Edge-aware Cloud-native Service for Enhancing Back Situation Awareness in 5G-based Vehicular Systems

Nina Slamnik-Kriještorac, F. Zarrar Yousaf, Girma M. Yilma, Rreze Halili, Marco Liebsch, and Johann M. Marquez-Barja

Abstract*—In the public safety sector, 5G offers immense opportunities for enhancing mission-critical services by provisioning virtualized service functions at the network edge, which enables achieving high reliability and low-latency. One of these mission-critical services is Back Situation Awareness (BSA) that supports Emergency Vehicles (EmVs) by increasing awareness about them on the roads. In this paper, we introduce an on-demand BSA application service, which has been developed for multi-domain Multi-Access Edge Computing (MEC) systems, enabling early notification for vehicles on the Estimated Time of Arrival (ETA) of an approaching EmV. The state-of-the-art approaches inform civilian vehicles about EmVs only when they are in a close proximity (up to 300m). However, in some situations (e.g., in congested areas), this may not be enough for the civilian vehicles to safely and timely maneuver out of the lane of an EmV. Our approach is, to the best of our knowledge, a unique way to significantly extend this awareness by creating an orchestrated 5G-based MEC deployment of BSA application service on optimally selected edges, thereby stretching over multiple edge domains and even countries. While consuming the real-time location, speed, and heading of an EmV, such application service affords the drivers with sufficient time to create a clear corridor, allowing the EmV to pass through unhindered in a safe manner thereby increasing the mission success. The detailed design and the performance analysis of the BSA application service that has been created following modern cloud-native principles based on Docker and Kubernetes, is presented in terms of the impact of emergency scale on the MEC system resources and service response time. Moreover, we also introduce a metric called panic indicator, which depicts how the proposed BSA service can potentially help in enabling drivers to calmly maneuver out of the path of an EmV, thereby increasing road safety.

***Index Terms*—MEC, 5G, multi-domain back situation awareness, NFV orchestration, vehicular communications, public safety**

I. Introduction

Multi-Access Edge Computing (MEC) and Network Function Virtualization (NFV) are considered as one of the key technology enablers for 5G and beyond [1], and MEC systems especially are leveraged for empowering applications with Ultra-Reliable and Low-latency Communication (URLLC) requirements. The flexible and agile service management features of the MEC/NFV systems have fostered new use cases and business models that were inconceivable with the previous generations of mobile network systems. Thus, in this paper, we present and evaluate an on-demand Back Situation Awareness (BSA) application service, which has been designed and developed for multi-domain MEC systems, to in-advance inform vehicles on the roads about an approaching Emergency Vehicle (EmV), with the ultimate goal of decreasing the overall response time of emergency responders. In the context of public safety, the high-level overview of the BSA use case is given in Fig. 1. In this scenario, MEC system is leveraged to notify the vehicles about the Estimated Time of Arrival (ETA) of an approaching EmV, whose presence is beyond the audio and visual range of those vehicles. Furthermore, to extend the range, the BSA service is dynamically made available in multiple MEC systems that might be in the same or different edge domains in order to cover the entire route-path of the EmV (Fig. 1). The edge domains might be a part of a single administrative domain or, when the emergency case happens close to the border, two administrative domains, i.e., mobile operators in different countries.

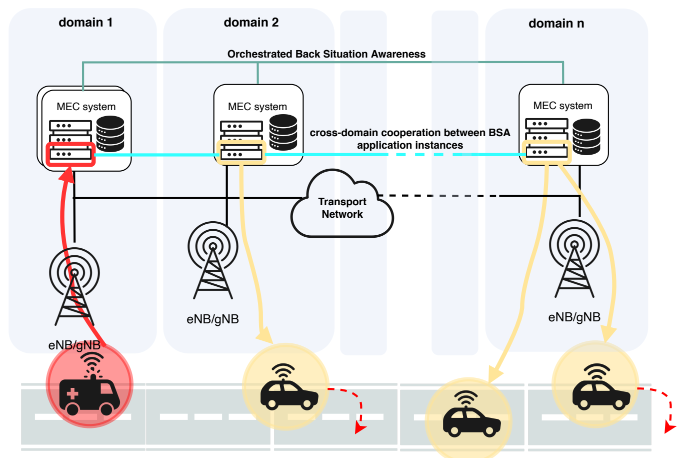


Fig. 1: Back situation awareness on the highways.

The application service is triggered upon the MEC systems receiving a notification message from an Emergency Management Authority (EMA), such as 112 (in EU) or 911 (in USA), providing the EmV ID, event location information, and the route path of the selected EmV. In response, the orchestration system selects the relevant MEC hosts along the route path and deploys the BSA service instances. In particular, such multi-domain deployment extends the range of notifications for civilian vehicles along the route path, informing them timely on the expected arrival of an EmV. The deployed application instances are then used by the dispatched EmV to periodically send Cooperative Intelligent Transport System (C-ITS) Cooperative Awareness Message (CAM) [2] towards the newly instantiated BSA application on the MEC systems (see red arrow line in Fig. 1), for EmV's each Global Positioning System (GPS) point on the road. Taking into account the EmV's ID, speed, location, and direction information, extracted from the CAM notifications, the BSA application computes the ETA values of the EmV for different

Nina Slamnik-Kriještorac, Rreze Halili, and Johann M. Marquez-Barja, are with the University of Antwerp-imec, IDLab-Faculty of Applied Engineering, Belgium. F. Zarrar Yousaf, Girma M. Yilma, and M. Liebsch, are with the NEC Laboratories Europe GmbH, Germany.

dissemination areas, which the BSA application derives along the route-path. The computed ETA values are then encoded in the C-ITS Decentralized Environment Notification Messages (DENMs) [3], which are broadcasted in the geographic regions bound to dissemination areas relevant to the encoded ETA value (see yellow arrow line in Fig. 1). In the following, we denote the distribution of DENMs in the dissemination areas as geo cast.

All the vehicles in the dissemination area shall decode the received DENM notifications to have the ETA values displayed to the driver. This process is repeated each time a CAM is received by the BSA service. For range extension, the BSA service that is directly receiving the CAM notifications from the EmV will forward the EmV's state/metadata information to the peering BSA service instances that are instantiated on the corresponding MEC hosts along the route path, in order to compute ETA values for the dissemination areas within their domain coverage. In other words, a federated multi-domain BSA service is created spanning over multiple MEC hosts. In addition, this multi-domain deployment is supported by BSA applications as they are edge-aware. This feature makes MEC applications aware of i) the edge location and the geographic area that they serve, as they can proactively inform orchestration entities about the need for an application instance in the other edge domains, as well as of ii) the other peering applications from the other edge domains to which they need to connect. The edge awareness also applies to the deployment of such MEC applications, as orchestration entities consider the availability of the network and computing resources in edge domains, as well as the location of MEC hosts, while making decisions on the application service deployment.

It is intuitive that decreasing the response time of emergency responders leads to a larger probability of successful interventions, and there are studies that assess the average response time of emergency responders [4–6], and how such a response time affects the success of emergency interventions and patients' mortality. Looking from a more technical perspective, there are also various approaches that leverage digital technologies and services to broadcast the information about the presence of an EmV on the roads, but they utilize the short-range Vehicle-to-Vehicle (V2V) communication that sends the required information about an EmV only in close vicinity from this EmV [7,8], thereby only addressing those vehicles that are approximately 300m away from them [9]. Thus, the V2V coverage of 300m is not enough for addressing emergency situations in an efficient manner by sending in-advance notifications, as emergency journeys are usually kilometers long and EmVs are moving at a high speed reducing the reaction time of the drivers. For example, the observational cohort study with 10,315 cases transported by four English ambulance services [6] reported that ambulance journey distances ranged from 0 to 58km (with a median of 5km). One effort to extend the awareness is given by Moroi and Takami [10], who propose a Vehicle-to-Infrastructure (V2I) approach, but this is still not enough given that the transmission range of roadside units is between 400 and 500m, with the average delay in message transmission of 487 ms and 574 ms [11].

To address the aforementioned gaps in existing approaches, our BSA system relies on the Vehicle-to-Network (V2N) communication, i.e., 5G-based MEC deployment where BSA application is running on the optimally selected edge cloud. Given that information such as the current location/speed of a vehicle needs to be timely delivered to the BSA application via CAM message updates, the longer uplink latency can significantly affect the efficiency of the V2X application service, i.e., the accuracy of estimating ETA values in case of the BSA application service. Further delays in such communication will produce more errors in the estimation of EmV's time of arrival. Thus, it is important that for our BSA deployment, an optimal MEC is selected taking into account both the computing and network resource availability, so that low-latency and high-reliability can be achieved.

This advance notification of the EmV's ETA shall afford the drivers enough time to calmly maneuver in a safe manner, i.e., without panicking, to create a clear corridor for the EmV to pass through unhindered, thereby enabling the EmV to reach the event location in time, enhancing mission success and road safety. However, in MEC systems, the multiple MEC applications are sharing a very limited pool of resources, and therefore it is important to understand the resource metering of MEC applications before they are hosted on the MEC platforms, in order to avoid degraded Quality of Service (QoS) of the respective MEC application and/or its adverse impact on other services due to extensive resource consumption during high load circumstances.

For the ETA algorithm as a part of our BSA application service, calculating ETA values and defining areas for ETA dissemination along the road, we conducted a detailed analytical analysis in our previous work [12], assessing the ETA accuracy and error estimation. In this paper, we focus on evaluating the overall functional and operational feasibility of the BSA service in a real environment, its management and service performance, and analyze i) the overall response time to emergency events, studying all the contributing factors, as well as ii) the impact of the BSA service on the MEC computing resources that will aid the service designer in deriving MEC system specifications for reliable hosting of this critical service. The experimental setup is created in a realistic environment, where we deployed the BSA application instances on top of the MEC hosts within an orchestrated vehicular system, i.e., the Smart Highway[1] testbed. This paper also introduces a new Key Performance Indicator (KPI) referred to as *panic indicator* indicating the level of panic experienced by the driver when notified of the ETA of an approaching EmV, and analyzes the factors for reducing the panic to ensure safe passage of the EmV(s) towards its destination. This indicator is determined by comparing the current ETA, and the difference between two successive ETA values, with the two thresholds. These thresholds are subjective as they depend on the drivers' perception, but the goal is to provide the notion of how panic can be preempted by MEC applications that assist vehicles on the road, in order to improve the efficiency of reaction of civilian vehicles to the arrival of an EmV. Therefore, the two main objectives of the proposed BSA system are i) to reduce the overall response time of emergency responders, and ii) to reduce the panic among drivers in the presence of emergency responders on the roads.

From the results that we obtained, we derive important conclusions on i) the design requirements of V2X services that are aimed for running on the MEC platforms in the 5G systems, with the goal to assist vehicles on the highways, and ii) the operations and management of such services, including the study of the factors that affect the service performance.

The rest of the paper is organized as follows. Section II gives an overview of the relevant related work. Section III presents the system design, the BSA service architecture, and the multi-domain aspects of BSA operation. In Section IV, we detail the testing methodology for the BSA application. This is followed by Section V providing detailed performance results, and analysis and discussion, based on an experimental setup. The paper finally concludes with Section VI.

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

II. Related Work

By exploring the opportunities of technological advancements, the Emergency Medical Service (EMS) providers are making an effort to optimize the use of existing resources and to offer high-quality medical services to patients. Some of the key goals of EMSs are reducing patient mortality, preventing disability, and improving chances of recovery [5]. Paving the way towards these goals, several studies have been conducted over the past years, examining the relationship between the reaction time, and the survival rate, thereby highlighting the significance of reducing the overall response time to emergency events. According to Sánchez-Mangas et al. [5], the probability of death decreases by one-third if there is a 10min reduction in the emergency response time. Furthermore, Vukmir [4] shows that 30min time is the upper time interval for the survival of a cardiac arrest patient. Hence, reduction of the response time plays a pivotal role in the emergency situations [4,5]. Another study, provided by Iannoni et al. [13], shows how the extensions of the hypercube model, combined with the hybrid genetic algorithms, optimize the configuration and operation of EMS on the highways. Considering the locations of the ambulance bases along the highway, Iannoni et al. [13] show how to minimize i) the average user response time, ii) imbalance of the ambulances workloads, and iii) the fraction of calls not serviced within a predetermined threshold. The study shows that the above-stated issues can be mitigated by relocating the ambulance bases, and simultaneously determining the sizes of district areas in the system.

