DATA PROCESSING ARCHITECTURE FOR C-ITS: USE CASES, TECHNICAL SOLUTIONS AND FUTURE DIRECTIONS

L. MENDIBOURE& S. MAALOUL[1,2] & T. SYLLA[1] & S. ERCAN[1] & H. ANISS[1] & A. HEDHLI[1]
[1]COSYS-ERENA Lab, University Gustave Eiffel, France
[2]JUNIA Engineering School, 59000 Lille, France

leo.mendiboure@univ-eiffel.fr, sassi.maaloul@univ-eiffel.fr, tidiane.sylla@univ-eiffel.fr, secil.ercan@univ-eiffel.fr, hasnaa.aniss@univ-eiffel.fr, abdelmename.hedhli@univ-eiffel.fr

ABSTRACT

Data processing is a major challenge for Cooperative Intelligent Transport Systems (C-ITS). Indeed, some applications related to connected and autonomous vehicles (CAV) are driving the development of ever more complex C-ITS services: automated driving, remote driving, platooning, etc. However, these applications have significant constraints in terms of Quality of Service (QoS): latency, reliability, bandwidth, etc. The data processing architecture that will be deployed for these C-ITS services must therefore be able to guarantee short delays both in terms of data analysis and processing, as well as a high level of availability. Without this condition, the proper functioning of C-ITS services will not be guaranteed, and the consequences could be significant for road safety and traffic fluidity.

Therefore, in this article, we propose to categorize C-ITS applications and highlighting their requirements in terms of Quality of Service. We will then study the existing architectural solutions for data processing (Cloud Computing, Edge Computing, etc.) and identify their advantages and drawbacks. As a third step, we will associate C-ITS applications and data processing technologies to propose a complete processing architecture ensuring an efficient operation of all the C-ITS applications. We will also define in this section some use cases proving the relevance of the proposed architecture. Finally, we will highlight the current and future challenges of implementing these new data processing architectures.

Keywords: C-ITS, Quality of Service, Data Processing, Cloud Computing, Edge Computing

1. INTRODUCTION

Cooperative Intelligent Transport Systems (C-ITS) [1] will play a key role in the future of Connected and Automated Vehicles (CAVs). Indeed, C-ITS, using different radio access technologies (ITS-G5, 4G/5G, etc.) [2], could allow these CAVs to exchange different pieces of information that could improve both road safety and traffic efficiency: obstacle detection, lane change, sudden braking, etc.

Existing C-ITS architectures, such as the one proposed in the SCOOP@F project [3], rely on a central (national) node managing the different C-ITS services. However, new emerging C-ITS services such as HD map sharing [4] for accurate vehicle positioning, vehicles platoons management [5] for energy savings and intersection control [6] to improve road safety present significant Quality of Service (QoS) requirements. Indeed, these C-ITS services imply important guarantees regarding bandwidth, latency and reliability [7].

The communication and data processing architectures used by C-ITS must therefore evolve today. The centralized and remote architectures (Cloud Computing) used so far do not seem able to meet the QoS requirements of these new C-ITS applications [8]. Among the solutions that could be considered, distributed computing architectures, called Edge Computing [9], seem to be able to play an essential role in the future of C-ITS [10]. Indeed, these architectures, widely promoted by mobile network operators, guarantee low latencies for high bandwidth and computationally demanding applications.

This is why, in this paper, we focus on C-ITS's new communication and data processing architectures. More specifically, we propose an analysis of the different solutions that could be considered for implementing these Edge Computing solutions and try to identify their limits and perspectives for each of them. This paper represents an interesting knowledge base for future works related to C-ITS services.

The rest of this paper is organized as follows. Section 2 briefly describes the context: C-ITS applications and architectures and the evolution of data processing architectures. Section 3 then proposes a quick study of the Edge Computing solutions applied so far in the vehicular environment. It also highlights their advantages and limitations. Finally, section 4 identifies different possible directions for further work on future C-ITS data-processing architectures.

2. BACKGROUND

This section presents some background information necessary to understand this article. We introduce important elements related to C-ITS (cf. Section 2.1) and concepts related to data processing architectures (cf. Section 2.2).

2.1. C-ITS Architecture Applications

The advances in wireless communications provide faster connections, lower latency, higher reliability, higher capacity and wider coverage. This enables the deployment of efficient C-ITS applications to increase road safety, traffic efficiency and the advent of connected and automated vehicles.

The communication architecture considered for C-ITS is based on two main technologies: ITS-G5 and 4G-V2X/5G. As a result, the communication architecture proposed for C-ITS is based on a hybrid architecture, as shown in Figure 1. The ITS-G5 is dedicated to the short-range communications between vehicles (vehicle-to-vehicle, V2V) and between vehicles and infrastructures through Road Side Unit (RSU) (vehicle-to-infrastructure, V2I). The long-range is allowed based on cellular technology 4/5G and can be used as a complement on roads not covered by short-range communications.

