuing strategy. If the packet queue is full, newly arriving packets are discarded.

Enhanced Distributed Channel Access (EDCA) extends DCF and introduces access categories (AC) to differentiate the channel access to different priorities. The differentiation is realized by several independent MAC entities per node with different configurations for inter-frame spaces, referred to as arbitrary IFS, and contention windows (CW) (see Tab. I) [14]. The ITS-G5-based protocol stack utilizes the EDCA queues to prioritize the medium access of messages, i.e. handles DENMs with a higher priority than other messages (Tab. I).

B. GeoNetworking and Contention-based Forwarding (CBF)

Contention-based Forwarding (CBF) is a forwarding algorithm for multi-hop distribution of packets in ad hoc networks [15]. With CBF, a node always broadcasts a data packet to all neighbors and these neighbors compete to become the forwarder of the packet. When the optimal forwarder is found, the others refrain from forwarding. Opposed to conventional forwarding algorithms, where the sender of a packet determines the forwarder, CBF is a receiver-based scheme. In a previous study we found that CBF, standardized as forwarding algorithm in of the GeoNetworking protocol ETSI EN 302 636-4-1 V1.2.1, performs as the best algorithm with respect to reliability, robustness and latency compared to the other GeoNetworking forwarding algorithms [15].

The sole requirement of CBF is the fact that the nodes need to know their ego-geographical position and the position of the sender; the first can be acquired by GNSS, the latter is carried in the packet header of single-hop packets and saved in the location table of the ego-nodes. Upon reception of a packet, a node buffers it and starts a timer. The timer duration depends on the own geographic position and the sender position, the node with the largest distance to the sender has the shortest timer. The minimum and maximum timer length is 1 ms and 100 ms, respectively. When the timer expires the packet is taken from the buffer, passed to the access layer, and re-broadcasted when the channel access is granted. This transmission is overheard by the neighbor nodes. If they receive the packet a second time, they stop their timers and remove it from their buffer. In order to recognize if a duplicate of a packet arrives, nodes maintain a packet duplication list that logs the identifier (ID) of the received packet (combination of source node identifier and sequence number). A flow diagram of CBF is provided in Fig. 2 (left part).

The CBF algorithm works as long as the duration of time between passing the packet to the access layer and reception of the packet at other nodes (access layer delay) is smaller than the CBF timer duration. When the access layer delay gets longer, typically because of the unrestricted medium access delay of CSMA/CA, the queuing time of a packet in the access layer queue can exceed the maximum CBF timeout. In this case, overhearing fails [15] and the efficient forwarding turns into an inefficient flooding of the network (sometimes also referred to as broadcast storm).

[1]ETSI EN 302 663 V1.2.0, ETSI TS 102 687 v1.1.2 (Draft)

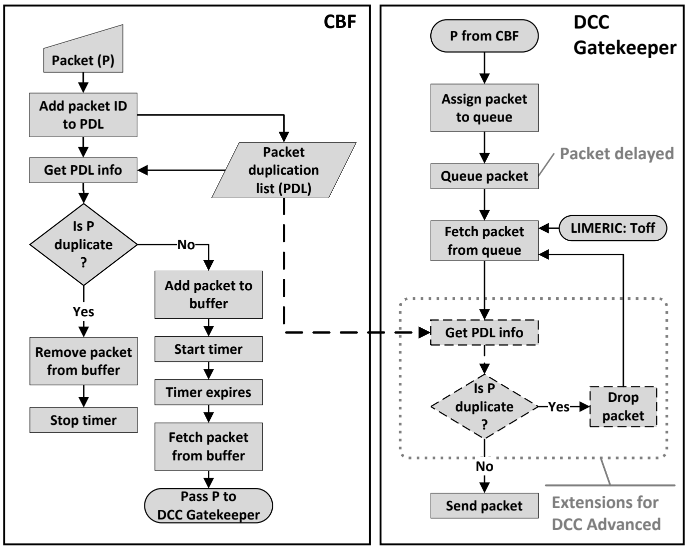

Figure 2 – CBF and DCC gatekeeper flow diagram. The diagram shows the flow of a packet through the networking & transport layer and gatekeeper extension.

