Building Zero-touch Service Management Framework for Automotive Services Using the Smart Highway Testbed

Raúl Cuervo Bello
Internet & Data Lab (IDLab)
University of Antwerp & imec
Antwerp, Belgium
raul.cuervobello@imec.be

Nina Slamnik Kriještorac
Internet & Data Lab (IDLab)
University of Antwerp & imec
Antwerp, Belgium
nina.slamnik@imec.be

Matic Pokorn
Department for Communication Systems
Jožef Stefan Institute
Ljubljana, Slovenia
matic.pokorn@gmail.com

Jernej Hribar
Department for Communication Systems
Jožef Stefan Institute
Ljubljana, Slovenia
Jernej.Hribar@ijs.si

Carolina Fortuna
Department for Communication Systems
Jožef Stefan Institute
Ljubljana, Slovenia
carolina.fortuna@ijs.si

Johann M. Marquez-Barja
Internet & Data Lab (IDLab)
University of Antwerp & imec
Antwerp, Belgium
johann.marquez-barja@imec.be

***Abstract*—In Beyond 5G (B5G) and towards the Sixth Generation (6G), the performance of vehicular communications can be evaluated and improved through services located in the cloud and closer to the users in the Multi-Access Edge Computing (MEC) units. The increasing demand for connected cars requires an optimal distribution of the available network resources to ensure required levels of End-to-End (E2E) latency and the reliability of the vehicular services which contributes to the safety of the participants in traffic. The allocation of these resources can be improved with the support of Zero-touch Network and Service Management (ZSM). In this paper, we present a ZSM framework for automotive services using the Smart Highway testbed to improve the performance of vehicular services in a realistic environment. The proposed framework consists of decision-making processes that follow the principles of ZSM and intent-driven management. The decisions are based on the MEC units' workload and availability to match the demand from vertical services. The collected datasets from the decision-making process algorithm are used to train a Deep Reinforcement Learning (DRL) model and compare its results with a simple rule-based algorithm. The results show that DRL can quickly adapt to the dynamic environment of a testbed and outperform the conventional rule-based approaches. This indicates that the DRL algorithm can improve the decision-making process and ultimately decrease the E2E latency of vehicular services.**

***Index Terms*—zero-touch service management, intent-driven management, testbed, Smart Highway, orchestration, deep reinforcement learning**

I. Introduction

The increasing demand for new business-oriented services in the Beyond 5G (B5G) towards Sixth-Generation (6G) networks is growing along with the constant evolution of vertical applications [1]. These applications are service-oriented and as such they continue to thrive as more IoT devices continue to add up to the networks [2]. In the case of vehicular services, features are brought closer to the user end devices from the cloud to the edge, by using communication standards like Intelligent Transportation System (ITS-G5) and Vehicle-to-Everything (V2X) and taking advantage of the computing capabilities of Roadside Units (RSUs) and modern vehicles that are equipped with sensors integrated into On-board Units (OBUs). As these sensors get smaller and cheaper they become more present and generate large amounts of data collected from surrounding and dynamic environments. This brings new opportunities for applications in vehicular systems with real-time features that can contribute to critical services like collision avoidance, which have a relevant impact on the safety of Vulnerable Road Users (VRUs).

The escalation of these vertical applications depends on the availability of computing and network resources, which can vary dynamically in a highly heterogeneous environment composed of different technologies and providers. This requires more agile management of programmable network functions with the use of Software Defined Networking (SDN) and Network Function Virtualization (NFV) by implementing Network Slicing (NS) based on the specific needs of the use case. The large-scale deployment of vehicular services in the cloud and Multi-Access Edge Computing (MEC) for the connected cars, implies the constant re-conditioning of computing and network resources to manage massive flow of traffic [3], to provide competitive Key Performance Indicators (KPIs) levels such as low latency that contributes to maintaining the Quality of Service (QoS) and eventually the expected Quality of Experience (QoE). However, the available resources at the network edge (i.e., MEC units) can be limited and distributed within complex and heterogeneous B5G/6G networks. To efficiently allocate computing and network resources where they are needed, traditional MANagement and Orchestration (MANO) operations need to cope with networks that become more complex because of the demanding performance of applications and devices. To tackle these challenges, Zero-touch Network and Service Management (ZSM) techniques can provide intelligent capabilities for automatic network and service management [4], to efficiently manage computing and network resources [1], along with the use of Artificial Intelligence and Machine Learning (AI/ML) practices [1].

