

An Age-Of-Information Perspective on Decentralized Congestion Control in Vehicular Networks

Ion Turcanu^{*}, Andrea Baiocchi[†], Nikita Lyamin[†] and Alexey Vinel^{*}

^{*}Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg

[†]Dept. of Information Engineering, Electronics and Telecommunications (DIET), University of Rome Sapienza, Italy

[‡]Volvo Car Corporation, Sweden

[§]School of Information Technology, Halmstad University, Sweden

ion.turcanu@uni.lu andrea.baiocchi@uniroma1.it

nikita.lyamin@volvocars.com alexey.vinel@hh.se

Abstract—Vehicular networking enables a wide range of emerging Cooperative Intelligent Transportation System (C-ITS) applications, from safety to traffic efficiency and infotainment. Many of these applications depend on the reliability and timeliness of status information periodically exchanged among vehicles on the same wireless communication channel. A major effort has been spent, especially by standardization bodies, to define congestion control algorithms for the vehicular networking environment. The picture is, however, more complex than simply controlling the load level on the channel, given the non-trivial interplay of delivery reliability, system throughput, and timeliness of updates. In this paper, we provide a comprehensive performance evaluation of the main state-of-the-art broadcast rate control algorithms from the point of view of channel load, utilization efficiency, and information freshness. We evaluate these algorithms in a realistic simulation environment and describe a centralized approach to define a bound on the performance. We show that controlling the congestion based on either channel load or information freshness only leads to sub-optimal performance.

Index Terms—Age of Information, Decentralized Congestion Control, Performance Evaluation, Vehicular Networks.

I. INTRODUCTION

Vehicular networking is a cornerstone of Cooperative Intelligent Transportation Systems (C-ITSs) and autonomous vehicle management. It is expected to provide significant improvement in terms of transportation safety and efficiency in a shorter time horizon. Cooperative awareness and sensing of the road environment is among the key services supported by vehicular networking. Specifically, we consider a vehicular network in which vehicles periodically exchange 1-hop broadcast messages, e.g., Cooperative Awareness Messages (CAMs) [1], Basic Safety Messages (BSMs) [2], Collective Perception Messages (CPMs) [3], or Maneuver Coordination Messages (MCMs) [4]. The effectiveness of safety applications, built upon these messages exchange, highly depends on the accuracy and timeliness of status information contained in these messages and shared periodically among vehicles [5].

The Age-of-Information (AoI) metric has been introduced to measure the average information freshness of systems characterized by nodes that periodically exchange time-critical status updates [6], [7]. In their early work, Kaul et al. [7] identify a number of service systems in which real-time status updates are necessary and propose general methods to compute the AoI, demonstrating the existence of an optimal rate at which nodes should send their updates. AoI is especially

important in wireless networks, which are characterized by unreliable communication links. Kadota and Modiano [8] address the problem of minimizing the AoI of a wireless network with a base station that serves multiple streams to different destinations. The authors derive a lower bound on the AoI performance, independently of the considered queuing discipline. Liu and Bennis [9] consider the use case of ultra-reliable and low-latency industrial Internet of Things (IIoT) and propose to characterize the maximal AoI using extreme value theory. Maatouk et al. [10] formulate an optimization problem and calibrate the back-off time in order to minimize the average AoI in a Carrier Sense Multiple Access (CSMA) network.

Kaul et al. [6] propose a broadcast rate adaptation algorithm that minimizes the system AoI of a vehicular network, demonstrating that simply adapting the contention window size does not reduce the AoI. In our previous works [11]–[13], we derive analytical models that – given the network connectivity graph – characterize the AoI metric when varying the broadcast rate. These models are able to capture the trade-off between channel load and the resulting AoI. Plöger et al. [14] suggest to exploit the number of interfering vehicles to generate AoI samples that can be used by cooperative driving applications, such as platooning. Other approaches consider controlling the transmission power instead of broadcast rate. For example, Abdel-Aziz et al. [15] focus on minimizing the transmit power to control the tail of AoI distribution in vehicular networks.

Broadcast (or transmission) rate control is also one of the three techniques proposed by ETSI to manage the network load in vehicular networks [16]. The other two are transmit power and data rate control. In particular, there are two main Decentralized Congestion Control (DCC) mechanisms standardized by ETSI: a reactive approach and an adaptive approach. Both these mechanisms are based on the measurement of so-called Channel Busy Ratio (CBR), which is directly related to the message delivery reliability in the vehicular networking environment. The adaptive approach is based on Linear Message Rate Integrated Control (LIMERIC) [17], an algorithm that uses linear feedback to adapt the message broadcast rate. A comparison of the ETSI DCC reactive approach and LIMERIC is provided in [18] and [19], where the superiority of LIMERIC in terms of stability and throughput is demonstrated.

Several recent works [20]–[22] evaluate the DCC perfor-

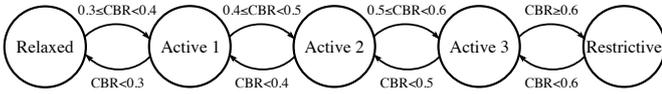


Figure 1. Outline of DCC Reactive state machine.

mance in cooperative vehicular applications, highlighting the importance of properly selecting the DCC parameters. In our previous work [20], we show that wrong DCC parametrization can have a negative impact on such applications, mainly because current DCC configuration focuses on controlling the CBR level, but not the system level application metrics, such as AoI. In fact, in another work [23], we propose a DCC state-machine and gatekeeper configuration for platooning applications that aims at meeting the requirements in terms of AoI while also controlling the channel congestion. However, in [23] we do not provide a comprehensive analysis and evaluation of the standardized ETSI DCC approaches, which is the focus of the current study.

