CVIP: A Protocol for Complex Interactions Among Connected Vehicles

Bernhard Häfner,[1,3] Josef Jiru,[2] Karsten Roscher,[2] Jörg Ott,[3] Georg A. Schmitt,[1] and Yagmur Sevilmis[2]
[1]BMW Group Development, Munich, Germany
[2]Fraunhofer Institute for Cognitive Systems IKS, Munich, Germany
[3]Chair of Connected Mobility, Technical University of Munich, Munich, Germany
Emails: {bernhard.haefner, georg.schmitt}@bmw.de, {firstname.lastname}@iks.fraunhofer.de, ott@in.tum.de

***Abstract*— Automated vehicles need to interact: to create mutual awareness and to coordinate maneuvers. How this interaction shall be achieved is still an open issue. Several new protocols are discussed for cooperative services such as changing lanes or overtaking, e.g., within the European Telecommunications Standards Institute (ETSI) and Society of Automotive Engineers (SAE). These communication protocols are, however, usually specific to individual maneuvers or based on implicit assumptions on other vehicles' intentions. To enable reuse and support extensibility towards future maneuvers, we propose CVIP, a protocol framework for complex vehicular interactions. CVIP supports explicitly negotiating maneuvers between the involved vehicles and allows monitoring maneuver progress via status updates. We present our design in detail and demonstrate via simulations that it enables complex inter-vehicle interactions in a flexible, efficient and robust manner. We also discuss open questions to be answered before complex interactions among automated vehicles can become a reality.**

I. Introduction

Automated vehicles will have to be connected. Besides internet connectivity for infotainment use cases, direct communication among traffic participants, Vehicle-to-Everything (V2X), is very close to becoming a reality. Even though this has been proclaimed for more than ten years, things have changed, recently. Standards regarding basic safety can be regarded as almost mature, and first Original Equipment Manufacturers (OEMs) start deploying basic V2X services in series production vehicles, despite not all questions having been answered, yet[1].

Attention has recently shifted from basic warning services to how vehicular communication can support cooperative and automated driving. Those so-called *Day 2* or *Day 3* services will change the way vehicles interact with each other [1], on urban and highway roads as well as on intersections [2], [3]. They use V2X to enable negotiations among vehicles and leverage distributed intelligence, further increasing safety and convenience. First standardization efforts of such advanced services have started in Europe [4] and the United States [5].

However, many challenges regarding complex interaction among vehicles are still unsolved, for example which is the best way to enable them. Recently, several protocols have been proposed, but all with certain shortcomings. Most often, protocols enable very specific maneuvers (e.g., lane changes), requiring adjustments for every new use case [6]. More importantly, a thorough investigation regarding communication parameters is missing for most protocol proposals.

This paper aims at mitigating some of the existing shortcomings. To this end, we present the generic Complex Vehicular Interactions Protocol (CVIP) for arbitrary complex interactions among Connected and Automated Vehicles (CAVs). Specifically, we describe the message flows and contents, ways to adjust for future use cases, and evaluate the proposal with regards to scalability and robustness. To the best of our knowledge, CVIP is the first solution that allows explicit joint maneuver negotiation, while still keeping general applicability. By including new fields in the known message set, future use cases can be easily realized.

II. Related Work

A lot of researchers started to look at complex interactions, recently. Burger *et al.* [7] defined a cooperative action to be *willingly and knowingly executed with the intention to work towards a common goal*, i.e., a joint optimum. They classify cooperation depending on exchanged information (*information-based*) and the optimization goal for a joint action (*maneuver-based*). In the highest stage, total utility is optimized by sharing state information, intentions, and individual utilities.

Cooperative behavior protocols can be divided into *implicit* and *explicit* ones. With implicit approaches, vehicles share intentions periodically and have to infer from the potentially changed intentions received from others, whether their proposal has been accepted or not [8]. In IMAGinE, every vehicle periodically broadcasts its *planned trajectory* as beacon. If a trajectory update is to be initiated, a *desired* trajectory is sent, additionally. Other vehicles evaluate a change of their own planned trajectory in order to make the desired trajectory possible. Based on received updated plans, the initiating vehicle can judge whether it can change its planned trajectory or not [9]. TransAID [10] adds the possibility of trajectory proposals by Road Side Units (RSUs) as traffic coordinators.