In this paper, we present a ZSM framework for vehicular services, based on European Telecommunications Standards Institute (ETSI) principles for Zero-touch service and Intent-driven management. The framework is deployed in the form of rule-based and Deep Reinforcement Learning (DRL)-based zero-touch services, which perform a decision-making process for the orchestration of vehicular services in a realistic environment, placed in the Smart Highway testbed[1]. In this testbed, we deployed a setup of Proof-of-Concept (PoC), to validate the Zero-touch services and their impact on the performance of vehicular services. In general, these Zero-touch services are AI/ML-based functions and/or algorithms to support the orchestration of services in vehicular systems by improving the decision-making process, based on the availability of computing and network resources. During the orchestration of vehicular services, data is collected from the CPU and memory workload at the edge computing units where the services are running, and End-to-End (E2E) latency measurements. These datasets are analyzed by the decision-making Zero-touch service and provide decisions based on the availability of computing and network resources. Along with the resulting decisions, the CPU, memory and latency data parameters are then used to train and validate a DRL model. The goal of this experiment is to evaluate how feasible the use of a DRL model is for making decisions to reduce E2E latency. Latency represents a key KPI value for the QoS of vehicular services. The DRL model is expected to adapt to the dynamic environment that the Smart Highway testbed provides and contribute to improving the performance of the ZSM framework.

II. Zero-touch Service Management and Intent-driven management

A. *Zero-touch Management*

The goal of ZSM is to automate the management of 6G network services based on service-level policies and rules to improve service performance [1]. This way, network operators can effectively manage network resources and enhance network performance. ZSM also enables a self-managing network system that can self-monitor, self-heal, and optimize its operations. The principles for autonomous network and services management rely on the foundation of technologies such as SDN, NFV, MEC and NS, which enable network programmability. Such programmability is essential for the flexibility in automated networks and offers great advantages to the B5G and 6G paradigms, such as the capability to optimize their processes to offer minimal latency during real-time operations, improved performance, high reliability, seamless connection and mobility support.

One of the key targets of ZSM is to enable automated network management across different vendors and domains [5]. To achieve this, a common interface is needed for data analytics and closed-loop control components. This interface facilitates the creation, execution, and governance of single or multiple closed-loops within end-to-end networks and enables the adoption of ZSM by the Mobile Network Operators (MNOs). In 2017, the ZSM Industry Specification Group (ISG) was created by the ETSI to coordinate the efforts toward standardization for the design of a framework for network automation based on the definitions of principles based on ZSM. ETSI provides descriptions for the ZSM architecture and references that describe the support for open interfaces, model-driven services and resource abstraction. These specifications are meant to grant ZSM the capabilities of modularity, flexibility, scalability and extensibility [5]. ZSM techniques are designed to improve the performance of the vertical services that are implemented in the B5G/6G networks. Their objectives are aligned with the vision of autonomous network and service management, that can be stable and highly reliable to reduce or eliminate human intervention [3].

B. *Intent-driven Management*

In a ZSM environment, autonomous network solutions can be conditioned by intent-driven approaches [6]. The operations carried out by the zero-touch services are expected to fulfill specific requirements such as service instantiation or termination, based on conditions that are triggered by a wide spectrum of circumstances like service disruption or demand escalation. Zero-touch service operations are meant to accomplish goals that can be defined by intents, that can be expressed in high-level but constrained language [6]. In an autonomous framework, intents can be created and used during runtime without previous implementation. This capability is essential for flexibility and is part of the nature of autonomous systems e.g., conflict detection, and conflict resolution.

The ETSI ZSM Group defines the intents as *"the formal specification of the expectations, including requirements, goals, and constraints, given to a technical system"* [6]. The formulation of the intents must be independent of the technology or the vendors, for this purpose the instructional language is constructed in an agnostic form. In [7], He et al. explain the importance of Intent-based Networking for Vehicular Edge Computing (VEC), as it enhances the capabilities of the approaches for dynamic resource management. The authors also propose a framework for VEC where intents are generated in the interaction between service providers and subscribers. These intents need processing and translation into a model that can be consumed by network controllers across the layers of device configurations. This approach presents an example of how Constrained Natural Language (CNL) models can be embedded in the communication flow where orchestration services provide instructions for managing resources. To test the performance of decision-making processes that allocate network resources for vehicular services based on RSU workload, we experiment on the Smart Highway testbed, thanks to the possibilities it brings for deploying PoC within realistic conditions. The resulting datasets are valuable for the training of the DRL model that would support the decision-making processes of the zero-touch services.