As indicated in our previous work [12], our proposed solution has several advantages compared with the state-of-the-art. The existing studies to predict travel times and investigate reliable routing optimizations use the uncertainty and time-dependent distribution of the speed (its mean and variance value respectively) [14,15]. This approach is considered complex and time-consuming for real implementation [14]. Our solution overcomes such a challenge as the BSA service is constantly updated with the real-time speed and position information from the EmV, which ensures the accurate and real-time observation of speed and travel times. Also, the BSA service ensures in-advance notifications of the EmV's arrival time, thereby expecting the vehicles in front to clear the required lane for the EmV to pass. In this situation, the proposed system can always select the shortest path, regardless of the traffic conditions, time of day/year, or other impact factors. In addition, in the existing studies [16,17], the historical data sets for travel time predictions are obtained from the large capacity historical dataset volumes maintained by the emergency medical institutions to predict arrival times of ambulances in different parts of the cities/urban/rural areas. In our solution, the historical data set is constantly updated using the received CAMs from the EmV. Therefore, we have updated accurate average speed values for 24 hours of the day, thus identifying rush hours, busy areas within cities, weather conditions, congestion, and ridership [12].

Nowadays there is a strong focus on the use of vehicular systems and the Vehicle-to-Everything (V2X) applications that are developed to improve safety on the roads [18,19], thereby enhancing the situation awareness [20]. A study conducted by Senart et al. [21] presents the need for broadcasting awareness messages and the dissemination of Time of Arrival (ToA) of emergency vehicles using a wireless medium. The idea is to disseminate information on EmV's arrival time to other vehicles and to have real-time feedback at the same time, in case the quality of the communication is degraded. Using the feedback information, the EmV becomes aware of those vehicles in front that have not been warned about its arrival yet, thus it accordingly slows down. The bottleneck of such an approach is that it increases the response time of an EmV. Another approach on this topic is also found in the work presented by Hadiwardoyo et al. [7] and by Metzner and Wickramarathne [8], where they propose a V2V application for disseminating the real-time information on the EmV's location, and the route path, in order to inform civilian cars about EmV's arrival. However, the short-range communication offered by V2V is not sufficient for extending the range of situation awareness. One attempt to facilitate the aforementioned issue is provided by Moroi and Takami [10], who propose using V2I communication, i.e., making use of Roadside Units (RSUs) installed along the road. However, the limited range provided by the V2I communication still lacks to comply with the requirements for vehicular applications that tackle multiple domains, e.g., distributed to different countries [18]. However, in such an approach, information about arriving EmV is only shared with the vehicles in close proximity to EmV, i.e., not in a larger region to increase awareness about EmV, which provides drivers with enough time to change their maneuver.

The efforts to extend the range of notifications by utilizing cellular technologies are presented by Shah et al. [18], where the use of 5G systems and MEC to support vehicular use cases is presented, thereby decreasing the delay in communication by deploying vehicular applications at the network edge. The involvement of cellular technologies to extend awareness on the roads comes as a solution to the limitations imposed by Dedicated Short Range Communication (DSRC) based on IEEE 802.11p technology [22]. In particular, the DSRC is seen as not capable to overcome the challenging conditions on the highways (e.g., high user mobility, high density of users) because of the short-range coverage, inefficient congestion control, and insufficient reliability [18,22]. On the other hand, cellular technologies are characterized by a larger coverage range, high network capacity, and technological maturity [22]. Despite the aforementioned benefits, the centralized control in cellular networks causes additional delay against the strict delay requirements of safety vehicular applications [22,23]. Thus, it is extremely important to carefully design the vehicular system and to consider what technologies are suitable for a specific use case.

In the research [24,25], 5G New Radio (5G NR) is recognized as an enabler of ultra-low latency and high reliability of network services. In particular, 5G NR supports a Uu interface in LTE and 5G, which is used for the transmission and reception of V2X messages over cellular infrastructure [25]. However, Wang et al. [26] point out that the use of the Uu interface is not always sufficient for the requirements of the V2X services, and they emphasize the importance of bringing those services closer to the users, i.e., at the network edge. Therefore, in our previous work [12], we studied the benefits of using 5G and MEC as a solution for supporting BSA in V2X scenarios, with the focus on the algorithm that estimates the time of arrival of an EmV, which is further disseminated to civilian vehicles along the road with a long-range distance. In this paper, we present the innovative edge-aware MEC application service that is developed for enhancing back situation awareness on the highways, relying on the 5G technology to provide connectivity to vehicles, thereby going beyond existing approaches that are short-range-based. Furthermore, we present the operational aspects of such service in a multi-domain deployment, thus, enabling the extended awareness about the EmV along the road that spans several edge domains.

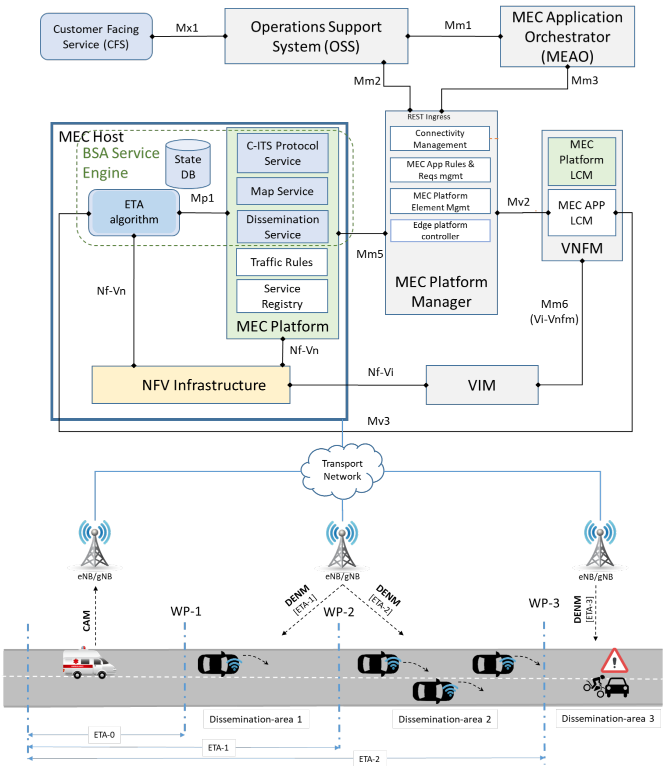


Fig. 2: Overview of the BSA service system design.

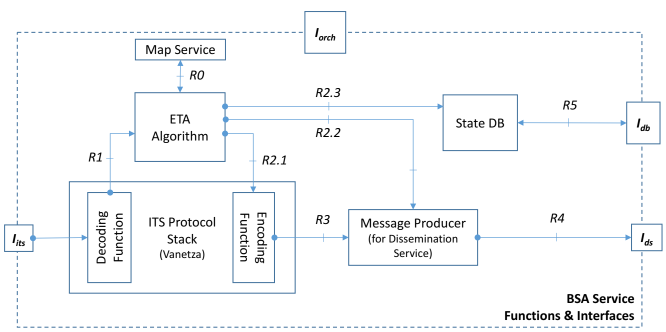


Fig. 3: Back Situation Awareness (BSA) Service Architecture - Functional Elements and Interfaces.

III. BSA Application - System Design and Architecture

In this section, we provide an overview of all service components that BSA application comprises and discuss the design principles as well as the functional architecture of the application. This is, to the best of our knowledge, the first attempt to fully design and develop a micro-service based MEC application, and a system around it, for a V2X use case addressing emergency situations on the roads, and later in Section IV to evaluate its performance in a realistic environment. Here we also detail the management aspects and operational feasibility of the BSA application service, stretching over multiple domains while being orchestrated by an optimized MEC orchestration system.

A. BSA Application Overview

Fig. 2 shows the system overview of the BSA application in the context of the standardized ETSI MEC system architecture [27], and also depicts it in the context of the use case illustrated in Fig. 1. It should be noted that Fig. 2 only shows functional elements and reference points that are relevant to the BSA system in order to reduce the complexity and improve readability.

The proposed BSA application, shown in Fig. 2, comprises of the following key components that are realized as independent and loosely coupled microservices:

1) **ETA Algorithm:** - This component is at the core of the BSA application, embodying the logic of assigning GPS Waypoints (WPs) on the EmV's route-path, and then computing ETA values from the speed, location, and direction information of the EmV encoded in the C-ITS CAM notifications [2] received periodically from the EmV with reference to these WPs. Such a mechanism of proactive notifications allows the drivers to deduce the maneuver recommendations. The ETA analytical model is further explained in Section IV-B1, while the detailed logic, including the analytical model and the performance evaluation of the ETA algorithm compared against the state-of-the-art approaches, have been described and analyzed in our previous work [12] in terms of the error in the estimation of ETA values.
2) **C-ITS Protocol Service:** This is proposed to be a microservice for decoding and parsing received C-ITS awareness and notification messages (CAM/DENM), as a part of the overall BSA application. The decoded information is relevant for the ETA algorithm to derive and encode ETA values, thereby preparing them in the DENM format for notifying the vehicles. This corresponds to the C-ITS protocol stack[2] and the decoding/encoding helper function entities in Fig. 3, which will be explained in III-B.
3) **Map Service:** This is proposed to be a microservice that can be consumed by the ETA algorithm for getting geospatial information about the road where EmV is traveling, which is determined based on its current location and the destination. Knowing such route-path information/plan, ETA algorithm can specify WPs along the route-path, and also get more information on the type of road the EmV is traveling on (e.g., highways).
4) **Database (DB) service:** This is proposed to be storage where the meta-data/state-information of the EmV decoded/parsed by the C-ITS protocol service from the periodically received CAMs are stored, which are then consumed by the ETA algorithm for calculating ETA values, and optionally maneuver recommendations. This corresponds to the State DB entity in Fig. 2.
5) **Dissemination Service:** This microservice is used for disseminating the EmV's ETA information to the vehicles within the relevant dissemination areas. The region of the route path between two successive WPs characterizes a dissemination area. The overall BSA application encodes the ETA values, calculated by ETA algorithm, in a C-ITS DENM [3], which is then geocasted in the respective dissemination area with the help of the dissemination service. For example, as depicted in Fig. 2, ETA-0, ETA-1, and ETA-2, are geocasted in Dissemination Area 1, Dissemination Area 2, and Dissemination Area 3, respectively. All the vehicles that are on the route path of the EmV and going in the same direction of the EmV will process the DENM received from the dissemination service, to extract the ETA value to be displayed on the control panel (e.g., human-machine interface) of a vehicle.

[2]Vanetza: an open-source implementation of the ETSI C-ITS protocol suite: https://www.vanetza.org/

The above functional elements comprising the BSA application service are hosted in a MEC host as MEC application. The other functional elements shown in Fig. 2 are specified in the ETSI GS MEC 003 v2.1.1 standard [27] and are used for the management and orchestration of the BSA related MEC applications and MEC services as defined above. It should be noted that other external entities, for example, EMA is able to access the BSA system via the Customer Facing Service (CFS) interface. The whole communication chain is done using the 5G mobile network infrastructure, the details of which are out of the scope of this paper.