All C-ITS services are available through a central server called Nfr-ITS-S, which forwards to vehicles or Traffic Management System (TMS) information issued from different servers (e.g. Road Operator's Platforms (PSA, SNCF, etc.), application server (for smartphone applications), MCTO server (for MCTO applications), traffic light system (for GLOSA application). The Nfr-ITS-S server manages the C-ITS services exchanged between all servers and each end user. Figure 1 represents a global view of the functional links between all actors.

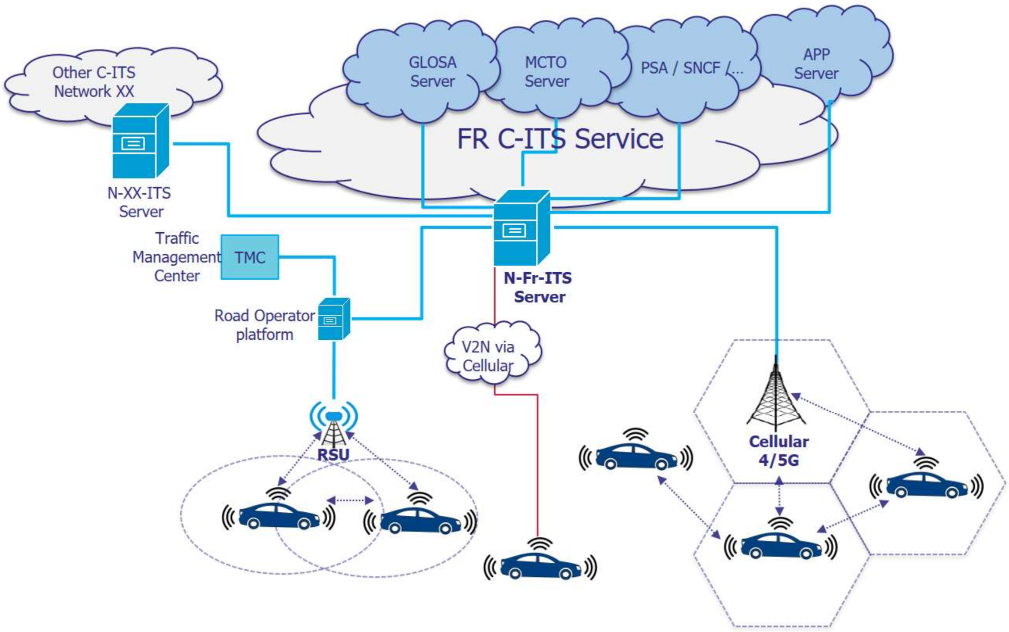

Figure 1: C-ITS functional architecture in France

Many use cases have been developed as part of the development of C-ITS projects, varying from convenience services (e.g., traffic information) to complex services expected for Connected and Automated Vehicles (CAVs). For each use case, stringent requirements must be supported by the underlying communication network. Different C-ITS services can have different requirements. In this context, the classification of V2X (vehicle-to-everything) services has been addressed by several organisations to classify V2X services:

- According to ETSI [11], all the use cases can be classified into 2 groups: Safety and Non-safety applications.
- 5G Automotive Association (5GAA) has identified the following groups of use cases: safety, vehicle operations management, convenience, autonomous driving, platooning, traffic efficiency and environmental friendliness, society and community [12].
- In the 5G Communication Automotive Research and innovation (5GCAR), all the use cases are categorized into five groups: cooperative manoeuvre, cooperative perception, cooperative safety, autonomous navigation, and remote driving [13].
- 3GPP [14] use cases are defined in 4 groups: Vehicle Platooning, Advanced Driving, Extended sensors and Remote Driving.

This diversity of classification allows the identification of the requirements and needs of the use case in terms of vehicular communication performances. Table 1 shows the main requirements of the advanced V2X services defined by 3GPP NR-V2X [14]. These advanced V2X services are grouped into Vehicle Platooning, Advanced Driving, Extended sensors and Remote Driving. The requirements depend on the specific use case.
For example, for Remote Driving that allows remote operation of a vehicle (either from a human operator or a V2X server), the maximum end-to-end latency must be lower than 5 *ms*. The Platooning use case group also requires low latency communications and high periodicity for broadcasting messages among a group of vehicles. Latency is a primary parameter for V2X services as a delay in the transmission or reception of messages can affect the proper functioning of the corresponding service. Other services for Extended Sensors, transmitting high-quality video and multi-sensor data, are often required by remote

drivers to assess rapidly changing traffic conditions around the vehicle. However, sharing full-featured ultra-high resolution or higher quality videos may require higher bandwidth, low latency and an important computation process. We can also note that the network's reliability must be higher than 90% for all use cases.