None of the existing works, however, analyzes the existing DCC approaches from the point of view of both channel load impact and system level performance metrics in the context of applications relying on periodic one-hop broadcast messages. In this paper, we fill this gap by providing a comprehensive performance evaluation of the current DCC standard approaches [16] in terms of both congestion levels and AoI performance. We compare these mechanisms with an approach that focuses only on minimizing the AoI [6]. We show that the existing solutions trade off channel load against information freshness, indicating that both CBR and AoI have to be considered in future designs of congestion control mechanisms. In addition, we provide a bound on performance based on an existing heuristic [24] that computes a centralized collision-free scheduling assuming the vehicular network connectivity graph is known.

II. ADAPTIVE RATE CONTROL ALGORITHMS

We describe three state-of-the-art distributed rate control algorithms for vehicular networks: DCC Reactive, DCC Adaptive, and AoI Adaptive. The first two approaches are proposed by ETSI [16] and represent the current standard for decentralized congestion control in vehicular networks. The latter is an adaptive broadcast rate algorithm proposed by Kaul et al. [6] that aims at minimizing the system AoI. We also present a centralized heuristic based on the work of Ramaswami and Parhi [24] that minimizes the number of interference-free slots in which every vehicle is given the opportunity to broadcast at least once. This heuristic, which we refer to as CentralSchedule, provides a bound on the minimum AoI that can be achieved in a broadcast-based distributed network.

A. DCC Reactive

DCC Reactive is described in [16] (clause 5.3) and consists of a state machine composed of several states a vehicle can be in, based on the current local CBR as measured by the vehicle itself (see clause 4.2.10 in [25]). The state machine has three main states: relaxed, active, and restrictive. The active state can be further divided in several sub-states. In this work, we focus on a configuration with three active states, as illustrated in Figure 1. Each vehicle evaluates its current state every T_{exec} and decides whether to switch to a neighboring state or not.

Table I
TRANSMISSION OPPORTUNITIES FOR EVERY STATE.

State	CBR	Packet rate
Relaxed	< 0.3	10 Hz
Active 1	0.3–0.39	5 Hz
Active 2	0.4–0.49	2.5 Hz
Active 3	0.5–0.59	2 Hz
Restrictive	> 0.6	1 Hz

Every state is characterized by a set of parameters that aim at controlling the congestion level. In general, there are different techniques to control the congestion level, i.e., by changing the transmit power, packet rate, and/or data rate. In this work, we only consider the transmit packet rate. Numerical values for the packet rate parameter associated to each state are given in Table I.

B. DCC Adaptive

DCC Adaptive is a distributed linear-rate control algorithm defined in [16] (clause 5.4) and based on LIMERIC, an algorithm proposed by Bansal et al. [17]. The main idea behind DCC Adaptive is to adjust every vehicle's packet rate so that the total channel load converges to a specified target. Each vehicle executes the algorithm periodically, every T_{exec} , and adapts the packet transmission rate accordingly. The algorithm itself consists of five main steps:

Step 1:

$$\overline{CBR} \leftarrow 0.5 \cdot \overline{CBR} + 0.5 \cdot \left(\frac{CBR_L + CBR_{\text{Prev}}}{2} \right) \quad (1)$$

Step 2:

If $\text{sign}(CBR_{\text{target}} - \overline{CBR})$ is positive **then**

$$\delta_{\text{off}} = \min(\beta \cdot (CBR_{\text{target}} - \overline{CBR}), G_{\text{max}}^+) \quad (2)$$

Else

$$\delta_{\text{off}} = \max(\beta \cdot (CBR_{\text{target}} - \overline{CBR}), G_{\text{max}}^-) \quad (3)$$

Step 3:

$$\delta \leftarrow (1 - \alpha) \cdot \delta + \delta_{\text{off}} \quad (4)$$

Step 4:

$$\mathbf{If} \ \delta > \delta_{\text{max}} \ \mathbf{then} \ \delta = \delta_{\text{max}} \quad (5)$$

Step 5:

$$\mathbf{If} \ \delta < \delta_{\text{min}} \ \mathbf{then} \ \delta = \delta_{\text{min}} \quad (6)$$

Here, \overline{CBR} represents the moving average of computed CBR values, CBR_L is the local CBR measured by the vehicle, CBR_{Prev} is the second most recent CBR_L . Also, CBR_{target} , α , β , G_{max}^+ , G_{max}^- are algorithm control parameters. Finally, δ is a non-dimensional parameter representing the maximum fraction of time the vehicle is allowed to occupy the wireless channel, and δ_{max} and δ_{min} are the maximum and minimum values of δ respectively.

Based on δ , the vehicle can calculate when the next transmission should occur by computing the waiting time right after the last transmission finished:

$$T_{\text{wait}} = t_{\text{last}} + \min \left(\max \left(\frac{T_{\text{tx}}}{\delta}, 25 \text{ ms} \right), 1 \text{ s} \right), \quad (7)$$

where T_{wait} is the waiting time before the next transmission, t_{last} is the time instance when the last transmission finished, and T_{tx} is the transmission duration of the last frame. To be noted that $0.025 \text{ s} \leq T_{\text{wait}} \leq 1 \text{ s}$.

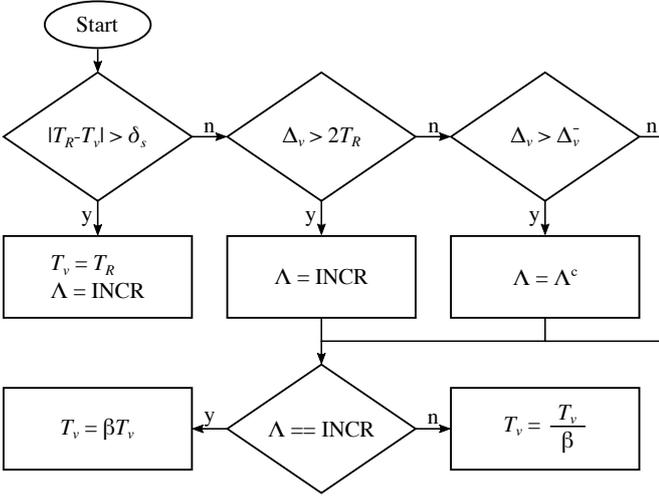


Figure 2. AoI Adaptive algorithm operation.

