A Modular Multi-Formalism Methodology for Performance Evaluation of Vehicle Applications

Mattia Tibaldi
Politecnico di Milano
Milano, Italy
mattia.tibaldi@polimi.it

Marco Gribaudo
Politecnico di Milano
Milano, Italy
marco.gribaudo@polimi.it

***Abstract*—Vehicle-to-Everything (V2X) applications are distributed systems that integrate heterogeneous computational components, both on-board and off-board, via communication stacks based on different technologies. Analyzing the performance of these applications remains a complex task, because it requires defining intricate simulation scenarios that are costly to design and configure. In this work, we propose a performance methodology that aims to simplify the study of vehicle applications by abstracting their structure into five modules (i.e., Sensors, Fusion, V2X sources, Logic, and V2X Receivers). The modular approach, derived from multi-formalism modeling, enables the construction of incremental models without requiring substantial modifications to it. This feature significantly reduces design time and enables faster, more flexible evaluation of application configurations. Experiments conducted on an assisted Advanced Driver Assistance Systems (ADAS) application show the effectiveness of the methodology, enabling rapid assessment of end-to-end latency and characterization of safety limitations.**

***Keywords*—V2X, ADAS, Multi-Formalism, Performance**

I. Introduction

Modern vehicle systems increasingly rely on vehicle-to-everything (V2X) communication to enhance safety, improve situational awareness, and facilitate collaborative interactions [1]. These communications are integral to the perception and decision-making processes (i.e., application logic) within the vehicle, forming comprehensive processing sequences from sensors to actuation in which delays can significantly affect critical safety operations and the communication itself [2].

Actual state-of-the-art simulators incorporate sensor [3], mobility [4], and communication protocol simulators [5], [6] to assess the effectiveness of V2X systems [7], [8]. Despite their widespread adoption, these simulators are difficult for non-experts to configure, and they often lag behind emerging communication standards such as IEEE 802.11bd and 3GPP 5G-NR. This difficulty stems from their incremental construction: users must learn the configuration languages and paradigms of each underlying simulator before conducting a study. Moreover, the level of abstraction they offer is limited. Evaluating a V2X application typically requires implementing much of its logic, which slows the exploration of new ideas.

This study aims to address current limitations by introducing a modular, multi-formalism performance methodology that enables comprehensive vehicle-application evaluations [9]. The methodology utilizes stochastic Petri nets to capture complex structures and interactions, queues to efficiently evaluate the processing phases, and integrates them into a modular structure directly linked to the V2X processing pipeline. Figure 1 shows the modules modeled in this work and how they interact in building the application. In general, a vehicle application reads data from sensors and combines it with external information (i.e., everything-to-vehicle (X2V) sources) in a fusion step before using it in the application logic. The application logic controls the vehicle's actions and can potentially notify external entities of decisions, such as the pedestrians and the cloud (i.e., V2X receivers).

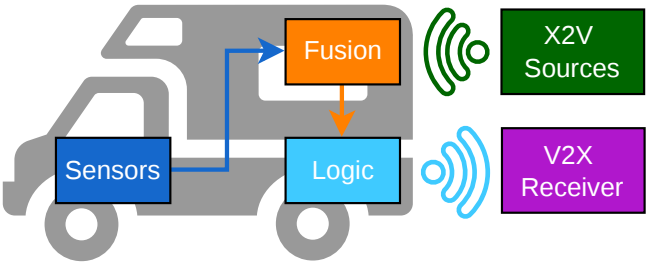


Fig. 1. A five modules in-vehicle application abstraction.

Furthermore, the multi-formalism paradigm allows us to build modular, parametric models that can be updated to support emerging technologies without extensive redesign, unlike purely analytical models, which typically require a full reformulation when system assumptions change [10]–[12].