B. Design Principles and Architecture

Since 5G networks and beyond are planned to be entirely software-driven, the need for a Cloud-Native architecture and design becomes a defacto choice for Communication Service Providers (CSPs). This approach allows CSPs to deploy services rapidly and flexibly, with reduced Capital Expenditure (CapEx) and Operating Expenditure (OpEx) through network automation. Hence, the MEC systems and MEC applications also need to follow the same principles, since resources are even more limited at the MEC systems and applications are expected to have a high level of flexibility and response time with minimum possible resource requirements. Thus, container-based applications become first-class citizens for MEC platforms. Given the stringent requirements for latency bounds and uplink bandwidth for V2X applications [28], they need to run at the 5G network edge but to be eligible for such placement, their design needs to adopt the same principles as for any MEC application. Such design requires complete flexibility, with the logic being decoupled into various microservices (as described for BSA application in Section III-A), which are loosely coupled via internal interfaces. Their external interfaces are being exposed towards i) end users, i.e., vehicles, so that they can connect to the application service and send their real-time messages, ii) dissemination services, which will be used for message dissemination towards vehicles, iii) orchestration entities that orchestrate MEC applications, and dynamically receive notifications from such applications to improve their life-cycle management and iv) peering application instances deployed in other edge domains, which are used for exchanging application metadata. In this section, we describe how the loose coupling of microservices that are detailed in Section III-A is achieved for the BSA application.

In Fig. 3, the detailed overview of the functional architecture of the BSA service is depicted, showing the internal interfaces between the various functional components of the BSA service, as well as the interfaces for interfacing with external services/functions. As depicted in Fig. 2, some of the functional elements and references apply also to the **Mp1** reference, to consume shared value-added MEC services, such as Map Service. These internal interfaces and external interfaces are depicted with arrow lines, where the direction of the arrow indicates the message producer/consumer relationship. That is the functional element from where the arrow line originates is the message producer and the functional element where the arrow line terminates is the message consumer. Functional elements linked by double arrow lines are both producer and consumer. Based on this, the following *internal* interfaces are specified:

- Interface R0 – on this interface the ETA algorithm can interface with a map service for determining the route-path information of the EmV and for related information. This interface will rely on the API exposed by the map service provider.
- Interface R1 – on this interface the Decoding Function of the C-ITS Protocol Stack sends the decoded event notification message (e.g., ETSI C-ITS CAM) received from the EmV towards the ETA Algorithm block.
- Interface R2.1 – on this interface the ETA Algorithm block sends the ETA value to the Encoding Function of the C-ITS Protocol Stack, for encoding it in the event warning notification message (e.g., ETSI C-ITS DENM).
- Interface R2.2 – on this interface the ETA Algorithm block sends the dissemination area towards the Message Producer, specifying the area where the event warning notification message encoded with the ETA (and received via R3) is supposed to be disseminated.
- Interface R2.3 – on this interface, the ETA Algorithm block sends the EmV state/meta information towards the State DB.
- Interface R3 – on this interface the encoding function sends out the event warning notification message with encoded ETA values to the Message Producer for dissemination to vehicles.
- Interface R4 – via this interface, the message producer interfaces with the external Message Dissemination Service block (see Fig. 3).
- Interface R5 – via this interface, the State DB is able to exchange state/meta information with peering BSA application service in another host/domain.

Furthermore, the *external* interfaces are also specified as follows:

- C-ITS protocol Interface (Iits) – via this interface, the BSA application service receives the event notification messages from the EmV.
- Dissemination Service Interface (Ids) – via this interface, the Message producer is able to communicate with the external Message Dissemination Service.
- Orchestration Interface (Iorch) – via this interface, the orchestration system is able to perform the lifecycle management of the BSA service instance.
- State DB Interface (Idb) – via this interface, the BSA application service instances in different domains exchange state/meta information with each other over the public network infrastructure.

C. Multi-domain/cross-border operation of the BSA service

In this section, we discuss the orchestration and operation aspects of the BSA service in the multi-domain cross-border scenario. The representation of the BSA application running in a distributed multi-domain environment is shown in Fig. 1, while Fig. 2 depicts the high-level architecture of the MEC orchestration system in each of these domains, and the BSA application running on top of it. The BSA application is a type of MEC application that is designed to address a V2X use case stretching over a long corridor on the highway, and as such, it requires proper management and orchestration to achieve a smooth cross-domain operation. Thus, in Fig. 4, we provide an overview of multi-domain operations of the BSA application, which are executed in the following three phases: i) Phase 1 is in charge of application deployment in the source domain from which the selected EmV starts its journey, ii) Phase 2 continues with the dynamic deployment of peering application instances in the other domains that are affected by EmV's route towards the destination, and shows the cross-domain collaboration between application instances, and iii) Phase 3 proceeds with the termination of application instances that are not used by the EmV anymore, thereby releasing MEC resources for other types of services.

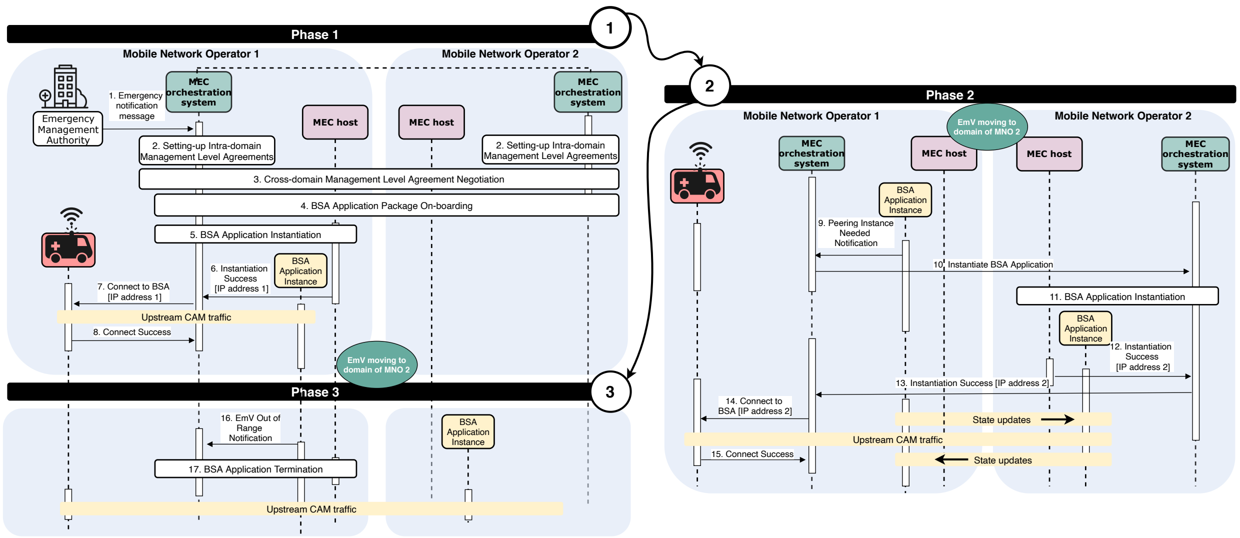


Fig. 4: Overview of multi-domain operations of the federated multi-domain BSA service; Yellow boxes imply the operations that are contributors to the KPIs we measured (e.g., upstream CAM traffic affected by communication latency, BSA application instance producing computational latency and CPU/RAM load, and state updates over network effected by state update delay).

a) Phase 1: Application deployment in the source domain: Prior to addressing the emergency situation on the road, the BSA application needs to be on-boarded and instantiated on the MEC platform, with the help of the orchestration system. As described in Section III-A, MEC applications such as BSA application, are instantiated upon the trigger received from the authorized customers/clients, for instance, public safety authorities (e.g., emergency management entity) via the customer interface (step 1, Fig. 4). Once the request is received by the MEC orchestration system, it proceeds with the application deployment (steps 2-5). In a multi-domain scenario (i.e., an inter-edge or inter-Mobile Network Operator (MNO)), the application service needs to be instantiated in multiple MEC hosts, and operated by different MNOs. In such a scenario, the application package needs to be on-boarded in all peering domains (i.e., all edge domains along the route path of the EmV) prior to application instantiation. This way, the proactive deployment of peering instances is facilitated, but it still requires certain agreements between orchestration entities in different domains to provide coordinated management and orchestration of BSA application service running in all domains simultaneously (steps 2-4, Fig. 4). Furthermore, as a part of Phase 1 in domain 1, the instantiation further proceeds with a target MEC system selection, in which the orchestration entities make optimal edge-aware decisions on resource selection and application placement, thereby taking into account: i) the real-time availability of computing resources in all MEC hosts within one edge domain, ii) geographical location of the MEC hosts, which is essential for the vehicular use cases with highly mobile users that need services to be deployed at geographically suitable MEC systems, and iii) the availability of network resources. Thus, to finalize Phase 1 in domain 1, BSA application is instantiated, and EmV is connecting to it (steps 5-8, Fig. 4). In particular, Fig. 4 hides the complexity of the MEC orchestration system, but Fig. 2 shows that it comprises as key elements an Edge Orchestration component, i.e., MEC Application Orchestrator (MEAO), and an Edge Platform Controller [29]. More details on the orchestration elements and operations are described in our work that studied collaborative orchestration for V2X services [30], explaining that the Edge Platform Controller extends the open-source container orchestration platform Kubernetes[3] to perform MEC Platform Management as well as connectivity control, based on an extension to the Container Networking Interface (CNI). This CNI extension supports Fast Data Input Output (FDIO) operations on additional and customized data plane interfaces for Kubernetes PODs[4]. Thus, the Edge Platform Controller enforces the tasks such as Life-Cycle Management (LCM) of the MEC applications. In the view of the BSA application, the additional interfaces are used for low-latency operations to receive C-ITS CAMs from an EmV (Upstream CAM traffic in Phase 1, Fig. 4), and to disseminate C-ITS DENMs to other vehicles, as described in Section III-B.

b) Phase 2: Dynamic deployment and runtime of peering application instances: The instantiation can be performed simultaneously on multiple edges in a coordinated manner between platform orchestrators, but it can be also proactively started on some specific edges in order to decrease latency in orchestration operation execution. The orchestration entities constantly monitor the deployed edge applications and allow these application instances to send notifications to orchestrators, as well as triggers for certain orchestration operations. This feature is significantly important for the orchestration platform as applications are edge-aware, and the platform can remain application-agnostic, allowing applications themselves to send application-specific triggers and start e.g., proactive BSA application deployment in the target domain, even before the EmV reconnects from the first MNO's network to the second. Thus, in Phase 2 depicted in Fig. 4, the peering BSA application is instantiated in domain 2, as per the trigger received from BSA application instance in domain 1 (steps 9-13, Fig. 4). Both application instances are edge-aware, i.e., aware of the environment where they run, and of the applications from other domains to which they need to connect. To exchange the real-time state updates about the EmV, there is a data-plane connection between peering instances, and to whichever instance the EmV is connected, it informs its peering instance about the EmV's current state (location, speed, destination), as shown in steps 14-15 in Phase 2 of Fig. 4.

[3]Kubernetes Project Portal: https://kubernetes.io/

[4]Kubernetes POD is the smallest deployable unit of computing that can be created and managed in Kubernetes.

TABLE I: System characteristics of the testbed machines.

System information	
Architecture	x86_64
CPU op-mode(s)	32-bit, 64-bit
CPU (s)	16
CPU (MHz)	1280.815
Memory	32GB
Processor	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
Storage	C610/X99 series chipset sSATA Controller [AHCI mode]
Disk	1TB Samsung SSD 860
Network	I350 Gigabit Network Connection

TABLE II: Implementation method for the BSA system.

BSA microservice	Implementation method
ETA algorithm	Python program based on SciPy & related libraries
C-ITS Protocol Service	Vanetza Open Source ETSI C-ITS Protocol Stack
Map Service	Python program based on gpxpy library that is processing external maps or Google maps
Database Service	SQLite database
Dissemination Service	Python program implementing AVRO message encoder, and interacting with Kafka REST API for disseminating DENM messages

c) Phase 3: Dynamic termination of application instances: During the application runtime, the orchestration entities make sure that application instances have a sufficient amount of resources to perform required operations (e.g., by performing scaling operations). Once the resources are not needed, such as in Phase 3 in domain 1 (steps 16-17, Fig. 4), orchestrators terminate the BSA application instance thereby permanently releasing the allocated resources. As long as the application instance in domain 2 is needed, i.e., while EmV is connected to it, Phase 3 in domain 2 represents the BSA application runtime.

IV. Testing methodology

In this section, we present the testing methodology for the BSA application, thereby i) describing the realistic experimental setup within the testbed environment, i.e., the Smart Highway testbed created for the V2X research, and ii) defining the set of metrics that reflect service performance.