Conceptually different is explicit coordination. The idea is that a vehicle's desired maneuver has to be explicitly *committed to* or *acknowledged* from relevant actors to be sure the proposal has been accepted. In [6], a lane change protocol is suggested. A *Lane Change Request* is broadcast, and answered via a unicast *Lane Change Response*. Based

[1]Audi press release Jan. 2020: media.audiusa.com/en-us/releases/384

on the feedback, a suitable peer vehicle is selected who will make space for the initiator, announcing having finished this with a *Lane Change Prepared* message. Now the initiator changes lane without further communication. Another way of explicit cooperation is the space-time reservation procedure [11]: a vehicle sends a *request* for some static or moving lane-level road space. Other vehicles will evaluate this in terms of inferred cost and send a *commit* message if they accept it. Vehicles not sending a commit message are either unwilling to participate or not able to take part in the negotiation; their behaviors have to be predicted based on an uncooperative movement model. Considering all received commits, the initiating vehicle will determine whether it is safe to enter the reserved road space and if yes do so, without further communication. Franke *et al.* present a protocol where each vehicle announces possible maneuvers and associated costs, and an optimal subset of maneuvers is chosen for execution [12].

Up to now, all approaches have limitations. For implicit ones, vehicles have to guess whether other vehicles understood the own proposal or are just coincidentally changing their trajectory. This may yield very conservative CAVs. Moreover, periodically broadcasting trajectories results in heavy bandwidth usage, unnecessary if no maneuver is to be performed. How a generation rule for such periodic beacons should look like is another unsolved issue [10]. Current explicit approaches are either very application-specific or based on road geometries. The former is suboptimal since each new maneuver would require a new message set. The latter is more general, but can only account for rather simple reservations of road space for the initiating vehicle. A disadvantage common to all current explicit maneuver coordination approaches is that they support only a single initiator and feedback from others. Not supported are maneuvers where two or more participants jointly negotiate and perform certain actions. To tackle this problem, Correa *et al.* [10] propose using infrastructure. They suggest a *Maneuver Coordination Message* via which RSUs can advise vehicles to follow certain trajectories. However, because they build upon intention beaconing, it is not possible to perform a truly joint maneuver among several vehicles, since no mechanism assures that either all or none of the addressed vehicles takes the action suggested by the infrastructure.

CVIP tackles these issues by enabling explicit maneuver negotiation. It is not use case specific and allows designing complex maneuvers for arbitrary many participants.

Some authors investigated group formation for complex maneuvers [13], but several disadvantages are related to that, cf. [8]. Due to these issues, we will, similar to [11], determine participants via the responses received from others.

III. Complex Interactions

Based on the above definition of cooperation, we will look at *complex* interactions. Those we define as message exchanges between *two or more* actors with *at least three messages* of which *at least one depends on another*. The rationale for this definition is as follows: clearly, a single vehicle cannot interact. Furthermore, even if simple interactions might exchange less than three messages, most cooperations should involve at least a *request/proposal*, a *response/acceptance*, and a *decision*. The dependency requirement reflects that an actual *inter*action needs to happen. Later messages must differ in content or addressee if something in the earlier chain of messages changes. Exchange of periodic beacons broadcasted independent of the received inputs, e.g., about the sender state or perceived objects, are not complex interactions, since they do not depend on other actors' actions or messages.