III. Smart Highway testbed and Experiment Setup

A. *The Smart Highway testbed*

The Cooperative Intelligent Transport System (C-ITS) testbed, also known as the Smart Highway [8], provides the playground where our experimentation is carried out. This testbed allows us to collect data based on realistic environments thanks to its seven RSUs deployed on a section spanning four kilometers long of the E313 highway in Antwerp, Belgium as shown in Figure 1. The test site has been designed to enable research in the scope of distributed/edge computing and vehicular services [9]. The RSUs are computing-capable units and they are part of the setup of our experiment by hosting containerized applications such as vehicular services, media content publishing, resource monitoring (e.g., RSUs availability based on their CPU, memory workloads and latency), decision-making processes and orchestration [2]. Each RSU features the following [10], as displayed in Figure 1:

1) Radio and antenna units to support wireless communications with Cohda MK5 and MK6c over IT-G5 and

[1] Smart Highway: https://www.fed4fire.eu/testbeds/smart-highway/

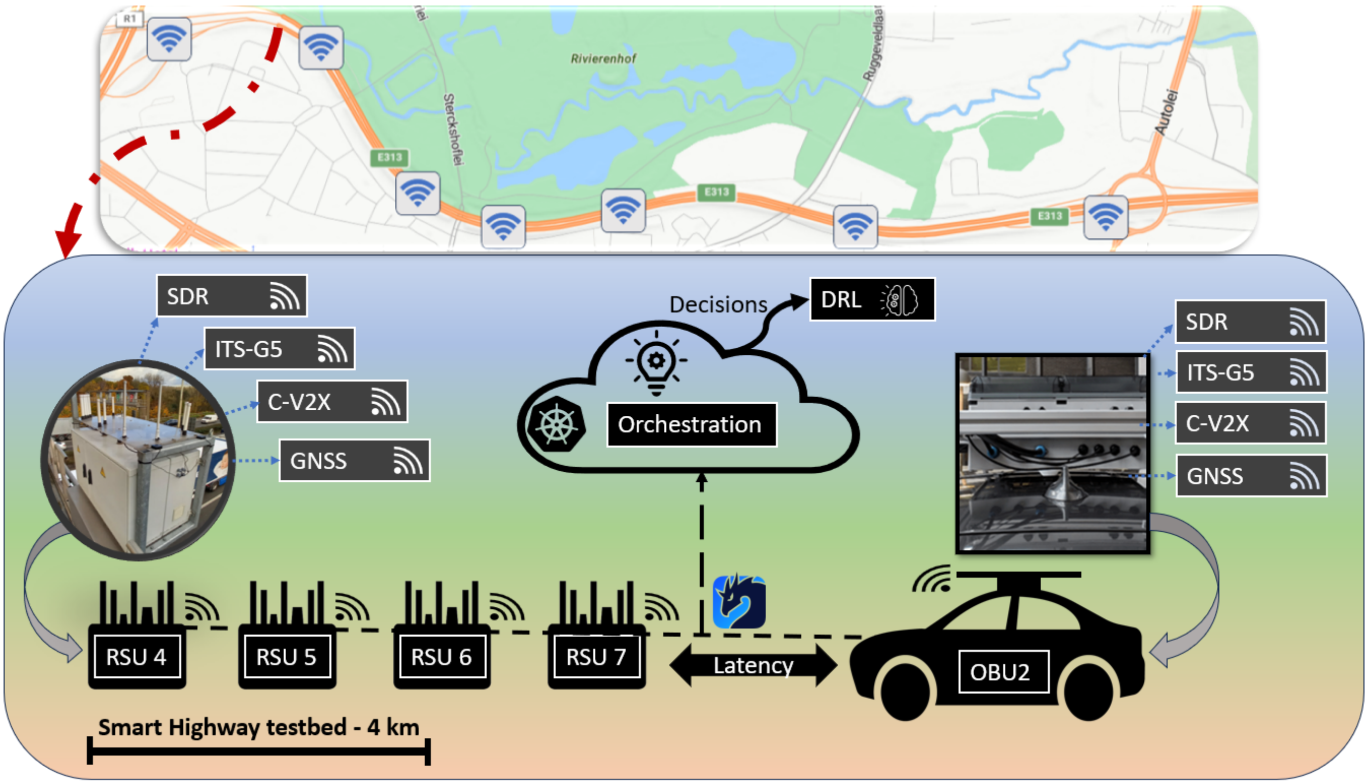

Fig. 1: Experiment setup on the Smart Highway testbed located at the E313 highway in Antwerp, Belgium.