C. AoI Adaptive

AoI Adaptive is a distributed broadcast rate control algorithm defined by Kaul et al. [6] that uses the local estimated AoI to decide whether to increase or decrease the current packet transmission rate. To be noted that the performance parameter used by this algorithm to control the broadcast rate is AoI, which is different than the metric used by DCC Reactive and DCC Adaptive, which use CBR instead. This means that, even though the control parameter is the same, the target system metric to be optimized is different.

AoI Adaptive operation is illustrated in Figure 2. The algorithm requires as input the current broadcast period of the vehicle T_v , the average broadcast period of all neighboring nodes T_R , the current local estimated AoI Δ_v , the second most recent estimated AoI Δ_v^- , the maximum allowed broadcast period spread δ_s , the action Λ , and the control parameter β . The action Λ and beacon interval T_v are state variables that can be updated each time the algorithm is run. The output of the algorithm is T_v – the updated broadcast period of vehicle v . To be noted that the value of Δ_v is computed according to Equation (9). The algorithm is executed by each vehicle with period T_{exec} .

AoI Adaptive assumes that vehicles include their current broadcast period in the exchanged beacon messages, in order to be able to calculate T_R . A vehicle first checks if the spread of the broadcast period, $|T_R - T_v|$, is greater than δ_s . If this is the case, then the current broadcast period T_v is set to the neighborhood average T_R and Λ is initialized. To be noted that the action Λ can have two values: (i) INCR, which indicates that the value of T_v has to be increased, and (ii) DECR, which indicates that T_v has to be decreased. If the spread is below the threshold, the vehicle checks if the current estimated AoI is greater than $2T_R$. If this is true, it means the medium is congested and T_v has to be increased, hence $\Lambda = \text{INCR}$. Otherwise, the vehicle verifies if the current estimated AoI, Δ_v , is increasing with respect to the previous estimated AoI, Δ_v^- . In this case, it means that whatever the current action Λ is, it has a negative impact on the estimated AoI, hence, it needs to reverse its current action. To be noted that Λ^c is the complement of Λ . Finally, if the value of Λ is INCR, the current broadcast period is increased by a factor β , otherwise it is decreased by the same factor.

D. Central Schedule

Central Schedule is a centralized scheduling mechanism that aims at finding an efficient and interference-free broadcast schedule in a wireless network. It is based on the work of Ramaswami and Parhi [24], who propose a heuristic based on sequential graph coloring to find the minimal-length interference-free schedule, assuming global knowledge of the network connectivity graph. We apply this heuristic to the vehicular networking context and use it to find bounds for relevant performance metrics. We assume the algorithm is executed periodically, every T_{exec} , by a central entity that has updated global knowledge about the network status at all times.

Let $G = (V, E)$ be the network connectivity graph, where V is the set of vehicles, and E is the set of edges. Let $u, v \in V$ be any two vehicles in V . Then, an edge $e = (u, v) \in E$ if and only if u and v are able to receive and correctly decode each other's broadcast messages. The algorithm assumes all edges are bidirectional, all nodes are globally synchronized, and broadcast packets have the same length. The time is split into slots of duration equal to the transmission time of a packet plus an additional guard band to account for propagation delays.

The main goal of Central Schedule is to assign transmission slots to each vehicle in a way to achieve a conflict-free (or interference-free) schedule using as few slots as possible. The definition of conflict is twofold:

- the same transmission slot s is assigned to two vehicles $u, v \in V$, and there is an edge $e = (u, v) \in E$ (i.e., the transmissions of two 1-hop neighbor vehicles overlap in time);
- the same transmission slot s is assigned to two vehicles $u, v \in V$, and there is a third vehicle z such that there are two edges $e_1 = (u, z) \in E$ and $e_2 = (z, v) \in E$ (i.e., the transmissions of two 2-hop neighbor vehicles overlap in time).

In other words, two vehicles are in a conflict-free situation, if they are more than two hops away in the connectivity graph. This definition is motivated by the assumption that reception disruption is only possible if the transmissions of two neighbors of a given node overlap. In other words, if node u transmits a message and no one of its 2-hop neighboring nodes transmits in the meantime, we assume that all neighbors of u decode the message successfully.

Defining a message schedule that avoids any such conflict, while giving every node an opportunity to send its messages, can be stated in terms of the 2-hop vertex coloring problem from graph theory. In general, the problem of finding the minimum number of slots for a conflict-free schedule is NP-complete.

Central Schedule is an heuristic that computes such a schedule in two main steps. In the first step, it finds the number of slots needed to schedule every vehicle exactly once (see Algorithm 1). Let $\mathcal{N}_v^{1,2}$ be the set of 1-hop and 2-hop neighbors of v . The algorithm iterates over all vehicles in V . At each iteration, it initializes the slot s (line 3). Then, it increments s for every $u \in \mathcal{N}_v^{1,2}$ that has already been assigned in slot s (lines 4–6). Finally, the algorithm assigns to v the first available slot that has not yet been assigned to any of the 1-hop and 2-hop neighbors of v (line 7). This process is repeated until all vehicles in V have an assigned

conflict-free transmission slot. The output of the first step is a schedule table S that maps every vehicle to exactly one specific slot s .

The second step of Central Schedule, summarized in Algorithm 2, maximizes the number of vehicles that can be scheduled in the same slot without creating conflicts. This step takes as input the schedule table S , obtained with Algorithm 1. For each $v \in V$ and for each $s \in S$, if there is no $u \in \mathcal{N}_v^{12}$ that is already scheduled in slot s , then assign slot s to v (lines 3–11). The output of this second step is a schedule table S in which the number of scheduling opportunities for each vehicle v is maximal.

