

Modelling and Examination of Collective Perception Service for V2X Supported Autonomous Driving*

Máté Herbert^a, András Váradi^b, László Bokor^a

^aBudapest University of Technology and Economics
Department of Networked Systems and Services

mherbert1@sch.bme.hu

bokor1@hit.bme.hu

^bCommsignia Ltd.

andras.varadi@commsignia.com

Abstract

Thanks to evolving cooperative V2X (Vehicle-to-Anything) communication, soon traffic participants do not have to use only their own perceptions of the environment to make a certain decision in a traffic situation, but also have the opportunity to use external originated data, which information could be hardly perceived or even totally unperceivable from their perspective. Collective Perception Service (CPS) is a Day 2 V2X service which can be used by an ITS-Station (ITS-S) to share information about its environment with other ITS-Ss by providing data about perceived objects (e.g., vehicles, obstacles, pedestrians) and available free space which can be safely allocated by other ITS-Ss (e.g., in an overtaking maneuver). Such extensive exchange of sensor data representations with growing bandwidth demand will aggravate the problem of the increasing load on the radio channels in future C-ITS use-cases. Motivated by this, we modeled CPS in an environment with changing vehicular traffic and communication conditions and examined the perceived Channel Busy Ratio (CBR) in multiple scenarios.

Keywords: V2X, Collective Perception Service, DCC, OMNeT++, Artery, SUMO

*The research reported in this paper was partly supported by the Higher Education Excellence Program in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI/FM) and by Commsignia Ltd.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Introduction

In recent times innovation of connectivity in all domains is continuing at an increasing pace. Information is becoming more reliable and is delivered faster than ever before, helping the IT ecosystem to perform at a similarly increasing speed. The transport industry has shown great interest in harvesting such connectivity-based solutions to deliver better C-ITS services. However, connectivity's significant role in ITS is to increase safety and efficiency radically. V2X (Vehicle-to-Everything) is a term used for low latency (and usually safety-relevant) information exchange between traffic participants. The concept of avoiding nearly all accidents involving all traffic participants by the use of V2X has been introduced in the early 2000s by the Automotive industry, followed by the founding of the Car2Car Communication Consortium and thereafter by initial cross-OEM pilot projects like CVIS [1], DriveC2X [2] and SimTD [3].

After nearly 20 years of research and validation, currently the so-called Day 1 system is being rolled out in Europe by major OEMs introducing (among other services) Cooperative Awareness (CA) where connected vehicles are made aware of each other to avoid accidents including non-line-of-site (NLOS) situations where classic ADAS systems are not able to help.

The Collective Perception Service (CPS) considered as a Day 2 service is foreseen as the biggest gamechanger when it comes to assisted and automated driving systems as it enables connected cars to not only disseminate their presence (via Cooperative Awareness) but the presence of all nearby (non-connected) entities whether it is a vehicle, a cyclist or an animal running across the street. CPS will digitalize traffic at a very early stage (meaning a meager penetration ratio of enabled cars). Collective Perception Message (CPM) used by CPS could include and share prosperous sensor data like position, speed, or dimension and even perceived objects and free space which describe the environment of the station around. ETSI ITS is currently standardizing the Collective Perception Message (CPM) under the Technical Specification number TS 103 324 with an initial Technical Report (TR 103 562) already published [5].

The message can be used to transmit up to 10 times per second the complete environmental perception of a single vehicle, providing sensor properties and capabilities, a database of identified objects, and occupiable free space within the range of its sensors. Consequently, CPS reduces the uncertainty of an ITS-S about the current environment, therefore supporting autonomous driving in the future.

With the growing penetration of V2X applications and V2X-capable devices, the load on the used channel will also increase. Frequent exchange of sensor data representations with higher size demand, like in the case of CPS, will further aggravate this problem. Therefore, simulation-based realistic modeling and examination of CPS and relating services is essential for further development of such advanced V2X solutions. This article introduces our initial simulation results gathered from Artery-based [13] CPS and sensor models we designed and developed relying on the framework capabilities [4]. Our implementation was then deployed and measured

in an environment with changing vehicular traffic and communication conditions and used in different traffic situations together with the mandatory and already deployed CA service.

The rest of the paper is organized as follows. First, we give a short overview of the background, focusing on the details of Decentralized Congestion Control (DCC) and CA and CP services. Then we introduce the simulation environment and the implemented scenarios and applied parameters we used in our experiments, followed by the gathered measurement results and their analysis. Finally, we conclude the paper and highlight our future work.