IV. DCC Operation

Core of the DCC is the gatekeeper, where a rate control algorithm is used to determine the gatekeeper's packet rate. Next we explain the gatekeeper functionality and the rate control algorithm.

A. DCC Gatekeeper

DCC aims at keeping the channel load below a predefined threshold in order to prioritize the data traffic and to ensure that high-priority safety messages are transmitted. For this reason, the gatekeeper is introduced, located below the networking & transport layer and above the EDCA-based access layer (see Fig. 1) [12]. The gatekeeper prioritizes the messages according to their traffic class and controls the transmission interval. It consists of three main functions, i.e. prioritization, queues and flow control.

DCC prioritization differentiates the packets coming from the upper layer according to their traffic class and assigns them to the corresponding queue, e.g. CAMs, regular DENMs and high-priority DENMs are buffered in the AC_BE, AC_VI and AC_VO queues, respectively. The traffic classes are mapped to the EDCA access categories and the respective queues with four priorities.

DCC queues are on top of the EDCA queues with the same priorities. The DCC flow control is in charge of the de-queuing process, whereas the queues depend on each other, which means that the queue with the highest priority is dequeued first.

DCC flow control is based on a closed-loop scheme where the node observes the network load and adapts the output variable to control the transmission rate accordingly. To achieve this, the DCC flow control function inserts a delay T_{off}, during which the gatekeeper is closed and subsequent packets are buffered in the DCC queues. When T_{off} expires the next packet is taken from the DCC queues and passed to the EDCA queues, whereby we assume a one-to-one mapping between DCC and EDCA queue access classes. The intention of T_{off} is to keep the channel load below a predefined threshold. With growing channel load T_{off} is increased and fewer packets are transmitted, which leads to a lower channel load. The dependency and prioritization among DCC queues can lead to the effect that packets 'starve' in lower priority queues when the channel is saturated and only higher prioritized packets are sent.

In order to cope with the malfunctioning of CBF caused by gatekeeper-induced high link-layer delay, we enable the DCC flow control to access the packet duplication list of CBF (see the dashed line in Fig. 2) [16]. When the DCC flow control component de-queues a packet, it acquires the information about the existence of a duplicate packet from the packet duplication list. If the packet is indeed a duplicate, the DCC flow control denies further processing

Table II – Relevant ETSI Standards

Standard	Version	Short Title
ETSI EN 302 663	V1.2.0	ITS-G5
ETSI EN 302 636-4-1	V1.2.0	GeoNetworking media-independent
ETSI TS 102 636-4-2	V1.1.1	GeoNetworking extensions for ITS-G5
ETSI TS 103 175	V1.1.1	Cross Layer DCC Management
ETSI TS 102 687	V1.1.2	DCC Access layer
ETSI EN 302 637-2	V1.3.2	Cooperative Awareness Message (CAM)
ETSI EN 302 637-3	V1.2.1	Decentralized Environmental Notification Message (DENM)

of the packet and thereby avoids erroneously forwarding (see the dashed box in Fig. 2). In the remainder of this paper, we refer to the extension of DCC for CBF as ”DCC Advanced” opposed to ”DCC Plain”.

B. Rate Control Algorithm

For the computation of the gatekeeper's T_{off} we choose the rate control algorithm LIMERIC [17] and modify the algorithm to use the channel busy ratio CBR, measured at the physical layer as adjustment value. The algorithm determines the node i's next packet rate $r_i(t_{k+1})$, which is the inverse of T_{off} (see Eq. (1), (2)). The algorithm works recursively; it calculates the tracking error between the actual CBR and the CBR threshold CBR_{th} and takes this deviation into account for the computation of the next message rate:

$$r_i(t_{k+1}) = (1-\alpha)r_i(t_k) + \frac{\beta}{T_{\text{on}}}\left(CBR_{\text{th}} - CBR_i(t_k)\right), \tag{1}$$

whereas $r_i(t_k)$ is node i's current packet rate and $CBR_i(t_k)$ indicates ith current channel busy ratio, α and β are constants that control the exponential forgetting and scaling, respectively. T_{on} is the transmission time, i.e. the time it takes to transmit the packet over the air. The rate converges to the value where CBR meets CBR_{th}.