III. PERFORMANCE METRICS

We define four main performance metrics used to evaluate the considered algorithms: (i) system AoI, (ii) beacon interval, (iii) CBR, and (iv) channel utilization efficiency. In the following, we describe how these metrics are computed when using both the distributed rate control algorithms and Central Schedule. The latter provides a bound on the considered metrics, since it is a centralized scheduling protocol based on global knowledge of the network connectivity graph. To be noted that these metrics bear a general meaning. In particular, they can be applied to different networking technologies (e.g., IEEE 802.11p, C-V2X), fully maintaining their meaning and definition.

A. System AoI

Every vehicle $v \in V$ generates broadcast messages with period T_v , where V is the total set of vehicles. Let Δ_{uv} be the average AoI of messages sent by u and received by v , as illustrated in Figure 3. The AoI of u 's information increases

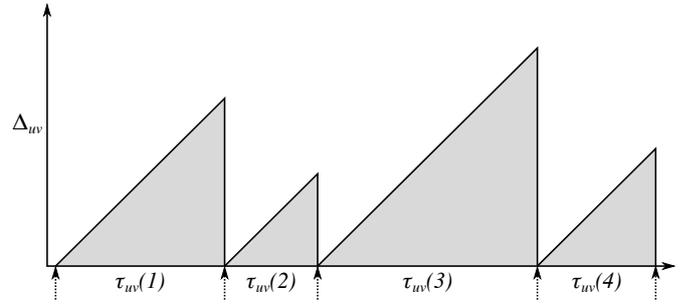


Figure 3. Average AoI (grey area) of messages sent by vehicle u and received by v . Dotted arrows represent arrival of messages from u at v .

linearly in time, and drops to zero¹, when v receives a new message from u .

Let us focus on an observation time interval \mathcal{I} and let $M_{uv}^{\mathcal{I}}$ be the set of messages v received from u during interval \mathcal{I} . Also, let $\tau_{uv}(i)$ be the time interval between the reception of messages m_{i-1} and m_i send by u and received by v . Then, Δ_{uv} related to the observation interval \mathcal{I} can be computed as follows:

$$\Delta_{uv} = \frac{1}{2} \frac{\sum_{i=1}^{|M_{uv}^{\mathcal{I}}|} \tau_{uv}^2(i)}{\sum_{i=1}^{|M_{uv}^{\mathcal{I}}|} \tau_{uv}(i)} \quad (8)$$

The average AoI over all neighbors u sending messages to a given node v can be found as follows:

$$\Delta_v = \frac{1}{|\mathcal{N}_v^1|} \sum_{u \in \mathcal{N}_v^1} \Delta_{uv} \quad (9)$$

where \mathcal{N}_v^1 represents the set of 1-hop neighbors of v . Finally, the average system AoI of all vehicles in the network, Δ , is found by means of a weighted average of the Δ_v 's, where the weight of Δ_v is the cardinality $|\mathcal{N}_v^1|$ of the neighbor set of v :

$$\Delta = \frac{\sum_{v \in V} |\mathcal{N}_v^1| \Delta_v}{\sum_{v \in V} |\mathcal{N}_v^1|} = \frac{\sum_{v \in V} \sum_{u \in \mathcal{N}_v^1} \Delta_{uv}}{\sum_{v \in V} |\mathcal{N}_v^1|} \quad (10)$$

In case of Central Schedule, the system AoI is computed based on the schedule table S . In particular, let $|S|$ be the size of the schedule table, representing the maximum number of unique slots identified by Central Schedule. Let $\{s_1, s_2, \dots, s_{|S_u|}\}$ be the subset of slots assigned to vehicle u , where $S_u \subseteq S$. Since all messages sent by u in the assigned slots are received by v , it is possible to see that $k_{uv}(i) = s_i - s_{i-1}$, i.e., it is the difference (in terms of number of slots) between two consecutive transmissions of u . Assuming T is the slot duration, computed as L/R (L being the message length and R the transmission data rate), then:

$$\Delta_{uv} = \frac{\frac{1}{2} T^2 \sum_{i=1}^{|S_u|} k_{uv}^2(i)}{T \sum_{i=1}^{|S_u|} k_{uv}(i)} = \frac{T}{2|S|} \sum_{i=1}^{|S_u|} k_{uv}^2(i) \quad (11)$$

since obviously it is

$$\sum_{i=1}^{|S_u|} k_{uv}(i) = |S| \quad (12)$$

To be noted that $\Delta_{uv} \equiv \Delta_u$, since Central Schedule guarantees that every message transmission leads to a successful reception

¹We are neglecting here the MAC access delay. Given the relatively low level of congestion maintained by the DCC algorithms, MAC access delay is quite small with respect to the considered beacon intervals, which are never below 25 ms and often much bigger.

Algorithm 1 Central Schedule: first step

Input: $G = (V, E)$; \mathcal{N}_v^{12} set of 1- and 2-hop neighbors of v

Output: Schedule table S

- 1: $S[v] \leftarrow 0$ for all $v \in V$
 - 2: **for all** $v \in V$ **do**
 - 3: $s \leftarrow 1$
 - 4: **while** $u \in \mathcal{N}_v^{12}$ AND $S[u] = s$ **do**
 - 5: $s \leftarrow s + 1$
 - 6: **end while**
 - 7: $S[v] \leftarrow s$
 - 8: **end for**
-