2. Background

2.1. Decentralized Congestion Control (DCC)

In the ETSI ITS-G5 C-ITS architecture DCC [11, 12] is a cross-layer entity with the main role of the congestion control of the packets in VANET (Vehicular Ad-hoc Network, implementing IEEE 802.11p [6] specifications) environments. The operation of DCC is based on CBR (Channel Busy Ratio) value measured on the selected channel. CBR is counted with the formula [7] $CBR = T_{busy}/T_{CBR}$, where T_{busy} is the duration of time in milliseconds, when the signal power of the measured channel exceeds -85 dBm through T_{CBR} , which equals to 100 ms. The essence of the DCC-based operation is that it delays message generation and packet transmission to control the congestion level of the channel. According to C2C-CC Basic System Profile [8], the Access layer part of DCC contains four queues, and each can store up to two messages. These queues have different priorities, and the messages fill them up according to the Traffic Class (TC) field value of each packet. The priority profiles of the queues are named from DP0 to DP3 (lower value means higher priority) [8]. DP0: TC = 0, only high priority DENM [9], DP1: TC = 1, only low priority DENM [9], DP2: TC = 2, only CAM [10], DP3: every other packet. When packet sending happens, the next message is chosen from a not empty queue, which has the highest priority. Behavior is not defined in the case when a packet arrives in a queue with no more free space. Still, in Artery (the simulation framework we use, see later), the oldest message is dropped in this situation so that messages with the most recent information shall be sent out the radio interface.

In our investigations, we used two reactive implementations of DCC, which means their operations are based on two Finite State Machines (FSM). One is called Fully Meshed, and the other one is called Gradual. Table 1 represent the states of these two FSMs. To each state, a T_{off} value is assigned, which is the minimum time interval between two message transmissions and generations.

These two implementations also operate quite differently. The Gradual one switches states step by step, and it uses only the latest value of CBR measurement. However, the Fully Meshed implementation can change rules by skipping one or more (for instance, it can shift Active 1 to Active 3 immediately). Still, it uses more

(a) States of Gradual DCC			(b) States of Fully Meshed DCC		
State	CBR	T_{off} (ms)	State	CBR	T_{off} (ms)
Relaxed	<0.3	50	Relaxed	<0.3	60
Active 1	0.3 - 0.39	100	Active 1	0.3 - 0.39	100
Active 2	0.4 - 0.49	200	Active 2	0.4 - 0.49	180
Active 3	0.5 - 0.65	250	Active 3	0.5 - 0.65	260
Restrictive	0.65 <	1000	Active 4	0.44 - 0.51	340
			Active 5	0.52 - 0.59	420
			Restrictive	0.59 <	460

Table 1: DCC FSMs

of the recent measurements to determine the proper state. According to C2C-CC BSP, the chosen implementation for deployments is the Gradual FSM.

2.2. CA and CP Services

Cooperative Awareness (Day 1) and Collective Perception (Day 2) services are periodic type ones; they send messages periodically with a rate between 1 and 10 Hz. Furthermore, these services send messages with significant packet size, so they mean relatively high stress to the used channel. CA Service is used to send information about the originating ITS station like position, heading, speed, or recent position points of the traveled path (path history). The latter is responsible for the highest amount of the allocated message size. Using CP Service, the ITS stations can inform each other about the perceived objects and free spaces that can be allocated by other vehicles. Thus, in general, the messages generated by this service are bigger than CA Messages (CAMs). Still, the message size is highly dependable on the number of perceived objects and the used area containers to encode free space information. For example, there are particular kinds of free space area shapes like circle, rectangle, or radial, which can be used to describe areas in simple ways with a low amount of consumed data. Still, there is the opportunity to use polygon type to cover complex areas, presumably with more bytes of data. CA and CP Services use position, heading, and speed change values to determine whether to generate a message or include a perceived object. In our investigations, we used three kinds of CP Message (CPM) content generation techniques:

- Objects and free space radials upon sensor detection: every perceived object included and to each one, a free space radial was assigned to cover available free space with a variable number of small shapes.
- Objects upon sensor detection with only one free space radial: every perceived object included, and each CPM contains a free space polygon with 16 vertices representing constant, but slightly high data usage.

- Every CPM is 1500 bytes large, which is the MTU size determined by 802.11 standards representing a kind of upper bound of penetration by CP Service in our simulations.