LTE-V2X in the bands 5.9GHz through a PC5 interface and 3.5GHz through a Uu interface and Peplink 5G[2].

2) Computing processing units provided by General Purpose Computing Units (GPCUs) that enhance the edge computing capabilities with Intel Xeon 8 Cores and 32 GB RAM.
3) Power control unit to provide and manage energy throughout the RSU components.
4) Backup and recovery to deal with events of failures.
5) Remote access to manage the deployed setups.

We also use an OBU, which is a mobile unit BMW X5 xDrive25d LO equipped with similar appliances as the RSUs, as shown in Figure 1, to evaluate the E2E latency of the RSUs which is fundamental for the performance of the vehicular services. For monitoring network and vehicular services, execution of stress tests, orchestration and data collection are run as services in a containerized environment supported by Kubernetes, as displayed in Figure 1. Our logical setup for the experiment is organized as follows:

- RSUs 4, 6 and 7 replicate multi-site hosting deployment while publishing status stats for monitoring purposes.
- RSU 5 contains algorithms that mimic the demands from users by performing requests to specific RSUs selected randomly in a specific time-lapse.
- RSU 7 also contains algorithms for orchestration, decision-making and data collection about performance and available resources e.g., Central Processing Unit (CPU), memory and latency. These algorithms can be run on any of the RSUs involved as they share similar technical characteristics. In this experiment, we deployed them in a single RSU as the rest of the vehicular services are distributed in the remaining RSUs.
- OBU 2 works as a mobile unit that measures the response time from the RSUs deployed in the Smart Highway testbed and collects data for latency records.

[2]Peplink 5G: https://www.peplink.com/technology/what-is-5g-with-peplink/

```
imec_orchestrator/rsu/decision    "decision": "deployment"
                                 }
imec_orchestrator/rsu/decision   {
                                   "host": "rsu5",
imec_orchestrator/rsu/decision     "decision": "terminate"
                                 }
```

Fig. 2: Exposed intent-based decision expressions published over Zenoh.

We relied on Zenoh[3] as a means of communication between the running applications across the RSUs and the OBU. Zenoh is a publish/subscribe technology that provides high compatibility within distributed systems, which makes it ideal for a typically diverse environment at the Edge. It helps to improve performance thanks to its noticeably low burden of extra head information in the messages [11].

B. *Intent-based decision expressions*

In an autonomous management framework, the interaction between the management services (e.g., between decision-making and resource management) can be supported by intents that are expressed in a declarative form [6] like the following: *deploy, terminate, scale up* or *scale down*. These intents

[3]Zenoh: https://zenoh.io/

represent the properties of the expected outcome as a goal, rather than containing a comprehensive description of the solution. The following example exposes the declaratively expressed decisions that were made in the context of the orchestration of network and services presented in Figure 2.

In the case of our experiment, we take into account the ETSI definitions for Intent-driven management in the scope of ZSM [6], as it follows: the availability of the resources for network and services in the RSUs is part of the input used during the decision-making processes. After that, the resulting output from the decision-making processes becomes a key factor for the orchestration services. The decisions made are presented as intentions that express the results expected from the orchestration actions, without defining how they are going to be executed by the defined deployment services. The intent-based interaction between the services for monitoring, decision-making and management services, brings flexibility and can expand static pre-defined rules for service operations such as scaling, instantiation or termination.

C. Experiment setup in realistic environment at the Smart Highway testbed.

The main goal of the testbed setup is to create an environment to deploy PoC for testing and validation of zero-touch services within the realistic environment conditions offered by the Smart Highway testbed. These services are AI-based functions and/or algorithms that support the orchestration service by improving the decision-making process. Its goal is to run algorithms that make decisions based on available computing and network resources.

Although the experiment setup can be replicated in a virtualized environment through simulation or modeling, the conditions that surround the RSUs deployment allow us to test values like latency under circumstances that produce datasets closer to reality. This is essential for the training of a DRL model that can later be used for the improvement of the decision-making process, as it is later explained in Section 3. The experimentation consists of deploying virtual machines on RSUs 4, 5, 6 and 7 and OBU 2 of a mobile unit. Within each RSU we have a deployment of Kubernetes-based nodes that provide vehicular services to be consumed through the network such as: assisted navigation, speed optimization, obstacle detection, collision avoidance and infotainment.