Algorithm 2 Central Schedule: second step

Input: $G = (V, E)$; \mathcal{N}_v^{12} ; Schedule table S

Output: S with maximal broadcasting set in each slot

- 1: **for all** $v \in V$ **do**
 - 2: **for all** $s \in S$ **do**
 - 3: $assigned \leftarrow \text{FALSE}$
 - 4: **for all** $u \in \mathcal{N}_v^{12}$ **do**
 - 5: **if** $S[u] = s$ **then**
 - 6: $assigned \leftarrow \text{TRUE}$
 - 7: **end if**
 - 8: **end for**
 - 9: **if** $\neg assigned$ **then**
 - 10: $S[v] \leftarrow s$
 - 11: **end if**
 - 12: **end for**
 - 13: **end for**
-

at all neighbors of the transmitting node u . Finally, the system AoI can be computed as:

$$\Delta = \frac{\sum_{v \in V} |\mathcal{N}_v^1| \Delta_v}{\sum_{v \in V} |\mathcal{N}_v^1|} \quad (13)$$

B. Beacon interval

The beacon interval of a vehicle v , T_v , is defined as the time interval between two consecutive message transmissions. T_v represents the main parameter the rate control algorithms can act on in order to control the congestion. Let $\{T_v^1, T_v^2, \dots, T_v^{|\mathcal{T}|}\} \in \mathcal{T}$ be the total set of transmission intervals of a given vehicle v . Then, the average beacon interval of vehicle v is simply

$$\bar{T}_v = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} T_v^i \quad (14)$$

To be noted that, in case of Central Schedule, $T_v^i \equiv k_{uv}(i) \cdot T$ for every $u \in \mathcal{N}_v^1$, where T is the slot time. Finally, the system mean beacon interval is

$$T_{\text{BI}} = \frac{1}{|V|} \sum_{v \in V} \bar{T}_v \quad (15)$$

C. Channel Busy Ratio

CBR represents the ratio of time the channel is sensed busy to the total measured time. Let CBR_v be the CBR value computed by the vehicle v as

$$CBR_v = \frac{T_{\text{busy}}}{T_{\text{busy}} + T_{\text{idle}}} \quad (16)$$

where T_{busy} is the total time the channel is sensed busy and T_{idle} is the time the channel is idle, as sensed by v . If \mathcal{I} is the observation interval, then we have $\mathcal{I} = T_{\text{busy}} + T_{\text{idle}}$. For Central Schedule, CBR_v is computed as

$$CBR_v = \frac{1}{|S|} \sum_{u \in \mathcal{N}_v^1} |S_u| \quad (17)$$

Finally, the mean system CBR is simply

$$\overline{CBR} = \frac{1}{|V|} \sum_{v \in V} CBR_v \quad (18)$$

D. Channel utilization efficiency

This metric measures how efficiently the channel is utilized, i.e., the ratio of time broadcast messages are received successfully. It is calculated by every vehicle v as follows:

$$U_v = \frac{T_{\text{rx}}}{\mathcal{I} - T_{\text{tx}}} \quad (19)$$

Here, \mathcal{I} is the total time span over which the metric is calculated, T_{rx} is the total time v correctly receives messages from neighbor vehicles, and T_{tx} is the total time v itself is sending messages. The mean system utilization efficiency is

$$\bar{U} = \frac{1}{|V|} \sum_{v \in V} U_v \quad (20)$$

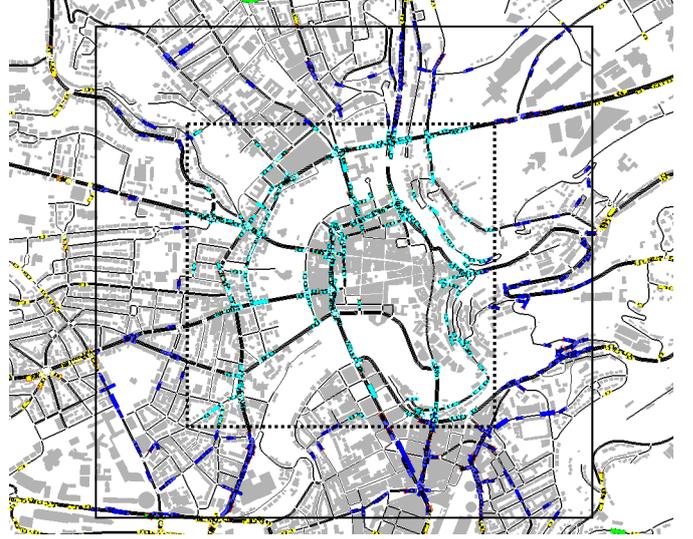


Figure 4. Simulation scenario: the outer rectangle (solid line) delimits the region of interest where the network is simulated (blue and cyan vehicles); the inner rectangle (dotted line) denotes the area from where statistics are collected (cyan vehicles).

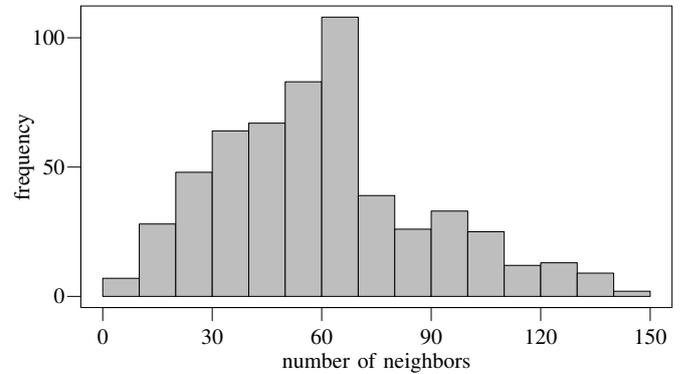


Figure 5. The distribution of number of 1-hop neighbors in the simulated scenario.

IV. SIMULATION SETUP

The considered rate control algorithms are implemented and evaluated in a realistic simulation framework composed of the discrete-event network simulator OMNeT++ 5.6.2 [26], the realistic road traffic and mobility simulator SUMO 1.3.1 [27], and the open-source vehicular network simulation framework Veins 5.0 [28].