V2X group	Latency (ms)	Size (Bytes)	Reliability (%)	Data rate (Mbps)	Range (m)
Advanced Driving	3-100	300-12000	90-99.999	10-50	360-500
Remote Driving	5	-	99.999	DL:25 UL: 1	-
Extended Sensors	3-100	1600	90-99.999	10-1000	50-1000
Vehicle Platooning	10-500	50-6000	90-99.99	50-65	80-350

Table 1. Performance requirements for use cases groups [14]

2.2. Data Processing Architectures: From Cloud Computing to Edge Computing

The most widely used data processing architectures today, including in the C-ITS domain, are based on a Cloud Computing model [15]: clients use the capabilities of remote servers, managed by Cloud Computing operators, to access an application or computer/network/storage resources. This model enables the customer to limit the costs associated with acquiring and managing a server farm. Nevertheless, the limits of this Cloud Computing model are numerous, especially for C-ITS services requiring both low latency and important computation processes (cf. Section 2.1). In particular, we can mention [16]:

1. The significant latency associated with Cloud Computing: The distance between the Cloud servers and the end users necessarily implies a high communication latency due to the non-consideration of the users' location. This approach cannot, therefore, meet the requirements of low-latency applications.
2. The lack of context-awareness of Cloud architectures: Cloud servers are not designed to consider the context and environment of the end user (weather, temperature, schedule, traffic, etc.). However, taking this type of information into account could improve the operation of deployed applications, including security [17].
3. The high bandwidth usage associated with this model: The transport of data to the Cloud servers necessarily implies a significant use of the available bandwidth in Backhaul and Core communication networks. This leads to a considerable additional cost (ecological and economic). This system, therefore, has limitations in terms of resource usage.
4. The potential security issues of this architecture: Sharing data with a remote Cloud Operator necessarily raises security and privacy concerns for the end-user at the network and server level. Moreover, the Cloud approach is centralized, and the risks of Denial-of-Service (DoS) attacks are high, as has already been demonstrated by numerous examples [18].

This is why new data processing architectures have been proposed to address the limitations of the Cloud model. The most widely considered solution today is Edge Computing [9]. The central idea of the Edge approach is to deploy Edge Servers as close as possible to users (cf. Figure 2). Such an approach could offer similar capabilities to Cloud environments in terms of storage/communication/computing as well as flexibility (hardware virtualization, simplified access to resources, etc.) while simultaneously bringing benefits associated with the local edge servers' deployment [10]: high responsiveness, real-time data processing, limitation of the Backhaul and Core network load, etc.