To connect the nodes we use Zenoh, which allows us to dispense with hypostatic addresses and lightweight transfer of information with a low burden of data overhead, to monitor the state of resource consumption on RSU 4-6 and 7 taking into account the load level of the processors and the E2E latency. We stress the computing and network resource usage to trigger different reactions and collect data for training. During this process, the selected RSUs will receive requests to access the available vehicular services deployed in one of the RSU nodes. In future work, we aim to interwind the stress tests with accurate inputs from the real-time state of the traffic of participants at the Smart Highway testbed.

There is a correlation between the increase of CPU load and E2E latency. As concurring requests are made to push the availability of the RSUs capacities to their limits, the generated stress increases CPU consumption and eventually affects the E2E latency of vehicular services depending on how busy the RSU computing units are while processing the requests. Under these conditions, the stressed RSU behaves as if it is under a high level of demand. To measure the E2E latency we use the OBU 2 (as shown in Figure 1) by sending requests to the RSUs and calculating the response time. This information is sent to the RSU 7 where the rest of the parameters like CPU and memory load are collected by the monitoring services.

The higher the latency of vehicular services running on the edge and the CPU load level of an RSU, the less availability would be left to provide the mentioned services. Each RSU will publish these parameters through Zenoh and they will be evaluated or analyzed by the algorithms that are running on RSU 7 involved in the decision-making processes. The decision algorithms review the data obtained by the rest of the RSUs and select the RSU that indicates the lowest latency and CPU consumption. The earlier implementations for the decision algorithms were rules-based and provided datasets of parameters and decisions that are later used for training a DRL based decision-making algorithm. The overall decisions are expected to be reinforced by using DRL, whose performance is later compared along with the rules-based algorithm in the next section of this paper. The resulting output from the decisions is expressed declaratively in sub-intents that contain the identity of the RSU that has better resource availability, to be later implemented by entities like Kubernetes master in our case, for optimal performance of the network and services.

IV. DRL-based Decision-Making to improve orchestration for RSU selection in the Smart Highway testbed.

Distributed edge computing in the testbed, available in the form of accessible RSUs, creates a complex and dynamic environment. In such environments, DRL has emerged as a suitable option for decision-making processes in recent years, as it has been proven to effectively learn optimal strategies in complex network environments while adapting to changing conditions [12]. Moreover, DRL offers a solution that can make near real-time decisions regarding which RSU should be used for the computational task of the user. In our experiment, we adapt the Deep Q-Network (DQN) algorithm [13], a well-known DRL approach, to tackle the problem of RSU selection. This is important for users, as it can affect the availability of computing and network resources for vehicular services.

To employ the DQN algorithm, we defined the state $\mathbf{s}(t)$, the action $a(t)$, and a reward signal $r(t)$, using the information available from the Smart Highway testbed, described in Section III-A. The states consist of a k long trace i.e., a sliding window, the CPU load, the memory usage, the available memory and the past latency for each RSU. The resulting state is very large, which leads us to use a neural network i.e., deep learning, for the Q-value approximation function. The available actions represent the selection of RSU i.e., $a(t) \in \{1, 2, \ldots, n\}$, where n is the number of RSUs. Finally, we define the equation for the reward signal as follows:

$$r(t) = -l(t-1), \tag{1}$$

where $l(t-1)$ represents the achieved latency in the previous time step. The objective of the DQN algorithm is to maximize the reward. Thus, by selecting the negative value of the latency as the reward signal, the algorithm will find a policy, i.e., a series of actions, that minimizes the latency in the system.

In Algorithm 1, which acts as a Zero-touch service, we summarize our implementation of the DQN approach for RSU selection in the Smart Highway testbed. The algorithm starts by initializing the policy and the target networks along with a replay memory and proceeds by observing the initial state. We employ an ϵ-greedy approach for action exploration (line

TABLE I: Notation Used in the DRL-based Solution.

Notation	Description	Notation	Description
$\mathbf{s}$	state	a	action
r	reward	k	sliding window size
n	number of RSUs	l	latency
Q	policy network	θ	policy network weights
Q'	target network	$\theta^{'}$	target network weights
λ	soft update	D	replay memory size

Algorithm 1 Proposed DRL solution for RSUs selection.