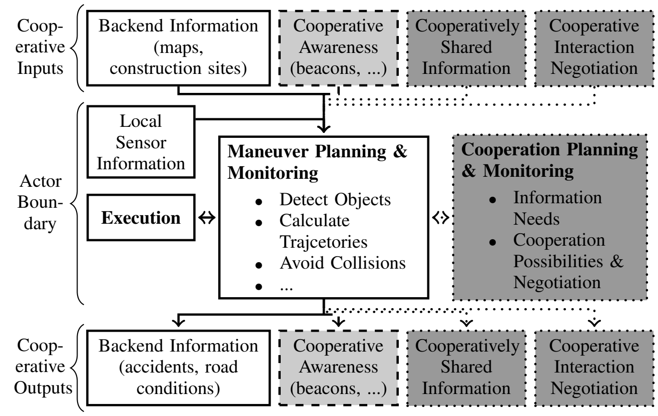

Fig. 1. A model for complex interactions showing an actor together with inputs and outputs in different stages of cooperation (Day 0-3). Solid/white elements depict entities present already in the co-existence phase, dashed/light gray elements enable cooperative awareness, while dotted/dark gray ones are needed for a fully cooperative environment.

This definition includes different applications like maneuver cooperation or information sharing, and is mostly concerned about the underlying protocol. The thinking is that in a fully-connected ecosystem, it may not be sensible to distinguish too many types of interactions, but rather generalize protocols towards applicability across a wide range of scenarios, as long as induced overhead allows. For example, a vehicle may need further information in order to start a cooperation proposal. It therefore may request information first (e.g., on objects perceived by the front vehicle), and then start a subsequent maneuver negotiation (e.g., about an overtake). This has another advantage that can be understood via Figure 1, showing a model for the functional split of interactions and the transition from co-existence to cooperation: especially the *Cooperation Logic* block will be OEM-specific, meaning the ego vehicle cannot rely on other vehicles taking decisions according to its own expectations. Here, explicitly sharing maneuver intentions and information needs makes it easier for other actors to process and decide on cooperation.

IV. Protocol Design

A. Assumptions and Constraints

In the vehicular context, we consider Cooperative Awareness Message (CAM) or Basic Safety Message (BSM)

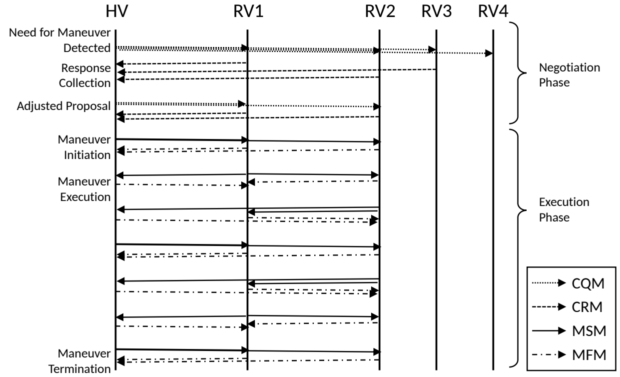

Fig. 2. An exemplary message flow between a Host Vehicle (HV) and several Remote Vehicles (RVs).

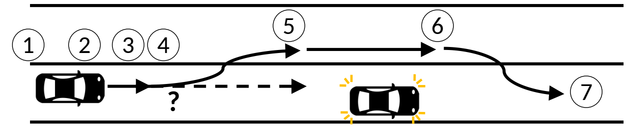

Fig. 3. Use case illustration for Stationary Vehicle Decision Assist. The numbers in circles are referenced in the text.

beacons as given. They are utilized to identify potential maneuver partners in an *Awareness Phase*, before the actual interaction involving a *Negotiation* and an *Execution Phase* starts. Additional information needed should be exchanged via dedicated messages. We only consider joint maneuvers performed by automated vehicles. They are able to share information about intentions, unlike manually-driven vehicles that can only measure their current status.

Vehicles have to be able to estimate others' dynamics to make reasonable maneuver proposals. In case a proposed maneuver is not possible for another vehicle, it can respond accordingly. As it is undecidable on protocol layer whether the content of a received message, e.g., a maneuver proposal, is realistic and feasible, higher layers have to evaluate incoming proposals, namely the "Cooperation Planning & Monitoring" unit in Figure 1. Regarding security, we assume that messages are signed and interactions are thus secured. This will potentially increase message sizes compared to the ones analysed in this paper. Security of the communication as well as of the interaction itself is important, but since ways like credentials exist that take care of preventing certain malicious actions [14], we defer a more detailed security analysis to future work.