$T_{\text{off},i}$ is the delay until DCC flow control of node i fetches one packet from the DCC queue:

$$T_{\text{off},i} = \frac{1}{r_i(t_{k+1})}. \tag{2}$$

The parameter values for the simulations are listed in Tab. III. T_{off} is restricted to values between 10 and 1,000 ms, which avoids that the interval decreases to 0 with lightly loaded channels and grows to (theoretically) infinity in saturated channels. For details on LIMERIC, e.g. convergence behavior and stability criterion, we refer to [17].

V. Evaluation Scenario and Metrics

For the performance evaluation, we consider a typical freeway scenario with three lanes per direction and a speed limit of 120 km/h, i.e. the maximum permitted velocity that vehicles are allowed to travel. The interesting geographical area is 1,500×100 m behind the transmitting vehicle and the DENMs, which are received outside this area, are discarded. The vehicles are uniformly distributed and the density of nodes varies between 5 to $60\,nodes/km/lane$, covering a range from free flow to traffic jam. Reasoning for selection of the vehicle densities can be found in [15]. To avoid edge effects at the end of a freeway, the simulated freeway forms a large circle with a diameter of 6 km. We further assume that all vehicles are equipped with the communication technology.

The simulator implements ITS-G5 PHY and data link layer, GeoNetworking with CBF, DCC with the gatekeeper, and the facilities layer protocols CAM and DENM, all following the respective ETSI standards (Tab. II).

The simulation is executed in steps: in the first 5 s of the simulation only CAMs are generated. After five seconds, 30 % of nodes start transmitting (regular) DENMs with an interval of 1 s for 90 s. Then, 2 % of the vehicles send 10 high-priority DENMs with an interval of 1 s, starting linearly distributed between 15 and 75 s. The overall simulation runs for 100 s. The nodes stop generating new DENMs 5 s before the simulation ends, which allows in-transit DENMs to reach their destination. Overall, the simulation has therefore transient phases at the beginning and the end of a simulation run. The simulation is executed for the three protocol options, i.e. EDCA, DCC Plain and DCC Advanced. CBR_{th} is set to 0.65 as defined in [12]. All queues in the simulated setup have a maximum

Table III – Simulation Parameters

Vehicle density	5 – 60 veh/km/lane (18.3 km, 3+3 lanes)
PHY and data link protocols	ITS-G5 (IEEE 802.11 OCB)
Data rate	6 $Mbit/s$
Tx Power & antenna gain	20 dBm + 9.1 dB
Channel bandwidth	10 MHz at 5.9 GHz
Path loss model	ThreeLogDistance: channel exponent 1.9 – 3.8
Fading model	Nakagami - shape factor 1
Energy detection threshold	-95 dBm
Carrier sense threshold	-99 dBm
Max - queue size | - queuing time	400 packets | 10 s
Simulation duration	100 s
CAM – interval | size | # of nodes	0.5 s | 512 bytes | 100 %
DENM – interval | size | # of nodes	1 s | 512 bytes | 30 %
High-priority DENM – interval | size | # of nodes	1 s | 512 bytes | 2 %
Geographical target area	1,500×100 m behind the vehicle
LIMERIC: α | β | T_{on}	0.01 | 0.001 | 872 μs
CBR threshold – CBR_{th}	0.65
$T_{\text{off,min}}$ | $T_{\text{off,max}}$	10 ms | 1,000 ms

queue size of 400 packets, except the BE-queue that is used for CAMs. The queues have a maximum queuing time of 10 s each, which corresponds to the default setting in ns-3. We use a simplified DENM protocol, where DENM options such as update, negate and cancel are not taken into account.