1: Randomly initialise policy network $Q(\mathbf{s}, a|\theta)$
2: Initialise target network Q' with weights $\theta^{'} \leftarrow \theta$
3: Initialise replay memory $\mathcal{D}$ to capacity D
4: Observe initial state $\mathbf{s}(t)$ at time-step $t = 0$
5: **for** $i = 1, T$ **do**
6: With probability ϵ select a random action
7: Otherwise select $a(t_i) = argmax_a Q(\mathbf{s}(t_i), a|\theta)$
8: Select RSU according to selected action a
9: Observe $\mathbf{s}(t_{i+1})$ and latency $l(t_i)$
10: Determine $r(t_i)$ with Eq. (1)
11: Store experience $\mathbf{s}(t_i), \mathbf{s}(t_{i+1}), r(t_i), a(t_i)$ in $\mathcal{D}$
12: Sample random batch of J experiences from $\mathcal{D}$
13: **for** every $\{\mathbf{s}(t_j), \mathbf{s}(t_{j+1}), r(t_j), a(t_j)\}$ in batch **do**
14: $y(t_j) = r(t_j) + \gamma max_{a(t_{j+1})} Q^{'}(\mathbf{s}(t_{j+1}), a(t_{j+1}))$
15: **end for**
16: Calculate loss: $\mathcal{Z} = \frac{1}{J} \sum_{j=0}^{J-1} (Q(\mathbf{s}(t_j), a(t_j)) - y(t_j))^2$
17: Update $Q(\mathbf{s}, a|\theta)$ by minimising the loss $\mathcal{Z}$
18: Softly update the target network: $\theta^{'} \leftarrow \lambda\theta$+(1-$\lambda$)$\theta^{'}$
19: **end for**

6). This means that the algorithm selects a random action, i.e., explore, with a probability of ϵ instead of using the reward with the highest Q-value (line 7). The system then offloads the task to the selected RSU, and after the time step passes, we observe the next state space and determine the reward using Eq. 1 (line 10). An experience, a tuple consisting of state, action, reward and next state, is then stored in the experience replay buffer. The algorithm then trains the Q networks by first randomly selecting experiences from the replay memory and calculating the target values for training the network (line 14). This is followed by the minimization of the loss function by gradient descent (line 16 and line 17). The latter allows the Q-network to learn to accurately predict the expected future rewards for each state-action pair and thus improve its performance over time. In the last step (line 18), the target network is softy updated with the factor λ. Note that the training (lines 12 and 16) can be performed in parallel to other tasks to ensure fast decision-making.

In our work, we have optimized the hyperparameters using a comprehensive grid search methodology. We systematically trained and evaluated the model over a predefined range of values for each hyperparameter to determine the optimal value. For example, for batch size, we considered values such as 16, 32, 64 and 128 and trained separate DQN models for each batch size. We identified 32 to be the best value as larger batches with more than 32 experiences did not perform better and smaller batches had lower performance. The architecture of both the policy and target networks, consisting of two dense hidden layers with 32 neurons, was found to be the most effective configuration, providing an optimal balance between performance and computational cost. For the optimization, we

TABLE II: DQN Hyperparameters.

Parameter	Value	Parameter	Value
n	3	k	8
episode length	5295	episodes	10
batch size	32	activation function	ReLU
γ (reward discount)	0.9	dense layers	2
dense layer size	32	memory capacity	2000

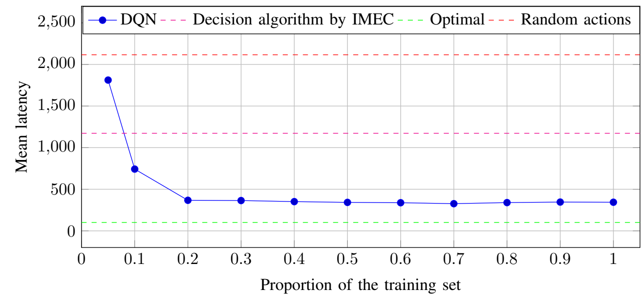

Fig. 3: Change in mean latency over the proportion of used measurements gathered from the Smart Highway testbed.

used the Adam optimizer, which is known for its efficiency in handling large datasets and complex architectures. The detailed specifications of the remaining hyperparameters are listed in Table II.