A. Smart Highway testbed setup for experimental evaluation

Prior to describing the BSA system evaluation results, we explain the implementation method. To implement the BSA system, we chose a realistic vehicular environment where we deployed Docker-based BSA application instances on top of the MEC hosts that are orchestrated by Kubernetes-based orchestration system. In addition, the ETA algorithm is running as a python-based application within the BSA Docker container, which apart from the ETA algorithm also embeds the following microservices: Vanetza ITS stack (C-ITS Protocol Service), Map Service, Database service, and Dissemination service, as described in Section III-A. The implementation method for each of those microservices is presented in Table II. The whole system is implemented using the Smart Highway testbed[5], which is a test site built on top of the E313 highway, located in Antwerp, Belgium. In this realistic testbed setup, the MEC hosts are collocated with RSUs, i.e., the wireless communication devices that are installed along the road to provide connectivity to the vehicles. For instance, the map in Fig. 5 showcases the locations of seven RSUs that are installed along the highway site, and those in red boxes are the ones used in our experimental setup. The system characteristics of these computing machines are listed in Table I.

As this research is conducted in the context of the 5G-CARMEN project, which is focused on leveraging 5G advancements to deliver safer and more intelligent transportation on the Bologna-Munich corridor, we designed our experiment on the Smart Highway testbed, emulating the mobility of the test vehicle using the location emulation service as if it is moving on the highway corridor between Italy and Austria. To create a multi-domain setup, we deployed two MEC hosts within RSUs, representing two different domains (e.g., countries). Hence, the deployment of MEC orchestrated applications in different RSUs, emulates the scenario with multiple administrative domains (e.g., two MEC systems in the vicinity of the border between two countries). As we have designed and developed the orchestrated MEC application for supporting back situation awareness on the highways, the example multi-domain scenario emulates the setup in two countries, with the highway corridor that connects them. Concerning the connectivity with vehicles in the Smart Highway testbed, it can be obtained via hybrid communication modules, either 3GPP LTE or Intelligent Transportation System (ITS-G5) and V2X. In our experimentation setup, one vehicle has been used for both sending CAM notifications and receiving DENM notifications, via long-range 4G. The involvement of more vehicles on the highway that will be in different dissemination areas is a part of our ongoing research and future work.

To emulate the movement of the EmV, we created and utilized an external service, i.e., a location emulation service, which generates the locations based on the Google map for the route between the starting point of the EmV and its destination. In our experimental evaluation, this location emulation service is running on the onboard unit of the physical vehicle we utilized in the Smart Highway setup (Fig. 5). Although the test vehicle is moving in Antwerp, the location emulation service is generating the geo-coordinates of the Italy-Austria corridor. In different testing rounds, we configure the service to generate CAM messages with different frequencies, i.e., 1 Hz, 5 Hz, and 10 Hz, thereby producing 1, 5, and 10, CAM messages per second, respectively. In all testing rounds, the speed of the EmV is constant, and it is 108km/h (30m/s), which is in the range of the speed limit for the European highways, and near realistic as in scenarios that we are targeting, i.e., in-advance clearance of the lane for an EmV, where the EmV will be able to maneuver with a stable speed. All CAM messages carrying real-time information on the EmV emulated location, speed, and heading, are sent from the on-board unit on the physical vehicle and are received by the BSA application running on the MEC host collocated in RSU on the highway E313. By processing this real-time information received in the CAM message, BSA application produces notifications for different areas on the road and disseminates them. Any vehicle located in such areas receives the notification, if it is equipped with communication capabilities and processing logic for parsing DENM messages with ETA information.

In addition, to evaluate the impact of the BSA system on the overall emergency response time, we have performed tests and data collection while driving on the E313 highway in Antwerp, thereby measuring the Actual Time of Arrival (ATA) of the testing vehicle (Smart Highway testbed vehicle equipped with sensors, GPS, and communication capabilities) on the road stretch of up to 3.5km. The obtained values are further studied in comparison with the presence of the BSA system, and the impact it would have on reducing the response time of ambulances or firetrucks.

[5]Smart Highway: https://www.uantwerpen.be/en/research-groups/idlab/infrastructure/smart-highway/.

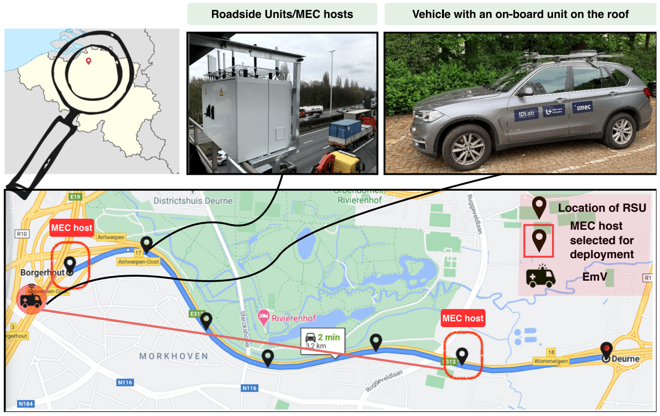


Fig. 5: The experimental setup consisting of the vehicle, and the MEC hosts deployed on the E313 highway (Antwerp, Belgium)

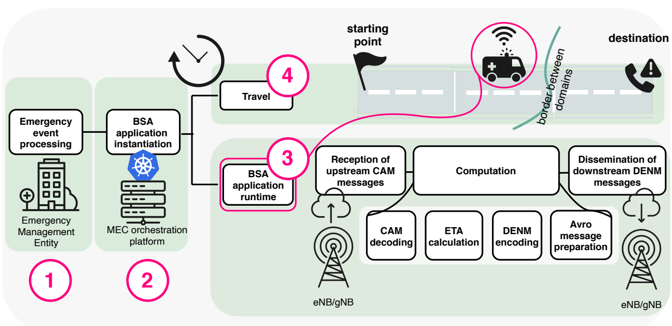


Fig. 6: The overall emergency response time in the BSA system.

B. *Key Performance Indicators*

1) ETA algorithm: Here we briefly describe the ETA model for estimating the time of arrival for EmV on different segments of the road. As presented in detail in our previous work [12], estimating the time of arrival for different dissemination areas is based on the EmV's current location with reference to each WP (i.e., WP_i), and it is denoted as ETA_i. The sum of all travel times of consecutive WPs is considered to be the total time of travel.

Predicting the travel time depends on the urban traffic environment which is highly dynamic, uncertain, stochastic, and time-dependent. Various factors such as random fluctuations in travel demands, weather conditions, interruptions caused by traffic control devices, and incidents, have an impact on travel time [14,31]. In our case, as denoted in equation (1), the ETA (t_{i+1}) distribution with reference from EmV's current position to a specific WP (WP_{i+1}), depends on i) the ETA (t_i) of the previous WP (WP_i), ii) the average speed of the EmV ($s_{v_i,v_{i+1}}(t_i)$), as well as iii) the distance between two successive WPs ($L_{i,i+1}$).

$$t_{i+1} = t_i + \frac{L_{i,i+1}}{s_{v_i,v_{i+1}}(t_i)}. \tag{1}$$

The distance $L_{i,i+1}$ is a constant parameter and it depends on the route path that the EmV selects to reach the destination. Therefore, the ETA varies only on the average speed of the EmV $s_{v_i,v_{i+1}}(t_i)$ maintained between the two successive WPs WP_{i+1} and WP_i, and its variance. This average speed value is obtained from the periodic real-time CAMs and historical speed values collected during the traveling experiences on the same route path. In particular, the Kalman filtering technique is used to improve ETA estimation by continuously updating or correcting the state variable, thereby creating the prediction with real-time measurements [12]. Further details on the filter implementation, analysis, and use, are available in our previously published work [12], which is entirely focused on the ETA algorithm and its performance.

In addition, this ETA algorithm is able to dynamically adjust the dissemination area sizes considering the relation between acceptable ETA estimation error or threshold ($e^{max}(t_{i+1})$), the speed of the EmV ($v(t_i)$), and the index ($n(t_{i+1})$), equation (2) [12].

$$L^{max}_{i+1} <= e^{max}(t_{i+1})\frac{v(t_i)}{n(t_{i+1})}. \tag{2}$$

The index n reflects the traffic conditions, weather conditions, time of the day/year, and other parameters that may impact the speed of the EmV. To forecast it for the upcoming time t_{i+1}, we use ATA ATA_i, speed of the vehicle $v(t_i)$, and the distance d_i, obtained at the time t_i, as represented in equation (3).

$$n(t_{i+1}) = ATA_i\frac{v(t_i)}{d_i}. \tag{3}$$

In this situation, the dynamically changing index n from every newly received CAM is used to dynamically resize the dissemination areas sizes $L(t_{i+1})$, so that the estimation error caused by this size stays under a defined threshold (e^{max}_{i+1}) [12].

The use of different filtering techniques, their error behavior, threshold values for emergency situations, the relation and dependency between estimation error ($e(t_i)$), dissemination area size ($L(t_i)$), speed of the EmV ($v(t_i)$) and index ($n(t_i)$) are modeled, analyzed, and evaluated in our previous work [12]. In addition to this, in [12] we have performed a comparative analysis of accuracy and run-time computational complexity of different filtering techniques, such as Simple moving average, Exponential moving average, and Filter-less method, compared to the Kalman filtering technique. Given the results that we obtained [12], the Kalman filtering technique proved to be superior in comparison with the other studied approaches, and as such, it has been selected for deployment within the ETA algorithm, and embedded in the BSA application service. Thus, as already said, the performance evaluation of the ETA algorithm itself is out of the scope of this paper, in which we rather focus on the performance of the overall BSA application service and its impact on the emergency response time.

2) BSA Service Performance: The main goal of deploying BSA application service on top of the orchestrated MEC systems is improving the safety, and efficiency of responding to emergency situations on the highways, thereby decreasing the overall emergency response time. Hence, in the following section, we present the impact of the emergency scale on the MEC system resources and service response time. In Fig. 6, we visualize the delay contributing factors to the overall emergency response time, as follows: i) processing of emergency event by an external EMA (contributor 1 in Fig. 6), ii) application instantiation on top of the orchestrated MEC system (contributor 2), iii) MEC application runtime while EmV is traveling (contributor 3), and iv) the total travel time of EmV from its starting point to the place of emergency event (contributor 4).

As described in Section III-C, in such BSA system, the trigger for instantiating a BSA application that will support an emergency event by generating event-specific notifications for all affected civilian vehicles on the road, comes from some external EMA. The processing of this request solely depends

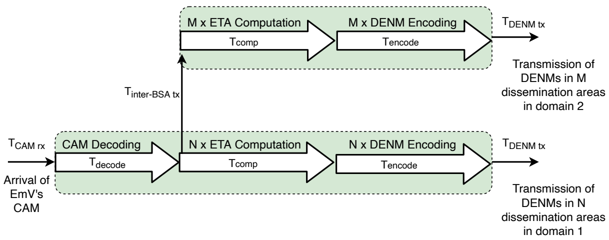


Fig. 7: Visualization of contributors to the overall BSA response time.

on this external EMA, and measuring its contribution to the overall response time is out of scope of our work.

Furthermore, once EMA generates a request for the BSA application, it sends the request to the corresponding orchestration system on the MEC. When the orchestration platform receives this request, the orchestrators proceed with a decision-making process to select the corresponding MEC system for hosting BSA application service. The selection of MEC system is described in Section III-C, and after the decision is made by orchestrators, the Edge Platform controller [30] applies this decision, and deploys the BSA service application on the selected MEC system.