The simulation are performed with the network simulator ns-3, version 3.23 (http://www.nsnam.org). We have extended the basic simulator by modules for GeoNetworking, CAM, DENM and gatekeeper. We note that the recent update of ETSI TS 102 687 v1.1.2 allows a node to bypass DCC queues and flow control for a single high-priority packet in every scheduling cycle to cope with emergency messages, here high-priority DENMs. Our gatekeeper implementation considers this features, however, we do not further detail the implications in the present paper. The channel is modeled with a log-distance propagation model with Nakagami fading [18]. Nakagami fading is a stochastic channel model; this model matches empirical data and it is commonly used for simulations of vehicle-to-vehicle communication. With the parameters setting in Tab. III the communication range ranges from 200 to 600 m, which corresponds to the distances where 95 % and 10 % of the packets are successfully received. The mobility of the vehicles is controlled by traces that were generated by the road traffic simulator SUMO (Simulation of Urban MObility) [19]. The simulations are performed with up to 6,500 nodes where, because of the very high number of nodes in the simulation setting, the change of the seed did not alter the results. Therefore, only a single seed was taken. A summary of all simulation parameters and corresponding values is given in Tab. III.

We define the following evaluation metrics to compare and evaluate the algorithms in terms of reliability, latency and channel load.

Inverse Node Coverage Ratio (Inverse NCR): Many safety and traffic efficiency applications, such as traffic jam ahead warning or emergency electronic brake light, require the dissemination of information to vehicles in a relevance area. This relevance area is defined by an application and mapped to a geographical area; a geometric shape with defined dimensions and orientation in which the communication system distributes the packet by multi-hop communication. The inverse NCR is a measure for the communication reliability and indicates whether a vehicle misses a packet. For example, if 10 vehicles are located inside the geographical area and two nodes miss the message, the inverse NCR would be 20 %.

Average End-to-End Delay (E2ED): This metric is defined as the duration of time between transmitting DENMs at the source until they arrive at the destination, averaged over all messages. It is worth noting that E2ED does not consider single-hop CAMs, but multi-hop DENMs only. It can be expressed as

$$E2ED = \frac{1}{N} \sum_{i=1}^{N} \left(t_{\text{Rx},i} - t_{\text{Tx},i} \right) , \tag{3}$$

where N is the number of received messages in the geographical area, $t_{\text{Tx},i}$ and $t_{\text{Rx},i}$ are the timestamps of the generated and received DENMs, respectively. The E2ED is measured from the source to the destination including all forwarders.

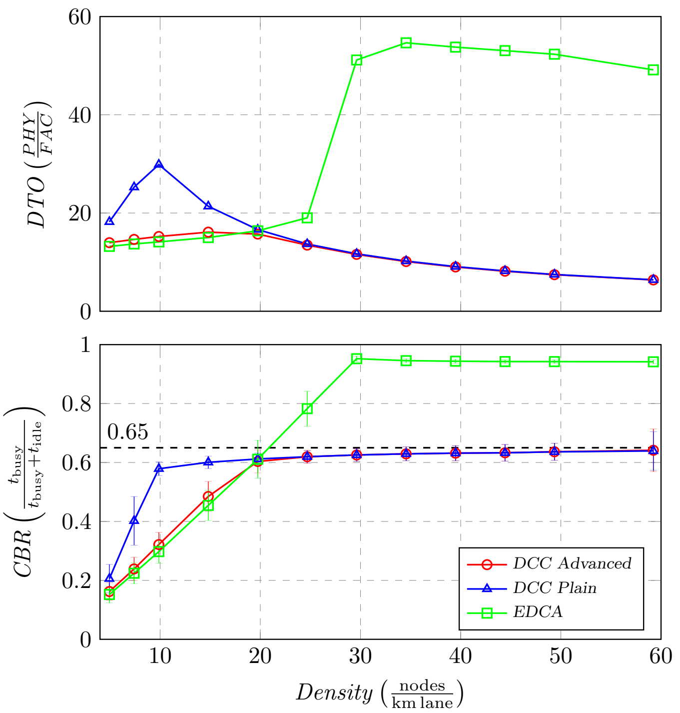

Figure 3 – Average data traffic overhead (DTO) and channel busy ratio (bottom) over node density for DENM.