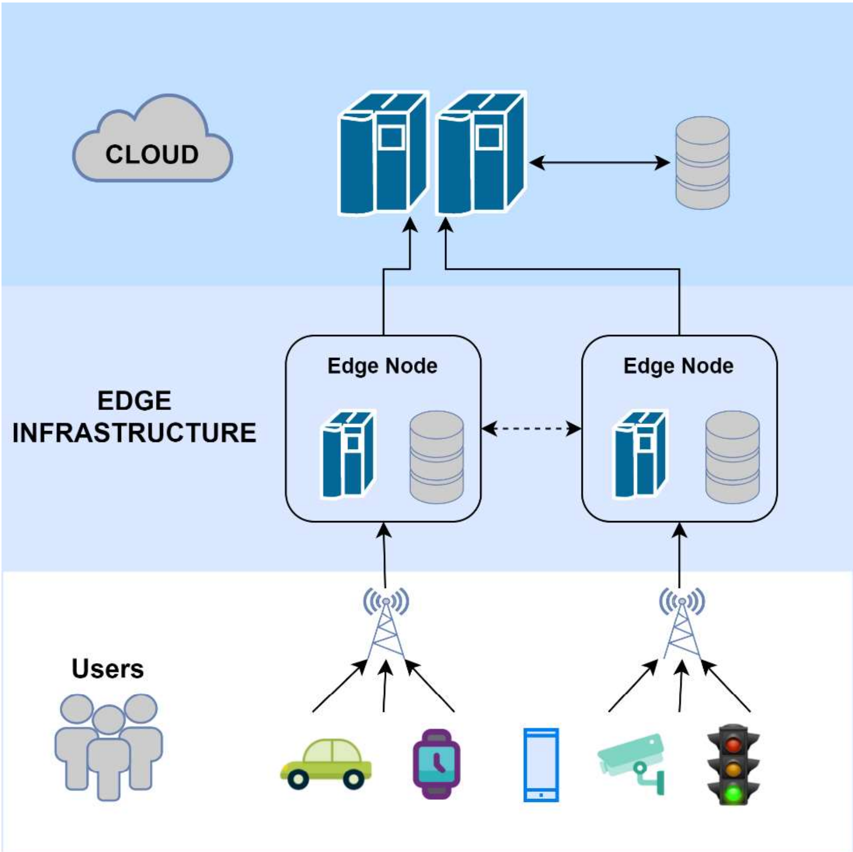

Figure 2: Edge Computing VS Cloud Computing

We can note that in the industrial world and the scientific literature, different technologies, including in the Edge Computing paradigm, have been proposed [19]:

- Multi-access Edge Computing (MEC) [20]: The idea of Multi-Access Edge Computing servers is mainly driven by Telecom operators. The principle associated with MEC consists in deploying computing servers at the level of (or a hop from) base stations. Thus, these operators could also be able to offer efficient edge services to end-users.
- Fog Computing [21]: This term was first introduced by Cisco. The idea is to use the processing and storage capacity available in the network equipment (routers for example) to offer services to end users. This means leveraging the existing infrastructure while bringing processing closer to the end users.
- Cloudlet – Micro Cloud Computing [22]: This concept was first proposed by Carnegie Mellon University, and a consortium was formed to develop it (Open Edge Computing Initiative including Intel, Nokia, Microsoft or Vodafone). The idea associated with Cloudlet is to deploy third-party small data centers one hop away from users (WLAN or LAN network). This could therefore be an interesting way for Cloud Operators to provide users with low-latency services.
- Mist Computing [23]: The idea is to use the computing/storage capacity of terminal devices (microcontrollers/microchips) to process the data of the surrounding sensors/actuators that do not necessarily have computing capacity. The application of this approach still needs to be improved. It aims at completing the solutions previously described (MEC, Fog, Cloudlet) in a highly resource-constrained environment (storage, computing, or even energy).

Thus, in a general context, many implementations of the Edge Computing idea have already been proposed (Fog, MEC, Cloudlet, Mist). These approaches aim to take advantage of the existing infrastructure, potentially adding compute servers to offer data processing services closer to the users.

3. EDGE COMPUTING ARCHITECTURES FOR C-ITS APPLICATIONS

In this section, we compare the different strategies proposed so far for implementing Edge Computing in vehicular networks. More specifically, we identify the devices that could be used to integrate this solution in the vehicular communication architecture (cf. Section 3.1) and the potential advantages/disadvantages of such implementations (cf. Section 3.2).

3.1. Potential deployment for Edge Computing architectures

Based on 1) the main principles of Edge Computing described in Section 2.2, 2) the implementations existing in other environments (MEC, Fog, Mist, Cloudlet) and 3) the characteristics of the vehicular architecture described in Section 2.1, the main Edge implementations considered so far for C-ITS services are (cf. Figure 3):

- Vehicle Edge Computing (**VEC**): This approach has been considered in many articles, such as [24, 25]. This is the lowest level that can be considered for implementing Edge Computing in vehicular networks. The idea of the VEC is to use vehicles' storage/computational capabilities to provide Edge Services to surrounding vehicles and devices with a more limited computing capacity. Thus, vehicles with significant processing capacity provide it to neighbouring vehicles. It is a direct exchange of services based on vehicle-to-vehicle communications.
- Roadside Infrastructure Edge Computing (**REC**): This idea has been considered by different authors in articles such as [26, 27]. The idea here is 1) to use the computing/storage capacity available within the RSUs or 2) to position computing servers at the RSUs level or as close as possible to them. Therefore, these RSUs can provide Cloud services closer to the users. This approach leverages the vehicle-specific communication infrastructure to provide Edge services to vehicles through Vehicle-to-Infrastructure communications.
- Base station Edge Computing (**BEC**): This approach is the most considered in the literature as it is one of the Edge approaches described in Section 2.2 and supported by the Telecom Operators (MEC) [28, 29]. The calculation servers are added at the Base Station level or as close to it as possible to guarantee low latency and high processing capacity. The operator thus becomes able to offer Edge Services to vehicles via Vehicle-to-Network communications.
- Distant Edge Computing (**DEC**): This is another type of deployment that has so far been considered mainly in the context of projects such as the InDiD pilot project [30]. The idea here is to deploy Edge servers close enough to the users and accessible by both RSUs and Base Stations. This could optimize the cost of edge server deployment while enabling low-latency services to be offered to end users.