We validate the proposed methodology in an assisted advanced driver assistance systems (ADAS) scenario, which offers practical, time-sensitive processing. In summary, this research makes the following contributions:

- **Data performance collection.** We collect detailed data on typical execution times for V2X processing pipeline, message generation rates, rates of hazardous road events, and the reliability of modern ADAS systems.
- **Vehicle application modeling.** We model the V2X application, encompassing sensing, fusion, logic, X2V sources, and V2X receivers, thereby providing a unified view of latency contributors across both computational and communication stages.
- **Testing on an assisted ADAS scenario.** We apply the model on a latency-critical assisted emergency braking use case, showing its applicability to realistic scenarios.

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

II. Proposed Methodology

We proposed a multi-formalist methodology comprising five modules as shown in Figure 1. Figure 2 shows how the proposed solution can be applied to a realistic case study (i.e., an assisted emergency braking system). We refer to this figure in the methodology description to make the narrative concise.

A. Sensor model

The sensor model mimics the behavior of an in-vehicle data source (e.g., radar, LiDAR, or camera) and the communication protocol used to transfer and process the data (the blue module in Figure 2). The source S_{source} generates message traffic with a certain frequency, while the place P_{reach} models the protocol's packet delivery ratio (PDR) by discarding some of the generated messages or forwarding others to the processing queue, thanks to the immediate transitions $t_{discard}$ and t_{reach}. The queue finally models the latency introduced by the type of protocol used (e.g., 802.11p vs. LTE-V2X). These three parameters, source frequency, latency, and PDR, allow the abstraction of the sensor while remaining faithful to its realistic behavior. By varying the frequency, we can mimic different sensors, while varying latency and PDR changes the type of protocol used and the situation being studied. In fact, since the PDR is correlated to vehicle density (i.e., the PDR decreases as density increases [13]), we can imagine urban rather than mountain scenarios simply by playing with this parameter.

B. X2V source model

The V2X model mimics the behavior of an roadside units (RSUs) along the route (the green modules in Figure 2). The model extends the sensor model with the reachability concept. The probability that the message reaches the destination is now conditioned not only by the PDR but also by the distance from the source, i.e., $P_{notAvail}$ and P_{Avail}. Two latencies model the vehicle's approach or departure from the source. The vehicle takes a time T_{up} to enter the source's range and a time T_{down} to exit it. T_{up} and T_{down} are determined by the vehicle's speed and the protocol under study. For example, LTE-V2X enables a stable connection to a source at longer ranges than 802.11p [13]. With this consideration, we can abstract the vehicle's mobility and the availability, A, or unavailability, $\overline{A}$, of the source is output to the block along with message M.

C. Fusion model

Sensor fusion combines multiple sensor data streams into a composite dataset that artificial intelligence algorithms can use to improve vehicle control. This step is among the most complex to simulate in state-of-the-art simulators because it requires synchronization among mobility, network, sensors, and application logic to be emulated correctly [14].

Our modeling combines two messages from different sensors, M_1 and M_2, into a single signal M_4 (the orange module in Figure 2). We use the Petri net extension with colored tokens to model a successfully combined data set that leads to better decisions (e.g., green token). At the same time, an unsuccessful combination leads to worse decisions (e.g., red token). What determines the two conditions is the availability or non-availability of the second source, i.e., the X2V source. T_{fusion} models the process execution time based on the number of fused sources. Finally, more than one fusion block can be instantiated in cascade to combine multiple sensors.

D. Logic model

The logic model is specific to each application. Figure 2 shows an example of an assisted emergency braking logic model and how colored tokens can be used in the process. Petri nets make it easy to abstract application logic, and changes to the model are immediate. This makes prototyping and testing vehicle applications faster and more accessible.

E. V2X receiver model

The V2X receiver model mimics the notification message to the pedestrian or infrastructure about the vehicle's decision. We model this behavior using a simple P_{reach}, which represents the probability that the message reaches its destination.