The BSA application runtime consists of several microservices, whose processes are highly relevant for the overall service performance. As we illustrate in Fig. 6, during the MEC application runtime, there are three distinct delay-incurring processes that are executed simultaneously:

- Reception of upstream CAMs that are sent by an EmV to the BSA application running on the MEC, i.e., T_{CAMrx} in Fig. 7. This process contributes towards the *communication latency*.
- Computation overhead involving the decoding of the periodically received CAMs in terms of speed/location/route of the EmV (i.e., T_{decode} in Fig. 7), for deriving ETA values for respective dissemination area (i.e., T_{comp} in Fig. 7). The derived ETA values are encoded inside the DENMs (i.e., T_{encode} in Fig. 7), which are generated for the respective dissemination areas to notify the civilian vehicles, and to prepare the required format[6] for the message dissemination service on the MEC system. All this accounts for the *computational delay*. Fig. 7 clearly depicts how each of these processes contributes to the overall computational delay.
- Dissemination of downstream DENM from the message dissemination services to all civilian vehicles in different dissemination areas on the road, i.e., T_{DENMtx} in Fig. 7. This process adds to *communication latency*.

Let us study the BSA system and its KPIs in a greater detail. If we consider all MEC hosts where BSA application service can be deployed as an undirected graph consisting of m edge nodes, i.e., $V = \{z_1, z_2, ..., z_m\}$, where i-th MEC host belongs to $\{1, 2, ..., m\}$, then T_{comm_i} is the communication latency for i-th MEC node that is hosting the BSA application for the j-th EmV ($j \in \{1, 2, ..., v\}$). Concerning the overall communication latency, described as T_{comm_i} in equation (4), it refers to the uplink and downlink latency for BSA application service, i.e., the time needed for CAMs to be sent from an EmV to the BSA application running on the MEC system, and the duration of dissemination of DENMs from BSA service to the civilian vehicles, respectively. The communication latency usually consists of the transmission and the propagation latency [30,32], which are described further in (4), as T_{t_i} and T_{p_i}, respectively. In particular, if x_i denotes the amount of data to be processed by the selected MEC deployment (i.e., the data carried by a CAM message), and B_{ij} the available bandwidth on the link between the i-th MEC host and the j-th EmV, then the transmission latency defined as $T_{t_{ij}}$ is the time needed for processing the x_i amount of data on the transmitter side (vehicle). In addition, the propagation latency $T_{p_{ij}}$ depends on the length of the link between the vehicle and the selected MEC deployment l_{ij} and the overall propagation speed over the wireless link. The parameters β and γ are defined as weighting factors that balance the networking characteristics [30,32], determining the variability of available bandwidth, in case of transmission delay (β), as well as the index of refraction in the medium different than a vacuum in case of propagation delay (γ). For the sake of simplicity, we can consider transmission delay stable, by using $\beta = 1$. On the other hand, as the signals between vehicle and MEC hosts propagate over the wireless network, γ could be also taken as 1.

[6]Apache Avro: https://avro.apache.org/docs/current/spec.html

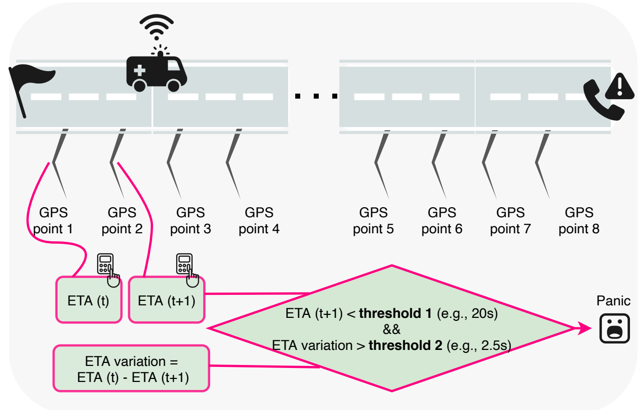


Fig. 8: Panic indicator evaluation per dissemination area.

$$\begin{aligned} T_{comm_i} &= T_{t_{ij}} + T_{p_{ij}} = T_{CAMrx_i} + T_{DENMtx_i} \\ T_{t_{ij}} &= \beta \cdot \frac{x_i}{B_{ij}} \\ T_{p_{ij}} &= \gamma \cdot \frac{l_{ij}}{s} \end{aligned} \tag{4}$$

On the other hand, for the computational latency illustrated in Fig. 7 and described as T_{comp_i} in equation (7), ETA calculation is performed in all domains affected by an emergency situation (e.g., domain 1 and domain 2), where respective BSA service instances calculate ETA for all dissemination areas in the domain (i.e., N, and M dissemination areas, in domain 1, and 2, respectively). It is important to notice that latency imposed by ETA calculation depends on the number of dissemination areas because BSA application performs calculation simultaneously for all areas in a particular domain.

To further study this type of latency, a MEC system is considered as a model where CAM messages are arriving as an $M|M|k$ queue model [33], in which the occupation of the processor on the MEC host can be defined as ρ in (5), whereas f_{CAM} is a CAM message arrival rate, k is the number of processors assigned to the BSA application service, and $\frac{1}{\mu}$ is the average time to process a single CAM message.

$$\rho = \frac{f_{CAM}}{k \cdot \mu} \tag{5}$$

$$P_w = \frac{(k \cdot \rho)^k}{k!} \cdot \left((1-\rho) \cdot \sum_{n=0}^{k-1} \frac{(k \cdot \rho)^n}{n!} + \frac{(k \cdot \rho)^k}{k!} \right)^{-1} \quad (6)$$

According to [33], the probability that a certain task needs to wait to be processed is described as P_w in (6), which further contributes to the definition of the computational latency in (7), consisting of the wait time and service time.

$$T_{comp_i} = T_{w_i} + T_{s_i} = P_{w_i} \cdot (1-\rho_i)^{-1} \cdot (k_i \cdot \rho_i)^{-1} + \frac{1}{\mu_i} \quad (7)$$

$$T_{comp_i} = T_{decode_i} + T_{eta_i} + T_{encode_i} \quad (8)$$

Given the insights from their study on CAM messages, Breu et al. [34] and Hoang et al. [35] consider the CAM rate/frequency a crucial input for the development of a series of vehicular network products, as it defines the rate of receiving updates on the position/speed/heading, thereby highly affecting the situational awareness of vehicles. Also, Lonc and Cincilla [36] and Kloiber et al. [37] confirm that the CAM frequency is an adjustable parameter that could take values from the range of 1 to 10 Hz, depending on the type of the C-ITS application. Thus, to study the impact that the frequency of the upstream CAMs has on the BSA service performance, we consider three different frequencies in our experimental setup (f_{CAM}), i.e., 1 Hz, 5 Hz, and 10 Hz. Higher frequency provides a more granular input (i.e., updating speed and location 10 times per second) for the ETA calculation, thereby increasing the accuracy of ETA estimation. However, such high CAM frequency might burden the service with an increased number of requests, which ultimately affects the system resource consumption. Thus, we define a constraint in (9), as the overall computational latency is bounded by the time needed for processing one single CAM message. In particular, if CAM frequency f_{CAM} is 10 Hz, the BSA application service has a 100 ms time frame to perform all operations (i.e., decoding, ETA calculation, encoding, and message preparation for dissemination), i.e., $T_{comp} \leq 100ms$. At the same time, a frequency of 1 Hz grants the application more time for computation before the next updated CAM arrives ($T_{comp} \leq 1s$). However, for some other services, such as autonomous driving ones, CAM frequency of 1 Hz might be too low, and impact the granularity of computation updates performed by the service (e.g., low ETA update granularity).

$$T_{comp_i} \leq \frac{1}{f_{CAM}} \quad (9)$$

Another important metric is the state update delay T_{sud} defined in (10), which is specific for multi-domain deployments where multiple peering BSA application services are running on different MEC hosts, while addressing the same emergency situation in a distributed way. This metric is equivalent to communication latency described in (4), which now depends on the amount of metadata to be sent over the network x_{meta}, bandwidth on the link between two application services $B_{m_1 m_2}$, and its length $l_{m_1 m_2}$. The x_{meta} is approximately 150 B (data exchanged between two BSA instances running on two edges, informing each other about the location/speed/heading of the vehicle). Concerning propagation delay, it is negligible in this case as the distance between adjacent MEC hosts is approximately 1 km, and since the hosts are connected via fiber, the propagation latency results in 5 μs.

$$T_{sud} = \beta \cdot \frac{x_{meta}}{B_{m_1 m_2}} + \gamma \cdot \frac{l_{m_1 m_2}}{s} \quad (10)$$

Moreover, our BSA application service is capable of serving multiple EmVs at the same time, with the opportunity to send EmV-specific notifications to all civilian vehicles in dissemination areas (the amount of data to be processed x_i increases). Thus, we also study the impact of a number of vehicles that consume BSA service simultaneously.

Finally, we introduce a metric called *panic indicator* to depict how our BSA application service can potentially help civilian vehicles to clear the lane for the EmV in a calm manner, thereby increasing road safety. In Fig. 8, we showcase how a panic indicator can be calculated for each dissemination area. In particular, for each GPS point on the road, the EmV sends an update on its speed/location/route via upstream CAMs, and based on that updated information, BSA application recalculates ETA for all dissemination areas. We compare the calculated ETA values for two successive updates from EmV on the road (e.g., GPS points will be more scarce with the lower frequency of the upstream CAMs), and the difference between them is then compared with the threshold 2 (Fig. 8). This threshold determines whether the difference between two successive ETA values can cause panic, such as an ETA of 20s dropping to 2s in the very next update. However, comparing two successive ETA values is not sufficient, because the difference of 18s from the previous case will not affect the driver in the same way if the current ETA is e.g., 10 minutes. In this case, the driver will most probably not even notice the difference between 10 minutes received in the previous update, and 9 minutes and 42s in the next update. Thus, the current value of ETA is important to consider as well (i.e., threshold 1 in Fig. 8), as it indicates whether EmV is approaching in a short time frame or not. In case the ETA variation from one update to another is higher than threshold 2, and the current ETA value is lower than threshold 1, the panic indicator will be turned on. This indicator is a Boolean data type, and if the previously defined criteria are not met, the indicator is equal to zero. It is important to note that thresholds 1 and 2 are subjective, as they depend on the drivers' perception, but in this paper, we provide the notion of how it can be preempted by MEC applications that assist vehicles on the road, in order to improve the efficiency of reaction of civilian vehicles to the arrival of an EmV.

In order to test the statistical significance of our results, presented in the following section, we apply the Kruskal Wallis test [38], a commonly used non-parametric test for two or more samples that do not necessarily follow a normal distribution. This test reflects whether the mean ranks between two or more measurement groups are statistically significant (i.e., p_{value} lower than 0.05) or not.

C. Limitations of the testing methodology

The whole MEC-based system for supporting emergency situations on the highways, which we built and presented in this paper, reflects on a real-life setup within highway environments. It is connecting different pieces of the MEC system, such as orchestration elements, containerized edge application deployments, and vehicles as consumers and producers of data/messages. Due to the complexity and diversity of the whole system, such implementation in real-life systems is quite challenging but very promising and valuable given the ever-increasing interest in 5G edge applications for diverse verticals [39]. Nevertheless, there are some limitations in the testing methodology that need to be further worked on, such as the lack of 5G deployment, the lack of more prototype/testbed vehicles that will be used for testing the message reception, and the static routes for EmV movement.

The lack of 5G impacts the results on the communication latency, but it is not limiting the overall evaluation of the

BSA impact, as the improvements brought by 5G over 4G can be anticipated based on the related studies [40]. The lack of multiple vehicles is mildly affecting the overall feasibility of the system, as the BSA application service is designed for multiple receiving vehicles (no limit on the number of vehicles). However, the performance of application service in terms of timely dissemination of DENM messages is tested in our approach as the same vehicle is used for both sending upstream CAM and receiving downstream DENM messages. In addition, the feasibility of the whole BSA system for multiple vehicles has been proven successful when tested in the 5G-CARMEN pilot sites with multiple prototype vehicles, as described in one of the final project deliverables [41]. Finally, the use of static routes in the Python program of the ETA algorithm (.gpx format) can be easily replaced by more advanced real-time services such as Google Maps API, which would update the route based on the current traffic circumstances. Such an approach would further improve the success of the BSA operation, but it does not add any additional complexity that has not been envisioned in the proposed system.