Thus, four main applications of the Edge Computing paradigm have so far been considered in the vehicular environment: VEC, REC, BEC and DEC. These allow servers to be deployed at different levels of the network architecture: at the vehicle level (VEC), at the RSU level (REC), at the Base Station level (BEC) or within the communication network itself (DEC).

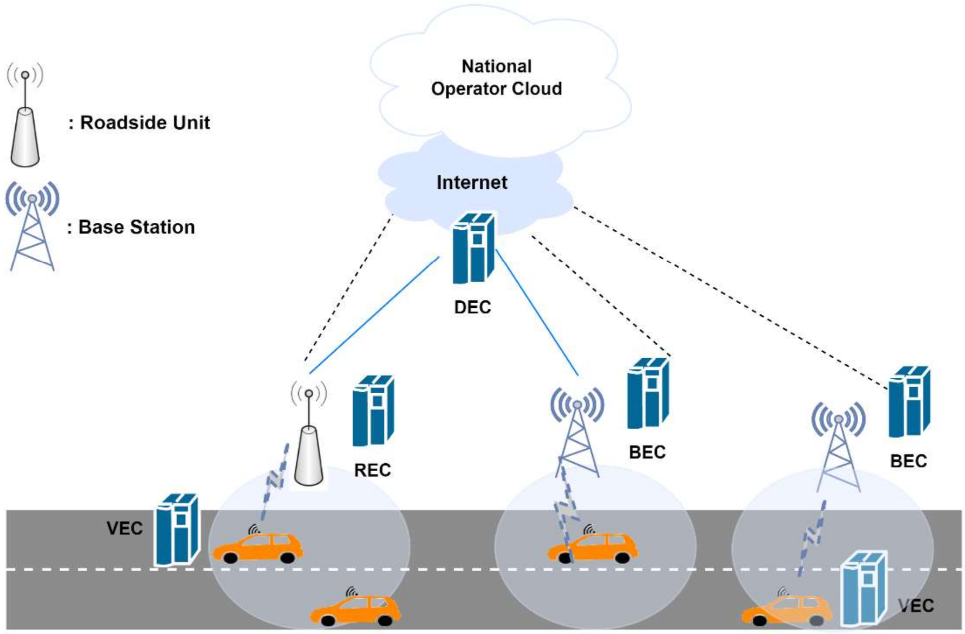

Figure 3: Presentation of the potential deployments of Edge Servers for C-ITS

3.2. Comparison of existing solutions: benefits and drawbacks

Since they are implemented at different levels of the vehicular communication architecture, the VEC, REC, BEC, and DEC approaches necessarily have different performance levels as well as different benefits and drawbacks. Considering one by one the solutions described in the previous section, their advantages/disadvantages seem to be the following (see Table 2):

- VEC: The idea of leveraging the processing capacity of neighbouring vehicles to perform data processing has two main advantages: 1) it guarantees high availability since, even in the absence of communication infrastructure, a vehicle can request support from neighbouring vehicles, and 2) it guarantees ultra-low latency owing to the small distance between vehicles and the low latency of vehicle-to-vehicle communication links. Moreover, this solution does not involve any additional cost since it is based on an existing computing capacity. However, the capacity of these vehicles will necessarily be limited, and the relevance of such a solution for high-demand services still needs to be verified. Moreover, it seems complex to implement such a solution in high-mobility scenarios. It mainly appears to be suitable for low-mobility scenarios (e.g. traffic jams).
- REC: The idea of relying on RSUs to provide computational capacity to vehicles seems to offer two main advantages: 1) higher computational/storage capacity than vehicles and, therefore, the ability to support more demanding services and 2) very low latency guaranteed by ITS-G5 communications. We can also add to this the fact that if this solution relies on the capacity of RSUs, without adding servers, its deployment cost will be low. However, the deployment of RSUs is still limited, and the accessibility of such services remains to be evaluated. The bandwidth available with ITS-G5 technology is also limited. Similarly, communication between RSUs, even if considered in some works such as [31], does not exist for the moment. Implementing global Edge solutions using RSUs, also remains a complex task.
- BEC: Using Base Stations to deploy Edge services, relying on low-latency 5G communication networks, seems to offer the same advantages as an RSU-based deployment: low latency and high computational capacity. Another advantage of this solution is the large-scale deployment of 5G base stations (important availability) and

the high bandwidth they can offer. Therefore, this solution seems able to meet the requirements of all C-ITS applications. However, it implies a dependency on the Telecom Operators, which is not very realistic in a context of vehicular communications aiming to guarantee road safety.