Through the investigations both CA and CP services applied the same generation rules, i.e., with the restrictions of DCC T_{off} (which can raise the applicable minimum generation rate) and the periodicity rules (messages shall be generated with a rate between 1 and 10 Hz), a message shall be created when the originating ITS station's position changes with 4 m, heading changes with 4° and speed changes with 0.5 m/s.

3. Simulation Environment and scenarios

3.1. Simulation Environment

The simulation environment we used is the Artery Framework [14] [13], where advanced simulations can be performed using ETSI ITS-G5 based protocols (e.g., GeoNet, BTP, etc.). Originally Artery was an extension of the Veins framework [16] [15] (which is for simulating WAVE, the US standards of V2X communication), but nowadays, it can also be used without Veins. The OMNeT++, the executor of the simulation models, is a C++ based discrete event simulator package [18] [17] interacting with the SUMO road traffic simulator [20] [19] via TraCI interface [21] using TCP socket.

To reach our goals, we performed several extensions, modifications and configurations in the existing Artery framework. Thanks to the Artery's well-designed architecture and the `asn1c` compiler [22], it was quite straightforward to extend the available message set using the ASN.1 description of CPM [5]. Based on that, a simple but rock solid CP service was created. To equip each vehicle appearing in the simulation scenarios with the required services, we needed to record those services in an XML file labeling them with a name and a listener (BTP) port. To mount radars simulating sensor detection, we also needed to configure them in XML files and to define the path of them in the `omnetpp.ini`, where other parameters of the scenario, like the field-of-view range of the radars, were also configured.

3.2. Simulation Scenarios

For simulations, we used the built-in `car2car-grid` scenario of Artery with our modifications. Each scenario was running for 22 simulation seconds, and the number of the vehicles followed a linear growing pattern from 0 to 90 during the simulation time. The deployment of the cars on the grid happened in pseudo-random positions but in a fixed way determined by an XML file so that the differences between each scenario shall be independent of this factor.

Each vehicle has two radars that detect in forward and backward directions to perceive other vehicles in the line of sight. Each sensor was configured to detect

in a radius of 150 meters and field-of-view of 180 degrees thus both sensors together detected the area in 360 degrees Every further measurement presented in this document was performed on the same vehicle which was deployed in $t = 0$ and present in during the whole simulation in every examined scenario. Since the standardization of Collective Perception began in the near past (ETSI TR [5] was just released in December 2019), the implementation of the service was necessary on our side. As a simplified generation rule of CP service, we utilized the one of CA service, which in our opinion is acceptable in the scenarios because every car followed the same behavior, thus in the meantime, they were going straight at the same speed. Furthermore, during our investigations, the standards describing the generation rule of CP messages were formed rapidly, and it was out of scope in our current analysis to follow that change. As we mentioned before, we used three types of message content generation techniques, and two of them result in variable message sizes that are in relation to the number of perceived objects, which is shown in Fig 1 (left) in the function of the simulation time as a reference. This kind of detection results in CPM sizes shown in Fig 1 (right) in function of the simulation time. According to this diagram, the size advantage of using radials against one polygon lasts until five simulation seconds in the applied scenario; after that, the overhead gets too big.

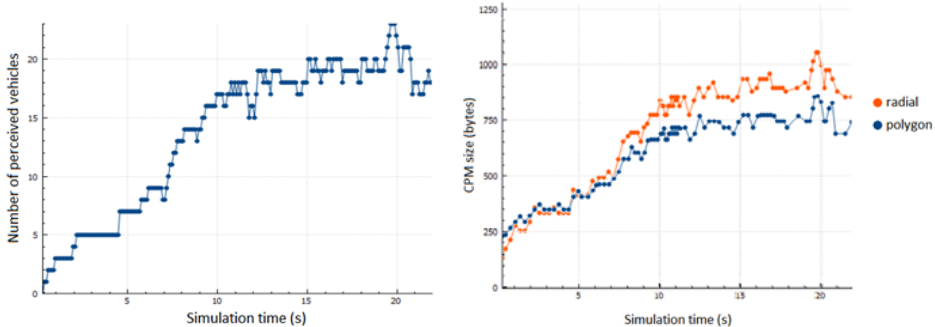


Figure 1: Number of perceived vehicles (left) and CPM sizes (right)

Several other parameters were used to define further elaborated simulation scenarios based on the above general details:

- Services running on each vehicle: 1) Only CA service (calibration measurement), 2) Only CP service, 3) Both CA and CP services.
- DCC-profile of each service: 1) Both CA and CP are on DP2, 2) CA is on DP2, CP is on DP3.
- DCC Finite State Machine: 1) Fully Meshed, 2) Gradual.
- Structure of CPM: 1) One free space radial for each perceived object, 2) One free space polygon with 16 vertices in each CPM, 3) Maximum size messages (1500 bytes)

Sizes of generated CAMs are between 148 and 350 bytes depending on that a packet contains a LowFrequencyContainer with path points or not. The following key performance indicators were taken into consideration in each scenario to analyze simulation results:

- Instantaneous CBR: measured on access layer level; to describe the channel load in relation to simulation time.
- number of dropped packages in each DCC queue: to describe packet loss caused by DCC restriction.
- number of sent packages on the radio: to describe the number of packets that got on the air interface, which is in connection with the channel load caused by the node where the measurements were made.
- number of received erroneous packets: measured on access layer level before any packet was transmitted to upper layers; to describe packet loss caused by channel degradation.

4. Results and analysis

Based on the above-introduced simulation environment and scenarios, we performed multiple simulation runs and examinations. We present the gathered results in subsections below grouped by the applied services.

4.1. Only CA service

These cases are for calibration measurements. In scenarios applying only CA service, just two cases can be distinct: using Fully Meshed DCC and using Gradual DCC. However, the results showed marginal differences to each other: the maximal CBR value did not reach 0.2, and no packets were dropped or received as erroneous.

4.2. Only CP service

The CBR values resulted from using Fully Meshed or Gradual DCC are depicted in Fig 2 (legend: FM: Fully Meshed, G: Gradual). The diagrams show that bigger packet sizes resulted in non-marginally higher CBR, which is conspicuous in higher traffic. Furthermore, Fully Meshed implementation led to lower CBR in general, mainly when maximum-sized packets were used. Using Gradual DCC, more packets were sent on radio, and fewer packets were dropped in DCC.

4.3. Both CA and CP service with different DCC-profiles

Both services were applied corresponding to C2C-CC BSP [8]: CA service on DP2, CP service on DP3. Looking at the diagrams in Fig 3, the differences in resulted CBR values are enormous between the Fully Meshed and Gradual scenarios. Fully

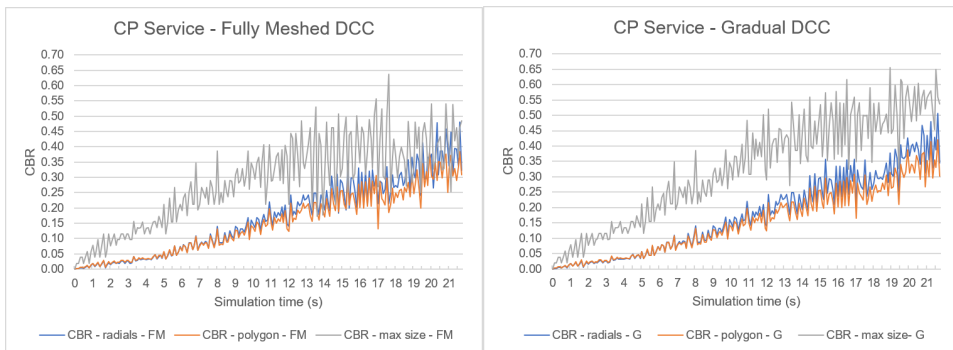


Figure 2: CBR values in case of CP service using Fully Meshed (left) and Gradual DCC (right)

	Fully Meshed			Gradual		
	Radials	Polygon	Max size	Radials	Polygon	Max size
Generated CPMs	90	89	79	93	93	89
Dropped in DCC	1	0	3	0	0	1
Sent on radio	89	89	75	93	93	88
Received erroneous	7	14	29	5	6	24

Table 2: Only CP service

Meshed DCC restricted CBR radically while using Gradual DCC resulted in high fluctuation in the congested part of the scenarios, which is in connection with the nature of Gradual DCC's operation because it does not use more recent CBR measurement values, only the most recent one. Table 3 shows that Fully Meshed DCC dropped more and sent fewer packets than Gradual DCC, but using the latter one resulted in more erroneous packets, especially when maximum-sized packets were used. Furthermore, almost every packet was dropped from the DP3 DCC queue; thus, we can say that CP service is handicapped by CA service.