The simulation study is conducted using a well-established and highly-realistic simulation scenario that models the traffic flow and vehicular mobility patterns in the city of Luxembourg over a 24h time span [29] (see Figure 4). We focus on a 30s time interval and the highest vehicular density of 205 veh/km², offered by this scenario. The simulated area is a 2.5 km x 2.5 km square covering the center of the simulation scenario. To avoid border effects, statistics are only collected from vehicles roaming inside a 1.5 km x 1.5 km area in the center of the simulated scenario. The distribution of the number of neighbors, obtained from a simulation run with fixed $T_v = 1$ s and $L = 100$ Byte, is presented in Figure 5. The main simulation and algorithm parameters are illustrated in Table II.

Table II
SIMULATION AND ALGORITHM PARAMETERS

Parameter	Value
Communication technology	IEEE 802.11p
Channel	5.89 GHz
Transmission power	100 mW
Bandwidth	10 MHz
Bitrate	6 Mbit/s
Beacon length L	100, 300, 600, 900, 1200 and 1500 Byte
DCC Reactive	
T_{exec}	0.1 s
DCC Adaptive	
T_{exec}	0.1 s
α	0.016
β	0.0012
CBR_{target}	0.62
δ_{max}	0.03
δ_{min}	0.0006
G_{max}^+	0.0005
G_{max}^-	-0.00025
AoI Adaptive	
T_{exec}	2 s
δ_s	0.05
β	1.1
Central Schedule	
T_{exec}	1 s

A. Implementation

DCC Reactive, DCC Adaptive, and AoI Adaptive are implemented as applications running in each vehicle's On-Board Unit (OBU). Vehicles broadcast beacon messages, with period T_v , containing basic status information. In addition, every vehicle maintains a local neighbor table where information coming from neighbor vehicles is stored. The value of the control parameter T_v is updated by the rate control algorithms every T_{exec} based on the measured performance parameters CBR (in case of DCC Reactive and DCC Adaptive) and Δ_v (in case of AoI Adaptive). The performance metrics (defined in Section III) are measured in the simulation environment, reflecting the actual status of the network.

Central Schedule runs as an *oracle* application that has a global overview of the network connectivity graph at all times. The oracle emulates a central scheduler that runs, e.g., in a 5G gNB or ITS-G5 Road Side Unit (RSU). In practice, the oracle takes periodic snapshots, every T_{exec} , of all vehicles' neighbor tables in order to create the graph G . To reflect link information as accurately as possible and make neighbor tables independent of the beacon interval, we use the same $T_v = 1$ s for all vehicles when simulating Central Schedule.

To be noted that we do not explicitly simulate the communication links between vehicles and oracle. In particular, we only simulate the beacon exchange process to obtain the connectivity graph G , while all performance metrics are drawn purely based on the results of Central Schedule (the schedule table S) rather than simulation. This is because we are interested in finding a theoretical bound on the performance metrics assuming global knowledge and conflict-free scheduling (i.e., no packet loss due to message collisions). In addition, since the results of the two heuristics used by Central Schedule (described in Algorithms 1 and 2) depend on the order in which the nodes are processed, we repeat the execution of Central Schedule 10 times for each new graph G and consider the schedule S that has the minimum cardinality.

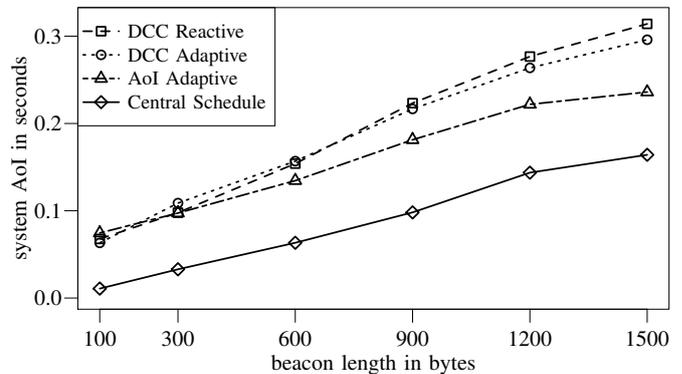


Figure 6. Mean system AoI (Δ) for different beacon length (L) values.

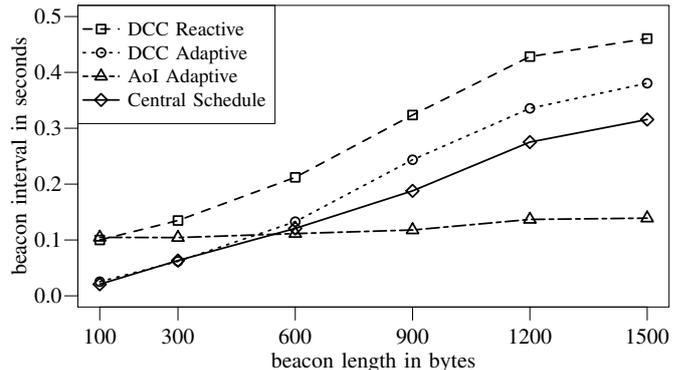


Figure 7. Mean system beacon interval (T_{BI}) for different beacon length (L) values.

V. PERFORMANCE EVALUATION

Figure 6 illustrates the mean system AoI (Δ) measured for different beacon length values and all four considered algorithms. As expected, Δ increases with the beacon length, as the communication channel becomes more congested. The best performance is achieved by Central Schedule, which indicates the lowest achievable AoI. Numerically, the mean system AoI that is obtained by Central Schedule when $L = 1500$ Byte is $\Delta \approx 0.16$ s. Clearly, all the distributed rate control algorithms yield an inferior performance in terms of Δ , since vehicles adapt their beacon intervals based on local knowledge only. Among these, AoI Adaptive achieves the best performance, since AoI is the performance parameter it tries to minimize by design. This is more visible when the beacon sizes are higher. For $L = 1500$ Byte, we obtain $\Delta \approx 0.24$ s with AoI Adaptive, which is a decrease of $\approx 25\%$ with respect to DCC Reactive and $\approx 20\%$ compared to DCC Adaptive. The reason is that DCC Reactive and DCC Adaptive are driven by the CBR performance rather than AoI, which means that they tend to increase the beacon interval in order to keep the channel congestion below certain operational levels.