B. Design Guidelines

The guiding principles of the Complex Vehicular Interactions Protocol (CVIP) are as follows: At first, we consider necessary a suitable amount of acknowledgements, since actors need to know whether others have heard, understood, and agreed to the proposed interaction. In our design, the execution status of a specific action can only be changed by the actor performing this action, in order to avoid inconsistencies. Secondly, even if some entity needs to express the initial request for a maneuver, the decisions on participation can only come from the actors themselves in a distributed way. Actors have to be able to abort maneuvers at any given point in time, for example if own safety is at risk. In general, we think cascading of requests (cf. [8]) is not to be encouraged, since this would considerably increase delays before a maneuver can be executed [11]. Cascading means that a request of vehicle A to vehicle B induces further requests from vehicle B to others in order to be able to accept vehicle A's request.

C. Protocol Flow

Based on these guidelines, we designed CVIP involving four types of messages: *Cooperative Request Message (CQM)*, *Cooperative Response Message (CRM)*, *Maneuver Status Message (MSM)*, and *Maneuver Feedback Message (MFM)*, as depicted in Figure 2. For brevity, we discuss the contents in an abstract manner first, and then show how this applies to a sample scenario involving a complex interaction named Stationary Vehicle Decision Assist (SVDA). Concrete sizes for message elements as described in the following can be found in Table I. SVDA works as follows, cf. Figure 3:

A stationary vehicle is located in front of a driving CAV. After becoming aware of the situation, in order to evaluate whether it is worth the risk of overtaking, the on-coming vehicle inquires about the estimated duration of stay of the stationary vehicle (① in Figure 3). If the duration is below a threshold or not known, the on-coming vehicle proposes to overtake, while the stationary vehicle should stay stationary (②). Both vehicles agree and the overtake is executed.

In general, after determining a need for information or a joint maneuver, some vehicle issues a CQM to potential partners. Its content can be depicted as tuple

$$\text{CQM} \triangleq \left(G,\, S,\, \mathcal{I}^Q,\, \mathcal{M}\right).$$

G contains basic information: protocol version, message ID i_{msg}, the initiating vehicle's station ID i_s, generation time stamp t_{gen}, and a sequence number n_{seq}. S contains basic status information, mainly the ego GPS position for reference and an Instance ID. Those can be used for referencing in subsequent messages. $\mathcal{I}^Q = (I_1^Q, ..., I_k^Q)$ and $\mathcal{M} = (M_1, ..., M_l)$ are containers for information requests and maneuvers, respectively, with $k + l \geq 1$. Each I^Q contains an Information Request Container ID $i_{\text{iqc}} = 1, ..., k$ for referencing, a destination station (vehicle) ID i_{dest} that should provide the information, the type of requested information T^Q as well as the update interval θ in which the information is requested. In SVDA, this could be the estimated duration in the current mobility state (i.e., with velocity zero) with $\theta = -1$ for one-time information. The elements M are Maneuver Containers describing the foreseen joint actions, each containing a container ID $i_{\text{mc}} = 1, ..., l$, a destination station ID i_{dest} that should perform the maneuver, the maneuver type T^m and related parameters P^m. By setting i_{dest} appropriately, even higher-authority parties like RSUs or emergency vehicles could propose maneuvers in which they themselves do not participate actively. Via T^m and

TABLE I
EXEMPLARY SIZE RANGES FOR THE PROTOCOL CONSTITUENTS MENTIONED IN THE TEXT.

Element	G	S	I^Q				I^R	M						
Sub-element			i_{iqc}	i_{dest}	θ	T^Q	V	i_{mc}	i_{pkt}	t_{start}	τ	T^m	S^m	P^m
Size [B]	16	46	4	4	4	4	1-100	4	4	4	4	4	4	1-500

P^m, maneuvers could be described as standardized names, parametrized functions, or also via trajectories. Those proposed maneuvers directly reflect assumptions on the physical capabilities and vehicle dynamics of other actors. A start time t_{start} – absolute or relative to another maneuver container – as well as the expected maneuver duration τ can also be included, if necessary. In the SVDA example, $\mathcal{M}$ could comprise one container per vehicle, stating T^m for the initiating vehicle as *overtake*, and for the stationary vehicle as *remain in current mobility state*, both with $t_{\text{start}} = 0$ and τ equal to the expected duration of the maneuver. Pending further investigation, also a more fine-grained statement of T^m's may make sense, e.g., having three containers for the ego vehicle – *acceleration*, *lane change*, *drive straight*, *lane change*, along with relative start times to ensure a common understanding of the order of execution.