III. Evaluation

We tested the effectiveness of our methodology on an ADAS assisted emergency braking system with V2X notification. This safety-critical case study requires stringent response times (e.g., less than 100 ms or 300 ms [15]) and is therefore ideal for a latency study. The application acquires information from the onboard radar, the vehicle in front, and an RSU. So, we instantiate a sensor module and two X2V sources. Note that only the data collected by the radar and the vehicle are fused. In contrast, the data from the RSU arrives directly in the application logic. Finally, the logic notifies two V2X receivers of the decision: the pedestrian to warn it of a potential danger and the cloud for diagnostics. Figure 2 shows the resulting model. Table I collects the parameters used in the simulation, and we used Java Modeling Tools (JMT) [16] to solve it and obtain performance values.

A. Results discussion

We compare the performance of two network protocols, IEEE 802.11p and C-V2X.

Figure 3A shows that throughput increases almost linearly with the message generation rate, but is modulated by the distance from the infrastructure. At the same frequency, C-V2X guarantees higher useful throughput thanks to a higher PDR; in contrast, IEEE802.11p, despite having lower physical latency, suffers in contention contexts: sensing, backoff, and collisions reduce the number of messages actually received. In practice, lower "raw" latency does not imply better application performance when the PDR is low: the superior reliability of C-V2X compensates for the higher physical latency, providing more useful information to the control logic.

Figure 3B confirms the trend by showing the average queue length in the model: the MAC directly affects wait times and congestion. Random channel access in 802.11p increases wait times, with queues up to 45% longer than C-V2X under the same conditions. With the adopted parameters (1–6 Hz, service