This is confirmed by the results highlighted in Figures 7 and 8. Figure 7 illustrates the mean system beacon interval T_{BI} of all vehicles in the network when varying the beacon length, while Figure 8 shows the mean CBR. As expected, the algorithms that trigger on average lower beacon intervals generally lead to higher CBR values. For example, DCC Reactive, that has the highest beacon intervals among all the algorithms, independently of the considered beacon length, has also the lowest CBR. At the same time, DCC Adaptive, that has been designed to use the channel more efficiently, tends to reach

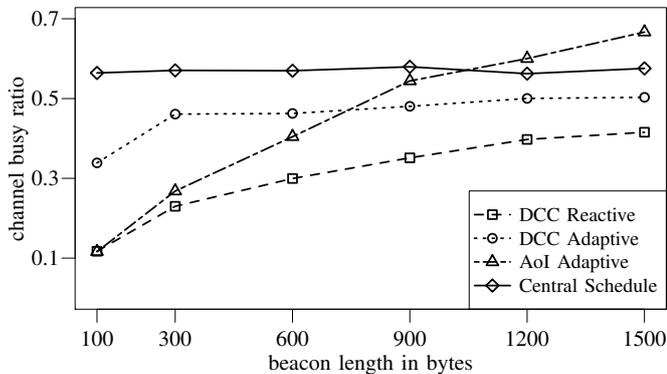


Figure 8. Mean system CBR for different beacon length (L) values.

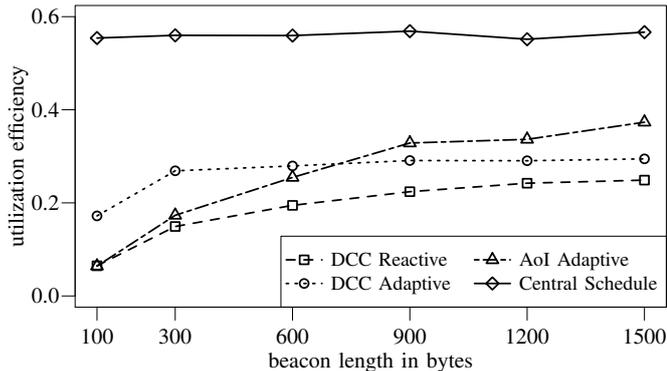


Figure 9. Mean system utilization efficiency (\bar{U}) for different beacon length (L) values.

the CBR_{target} faster than DCC Reactive by keeping the beacon intervals lower. Interestingly, the slope of the beacon interval curve of AoI Adaptive is much smaller when compared to all other algorithms, which leads to the biggest slope in terms of CBR. This means that AoI Adaptive is less sensitive to the CBR variations, focusing mostly on keeping the system AoI low. Central Schedule represents the best case scenario of what an adaptive rate control algorithm is supposed to do: it adapts the beacon interval so as to keep the CBR constant while minimizing the system AoI. In our case, the CBR achieved with Central Schedule is always ≈ 0.57 .

The channel utilization efficiency is illustrated in Figure 9. Here, we can see that the highest utilization efficiency is achieved by Central Schedule, which, in fact, provides an upper bound on \bar{U} . As expected, the results show also that DCC Adaptive utilizes the channel more efficiently when compared to DCC Reactive, mainly due to the lower beacon intervals generated by DCC Adaptive. For $L \leq 600$ Byte, the utilization efficiency of AoI Adaptive is in between that of DCC Reactive and DCC Adaptive, while for $L > 600$ Byte AoI Adaptive yields the highest efficiency among the distributed algorithms. This result is consistent with the beacon interval results presented in Figure 7.

However, a low beacon interval does not necessarily mean a low AoI as well. This can be seen when looking at the difference between AoI Adaptive and Central Schedule. Despite the fact that, for $L > 600$ Byte, the mean beacon interval of Central Schedule is higher than AoI Adaptive, the system AoI is still smaller. The main reason is the better channel utilization efficiency achieved by Central Schedule, which, by design, does not generate packet collisions, hence,

all broadcast messages are correctly received. This indicates that many of the broadcast messages are lost when using the distributed rate control algorithms, which decreases the overall utilization efficiency and increases the system AoI.

VI. CONCLUSION

In this paper, we provided a detailed performance evaluation of the main state-of-the-art congestion control mechanisms for vehicular networks. In particular, we evaluated the standardized ETSI Decentralized Congestion Control (DCC) algorithms [16] – DCC Reactive and DCC Adaptive – and compared their performance with an adaptive broadcast rate control algorithm that focuses on minimizing the system Age-of-Information (AoI) [6]. These mechanisms typically adapt the broadcast rate in order to optimize a certain performance metric: either the channel congestion level, or the information freshness. Most of the existing related works evaluate these algorithms focusing on the corresponding target performance metric only.

In this paper, we filled this gap by providing a comprehensive evaluation of these mechanisms from both channel load and information freshness perspective. We also measured how efficiently is the channel used by each of these mechanisms. In addition, we described a heuristic that provides performance bounds to each of the considered metrics. Our results suggest that only focusing on keeping low the channel load can lead to higher AoI values. This means that safety-critical information could become obsolete. At the same time, adapting the broadcast rate focusing on minimizing the information freshness only, can lead to high Channel Busy Ratio (CBR) values, which has a negative impact on information reliability. In fact, striving for guaranteeing update freshness of delivered messages does not necessarily imply that *all* relevant messages are gathered at every receiver. In addition, other applications that share the same wireless channel (e.g., infotainment, multimedia) could suffer a quality degradation if the channel is congested. As a consequence, new congestion control mechanisms that consider both the CBR and AoI variations are needed.