After reception of the CQM, other vehicles evaluate the included information in their Cooperation Logic. They respond with a CRM defined as

$$\text{CRM} \triangleq \left(G,\, S,\, \mathcal{I}^R,\, \mathcal{M}\right).$$

While G and S contain the same types of sub-elements as in the CQM (i.e., updated t_{gen}, etc.), $\mathcal{I}^R = \left(I_1^R, \ldots, I_k^R\right)$ are now Information Response Containers containing a referenced i_{iqc}, along with requested values V or an error. The vehicle can also state when a given T^Q was not understood or is not available. The $\mathcal{M}$ are maneuver containers including a packet ID i_{pkt}, based on i_s and n_{seq} of the original CQM for stating references. Besides, a maneuver status S^m set to `Planned` or a respective error status is contained. If needed, updated values for t_{start}, τ or even T^m can be given. The combination of request and received responses gives a clear picture to the initiator on the others' willingness to participate: if no changes were proposed, all involved vehicles are willing to participate.

This iteration of CQM and CRM will be repeated until no changes are proposed and no errors are sent any more. This is the sign for convergence and every vehicle can be sure that all others also have agreed to a maneuver. As depicted in Figure 2, CAVs not implementing the protocol will not send a CRM (RV4). Vehicles that implement the protocol, but that either do not implement some of the requested T^Q or T^m, or that cannot fulfill requirements from the CQM, will send a response stating this (RV3). The initiating vehicle then updates the proposal according to the feedback and sends a new CQM involving only willing, capable, and necessary vehicles.

After convergence, the initiating vehicle will send an MSM

$$\text{MSM} \triangleq (G, S, \mathcal{M}),$$

with the agreed maneuvers in $\mathcal{M}$, together with a maneuver status S^m as `Planned` for each container in $\mathcal{M}$ (③). Other possible status values are `InProgress`, `Finished`, and `Cancelled`. Via those status, all vehicles will always know the execution status of each participating actor. Upon reception of this first MSM, every vehicle has to store the current status of all l maneuvers internally in order to keep track of the other actors' execution and to know when to trigger own ones.

In order to inform the sender of the MSM that all participating vehicles have received the MSM and to make sure all participators' internal state directories are synchronized, an MFM is sent as acknowledgement, which can be described by

$$\text{MFM} \triangleq (G, S, \mathcal{M})$$

and repeats the content of the MSM just received. While this consumes a considerable amount of the overall necessary bandwidth, the authors think that it provides an efficient mechanism to prevent diverging internal execution states across actors, for example due to messages not received. When diverging states for maneuver container $M_{l'}$ are detected from the received MSMs and MFMs, the vehicle executing $M_{l'}$ can send a clarifying MSM containing the currently correct execution state.

Whenever subsequently an actor changes its maneuver execution state – the sequence is known via a consistent use of the t_{start} and τ – it will send an MSM with the updated status to let every other vehicle know that it entered a new state (e.g., at ④/⑥ where *Lane Change* switches to `inProgress`, or at ⑤/⑦ where it is `Finished`). Once all maneuvers' status has been set to `Finished`, the joint maneuver is completed.

As the reception of messages is essential for synchronized state transitions among the vehicles, a resend mechanism based on a timeout has been implemented. If a CQM is dropped and thus no CRMs are received, the initiator resends its CQM after $t_{\text{wait}}^{\text{cqm}}$, up to c^{cqm} times. If a CRM is dropped, then the vehicle has to be regarded uncooperative. In case an MSM is dropped, no MFMs are received at all and the message is thus resent after $t_{\text{wait}}^{\text{msm}}$, at most c^{msm} times. In case of missing MFMs, the respective MSM is also resent after a timeout of $t_{\text{wait}}^{\text{msm}}$, in order to ensure synchronized state updates. If MFMs are missing after c^{msm} retries, the maneuver will be cancelled. The timeouts and maximum resends may be adjusted for example based on different application scenarios, driving conditions, or traffic with surrounding vehicles potentially shadowing signals.