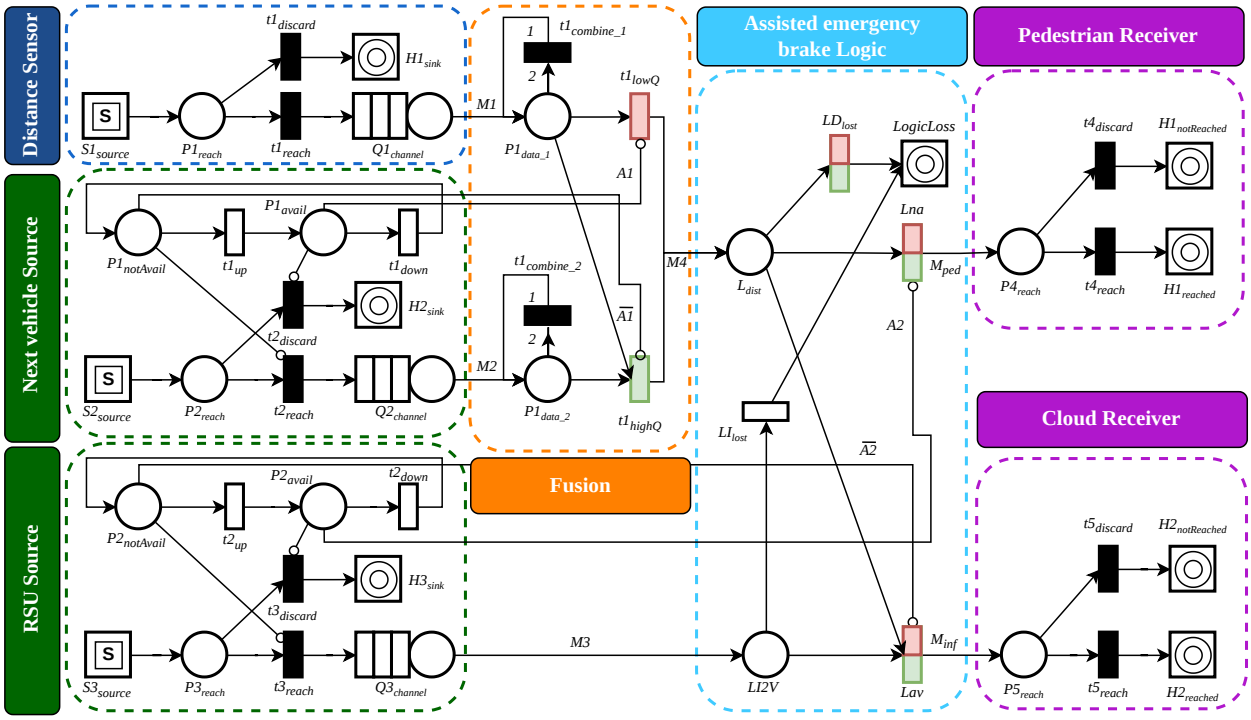


Fig. 2. The complete modular multi-formalism model of an assisted emergency braking application.

TABLE I
TYPICAL RANGES AND USED VALUES IN THE MODEL.

	Parameter	Description	Typical	Used	Ref.
	$S1_{source}$	Radar frequency	1–10 Hz	5 Hz	[17]
	$t1_{discard}$	Protocol PDR	0–1	0.4	[13]
	$t1_{reach}$	Protocol PDR	0–1	0.6	[13]
	$Q1_{channel}$	Proces. time	10.64–22.28 ms	16 ms	[15]
	$S2_{source}$	V2V frequency	1–10 Hz	5 Hz	[17]
	$S3_{source}$	I2V frequency	1–10 Hz	4 Hz	[17]
	$t2_{discard}$	Protocol PDR	0–1	0.4	[13]
	$t2_{reach}$	Protocol PDR	0–1	0.6	[13]
	$t3_{discard}$	Protocol PDR	0–1	0.4	[13]
	$t3_{reach}$	Protocol PDR	0–1	0.6	[13]
	$t1_{up}$	Time to enter vehicle	-	10 ms	-
	$t1_{down}$	Time to exit vehicle	-	10 ms	-
	$t2_{up}$	Time to enter RSU	-	10 ms	-
	$t2_{down}$	Time to exit RSU	-	10 ms	-
	$Q2_{channel}$	V2V proces. time	10.64–22.28 ms	16 ms	[15]
	$Q3_{channel}$	I2V proces. time	15.35–33.04 ms	23 ms	[15]
	$t1_{lowQ}$	LQ proces. time	24.56–89.36 ms	82 ms	[18]
	$t1_{highQ}$	HQ proces. time	77.72–161 ms	95 ms	[15]
	LD_{lost}	Loss from distance	-	0.2	-
	LI_{lost}	Loss from RSU	-	0.2	-
	Lna_{HQ}	HQ proces. time	-	0.05	-
	Lna_{LQ}	LQ proces. time	-	0.1	-
	Lav_{HQ}	HQ proces. time - I2V	-	0.02	-
	Lav_{LQ}	LQ proces. time - I2V	-	0.05	-
	$t4_{discard}$	Failure rate	0.4–0.5	0.4	[19]
	$t4_{reac}$	Success rate	0.5–0.6	0.6	[19]
	$t5_{discard}$	Failure rate	0.4–0.5	0.4	[19]
	$t5_{reach}$	Success rate	0.5–0.6	0.6	[19]

time 16–23 ms, fusion 82–95 ms), the system remains stable: arrival and service are comparable, and typical queues remain under four messages, confirming the correct sizing of the pipeline. However, higher frequencies or a worse PDR could quickly lead to instability, a critical risk in safety applications.

Figure 3C reports the average queue residence time, which is directly related to end-to-end latency. With the current configuration, keeping the total latency under 100 ms requires a message rate below 4 Hz; extending the limit to 300 ms offers only 1 Hz of headroom, because fusion, in particular, introduces structural delays of tens of milliseconds that are difficult to reduce without modifying the architecture or compute capacity. This result is consistent with [14], which highlights how fusion can generate significant delays.