ACKNOWLEDGEMENT

The research leading to the results reported in this work has received funding from the H2020 5G-DRIVE project (ID: 814956), the Knowledge Foundation (KKS) in the framework of "Safety of Connected Intelligent Vehicles in Smart Cities – SafeSmart" project (2019–2023), Swedish Innovation Agency (VINNOVA) in the framework of "Emergency Vehicle Traffic Light Pre-emption in Cities – EPIC" project (2020–2022) and the ELLIIT Strategic Research Network. This support is gratefully acknowledged.

REFERENCES

- [1] ETSI EN 302 637-2 V1.3.2 (2014-11), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," Nov. 2014.
- [2] SAE Int., "Dedicated Short Range Communications (DSRC) Message Set Dictionary," Tech. Rep., Mar. 2016.
- [3] K. Garlich, H.-J. Günther, and L. C. Wolf, "Generation Rules for the Collective Perception Service," in *2019 IEEE Vehicular Networking Conference (VNC)*, IEEE, 2019, pp. 1–8.
- [4] D. Bischoff, F. Schiegg, T. Meuser, and R. Steinmetz, "Impact of Imperfect Communication on Cooperative Vehicular Maneuvering at Intersections," in *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, IEEE, 2020, pp. 1–5.
- [5] 5G Automotive Association (5GAA), "C-V2X Use Cases Volume II: Examples and Service Level Requirements," Tech. Rep., Oct. 2020.

- [6] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Jun. 2011, pp. 350–358.
- [7] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" In *Proceedings IEEE INFOCOM 2012*, Mar. 2012, pp. 2731–2735.
- [8] I. Kadota and E. Modiano, "Minimizing the Age of Information in Wireless Networks with Stochastic Arrivals," in *Proceedings of the 20th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. Mobihoc '19, Catania, Italy: ACM, 2019, pp. 221–230.
- [9] C.-F. Liu and M. Bennis, "Taming the Tail of Maximal Information Age in Wireless Industrial Networks," *IEEE Communications Letters*, vol. 23, no. 12, pp. 2442–2446, Dec. 2019.
- [10] A. Maatouk, M. Assaad, and A. Ephremides, "On the age of information in a CSMA environment," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 818–831, 2020.
- [11] A. Baiocchi and I. Turcanu, "A Model for the Optimization of Beacon Message Age-of-Information in a VANET," in *29th International Teletraffic Congress (ITC 29)*, vol. 1, Genoa, Italy: IEEE, Sep. 2017, pp. 108–116.
- [12] A. Baiocchi, I. Turcanu, N. Lyamin, K. Sjöberg, and A. Vinel, "Age of Information in IEEE 802.11p," in *17th IFIP/IEEE International Symposium on Integrated Network Management (IM): ITAVT Workshop*, Virtual Conference: IEEE, May 2021.
- [13] A. Baiocchi and I. Turcanu, "Age of Information of One-Hop Broadcast Communications in a CSMA Network," *IEEE Communications Letters*, vol. 25, no. 1, pp. 294–298, 2021.
- [14] D. Plöger, M. Segata, R. L. Cigno, and A. Timm-Giel, "Markov-modulated models to estimate the age of information in cooperative driving," in *IEEE Vehicular Networking Conference (VNC)*, IEEE, 2019, pp. 1–4.
- [15] M. K. Abdel-Aziz, S. Samarakoon, C.-F. Liu, M. Bennis, and W. Saad, "Optimized age of information tail for ultra-reliable low-latency communications in vehicular networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1911–1924, 2019.
- [16] ETSI TS 102 687 V1.2.1 (2018-04), "Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," Apr. 2018.
- [17] G. Bansal, J. B. Kenney, and C. E. Rohrs, "LIMERIC: A linear adaptive message rate algorithm for DSRC congestion control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, 2013.
- [18] 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 6th International Symposium on Wireless Vehicular Communications (WiVeC 2014)*, IEEE, 2014, pp. 1–7.
- [19] A. Rostami, B. Cheng, G. Bansal, K. Sjöberg, M. Gruteser, and J. B. Kenney, "Stability challenges and enhancements for vehicular channel congestion control approaches," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2935–2948, 2016.
- [20] 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.
- [21] A. Balador, C. Bai, and F. Sedighi, "A comparison of decentralized congestion control algorithms for multiplatooning communications," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, IEEE, 2019, pp. 674–680.
- [22] Z. Liang, F. Sedighi, and A. Balador, "Evaluation and optimization of decentralized congestion control algorithms for vehicular networks," in *IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, IEEE, 2019, pp. 1–8.
- [23] N. Lyamin, B. Bellalta, and A. Vinel, "Age-of-Information-Aware Decentralized Congestion Control in VANETs," *IEEE Networking Letters*, vol. 2, no. 1, pp. 33–37, 2020.
- [24] R. Ramaswami and K. K. Parhi, "Distributed scheduling of broadcasts in a radio network," in *IEEE INFOCOM '89, Proceedings of the 8th Annual Joint Conference of the IEEE Computer and Communications Societies*, IEEE, 1989, pp. 497–504.
- [25] ETSI EN 302 571 V2.1.1 (2017-02), "Intelligent Transport Systems (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," Feb. 2017.
- [26] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," in *1st ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008)*, Marseille, France: ACM, Mar. 2008.
- [27] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *IARIA International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, Dec. 2012.
- [28] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, pp. 3–15, Jan. 2011.
- [29] L. Codeca, R. Frank, and T. Engel, "Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research," in *7th IEEE Vehicular Networking Conference (VNC 2015)*, Kyoto, Japan: IEEE, Dec. 2015.