- DEC: Deploying Edge servers accessible by both RSUs and Base Stations seems to offer many advantages, such as having a large processing capacity accessible via different networks (high availability, high bandwidth) as well as the possibility to optimize the deployment of servers according to both the RSU and Base Station infrastructure (cost optimization). The dependency on the Telecom operator in such an architecture could also be limited. However, this necessarily implies deploying these servers at a more distant location in the network, thus increasing latency. In addition, sharing Edge servers among different network operators could lead to higher security risks, and the implementation of such an architecture could be complex on a large scale.

Thus, as seen in Table 2, each of the Edge Computing approaches described in this section (VEC, REC, BEC, DEC) has advantages and disadvantages. Communication latency, computational/storage capacity and availability are the three differentiating criteria between these different solutions.

Approach	Advantage	Drawbacks
VEC	Ultra-Low Latency Potentially High Availability Zero Cost Deployment	Limited resources (Computation/Storage) High mobility
REC	Important Resources (Computation/Storage) Ultra-Low Latency Potentially Zero Cost Deployment	Limited Bandwidth Potentially Limited Availability
BEC	Important Resources (Computation/Storage) Low Latency High Bandwidth	Operator-Dependency
DEC	Very important resources (Computation/Storage) High Bandwidth High availability Potentially Optimized Cost Deployment	Higher Latency Difficult Integration Potential Security Issues

Table 2: Comparison of the different Edge Computing Solutions for Vehicular Networks

4. FUTURE DIRECTIONS FOR DATA PROCESSING ARCHITECTURES FOR C-ITS

Numerous solutions are currently being considered for deploying Edge Computing solutions in vehicular networks to enable future C-ITS services. In this section, we highlight some future directions that must be considered to make deploying such services a reality.

4.1. Multi-layer architectures

The VEC, REC, BEC, and DEC architectures presented in Section 3 each appear to have the potential to provide benefits and play an important role in the deployment of future C-ITS services. As a result, it seems important today to propose new, multi-layer architectures that combine these different Edge Computing approaches (e.g. a REC+VEC-based architecture). This would allow the implementation of a data processing architecture 1) with a high level of availability (in the presence or not of the infrastructure), 2) designed for applications with different latency requirements (ultra-low to low), 3) with a high computing/storage capacity, 4) with an optimized deployment cost. This idea of multi-layer architecture has already been studied in other contexts, particularly for connected devices [32, 33]. Some projects such as InDiD [30] have also introduced it in the C-ITS environment. However, this approach is still in its infancy and there seems to be a lot of work in this area to ensure multi-layer edge servers' orchestration.

4.2. Multi-operators architectures

This second issue is indirectly related to the previous one. Considering multiple layers of Edge Computing or combining different Telecom operators to guarantee a wide 5G coverage and continuity of Edge services will ultimately appear necessary. Indeed, without implementing one of these solutions, it will not be possible to guarantee a high availability of C-ITS services and to ensure the safety of road users. This is why, over time, Edge architectures for vehicular networks will rely on a set of operators, each providing Edge services. However, some services require continuity and migration between several edge servers [34].

Similarly, users can simultaneously request services from different edge servers [35]. Therefore, it seems essential to implement protocols/services that will allow these different operators to coordinate themselves to guarantee the continuity of the services requested by the users. However, this idea has been little consideration for the moment in the literature, and there seems to be a lot of work in this area as noted in [36].

4.3. Green edge computing

Minimizing the energy consumption of communication networks (including vehicular communication networks) and data processing systems seems to be an important issue. Edge Computing brings data processing closer to users. Thus, it could potentially reduce the amount of bandwidth and energy required to transport and remotely process data. Nevertheless, an architecture as complex as Edge Computing, composed of an important number of distributed Edge Servers, necessarily leads to important issues regarding services management and orchestration [37]: load balancing, mobility, etc. Inefficient management could result in significant and unnecessary energy consumption. This is why the implementation of future Edge architectures must take into account energy optimization. Therefore, it seems important today to pursue ongoing work to optimize the energy consumption of Edge Computing architectures [38]. It could also be important to evaluate the impact of these distributed architectures in terms of overall energy consumption.

4.4. Security and privacy

The distribution of computation in Edge architectures seems to eliminate the Denial-of-Service issues previously mentioned for Cloud architectures, even if the Edge orchestrator remains vulnerable. However, the distribution of servers in heterogeneous devices (vehicles, RSUs, Base Stations, Routers, etc.), possibly managed by different operators, seems to necessarily open the door to numerous attacks [39], whether against WiFi interfaces (e.g. sybil, spam), hardware (e.g. side channels) or software (e.g. malware). In addition, independent nodes (vehicles) and edge service providers could be malicious and seek to retrieve/modify the data transmitted to them. Therefore, it seems essential to design new solutions specifically designed for this distributed environment, taking into account 1) the heterogeneity of the nodes (resources, vulnerabilities, etc.) 2) the involvement of different actors (operators, individuals, etc.) 3) the high distribution level of this architecture 4) the characteristics of the vehicular environment.