Since the requirements for different use cases may differ substantially, specific message contents (such as the number and type of containers, which information is transmitted, etc.) can be adjusted according to respective needs. This gives CVIP the flexibility to support different scenarios without the need to define specialized messages. New scenarios are enabled easily by augmenting the set of defined values for T^Q, T^m, P^m or adding new fields. The information in Table I and the analysis in Section V show that message sizes currently do not exceed a few hundred Bytes. Thus, today's vehicular communication technologies like ITS-G5 or LTE-V2X can easily support this extensibility.

V. Numerical Experiments and Evaluation

In order to show the general applicability of CVIP, we abstracted the evaluation setting from specific applications. Instead, we will answer the following questions via a set of simulations of the protocol flow: Is the protocol

- *scalable* with respect to the number of nodes participating in a complex interaction?
- *feasible*, i.e., is the message size constrained within sensible limits, even for high k, l?
- *robust*, i.e., is it able to cope with lost packets?

To this end, we set up a simple simulation scenario: a set of N static vehicle nodes is placed on a straight lane in the simulation area. An initiating node sends the first CQM to all other nodes. As we do not want to investigate the details of the logic deciding on incoming cooperation requests, all vehicles send a CRM with positive feedback. In reality, the higher-level cooperation logic would have to evaluate incoming CQMs' feasibility. Then, a simple maneuver is performed as described in Algorithm 1: one node after the other starts their maneuver of duration τ. With this setup, we evaluate scalability regarding N and l, as well as robustness by inserting packet losses with probability p_{drop}.

All our simulations have been performed using the Intelligent Transportation System (ITS) framework ezCar2X[2] for connected applications [15] in combination with the simulators SUMO[3] and ns-3[4]. For significant results, we performed 100 simulation runs for each parameter set described later.

The number of messages exchanged in one interaction can be formalized depending on the number of nodes N, of negotiation rounds ν, and of maneuver containers l as

$$n_{\text{msgs}}^{\text{cvip}} = \nu + \nu \cdot (N-1) + 2 \cdot l + 2 \cdot (N-1) \cdot l. \quad (1)$$

The factor 2 is because an MSM with status `inProgress` and `Finished` will be sent for each of the l maneuver containers, respectively, and each of them will be confirmed via MFMs by the $N-1$ other actors. It can be easily seen that $n_{\text{msgs}}^{\text{cvip}} = \mathcal{O}\left(N \cdot (\nu + 2l)\right)$.

In contrast, the number of messages transmitted with frequency f in beaconing-based intention sharing,

$$n_{\text{msgs}}^{\text{beac}} = f \cdot \Theta \cdot N, \quad (2)$$

is dependent on the overall maneuver execution time Θ. For a reasonable set of values, e.g., $N = 10$, $\nu = 1$, $l = 20$, $f = 5Hz$, and $\Theta = 10s$, CVIP reduces the number of exchanged messages by almost 20%. This reduction is all the more significant as beacons are foreseen to be used even in times no maneuver is executed, as the whole concept of implicit maneuver negotiation is based on this periodic broadcast of intentions. With CVIP, messages are only exchanged if there is a need to. For a pure information exchange (i.e., $l = 0$, $k > 0$, $\nu \geq 1$), only very few messages will be sent, exactly satisfying information needs of the requesting vehicle. The size of the involved messages increases only linearly with the numbers k and l of included containers and, as Table I shows, the size of each I^Q, I^R, and M is relatively small. Via serialization techniques the overall sent message size can be further reduced. For example, for MFMs with $l = 7$, the sent message size was on average 137 Byte, while the message size within the source code was 203 Bytes.