4.4. Both CA and CP service with the same DCC-profile

These scenarios are for presenting the situation when messages of two services with similar generation rules are sent out on the radio interface with equal chances. The diagrams of Fig 4 show the increased CBR values resulted from more actu-

	Fully Meshed			Gradual		
	Radials	Polygon	Max size	Radials	Polygon	Max size
Generated CAMs	97	97	88	99	100	95
Generated CPMs	89	89	82	91	92	87
Dropped in DCC-DP2 (CAM)	0	0	2	0	0	0
Dropped in DCC-DP3 (CPM)	23	20	45	7	6	19
Sent on radio	160	164	120	182	185	161
Received erroneous	18	19	28	51	57	122

Table 3: CA and CP services with different DCC profiles

	Fully Meshed			Gradual		
	Radials	Polygon	Max size	Radials	Polygon	Max size
Generated CAMs	96	97	89	98	99	94
Generated CPMs	89	89	82	90	91	87
Dropped in DCC	31	22	51	8	7	29
Sent on radio	152	162	118	180	183	150
Received erroneous	27	24	29	93	58	213

Table 4: CA and CP services with same DCC profiles

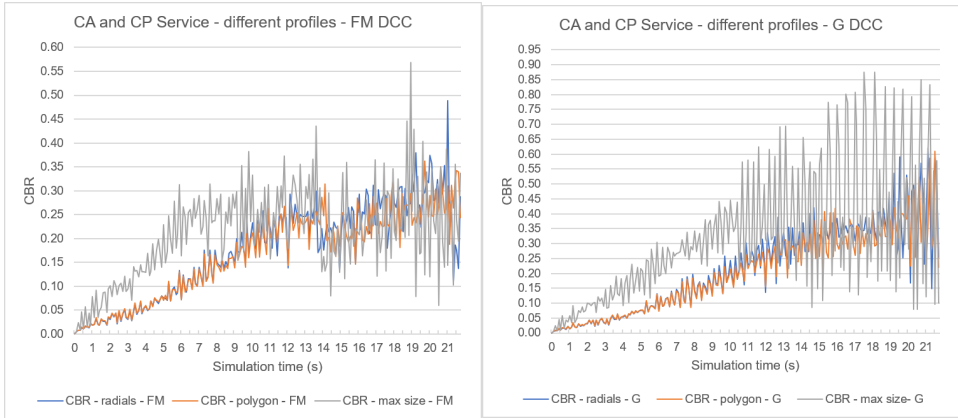


Figure 3: CBR values in case of CA and CP services with different DCC-profiles using Fully Meshed (left) and Gradual DCC (right)

ally transmitted CPMs – which are more significant than CAMs in general. In Gradual cases, the fluctuation of CBR is higher than in cases with different DCC profiles, and the number of erroneous packets is more prominent, too. Based on our experiences, we can say that Gradual DCC implementation with more transmitted packets consequences much higher amounts of received erroneous packets than Fully Meshed implementation because the latter one holds packets back more drastically, thus in case of full DCC queues drops more packets, resulting in lower CBR values.

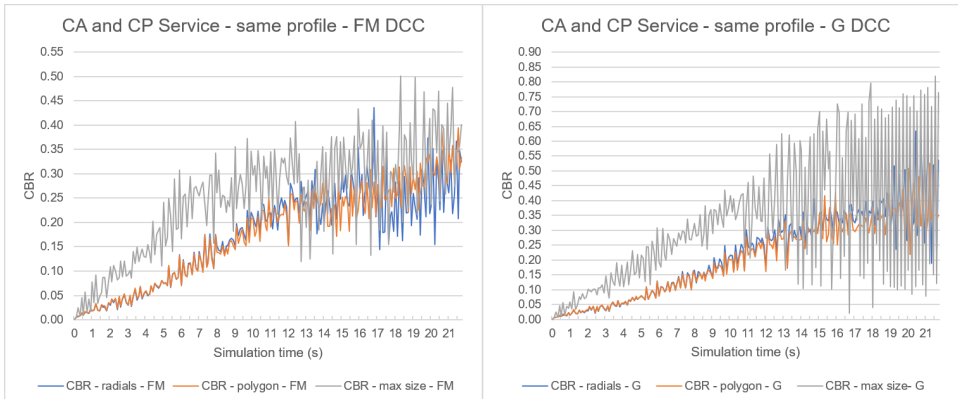


Figure 4: CBR values in case of CA and CP services with same DCC-profiles using Fully Meshed (left) and Gradual DCC (right)

5. Conclusions

Our simulation results are focused on the Channel Busy Ratio (CBR) and showed that the quality of the provided CP service degraded significantly when the two services were used simultaneously on the same channel, therefore motivating deployment efforts of the more sophisticated multichannel V2X solutions. Data-intensive applications should use distinct channels, and each channel – if it is feasible – may have its own DCC.