5. CONCLUSIONS

The latest applications intended for Cooperative and Intelligent Transport Systems (C-ITS), such as platooning or Ultra High Definition (UHD) maps' sharing, for example, present important requirements in terms of Quality of Service (latency, bandwidth, calculation). The traditional Cloud Computing architecture, necessarily inducing important communication delays, does not seem to be able to support such services.

This is why in this paper we focus on new data processing architectures designed for high performance and low latency applications: Edge Computing architectures. After a quick overview of the main C-ITS applications and the principle of Edge Computing, we present the main implementations of this paradigm in the vehicle environment: Vehicle Edge Computing (VEC), RoadSide Infrastructure Edge Computing (RED), Base Station Edge Computing (BEC) and Distant Edge Computing (DEC). We also highlight the main advantages and disadvantages of these different approaches focusing on four parameters: the cost of deployment, the latency associated with the technology, the computing/storage capacity offered by the approach and its level of availability.

In this paper, we have also identified some of the future directions that could be considered for Edge Computing in vehicular networks: 1) the definition of an approach integrating VEC, REC, BEC and DEC, 2) the design of architectures/protocols involving multiple operators, 3) the implementation of Green Edge Computing services and 4) the implementation of security solutions in this environment. This paper identifies the current state-of-the-art of Edge architectures in vehicular networks and provides an interesting basis for future work in this area.

REFERENCES

[1] Sjoberg, K., Andres, P., Buburuzan, T., & Brakemeier, A. (2017). Cooperative intelligent transport systems in Europe: Current deployment status and outlook. IEEE Vehicular Technology Magazine, 12(2), 89-97.
[2] Maaloul, S., Aniss, H., Mendiboure, L., & Berbineau, M. (2022). Performance Analysis of Existing ITS Technologies: Evaluation and Coexistence. Sensors, 22(24), 9570.
[3] Aniss, H. (2016, June). Overview of an ITS project: SCOOP@ F. In International Workshop on Communication Technologies for Vehicles (pp. 131-135). Springer, Cham.
[4] Massow, K., Kwella, B., Pfeifer, N., Häusler, F., Pontow, J., Radusch, I., ... & Haueis, M. (2016, November). Deriving HD maps for highly automated driving from vehicular probe data. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) (pp. 1745-1752). IEEE.
[5] Balador, A., Bazzi, A., Hernandez-Jayo, U., de la Iglesia, I., & Ahmadvand, H. (2022). A survey on vehicular communication for cooperative truck platooning application. Vehicular Communications, 100460.
[6] Baras, S., Saeed, I., Tabaza, H. A., & Elhadef, M. (2018). VANETs-based intelligent transportation systems: An overview. In International Conference on Ubiquitous Information Technologies and Applications, International Conference on Computer Science and its Applications (pp. 265-273). Springer, Singapore.
[7] Maaloul, S., Aniss, H., Kassab, M., & Berbineau, M. (2021). Classification of c-its services in vehicular environments. IEEE Access, 9, 117868-117879.