V. Results and discussion

A. Computational and communication latency

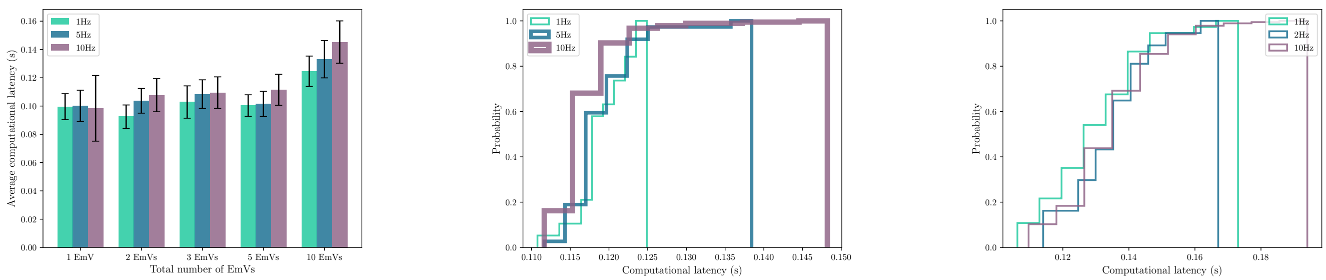
As described in Section IV-B, BSA application service can receive CAMs from EmVs with different frequencies, and accordingly, in Fig. 9 we show the average computational latency of BSA application service, with reference to a single EmV, depending on the number of EmVs that are simultaneously served, and the upstream CAM frequency. In particular, Fig. 9a depicts the average computational latency with reference to single EmV, and although there is a slight increase in average latency with the increase of upstream CAM frequency and a number of vehicles, a more visible difference between cases can be seen in Figures 9b and 9c that show the Cumulative Distribution Function (CDF) of latency with reference to single EmV, in case there is one EmV, and ten EmVs (i.e., large scale emergency), respectively.

From the results presented in Figures 9b and 9c, we can see that computational latency increases with an increase in CAM frequency, as well as in case of an increased number of EmVs being served by a single BSA application. In Fig. 9b, there is an evident increase in latency, as all values are below 125 ms, 138 ms, and 148 ms, for 1 Hz, 5 Hz, and 10 Hz, respectively. The same behavior is observed for 10 EmVs going in the same direction, however, due to the increased processing from the BSA application, there will be an increase in the overall computational latency for the case of 10 EmVs heading to the same destination (Fig. 9c), where all values of latency are lower than 175 ms, 185 ms, 275 ms, for 1 Hz, 5 Hz, and 10 Hz, respectively. In particular, for the frequency of 10 Hz, the computational latency is always below 150 ms in case there is only one EmV (Fig. 9b). In the same case, the probability drops to 0.65 if the BSA service is serving 10 EmVs simultaneously (Fig. 9c), meaning that there is even a 35% chance that computational latency is above 150 ms, which leads to the insufficient time frame for processing CAM message and informing civilian vehicles about the update in ETA. Thus, for 10 Hz, the time frame of 100 ms (equation (9)) is not sufficient for the service to perform all operations illustrated in Fig. 6 until the next message with update speed/location is received. Applying the Kruskal Wallis test on the collected results for different CAM frequencies results in $p_{value} = 0.0024$, which is lower than 0.05, thus showing the statistical significance of the difference between the computational delay in these three samples. Similarly, comparing samples across different numbers of EmVs, the result is $p_{value} = 0.00026$ for the frequency of 1 Hz, with negligibly small p_{values} for the other two frequencies.

If we now take a look at the average computational latency per specific BSA operation (Fig. 10), it can be seen that in all cases less than 20% of the overall computational delay is incurred by the algorithm that evaluates ETA for all dissemination areas, based on the latest data on the speed/latency. Thus, knowing that other processes such as encoding/decoding of C-ITS messages (e.g., CAMs and DENMs), and preparing messages in a required format for dissemination, take most of the time, it is important to ensure enough resources for these processes to run properly. Therefore, for the MEC application such as BSA, it is better to deploy all services in separate containers, which can be further separately scaled by orchestration entities, thus, potentially saving more computing resources than in the case of scaling the whole single container BSA application.

The frequency of sending upstream CAM messages from an EmV to the BSA application substantially hinders the time given to the application to perform computation, i.e., i) to decode the received CAM message and resolve EmV's speed/location, ii) to calculate ETA for all dissemination areas, and iii) to prepare DENM messages for dissemination, following a requested message format. In the case of 1 Hz, the time frame for computation is 1s, which is sufficient, according to the results presented in Fig. 9. However, if the frequency is 10 Hz, 100 ms seems not to be enough for BSA to perform all operations. To address this issue, BSA application can adjust the reception of upstream CAM messages from an EmV, by filtering out a certain number of messages sent within a 1s time frame, but taking care of its impact on the accuracy of ETA calculation at the same time. This way, although not configuring the CAM generation frequency at the vehicle side, the application itself should dynamically adjust the upstream frequency so it can adequately and timely respond to each new message.

Tackling the multi-domain deployment of BSA application service, which is described in Section III-C, once the peering BSA application instance is deployed in the second domain (e.g., first application instance in Italy, and second in Austria), the source application instance needs to proactively inform its peering application instance about the changes in speed/location of the EmV. This way, even while not receiving CAMs directly from the EmV yet, the peering application instance can derive the ETA values for the dissemination areas under its control. Thus, we measured the average state update latency for two application instances running on two different MEC systems, as presented in Fig. 11c. This metric needs to be taken into account while performing the BSA operations in the peering application instance because the actual speed/latency from the EmV are derived from the CAM before the time indicated by state update latency. Thus, neglecting the state update duration might affect the accuracy of calculating the ETA values for dissemination areas. Let us consider the case when CAM frequency is 10 Hz and there are 10 EmVs simultaneously using BSA application service. As it can be seen in Fig. 11c, the average state update latency is 31.4 ms, which means that less than 70 ms is left for BSA application to calculate ETAs and prepare messages for vehicles in different areas. Considering the average computational latency shown in Fig. 9a, 70 ms is not sufficient even for the cases of lowest CAM frequency and only one EmV in the system. Therefore, it is essential for such application services to constantly monitor all performance parameters, and thus, generate application-specific alarms for orchestration entities to allocate more resources or migrate applications from one edge to another. The

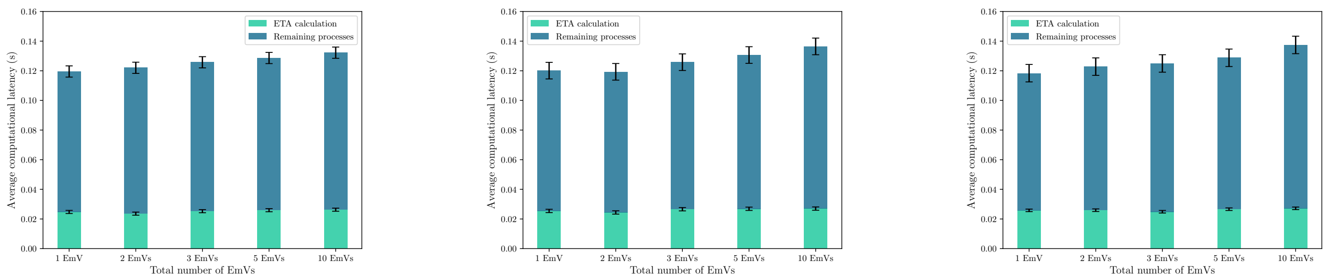


(a) Average computational latency with reference to single EmV being served by BSA application.

(b) CDF with reference to single EmV being served by BSA application; One EmV in total served by BSA application.

(c) CDF with reference to single EmV being served by BSA application; Ten EmVs in total served by BSA application.

Fig. 9: The overall computational latency with reference to single EmV.

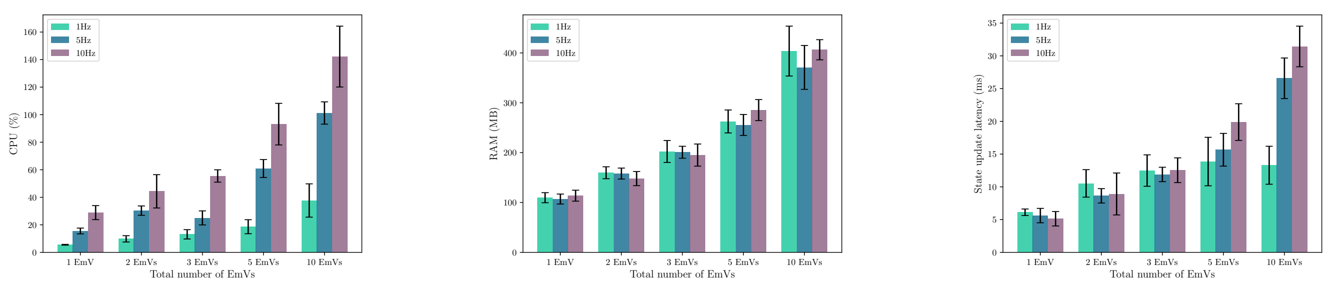


(a) CAM frequency 1 Hz.

(b) CAM frequency 5 Hz.

(c) CAM frequency 10 Hz.

Fig. 10: The overall computational latency per process.



(a) Average CPU consumption.

(b) Average RAM consumption.

(c) Average state update latency.

Fig. 11: The resource consumption and state update latency.

statistical test also supports the previously presented result, showing that the state update latency significantly changes with the CAM frequency ($p_{value} = 2.2845e{-}05$), and with an increasing number of EmVs ($p_{value} = 0.00031$ for the CAM frequency of 1 Hz).

Therefore, for the two peering BSA application service instances running on two MEC platforms, the state update latency needs to be minimized in order to keep the application instance 2 (which is still not receiving CAM messages directly from EmV) updated on the EmV's speed/location. Although low (Fig. 11c), the state update duration might affect the accuracy of calculating the ETA values for dissemination areas, it needs to be accounted for in ETA algorithm. Concerning the communication latency described in Section IV-B, i.e., uplink and downlink latency for CAM reception, and DENM dissemination, respectively, we have collected measurements within the same experimental setup described in Section IV-A. In particular, the vehicle used in our experimental setup has been used for both: i) sending CAMs to the BSA application service, and ii) receiving DENMs from the BSA application service. The client application deployed on the computing engine in the Onboard Unit (OBU) of the vehicle is connected to the BSA application services via long-range 4G. Within 10 series of measurements, the results that we obtained indicate an average uplink latency of 28.84ms, with a standard deviation of 18.64 ms, an average downlink latency of 18.63 ms, and a standard deviation of 6.39 ms.

Measuring uplink latency for the reception of CAM messages is important for V2X services, as it indicates the time from the moment when the important data is generated on the vehicle side, to the moment when this data is processed by V2X service on the MEC platforms. For information such as the current location of a vehicle, the longer uplink latency can significantly affect the efficiency of the V2X service. For example, if a vehicle is driving at the speed of 100 km/h, the uplink latency of 50 ms will affect the quality of data,

TABLE III: Result analysis of reducing the overall emergency response time.