Figure 3 illustrates how the DQN algorithm learns from the data collected from the Smart Highway testbed. We split the collected data into training and test data and then used different percentages of the training data to test the performance of DQN. When only a fraction of the training dataset is used, about five percent, the performance of DQN is only slightly better than that of a random selection approach i.e., selecting RSUs at random. However, the swift improvement leads to better performance results than the conventional decision algorithm. This indicates that DRL can quickly adapt to the dynamic environment of a testbed, as it only requires a relatively small number of data samples to outperform conventional rule-based approaches. The performance of DRL using the datasets from the decision-making process during the orchestration of the vehicular services indicated promising results based on the rapid adaptability of this algorithm in the dynamic environment of the Smart Highway testbed. This characteristic made DRL outperform the conventional rule-based approach for decision-making, presenting it as a feasible candidate for the support and improvement of the orchestration of vehicular services.

V. Conclusion and Future Work

This work presents promising results of the use of DRL based algorithm in dynamic environments like the Smart Highway testbed, to perform decision-making processes for improving the orchestration of computing and network resources for vehicular services. The adaptability of this algorithm can outperform the conventional rules-based decision-making algorithm variant. These results contribute to showcasing how Zero-touch services based on Machine Learning (ML) practices like DRL can have a positive impact on automated and intelligent orchestration/network service management. However, using these techniques presents also challenges to be tackled, such as the intense consumption of computing resources. As the current PoC continues to expand its features, we will focus also on optimizing the use of DQN algorithms by extended at Iasi.png

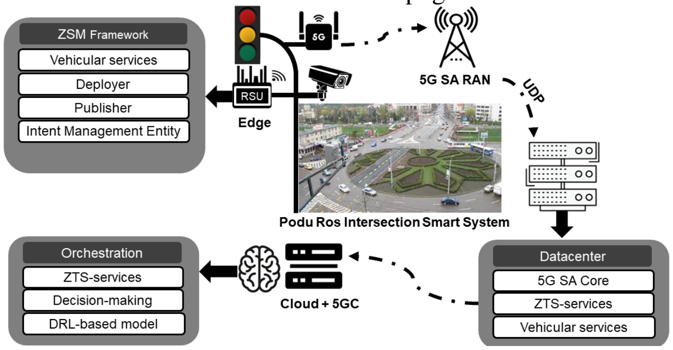

Fig. 4: PoC planned scale-up at Podu Roș intersection, Iași, Romania

exploring strategies to reduce the computational overhead of the DRL model. Additionally, improving the interpretability and security of DRL algorithms for real-world applications is proving to be a key challenge that can be addressed with PoC. Furthermore, such works will be crucial to demonstrate that decisions in dynamic environments can be both reliable and comprehensible.

Several approaches have been taken to reduce the costs of processing data like power, and bandwidth while offloading computing tasks to a remote cloud. The use of MEC brings computing resources to the user devices as it can reduce energy consumption and response time [14]. Another related work [15] proposes a scheme that uses the least absolute shrinkage to provide a lower variation and higher rewards while solving the data sparsity problem. As a final mention, reward shaping can improve the performance of deep reinforcement learning in perishable inventory management [16], and also increase training stability and trust in the policies obtained by black box DRL algorithms.

One of the main goals of the presented experiment was to obtain data related to the demand for computing and network resources under a realistic environment like the one provided by the Smart Highway testbed. For future experiments, the current PoC is planned for scale-up as part of the TrialsNet[4] project, in the context of Smart Traffic Management (STM), at the Podu Roș intersection in Iasi, Romania as shown in Figure 4. This will allow us to obtain more accurate data from the demand received by the RSUs. This will also contribute to validating and reproducing the PoC in different scenarios outside of the Smart Highway testbed environment.

The aforementioned use case aims to develop applications of B5G/6G networks supported by computing resources to provide optimal traffic flow and improve the VRUs safety to prevent accidents. In the case of autonomous intent-driven management, a higher level of specialization should be designed to enhance the decomposition and interpretation of the intents across more complex and deep layers in the scope of the orchestration. Also, extra measures for coordinating and monitoring the intent-based interaction would assure its proper performance and failure recovery to prevent human intervention as it follows in the ZSM principles.

Acknowledgement

This work has been performed within the European projects: Horizon 2020 Vital-5G project (Grant Agreement No. 101016567) and SNS project: TrialsNet (Grant Agreement No. 101095871). This work was also supported in part by the HORIZON-MSCA-IF project TimeSmart No. 1010637.