Finally, Figure 3D–E show how the throughput of low-quality (LQ) and high-quality (HQ) fusions varies with data rate and radio conditions. C-V2X, thanks to the higher PDR, produces more HQ fusions: more complete and synchronized messages, therefore more reliable decisions. With 802.11p, however, the throughput of LQ increases; that is, decisions based on incomplete information (for example, when the message from the vehicle in front does not reach the one behind). The trade-off is clear: HQ improves decision quality but requires more time (95 ms vs. 82 ms), increasing latency; LQ is faster but imposes more conservative control behaviors.

IV. Conclusion and Future Work

We propose a modular, multi-formalism methodology for evaluating the performance of vehicle applications. The multi-formalism utilizes time Petri nets to capture complex structures and interactions, and queues to efficiently evaluate process performance. We have identified five modules that can be combined to facilitate the study and development of these applications, making this type of research more accessible. Experiments conducted on an ADAS scenario show the methodology's effectiveness, enabling rapid assessment of end-to-end latency and characterization of related safety limitations.

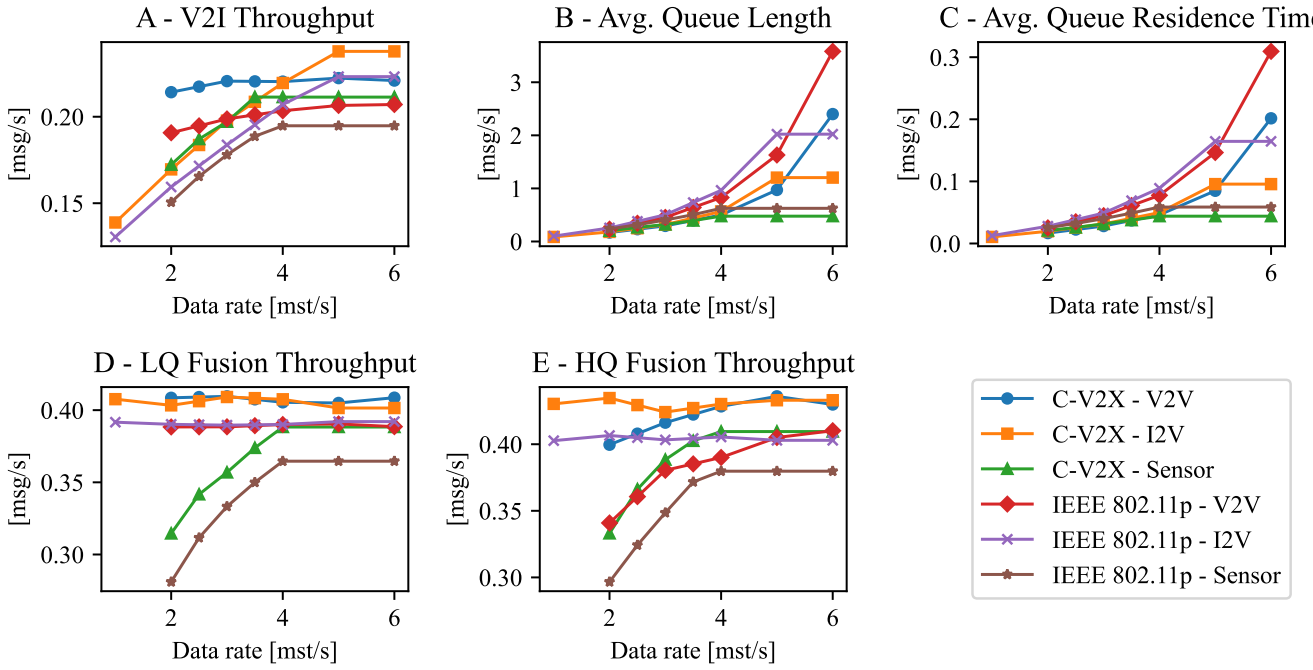


Fig. 3. Comparison of the IEEE 802.11p and C-V2X with the LTE-V2X protocol as the data rate varies.