Channel Busy Ratio (CBR): The CBR describes the data load on a wireless channel as the ratio between the fraction of time the channel is detected to be busy, i.e., a packet transmission is ongoing, and the fraction of time a channel is idle (no transmission is ongoing), for a pre-defined time interval. For example, a CBR of 0.8 means that for 80 % of the time the channel is busy. The CBR is measured at the PHY in intervals of 25 ms and averaged over 10 values, whereas the average was calculated by a sliding window.

Data Traffic Overhead (DTO): The data traffic overhead exhibits the number of sent frames at the PHY divided by the number of generated DENMs at the facilities layer. Compared to CBR, the DTO considers the amount of data whereas CBR only measures the channel state, and does not count the number of generated messages.

VI. Performance Evaluation

This section describes the results of the simulation-based performance evaluation.

A. Channel Load

Figure 3 shows two metrics for the channel load – the data traffic overhead (DTO) and the average channel busy ratio (CBR) at the top and bottom sub-figure, respectively.

As expected, EDCA has a linearly increasing CBR up to almost 1 and hence fully saturates the wireless channel for high node densities. We note that a CBR of 1 can hardly be achieved due to inter-frame spaces used in the CSMA/CA scheme. The critical point lies between a vehicle density of 25 and 30 $nodes/km/lane$. Here, the DTO shows a steep slope, which indicates that the CBF algorithm breaks and the forwarding nodes generate a huge number of retransmissions. DCC Plain, for a node density of 10 $nodes/km/lane$, utilizes the channel twice as much as the other algorithms, EDCA and DCC Advanced, and quickly reaches the CBR threshold of 0.65. This behavior can be explained by the malfunctioned CBF mechanism for low densities, while the DCC gatekeeper effectively throttles the packets (locally generated and forwarded) at higher densities. DCC Advanced has the same behavior as EDCA up to a node density of 20 $nodes/km/lane$. Beyond this point, the LIMERIC algorithm increases T_{off} and the gatekeeper reduces the packet rate. Overall, for the whole range of node densities, the average CBR for DCC Plain and DCC Advanced is below the predefined channel busy ratio threshold of 0.65, and the DTO slightly

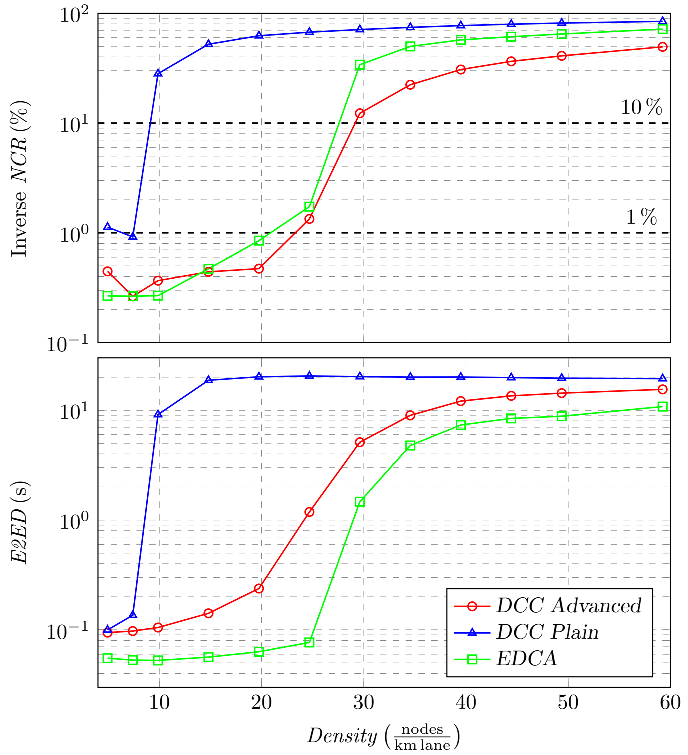

Figure 4 – Inverse node coverage ratio (NCR) and average end-to-end delay (E2ED) for DENM.

declines with the node density. While single nodes can temporarily exceed CBR_{th}, the LIMERIC algorithm ensures that in average the CBR sticks to the threshold. The advantage of DCC Advanced compared to DCC Plain is that it has a smaller CBR and DTO for low node densities.