[2]ezcar2x.fraunhofer.de
[3]sumo.dlr.de
[4]nsnam.org

Algorithm 1 Sample application on node j.

Input: MSM from node $j-1$ with $S^m =$ `inProgress`

1: Send MFM, then wait for t_{start}
2: Send MSM, setting M_j's S^m to `inProgress`
3: Wait for τ
4: Send MSM, setting M_j's S^m to `Finished`

In a simulation, even for $l = 7$, the overall message size did never exceed 225 Byte, transmitting all essential elements of a maneuver container. Even for today's technologies like ITS-G5 or LTE-V2X, this makes our protocol feasible. For future use cases, there is still space to include additional fields, e.g., in P^m, without message sizes becoming infeasible. This makes even more complex maneuvers possible.

By introducing packet losses with probability p_{drop}, we investigate how robust the protocol is towards transmission outages, even with our basic resend mechanism. For evaluation, we count maneuvers as successfully completed if the last MSM containing all maneuver status set to `Finished` was sent and received. The ratio of successful maneuvers to all maneuvers is called *success rate*. If a single CRM is dropped, the maneuver will not be successful, since the initiating node cannot find out that a CRM has been dropped. Depending on the application, the initiating vehicle may thus choose to send up to c^{cqm} CQMs in order to minimize the probability that CRMs get lost. As Figure 4 shows, this significantly improves the success rate, reaching 0.8 even for $p_{\text{drop}} = 0.2$ for $c^{\text{cqm}} = 2$.

Figure 5 shows for scenarios with $N = 3$ and one maneuver per vehicle, that the number of messages that need to be exchanged increases with p_{drop}, the base line case being $p_{\text{drop}} = 0$. This is expected, but even for $p_{\text{drop}} = 0.2$, not even twice as many messages will be sent. In general, CQMs are sent relatively more often than CRMs, since a vehicle has to resend them for a lost CQM as well as for a lost CRM. The same holds for comparing MSMs and MFMs.

VI. Discussion and Future Research

The preceding analysis shows that CVIP is feasible for complex interactions. However, further research is needed.

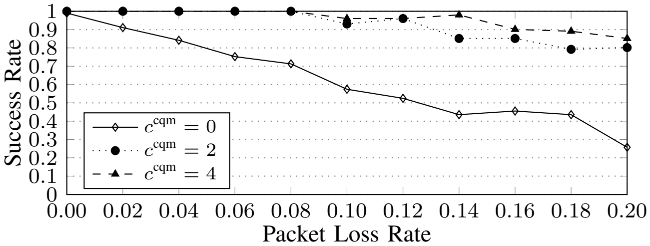

Fig. 4. Ratio of successfully completed maneuvers vs. packet loss rate.

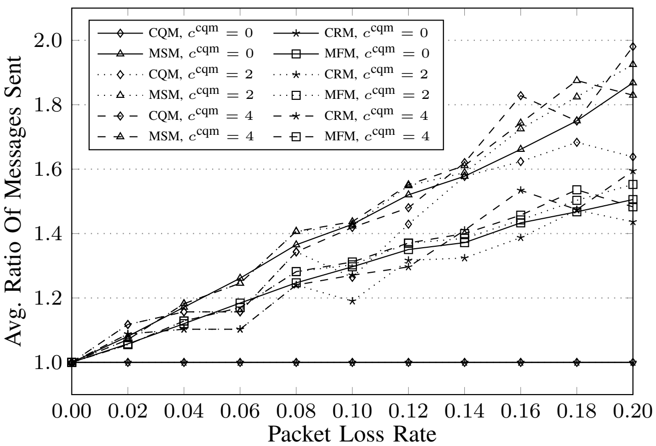

Fig. 5. Messages sent in successfully completed maneuvers vs. p_{drop}, for each message type compared to the base line case of $p_{\text{drop}} = 0$.

At first, the retransmission mechanism adopted in this paper is rather simple. Advanced mechanisms may be able to reduce network load and increase efficiency as well as success rates.