In the future, we intend to continue the study by comparing it with solutions proposed by other simulators.

V. Acknowledges

This work has been funded by the Shift2SDV project (Grant Agreement No. 101194245), supported by the Chips Joint Undertaking and its members, including top-up funding by the national authorities of Austria, Denmark, Germany, Greece, Finland, Italy, Netherlands, Poland, Portugal, Spain, and Turkey.

References

[1] Z. Xinyu, L. Junxian, Z. Jingyi, Z. Shiyan, W. Jingyuan, Y. Yi, L. Jiale, and L. Jun, "Vehicle-to-everything communication in intelligent connected vehicles: A survey and taxonomy," *Automotive Innovation*, vol. 8, no. 1, pp. 13–45, 2025.

[2] Z. Bai, G. Wu, M. J. Barth, Y. Liu, E. Akin Sisbot, K. Oguchi, and Z. Huang, "A survey and framework of cooperative perception: From heterogeneous singleton to hierarchical cooperation," *Trans. Intell. Transport. Sys.*, vol. 25, no. 11, p. 15191–15209, Nov. 2024.

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[4] *Simulation of Urban MObility*, Eclipse SUMO, 2024. [Online]. Available: https://sumo.dlr.de

[5] A. Varga and R. Hornig, "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*, ser. Simutools '08. ICST, 2008.

[6] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34.

[7] R. Riebl, C. Obermaier, and H.-J. Günther, *Artery: Large Scale Simulation Environment for ITS Applications*. Cham: Springer International Publishing, 2019, pp. 365–406.

[8] A. Vladyko, P. Plotnikov, and G. Tambovtsev, "Simulation-based evaluation of v2x system with variable computational infrastructure," *Network*, vol. 5, no. 1, 2025.

[9] G. Franceschinis, M. Gribaudo, M. Iacono, N. Mazzocca, and V. Vittorini, "Drawnet++: Model objects to support performance analysis and simulation of systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2324 LNCS, pp. 233 – 238, 2002, cited by: 28.

[10] M. Tibaldi, G. Palermo, and C. Pilato, "Dynamically-tunable dataflow architectures based on markov queuing models," *Electronics*, vol. 11, no. 4, 2022.

[11] W.-S. Choi and S.-G. Choi, "Thread-based modeling and analysis in multi-core-based v2x communication device," *Sustainability*, vol. 14, no. 14, 2022.

[12] Annu and P. Rajalakshmi, "Distance-based queue modeling in sidelink c-v2x communication for platooning: Towards 6g-v2x," in *2024 IEEE 100th Vehicular Technology Conference*, 2024, pp. 1–5.

[13] R. Molina-Masegosa, J. Gozalvez, and M. Sepulcre, "Comparison of ieee 802.11p and lte-v2x: An evaluation with periodic and aperiodic messages of constant and variable size," *IEEE Access*, vol. 8, pp. 121 526–121 548, 2020.

[14] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, 2021.

[15] H. Park, M. Cho, Y. Jang, and J.-W. Choi, "Latency analysis of in-vehicle network for advanced driver assistance system," in *2024 IEEE 100th Vehicular Technology Conference*, 2024, pp. 1–7.

[16] M. Bertoli, G. Casale, and G. Serazzi, "Jmt: performance engineering tools for system modeling," *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 10–15, 2009.

[17] ETSI EN 302 637-2 V1.4.1. (2019) Intelligent transport systems (its); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf

[18] J. Park, V. Kavathekar, S. Bhuduri, M. H. Amin, and S. S. Devaraj, "A co-operative perception system for collision avoidance using c-v2x and client–server-based object detection," *Sensors*, vol. 25, no. 17, 2025.

[19] J. B. Cicchino, "Effectiveness of forward collision warning and autonomous emergency braking systems in reducing front-to-rear crash rates," *Accident Analysis & Prevention*, vol. 99, pp. 142–152, 2017.