B. Reliability and Latency

Figure 4 exhibits the inverse node coverage ratio (inverse NCR) and the end-to-end delay (E2ED) over the node density at the top and bottom sub-figure, respectively. The dashed lines indicate the threshold, when 1 and 10 % of nodes located in the geographical area miss the packet.

DCC Plain has an inverse NCR of 1 % for very low node densities. For densities greater than $5\,nodes/km/lane$, the curve value grows to 50 % for $15\,nodes/km/lane$. The reason is that even at low node densities the CBR (see Fig. 3) is already at 0.6 and all nodes attempt to forward the received packets. The forwarding causes a highly loaded channel (channel usage is at the permitted threshold of 0.65, see Fig. 3) and increases the gatekeeper's queue size. In these queues the packets wait for channel access, which is not granted due to the congested channel, and therefore many packets are discarded when they exceed their queuing lifetime. We will further illustrate the queue behavior below. There is an additional effect of the long queuing time: When channel access is granted, all packets with an expired queuing lifetime are discarded and those packets with a short remaining queuing lifetime are forwarded. As a result, we yield a very high end-to-end delay of almost $20\,s$ at $15\,nodes/km/lane$.

For EDCA and DCC Advanced the inverse NCR remains at low values, i.e. below 1 %, up to a density of 20 – $25\,nodes/km/lane$. Beyond this point the inverse NCR increases almost monotonically since the channel starts to be overloaded. This behavior results in a higher number of packet drops and DCC restricts the channel usage, respectively. For EDCA with higher node densities more and more nodes cannot overhear the retransmissions and therefore the number of redundant packets increases, which causes a lower reliability. With DCC Advanced, the overhearing works, however, the packet losses appear because of the highly overloaded channel.

Figure 5 illustrates the behavior of the queue size for DCC Plain, here for the VI-queue with regular (non-priority) DENMs: The top figure plots T_{off} over simulation time; we can see that the simulation reaches a steady state after 10 – 20 s and the value of T_{off} remains approximately constant for the remaining time. Considering the vehicle