[8] Jararweh, Y., Doulat, A., AlQudah, O., Ahmed, E., Al-Ayyoub, M., & Benkhelifa, E. (2016, May). The future of mobile cloud computing: integrating cloudlets and mobile edge computing. In 2016 23rd International conference on telecommunications (ICT) (pp. 1-5). IEEE.
[9] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE internet of things journal, 3(5), 637-646.
[10] Mendiboure, L., Chalouf, M. A., & Krief, F. (2019). Edge computing based applications in vehicular environments: Comparative study and main issues. Journal of Computer Science and Technology, 34(4), 869-886.
[11] Gao, L., Li, Y., Misener, J. and Patil, S. (2017). C-V2X based basic safety related ITS spectrum requirement analysis. IEEE 86th Veh. Technol. Conf. (VTC-Fall), 1-5.
[12] 5GAA, Germany. (2019). C-V2X Use Cases: Methodology: Examples and Service Level Requirements White Paper.
[13] Fernandez, A. E., Fallgren, M., and Brahmi, N. (2019). 5GCAR scenarios, use cases, requirements and KPIs. 5GCAR, Germany, Tech. Rep. 5GCAR/D2.1.
[14] 3GPP TR 22.886 (2018). Technical Specification Group Services and System Aspects; Study on Enhancement of 3GPP Support for 5G V2X Services (Release 16).
[15] Kim, W. (2009). Cloud computing: Today and tomorrow. J. Object Technol., 8(1), 65-72.
[16] Islam, M. M., Morshed, S., & Goswami, P. (2013). Cloud computing: A survey on its limitations and potential solutions. International Journal of Computer Science Issues (IJCSI), 10(4), 159.
[17] Grimm, D., Stang, M., & Sax, E. (2021). Context-Aware Security for Vehicles and Fleets: A Survey. IEEE Access.
[18] Bhardwaj, A., Mangat, V., Vig, R., Halder, S., & Conti, M. (2021). Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions. Computer Science Review, 39, 100332.
[19] Ketu, S., & Mishra, P. K. (2022). Cloud, fog and mist computing in IoT: an indication of emerging opportunities. IETE Technical Review, 39(3), 713-724.
[20] Shahzadi, S., Iqbal, M., Dagiuklas, T., & Qayyum, Z. U. (2017). Multi-access edge computing: open issues, challenges and future perspectives. Journal of Cloud Computing, 6(1), 1-13.
[21] Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015, November). Fog computing: Platform and applications. In 2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb) (pp. 73-78). IEEE.
[22] Babar, M., Khan, M. S., Ali, F., Imran, M., & Shoaib, M. (2021). Cloudlet computing: recent advances, taxonomy, and challenges. IEEE Access, 9, 29609-29622.
[23] Uehara, M. (2018). Mist computing: Linking cloudlet to fogs. Computational Science/Intelligence and Applied Informatics, 201-213.
[24] Feng, J., Liu, Z., Wu, C., & Ji, Y. (2017). AVE: Autonomous vehicular edge computing framework with ACO-based scheduling. IEEE Transactions on Vehicular Technology, 66(12), 10660-10675.
[25] Sun, Y., Guo, X., Song, J., Zhou, S., Jiang, Z., Liu, X., & Niu, Z. (2019). Adaptive learning-based task offloading for vehicular edge computing systems. IEEE Transactions on vehicular technology, 68(4), 3061-3074.
[26] Ning, Z., Huang, J., Wang, X., Rodrigues, J. J., & Guo, L. (2019). Mobile edge computing-enabled Internet of vehicles: Toward energy-efficient scheduling. IEEE Network, 33(5), 198-205.
[27] Javed, M. A., & Zeadally, S. (2021). Ai-empowered content caching in vehicular edge computing: Opportunities and challenges. IEEE Network, 35(3), 109-115.
[28] Wang, G., & Xu, F. (2020). Regional intelligent resource allocation in mobile edge computing based vehicular network. IEEE Access, 8, 7173-7182.
[29] Al-Badarneh, J., Jararweh, Y., Al-Ayyoub, M., Fontes, R., Al-Smadi, M., & Rothenberg, C. (2018). Cooperative mobile edge computing system for VANET-based software-defined content delivery. Computers & Electrical Engineering, 71, 388-397.
[30] InDiD Pilot Project – URL: https://www.c-roads.eu/pilots/core-members/france/Partner/project/show/indid.html
[31] Su, Z., Hui, Y., Xu, Q., Yang, T., Liu, J., & Jia, Y. (2018). An edge caching scheme to distribute content in vehicular networks. IEEE Transactions on Vehicular Technology, 67(6), 5346-5356.
[32] Wang, P., Yao, C., Zheng, Z., Sun, G., & Song, L. (2018). Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems. IEEE Internet of Things Journal, 6(2), 2872-2884.
[33] Wang, P., Zheng, Z., Di, B., & Song, L. (2019). HetMEC: Latency-optimal task assignment and resource allocation for heterogeneous multi-layer mobile edge computing. IEEE Transactions on Wireless Communications, 18(10), 4942-4956.
[34] Wang, S., Xu, J., Zhang, N., & Liu, Y. (2018). A survey on service migration in mobile edge computing. IEEE Access, 6, 23511-23528.
[35] Ksentini, A., & Frangoudis, P. A. (2020). Toward slicing-enabled multi-access edge computing in 5G. IEEE Network, 34(2), 99-105.
[36] Sylla, T., Mendiboure, L., Maaloul, S., Aniss, H., Chalouf, M. A., & Delbruel, S. (2022). Multi-Connectivity for 5G Networks and Beyond: A Survey. Sensors, 22(19), 7591.
[37] Lyu, X., Tian, H., Jiang, L., Vinel, A., Maharjan, S., Gjessing, S., & Zhang, Y. (2018). Selective offloading in mobile edge computing for the green internet of things. IEEE network, 32(1), 54-60.

[38] Hou, W., Ning, Z., & Guo, L. (2018). Green survivable collaborative edge computing in smart cities. IEEE Transactions on Industrial informatics, 14(4), 1594-1605.
[39] Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J., & Lv, W. (2019). Edge computing security: State of the art and challenges. Proceedings of the IEEE, 107(8), 1608-1631.