Developing a version of CP service more accurate to its generation rules declared in its published TR and using MCO implementation features of the newest version of Artery could show further exciting and useful information.

Acknowledgements The authors would like to thank the experts from Commignia Ltd. and the Department of Networked Systems and Services of BME for their valuable comments and guiding discussions.

References

- [1] The Cooperative Vehicle-Infrastructure Systems Project <http://www.cvisproject.org/>
- [2] The Drive-C2X Project <http://www.drive-c2x.eu/>
- [3] The SimTD Project <http://www.simtd.de/>
- [4] K. GARLICH, M. WEGNER AND L. C. WOLF, Realizing Collective Perception in the Artery Simulation Framework, 2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan, 2018, pp. 1-4, doi: 10.1109/VNC.2018.8628412.
- [5] ETSI TR 103 562 V2.1.1 (2019-12) Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2, 2019.
- [6] IEEE, 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 Standard 802.11-2016, December 2016.
- [7] ETSI EN 302 571 - 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 - v2.1.1. Feb-2017.
- [8] CAR 2 CAR Communication Consortium, “Basic System Profile - v1.4.0.” 13-Sep-2019., <https://www.car-2-car.org/documents/basic-system-profile/>
- [9] ETSI EN 302 637-3 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service - v1.3.1.” Apr-2019.
- [10] ETSI EN 302 637-2 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service - v1.4.1.” Apr-2019.

- [11] ETSI TS 102 687 - Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part - v1.2.1. Apr-2018.
- [12] ETSI TS 103 175 - Intelligent Transport Systems (ITS); Cross Layer DCC Management Entity for operation in the ITS G5A and ITS G5B medium - v1.1.1. Jun-2015.
- [13] GitHub - rieb1/artery: OMNeT++ V2X simulation framework for ETSI ITS-G5 [Online]. Available: <https://github.com/rieb1/artery>, Last accessed on 01/2020.
- [14] RIEBL R., OBERMAIER C., GÜNTHER HJ. (2019) Artery: Large Scale Simulation Environment for ITS Applications. In: Viridis A., Kirsche M. (eds) Recent Advances in Network Simulation. EAI/Springer Innovations in Communication and Computing. Springer, Cham
- [15] GitHub - sommer/veins [Online]. Available: <https://github.com/sommer/veins>, Last accessed on 01/2020.
- [16] SOMMER C. ET AL. (2019) Veins: The Open Source Vehicular Network Simulation Framework. In: Viridis A., Kirsche M. (eds) Recent Advances in Network Simulation. EAI/Springer Innovations in Communication and Computing. Springer, Cham
- [17] GitHub - omnetpp/omnetpp [Online]. Available: <https://github.com/omnetpp/omnetpp>, Last accessed on 01/2020.
- [18] ANDRÁS VARGA AND RUDOLF HORNIC, 2008. An overview of the OMNeT++ simulation environment. In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (Simu-tools '08). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, Article 60, 1–10.
- [19] GitHub - eclipse/sumo [Online]. Available: <https://github.com/eclipse/sumo>, Last accessed on 01/2020.
- [20] PABLO ALVAREZ LOPEZ, MICHAEL BEHRISCH, LAURA BIEKER-WALZ, JAKOB ERDMANN, YUN-PANG FLÖTTERÖD, ROBERT HILBRICH, LEONHARD LÜCKEN, JOHANNES RUMMEL, PETER WAGNER, AND EVAMARIE WIESSNER, Microscopic Traffic Simulation using SUMO, IEEE Intelligent Transportation Systems Conference (ITSC), 2018.
- [21] TraCI - SUMO Documentation [Online]. Available: <https://sumo.dlr.de/docs/TraCI.html>, Last accessed on 01/2020.
- [22] GitHub - vlm/asn1c: The ASN.1 Compiler [Online]. Available: <https://github.com/vlm/asn1c>, Last accessed on 01/2020.