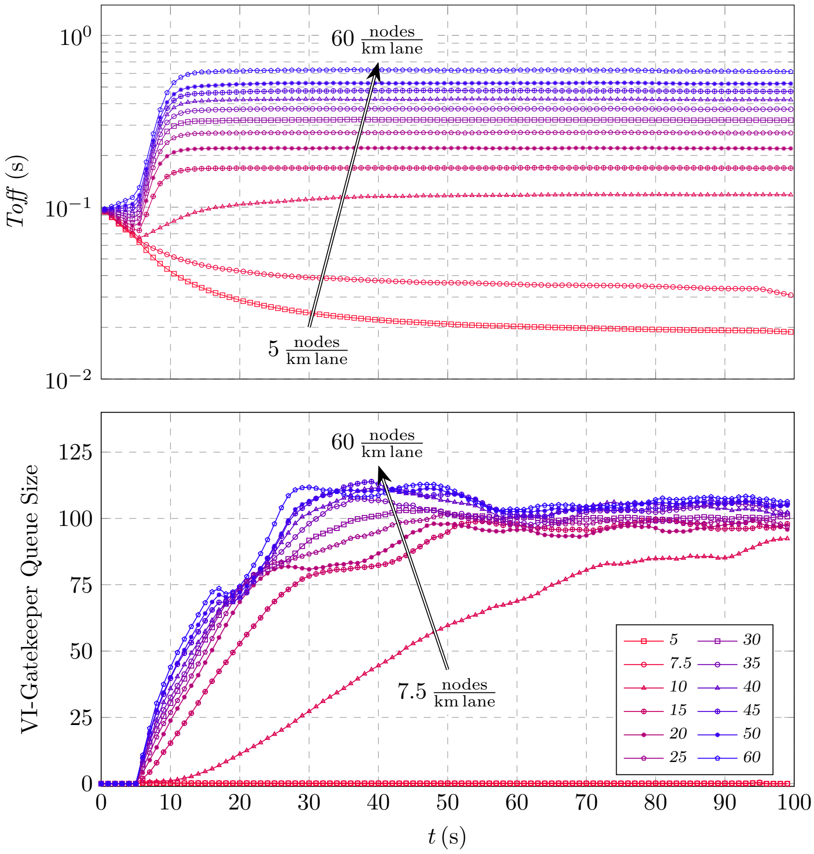

Figure 5 – Average T_{off} time (top) and VI-queue size (bottom) over simulation time for DCC Plain.

density as parameter, T_{off} grows from 40 to 600 ms for a density of 5 to 60 $nodes/km/lane$, respectively, which indicates that DCC Plain effectively restricts the channel access with increasing channel load. However, as it can be seen in the bottom part of Fig. 5, the queue size of the VI-queue quickly grows for all densities greater than 7.5 $nodes/km/lane$.[2] This behavior leads to the effect described above: More and more nodes cannot overhear the transmissions and CBF fails.

Comparing EDCA and DCC Advanced in terms of inverse NCR, EDCA is slightly better for low densities up to 15 $nodes/km/lane$. Beyond 15 $nodes/km/lane$, DCC Advanced shows a better performance, for selected density values up to 30 %. This number indicates that with the proposed mechanism, where the gatekeeper's flow control checks the packet duplication list and prevents packets from being forwarded when they are identified as duplicates, the overhearing works despite potential long queuing times.

Considering E2ED, values for DCC Advanced are considerably longer than for EDCA. The reason is the CBR threshold that is enforced by the gatekeeper: Only 65 % of the time the channel is used for packet transmission and the remaining time the channel is idle. The disadvantage is an increasing queue size and consequently longer queuing time, which results in a long E2ED. Thus, for DCC Advanced vs. EDCA we can state that the lower CBR comes at the costs of a longer E2ED at a comparable NCR.

Among the considered schemes, DCC Plain has by far the worst performance and disqualifies as a candidate scheme for multi-hop forwarding with gatekeeper-based DCC. DCC Advanced shows clearly the best performance in terms of low inverse NCR. However, compared to EDCA this advantage comes at the cost of a higher E2ED. Therefore, we can summarize that DCC Advanced implies a tradeoff between NCR and delay. However, considering safety applications, also the channel load need to be taken into account. For high node densities, EDCA is not able to restrict the channel load and (almost) fully saturates the wireless channel. In contrast, DCC Advanced would be able to send high-priority messages through the gatekeeper's higher-priority VO-queue, which is empty and therefore no delay occurs.

[2] Note that the transmission of DENMs start at 5 s simulated time, therefore the VI-queue for DENMs is empty for the first 5 s.

VII. Conclusions

In this paper we have studied mechanisms for decentralized congestion control (DCC) in vehicular communication systems based on IEEE 802.11 OCB / ITS-G5. We have presented the DCC architecture for the European C-ITS standardized by ETSI and in particular addressed the interaction between medium access control, decentralized congestion control (DCC) and contention-based forwarding (CBF) for multi-hop packet transport.