Drive	**Length (m)**	**Statistical analysis of the difference between (BSA system/BSA system with 30% speed variation) and measured ATA values**					
		Average (s)	**St. deviation (s)**	**95th percentile (s)**	**50th percentile (s)**	**Max (s)**	**p-value**
1	3501.79	(51.47, 50.38)	(26.68, 25.98)	(94.56, 92.51)	(51.53, 50.44)	(103.39, 101.29)	(2.86e-18, 1.44e-17)
2	3413.71	(65.75, 64.78)	(30.35, 29.67)	(107.28, 105.04)	(70.98, 70.25)	(112.95, 110.61)	(6.4e-28, 3.75e-27)
3	56.07	(3.53, 3.48)	(1.95, 1.95)	(6.05, 6.01)	(3.49, 3.44)	(6.34, 6.29)	(0.0033, 0.0033)
4	3451.92	(48.55, 47.26)	(25.22, 24.51)	(88.44, 86.17)	(49.88, 48.62)	(96.86, 94.57)	(5.31e-17, 3.45e-16)
5	3064.35	(38.45, 37.47)	(21.22, 20.75)	(73.23, 71.56)	(36.7, 35.58)	(77.45, 75.71)	(5.24e-13, 1.91e-12)
6	3305.86	(39.76, 38.79)	(20.43, 19.89)	(70, 68.06)	(41.3, 40.47)	(74.11, 72.11)	(1.59e-13, 5.96e-13)
7	3300.56	(37.54, 36.48)	(21.09, 20.51)	(73.75, 71.89)	(36.46, 35.41)	(79.32, 77.42)	(6.65e-12, 2.5e-11)
8	992.6	(9.71, 9.41)	(5.24, 5.09)	(17.51, 16.99)	(9.85, 9.53)	(18.35, 17.83)	(0.00065, 0.0009)
9	1300.43	(12.54, 12.19)	(7.59, 7.39)	(25.19, 24.44)	(11.85, 11.53)	(27.11, 26.36)	(0.00017, 0.00025)

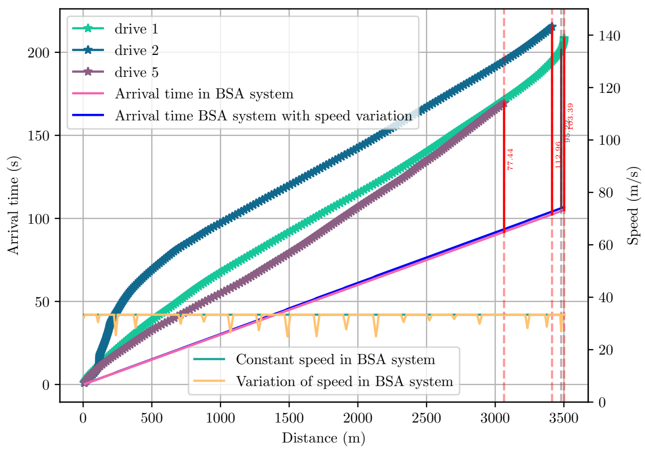


Fig. 12: Actual time of arrival with and without the BSA system.

because the vehicle will already have moved for an additional ca. 1.4 m until the V2X service receives this data. In the case of autonomous vehicles, such delay is of course not tolerable, and that is why the ultra-low latency promised by 5G and MEC is important. Concerning the BSA service, such delay can affect the accuracy of the ETA algorithm [12], and thus, it is important to inform the service about the average latency on the uplink, so it can adjust the ETA algorithm that will accordingly correct the estimation of ETA values, taking into account the speed of the vehicle and the measured latency. When it comes to the downlink latency, less than 20 ms latency that we obtained and presented in Section V will not significantly affect the accuracy of the ETA value presented in each civilian car, but for some other types of V2X services that e.g., provide maneuver recommendations, this delay is also important to consider and to decrease. As in 4G, this average one-way latency is around 28 ms for uplink, and 18 ms for downlink, improvements in latency brought by 5G play a significant role in V2X services.

B. *Resource consumption*

The increase in CAM frequency and number of EmV concurrently served not only increases the computational latency but also highly affects the resource consumption of the containerized application. As it can be seen in Fig. 11, both CPU and RAM load increase with the CAM frequency and number of EmVs, which needs to be taken into account when deploying BSA application service. For example, when 10 EmVs are being served by BSA service running on the MEC system, for frequencies higher than 1 Hz, more than one CPU core is needed, and if not properly managed and orchestrated, such increase in load can result in service failure. Due to the resource-constrained nature of edge nodes where BSA service is running, the resource consumption needs to be carefully assessed and monitored in order to prevent disruptions in application performance (e.g., service unavailability, longer computational latency, low accuracy of ETA evaluation).

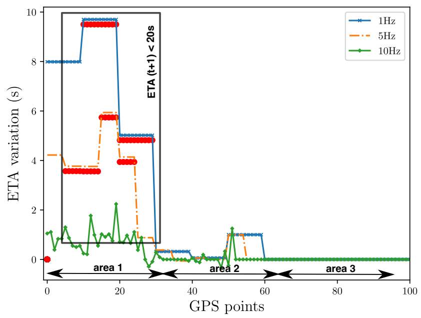
Thus, the higher frequency of CAM messages, the higher CPU and RAM load. Also, the more EmVs are served by the same BSA application instance concurrently, the more resources are needed. As both computing and networking resources need to be efficiently consumed in MEC platforms, this increase might severely disrupt the service performance, increasing the average response time of BSA application service.

C. *The overall emergency response time*

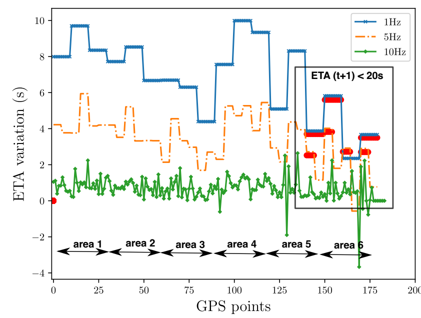
The largest portion of the overall emergency response time is mainly determined by the travel time of the EmV, as indicated in Fig. 6. Thus, in addition to the performance of the BSA system in terms of communication/computational latency and resource consumption at the network edge, here we briefly discuss the impact of the BSA system on overall emergency response time. In Tab. III, and Fig. 12, we showcase the analysis of the sample of real drives performed on the E313 highway in Antwerp, Belgium. During these drives, the ATA values are collected for different trip lengths (Tab. III). To compare the obtained values of ATA for an emergency responder with the time of arrival using the BSA system, we considered two scenarios: i) no interference from the civilian vehicles, i.e., all civilian vehicles clearing the lane in time so that the EmV can pass with the maximum speed of 120 km/h (BSA system in Fig. 12), and ii) interference from the other vehicles causing random speed variations of up to 30% (BSA system with speed variations) each 10 m on the road. The results show a statistically significant reduction of the overall response time with the BSA system compared to the drives with no BSA system used on the highway (p-value in Kruskal-Wallis tested lower than 0.05, Tab. III). The maximum reduction of the arrival time of more than 1.5 min with the BSA system is visible for the drives that are close to 3.5 km, which illustrates realistic scenarios and driving routes of emergency responders. Such a reduction of emergency response time is crucial for increasing the efficiency of emergency responders on busy roads.

D. *Panic indicator*

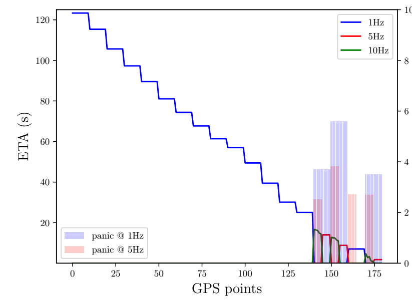
As elaborated in Section IV-B, studying the panic indicator for services such as BSA can help to improve the overall performance, as notifications for vehicles/drivers can be generated more efficiently, thereby preempting their reaction and its potential outcome (e.g., increased stress that might result in uncoordinated and incautious response to the approaching EmV). To derive conclusions on the occurrence of panic, we considered the variation of ETA values that are collected on the testbed along the road (ETA variation illustrated in Fig. 8), considering the route with six dissemination areas in total. We



(a) Civilian vehicles in area 2.



(b) Civilian vehicles in area 6.



(c) Panic avoidance with dynamic frequency adjustment (area 6).

Fig. 13: ETA Variation ($ETA(t) - ETA(t+1)$); The values highlighted in red imply that the panic indicator is on, for civilian vehicles in dissemination areas 1 to 6.

calculate ETA variation that would be experienced by civilian vehicles in these six dissemination areas, and present it in Fig. 13. According to the description of panic indicator as a metric, which is provided in Section IV-B, we indicate that panic happens in situations in which the current ETA value for a specific area is less than the 20s (i.e., indicating a soon arrival of EmV), and ETA value drops for more than 2.5s comparing to the previously received notification (i.e., ETA variation larger than 2.5s). If these criteria are met, the panic indicator is turned on (i.e., Boolean value 1) for all civilian vehicles in a specific dissemination area.

The thresholds used in this criteria are subjective, but here we try to use some close to realistic values and assess how often panic, due to the EmV, might occur on the highways. Figures 13a) and 13b) show the ETA variation ($ETA(t) - ETA(t+1)$) for civilian vehicles in areas 2 and 6, while the trends in the remaining areas are not shown due to the space limitations in the manuscript. For example, in Fig. 13b, the GPS points on the x-axis indicate the location of EmV that is moving through areas 1-6. In this particular case, the y-axis displays the ETA variation experienced by civilian vehicles in dissemination area 6, depending on the current location of EmV, which can be in any of the six areas (as displayed on the x-axis). The portions of the graphs that are highlighted in grey represent the cases when the ETA variation is larger than 2.5s, and the current ETA is lower than the 20s (i.e., both criteria from Fig. 8 met). With reference to Fig. 13b, civilian vehicles in area 6 start experiencing panic only when EmV is in areas 5 and 6, i.e., closer to them, and when BSA application notifies them about EmV's arrival with a lower frequency, i.e., 1 and 5 Hz.

From the obtained results (Fig. 13), it can be seen that panic mostly happens for 1 Hz CAM frequency, which is somewhat expected, due to the least frequent updates on the current speed and location of an EmV. In our scenario, panic never occurs in case the frequency is 10 Hz, but from the results studied above, we clearly identified several bottlenecks of having such a high frequency of upstream messages. Therefore, it is important for BSA application to dynamically adjust the frequency of sending notifications to civilian vehicles/drivers for different dissemination areas, to decrease the probability of panic, thereby improving their efficiency of responding to EmV's arrival. Therefore, Fig. 13c shows the BSA mechanism of dynamically adjusting the frequency of sending DENM messages to civilian vehicles in cases panic is expected. This way, the ETA notifications are sent with an increased frequency (either 5 or 10Hz) where panic occurs in Fig. 13b, but not in other cases in order to save edge resources. Thus, with such a feature, BSA system is both edge-aware/resource-aware and capable of reducing panic in the presence of emergency responders on the roads.

E. Design requirements for V2X applications

Given the resource constraints in edge networks, the design of MEC applications such as BSA is highly important because of resource consumption. If MEC application is designed to perform all separate processes, or groups of processes, in separate containers, the orchestration entities can scale containers independently, and potentially save more computing resources [42] than is the case of scaling all processes inside one container at once. Such a design, which decouples the main application logic into several independent and loosely coupled microservices, allows MEC orchestrators to rapidly and flexibly deploy services and make sure that application performance matches the required level of quality of service. As such, V2X applications become suitable for running within orchestrated MEC systems, as described in Section III. With CNI extension for Kubernetes, our MEC orchestration system dynamically creates external interfaces for V2X application deployment and makes it accessible for the vehicles, dissemination services, orchestrators, and peering application instances in other MEC platforms. This interface towards orchestration entities can be further used for informing orchestration layers about some internal application procedures (e.g., a vehicle is approaching the border between two countries) so that the life-cycle management of applications can be improved by deploying additional instances in other relevant domains (described in Section III-C). Finally, if the V2X application is expected to run in distributed MEC environments, the dynamic setup of a data plane communication between peering instances should be enabled, so that the necessary metadata or application context can be timely transferred.

VI. Conclusion

In this paper, we introduced an on-demand edge-aware MEC application service to enhance back situation awareness on the highways, thereby enabling early notifications for vehicles about the ETA of an approaching EmV. This cloud-native application service provides drivers with sufficient time to create a safety corridor for the EmV by clearing the lane and allowing the EmV to pass through unhindered in a safe manner, thus, increasing the mission's success. This impact is also measured in terms of the reduction of overall response time of emergency responders, and panic levels among civilian drivers. Due to the significant importance of decreasing the overall response time to emergency events, we performed a thorough performance analysis of the BSA application service, measuring the impact of an emergency on the MEC system resources, and service response time. Moreover, we introduced a metric called *panic indicator* that provides a notion of how the proposed BSA service can potentially help in enabling drivers to calmly maneuver out of the path of an EmV, thereby increasing the road safety with a more efficient reaction to EmV's arrival.