Secondly, our simulation showed that current technologies are sufficient to enable complex interactions. New communication technologies like 5G-V2X may support further use cases, e.g., by enabling unicast or bigger message sizes in order to include very data-intensive information V or maneuver parameters P^m. Vice-versa, investigating new use cases and application possibilities for the protocol could yield requirements for technological development.

Next, independent of the protocol, only very little work exists about the cooperation logic (cf. Figure 1). Finding general rules how to decide on cooperation willingness would allow an investigation of computational loads for such decisions and whether the time budgets required by concrete maneuvers will be violated or not. Related is the question on how to cope with the cooperation logic being OEM-specific.

Additionally, since many approaches for complex interactions exist, a thorough comparison is needed in order to determine which one best fulfills given and foreseen future requirements. Especially the trade-off between flexible and specialized, and explicit and implicit protocols has to be quantified.

Lastly, an investigation about security is still missing. In our and many other approaches, the cooperation logic together with scene understanding sensor fusion could yield some protection against malicious actors, but a thorough investigation of how to protect against disadvantageous or dangerous situations is needed.

VII. Conclusion

In this paper, we presented CVIP, a protocol for complex interactions among CAVs. After introducing message contents and flows, we conducted a thorough analysis of its scaling performance and robustness. While there still exists a need for further research, this protocol has the potential to enable vehicular information exchanges and maneuvers in a flexible and efficient manner, going beyond what current protocol proposals can achieve.

References

[1] K. Sjöberg, P. Andres, T. Buburuzan, and A. Brakemeier, "Cooperative Intelligent Transport Systems in Europe: Current Deployment Status and Outlook," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 89–97, 2017.

[2] L. Chen and C. Englund, "Cooperative Intersection Management: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.

[3] J. Rios-Torres and A. A. Malikopoulos, "A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[4] *Intelligent Transport Systems (ITS); Vehicular Communication; Informative Report for the Maneuver Coordination Service*, ETSI TR 103 578 V0.0.4 (2019-11), (Draft).

[5] *Application Protocol and Requirements for Maneuver Sharing and Coordinating Service*, SAE J3186 (Draft).

[6] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enhancements of V2X Communication in Support of Cooperative Autonomous Driving," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 64–70, 2015.

[7] C. Burger, P. F. Orzechowski, O. S. Tas, and C. Stiller, "Rating Cooperative Driving: A Scheme for Behavior Assessment," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1–6.

[8] B. Lehmann, H. J. Günther, and L. Wolf, "A Generic Approach Towards Maneuver Coordination for Automated Vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3333–3339.

[9] I. Llatser, T. Michalke, M. Dolgov, F. Wildschütte, and H. Fuchs, "Cooperative Automated Driving Use Cases for 5G V2X Communication," in *IEEE 5G World Forum (5GWF)*, 2019, pp. 120–125.

[10] A. Correa, R. Alms, J. Gozalvez, M. Sepulcre, M. Rondinone, R. Blokpoel, L. Lucken, and G. Thandavarayan, "Infrastructure Support for Cooperative Maneuvers in Connected and Automated Driving," in *IEEE Intelligent Vehicles Symposium*, 2019, pp. 20–25.

[11] D. Heß, R. Lattarulo, J. Pérez, T. Hesse, and F. Köster, "Negotiation of Cooperative Maneuvers for Automated Vehicles: Experimental Results," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 1545–1551.

[12] K. Franke, M. Düring, R. Balaghiasefi, M. Gonter, K. Lemmer, and F. Kücükay, "A Reference Architecture for CISS/CDAS within the Field of Cooperative Driving," in *IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2014, pp. 357–363.

[13] C. Frese, J. Beyerer, and P. Zimmer, "Cooperation of Cars and Formation of Cooperative Groups," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 227–232.

[14] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A Security Credential Management System for V2X Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3850–3871, 2018.

[15] K. Roscher, S. Bittl, A. A. Gonzalez, M. Myrtus, and J. Jiru, "ezCar2X: Rapid-Prototyping of Communication Technologies and Cooperative ITS Applications on Real Targets and Inside Simulation Environments," in *11th Conference Wireless Communication and Information*, 2014, pp. 51–62.