[4]TrialsNet: https://trialsnet.eu/deliverables/

References

[1] M. Liyanage, Q.-V. Pham, K. Dev, S. Bhattacharya, P. K. R. Maddikunta, T. R. Gadekallu, and G. Yenduri, "A survey on zero touch network and service management (zsm) for 5g and beyond networks," *Journal of Network and Computer Applications*, vol. 203, 2022. doi: https://doi.org/10.1016/j.jnca.2022.103362.

[2] N. Slamnik-Kriještorac and J. M. Marquez-Barja, "Unraveling edge-based in-vehicle infotainment using the smart highway testbed," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–4, 2021. doi: https://doi.org/10.1109/CCNC49032.2021.9369622.

[3] N. Slamnik-Kriještorac, M. Camelo, C.-Y. Chang, P. Soto, L. Cominardi, D. De Vleeschauwer, S. Latré, and J. M. Marquez-Barja, "Ai-empowered management and orchestration of vehicular systems in the beyond 5g era," *IEEE Network*, vol. 37, no. 4, pp. 305–313, 2023. doi: https://doi.org/10.1109/MNET.008.2300024.

[4] E. T. S. Institute, *Zero-touch network and Service Management (ZSM); Terminology for concepts in ZSM*. 2023. Online [Available]: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/007/02.01.01_60/gs_ZSM007v020101p.pdf, Date accessed: 2023-09-01.

[5] ETSI Industry Specification Group (ISG), "Zero-touch network and Service Management (ZSM); Reference Architecture." ETSI GS ZSM 002 V1.1.1 (2019-08), 2019. Online [Available]: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf, Date accessed: 2023-09-07.

[6] E. T. S. Institute, *Zero-touch network and Service Management (ZSM); Intent-driven autonomous networks; Generic aspects*. 2023. Online [Available]: https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/011/01.01.01_60/gr_ZSM011v010101p.pdf, Date accessed: 2023-12-14.

[7] T. He, A. N. Toosi, N. Akbari, M. T. Islam, and M. A. Cheema, "An intent-based framework for vehicular edge computing," in *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 121–130, 2023. doi: https://doi.org/10.1109/PERCOM56429.2023.10099081.

[8] Marquez-Barja, J. and Lannoo, Bart and Naudts, Dries and Braem, B. and Donato, C. and Maglogiannis, Vasilis and Mercelis, S. and Berkvens, R. and Hellinckx, P. and Weyn, M. and Moerman, Ingrid and Latré, Steven, "Smart Highway : ITS-G5 and C2VX based testbed for vehicular communications in real environments enhanced by edge/cloud technologies," in *2019 European Conference on Networks and Communications (EuCNC), Abstracts*, p. 2, IEEE, 2019. [Online] Available: https://biblio.ugent.be/publication/8642435.

[9] V. Charpentier, N. Slamnik-Krijestorac, and J. Marquez-Barja, "Latency-aware c-its application for improving the road safety with cam messages on the smart highway testbed," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2022. doi: https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798350.

[10] I. University of Antwerp, "Smart highway testbed." https://www.uantwerpen.be/en/research-groups/idlab/infrastructure/smart-highway/. Online [Available]: https://www.uantwerpen.be/en/research-groups/idlab/infrastructure/smart-highway/, Date accessed: 2023-11-24.

[11] G. Baldoni, J. Loudet, L. Cominardi, A. Corsaro, and Y. He, "Zenoh-based dataflow framework for autonomous vehicles," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 555–560, 2021. doi: https://doi.org/10.1109/QRS-C55045.2021.00085.

[12] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019. doi: https://doi.org/10.1109/COMST.2019.2916583.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level Control Through Deep Reinforcement Learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015. doi: https://doi.org/10.1038/nature14236.

[14] Z. Zabihi, A. M. Eftekhari Moghadam, and M. H. Rezvani, "Reinforcement learning methods for computation offloading: A systematic review," *ACM Comput. Surv.*, vol. 56, aug 2023. doi: https://doi.org/10.1145/3603703.

[15] J. Park and K. Chung, "Distributed drl-based computation offloading scheme for improving qoe in edge computing environments," *Sensors*, vol. 23, no. 8, 2023. doi: https://doi.org/10.3390/s23084166.

[16] B. J. De Moor, J. Gijsbrechts, and R. N. Boute, "Reward shaping to improve the performance of deep reinforcement learning in perishable inventory management," *European Journal of Operational Research*, vol. 301, no. 2, pp. 535–545, 2022. doi: https://doi.org/10.1016/j.ejor.2021.10.045.