We have verified that for the medium access scheme EDCA the channel busy ratio and the data traffic overhead, both measures for the load in the wireless channel, significantly grow with the vehicle density, and hence make DCC mechanisms a necessity. We have further demonstrated that the proposed gatekeeper-based DCC approach is able to effectively control the channel load in terms of channel busy ratio and data traffic overhead. While CBF over EDCA shows good performance in terms of node coverage ratio and end-to-end delay, the use of the gatekeeper significantly lowers the CBF performance. Reason for the performance degradation is the fact that the gatekeeper queues packets with a delay that is in the same order of magnitude or longer compared to the maximum CBF timer duration. We have shown that a DCC extension, which enables the DCC flow control to access the packet duplication list of CBF, restores CBF's overhearing functionality. Our simulations indicate that CBF with the DCC extension considerably improves the inverse node coverage ratio and end-to-end delay.

References

1. J. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proc. of the IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
2. A. Festag, "Cooperative Intelligent Transport Systems Standards in Europe," *Communication Magazin*, vol. 12, no. 52, Dec. 2014.
3. *IEEE 802.11-2012 - IEEE Standard for Information technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std., Mar. 2012.
4. D. Eckhoff, N. Sofra, and R. German, "A Performance Study of Cooperative Awareness in ETSI ITS G5 and IEEE WAVE," in *IEEE WONS 2013*, Banff, Alberta, Canada, Mar. 2013.
5. S. Subramanian *et al.*, "Congestion Control for Vehicular Safety: Synchronous and Asynchronous MAC Algorithms," in *ACM VANET 2012*, Low Wood Bay, Lake District, UK, 2012, pp. 63–72.
6. 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 *IEEE WoWMoM 2015*, Boston, MA, USA, Jun. 2015, pp. 1–6.
7. 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 *IEEE WiVeC '14*, Vancouver, Canada, Sep. 2014, pp. 1–7.
8. G. Bansal, J. Kenney, and C. Rohrs, "LIMERIC: A Linear Adaptive Message Rate Algorithm for DSRC Congestion Control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, Nov. 2013.
9. *ETSI TS 102 687 V1.1.1 ITS DCC Mechanisms for Intelligent Transport Systems Operating in the 5 GHz Range; Access Layer Part*, ETSI Std., Jul. 2011.
10. *ETSI TS 103 175 V1.1.1 ITS; Cross Layer DCC Management Entity for Operation in the ITS G5A and ITS G5B Medium*, ETSI Std., Jun. 2015.
11. *ETSI EN 302 571 V2.0.3 (Draft) ITS; Radiocommunications Equipment Operating in the 5 855 MHz to 5 925 MHz Frequency Band; Harmonised Standard Covering the Essential Requirements of Article 3.2 of Directive 2014/53/EU*, ETSI Std., Mar. 2016.
12. ETSI, *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 (ETSI TR 101 612 V1.1.1)*, Sep. 2014.
13. G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
14. A. Brachman and L. Chrost, "On the Shortcoming of EDCA Queues in the Presence of Multiservice Traffic," in *2nd IFIP WG 6.8 / WMNC 2009*, Gdansk, Poland, Sep. 2009, pp. 69–79.
15. S. Kühlmorgen, I. Llatser, A. Festag, and G. Fettweis, "Performance Evaluation of ETSI GeoNetworking for Vehicular Ad Hoc Networks," in *IEEE VTC Spring '15*, Glasgow, UK, May 2015, pp. 1–6.

16. S. Kühlmorgen, A. Festag, and G. Fettweis, “Impact of Decentralized Congestion Control on Contention-based Forwarding in VANETs,” in *IEEE SmartVehicles 2016*, Coimbra, Portugal, Jun. 2016.
17. J. Kenney, G. Bansal, and C. Rohrs, “LIMERIC: A Linear Message Rate Control Algorithm for Vehicular DSRC Systems,” *ACM VANET 2011*, pp. 1–10, Sep. 2011.
18. T. Abbas, K. Sjberg, J. Karedal, and F. Tufvesson, “A Measurement Based Shadow Fading Model for Vehicle-to-Vehicle Network Simulations,” *arXiv:1203.3370 [cs]*, Mar. 2012. [Online]. Available: http://arxiv.org/abs/1203.3370
19. D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent Development and Applications of SUMO – Simulation of Urban MObility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.