From the results presented in this paper, we see that it is important for the BSA application to dynamically adjust the frequency of sending ETA updates to civilian vehicles, as panic is more likely to happen if the frequency is low. The performance evaluation of the BSA application service is obtained in a realistic environment, i.e., on top of the distributed MEC hosts within the Smart Highway testbed, which is deployed along E313 highway in Antwerp, Belgium. We show that the frequency of sending CAMs from an EmV to the BSA application significantly affects the overall computing delay, hindering the time given to the application to perform computation before an updated CAM is received. As discussed, this issue can be mitigated by adjusting the reception of upstream CAMs at the application side, but taking into account the accuracy of calculating ETA for different areas. A similar effect on the computing delay is also noticed in the case of an increased number of simultaneously served EmVs, which can be solved by performing application scaling. Concerning the scaling of BSA application, reserving more resources needs to be properly managed due to the resource constraints in MEC systems, especially in the case of the higher CAM frequencies that showed an increased CPU and memory load. As in the Smart Highway testbed the connectivity with vehicles can be achieved via hybrid communication modules (e.g., LTE, ITS-G5, and V2X), we have also utilized the 4G long range to establish a communication between client application in vehicle and the BSA running on the MEC hosts. Concerning the BSA service, such delay can affect the accuracy of ETA algorithm, and it is important to inform service about the average latency on the uplink, so it can adjust the ETA algorithm that will accordingly correct the estimation of ETA values, taking into account the speed of the vehicle and the measured latency.

Thus, in this paper, we derive important conclusions i) about the design of V2X services that are aimed for running on the MEC platforms in the 5G systems, with the goal to assist vehicles on the highways, and ii) about the operations of such services, including the study of the factors that affect the service performance. As a part of our future work, we plan to also study the impact of all contributors to the computing delay (e.g., CAM frequencies, number of EmVs, state update delay across domains) on the accuracy of estimating time of arrival of an EmV.

VII. Acknowledgement

This work is performed in the H2020 projects 5G-CARMEN (No. 825012) and 5G-BLUEPRINT (No. 952189), and supported by the Flemish Ministry of Mobility and Public Works (MOW) and the regional Agency for Roads and Traffic (AWV), Belgium.

References

[1] D. Soldani and A. Manzalini, "Horizon 2020 and Beyond: On the 5G Operating System for a True Digital Society," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 32–42, 2015. doi: http://dx.doi.org/10.1109/MVT.2014.2380581.

[2] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," *ETSI ISG ITS, ETSI EN 302 637-2 V1.4.1*, 2019. Online [Available]: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.02_60/en_30263702v010302p.pdf.

[3] ETSI., "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specification of Decentralized Environment Notification Basic Service," *ETSI ISG ITS, ETSI EN 302 637-3 V1.3.0*, 2018. Online [Available]: https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.02.01_30/en_30263703v010201v.pdf.

[4] R. B. Vukmir, "Survival from pre-hospital cardiac arrest is critically dependent upon response time.," in *Resuscitation 69(2)*, pp. 229–234, 2006. Available [Online]:doi:10.1016/j.resuscitation.2005.08.014.

[5] R. Sánchez-Mangas, A. García-Ferrrer, A. de Juan, and A. M. Arroyo, "The probability of death in road traffic accidents. how important is a quick medical response?," *Accident Analysis & Prevention*, vol. 42, no. 4, pp. 1048 – 1056, 2010. doi: https://doi.org/10.1016/j.aap.2009.12.012.

[6] J. Nicholl, J. West, S. Goodacre, and J. Turner, "The relationship between distance to hospital and patient mortality in emergencies: An observational study," *Emergency medicine journal : EMJ*, vol. 24, pp. 665–8, 10 2007. doi: http://dx.doi.org/10.1136/emj.2007.047654.

[7] S. A. Hadiwardoyo, S. Patra, C. T. Calafate, J.-C. Cano, and P. Manzoni, "An intelligent transportation system application for smartphones based on vehicle position advertising and route sharing in vehicular ad-hoc networks," *Journal of Computer Science and Technology*, vol. 33, no. 2, pp. 249–262, 2018. doi: https://doi.org/10.1007/s11390-018-1817-4.

[8] A. Metzner and T. Wickramarathne, "Exploiting Vehicle-to-Vehicle Communications for Enhanced Situational Awareness," in *2019 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, pp. 88–92, 2019. doi: http://dx.doi.org/10.1109/COGSIMA.2019.8724309.

[9] U. D. of Transportation, "Vehicle-to-Vehicle Communication Technology," *National Highway Traffic Safety Administration Report*, 2014. Online [Available]: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/v2v_fact_sheet_101414_v2a.pdf.

[10] Y. Moroi and K. Takami, "A Method of Securing Priority-Use Routes for Emergency Vehicles using Inter-Vehicle and Vehicle-Road Communication," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, 2015. doi: http://dx.doi.org/10.1109/NTMS.2015.7266466.

[11] L. Xue, Y. Yang, and D. Dong, "Roadside infrastructure planning scheme for the urban vehicular networks," *Transportation Research Procedia*, vol. 25, pp. 1380–1396, 12 2017. doi: http://dx.doi.org/10.1016/j.trpro.2017.05.163.

[12] R. Halili, F. Z. Yousaf, N. Slamnik-Krijestorac, G. M. Yilma, M. Liebsch, R. Berkvens, and M. Weyn, "Self-correcting Algorithm for Estimated Time of Arrival of Emergency Responders on the Highway," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2022. doi: https://doi.org/10.1109/TVT.2022.3209100.

[13] A. P. Iannoni, R. Morabito, and C. Saydam, "An optimization approach for ambulance location and the districting of the response segments on highways," *European Journal of Operational Research*, vol. 195, pp. 528–542, June 2009. doi: https://doi.org/10.1016/j.ejor.2008.02.003.

[14] W. Min, L. Yu, P. Chen, M. Zhang, Y. Liu, and J. Wang, "On-Demand Greenwave for Emergency Vehicles in a Time-Varying Road Network With Uncertainties," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 3056–3068, 2020. doi:http://dx.doi.org/10.1109/TITS.2019.2923802.

[15] Y. Pan, L. Sun, and M. Ge, "Finding reliable shortest path in stochastic time-dependent network," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 451–460, 2013. Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013).

[16] M. Poulton, A. Noulas, D. Weston, and G. Roussos, "Modeling Metropolitan-Area Ambulance Mobility Under Blue Light Conditions," *IEEE Access*, vol. 7, pp. 1390–1403, 2019. doi: http://dx.doi.org/10.1109/ACCESS.2018.2886852.

[17] A. B. Patel, N. M. Waters, I. E. Blanchard, C. J. Doig, and W. A. Ghali, "A validation of ground ambulance pre-hospital times modeled using geographic information systems," *International Journal of Health Geographics*, vol. 11, no. 1, p. 1, 2012. doi: https://doi.org/10.1186/1476-072X-11-42.

[18] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for Vehicular Communications," *IEEE Communications Magazine*, vol. 56, pp. 111–117, Jan 2018. doi: http://dx.doi.org/10.1109/MCOM.2018.1700467.

[19] S. Joerer, B. Bloessl, M. Segata, C. Sommer, R. L. Cigno, A. Jamalipour, and F. Dressler, "Enabling Situation Awareness at Intersections for IVC Congestion Control Mechanisms," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1674–1685, 2016. doi: http://dx.doi.org/10.1109/TMC.2015.2474370.

[20] E. Uhlemann, "Introducing Connected Vehicles [Connected Vehicles]," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 23–31, 2015. doi: http://dx.doi.org/10.1109/MVT.2015.2390920.

[21] A. Senart, M. Bouroche, and V. Cahill, "Modelling an Emergency Vehicle Early-warning System using Real-time Feedback," *IJIIDS*, vol. 2, pp. 222–239, 01 2008. doi: http://dx.doi.org/10.1504/IJIIDS.2008.018256.

[22] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9457–9470, 2016. doi: http://dx.doi.org/110.1109/TVT.2016.2591558.

[23] R. Molina-Masegosa and J. Gozalvez, "LTE-V for Sidelink 5G V2X Vehicular Communications: A New 5G Technology for Short-Range

Vehicle-to-Everything Communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017. doi: http://dx.doi.org/10.1109/MVT.2017.2752798.

[24] E. Uhlemann, "Initial Steps Toward a Cellular Vehicle-to-Everything Standard [Connected Vehicles]," *IEEE Vehicular Technology Magazine*, vol. 12, no. 1, pp. 14–19, 2017. doi: http://dx.doi.org/10.1109/MVT.2016.2641139.

[25] A. Kousaridas, C. Zhou, D. Martín-Sacristán, D. Garcia-Roger, J. F. Monserrat, and S. Roger, "Multi-Connectivity Management for 5G V2X Communication," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–7, 2019. doi: http://dx.doi.org/10.1109/PIMRC.2019.8904431.

[26] Y. Wang, J. Wang, Y. Ge, B. Yu, C. Li, and L. Li, "MEC support for C-V2X System Architecture," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, pp. 1375–1379, 2019. doi: http://dx.doi.org/10.1109/ICCT46805.2019.8947060.

[27] ETSI, "Multi-Access Edge Computing (MEC); Framework and Reference Architecture," *ETSI ISG MEC, ETSI GS MEC 003 V2.1.1*, 2019. Online [Available]: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf.

[28] 5GAA, "Use Case Implementation Description," *5GAA Automotive Association*, 2021. Online [Available]: https://5gaa.org/wp-content/uploads/2021/04/use-case-t21001.pdf.

[29] 5G-CARMEN, "Deliverable 4.1 - Design of the secure, cross-border, and multi-domain service orchestration platform," *H2020 5G-CARMEN Project Consortium*, 2020. Online [Available]: https://5gcarmen.eu/wp-content/uploads/2020/11/5G_CARMEN_D4.1_FINAL.pdf.

[30] N. Slamnik-Krijestorac, G. M. Yilma, M. Liebsch, F. Z. Yousaf, and J. Marquez-Barja, "Collaborative orchestration of multi-domain edges from a Connected, Cooperative and Automated Mobility (CCAM) perspective," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021. doi: http://dx.doi.org/10.1109/TMC.2021.3118058.

[31] S. I. Chien and C. M. Kuchipudi, "Dynamic travel time prediction with real-time and historic data," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 608–616, 2003. doi: https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(608).

[32] W. Li, Y. Zi, F. Lei, F. Zhou, Y. Peng, and Q. Xuesong, "Latency-Optimal Virtual Network Functions Resource Allocation for 5G Backhaul Transport Network Slicing," *Applied Sciences*, vol. 9, p. 701, 02 2019. doi: http://dx.doi.org/10.3390/app9040701.

[33] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, K. Mizutani, T. Inoue, and O. Akashi, "Towards a Low-Delay Edge Cloud Computing through a Combined Communication and Computation Approach," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, 2016. doi: https://doi.org/10.1109/VTCFall.2016.7881581.

[34] J. Breu, A. Brakemeier, and M. Menth, "A quantitative study of Cooperative Awareness Messages in production VANETs," in *J Wireless Com Network 2014*, 2014. doi: https://doi.org/10.1186/1687-1499-2014-98.

[35] G.-M. Hoang, B. Denis, J. Härri, and D. T. Slock, "On communication aspects of particle-based cooperative positioning in GPS-aided VANETs," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 20–25, 2016. doi: https://doi.org/10.1109/IVS.2016.7535358.

[36] B. Lonc and P. Cincilla, "Cooperative ITS security framework: Standards and implementations progress in Europe," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6, 2016. doi: http://dx.doi.org/10.1109/WoWMoM.2016.7523576.

[37] B. Kloiber, T. Strang, M. Röckl, and F. de Ponte-Müller, "Performance of CAM based safety applications using ITS-G5A MAC in high dense scenarios," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 654–660, 2011. doi: http://dx.doi.org/10.1109/IVS.2011.5940461.

[38] K. Weaver, V. Morales, S. Dunn, K. Godde, and P. Weaver, *Kruskal—Wallis. In An Introduction to Statistical Analysis in Research*, ch. 8, pp. 353–391. John Wiley & Sons, Ltd, 2017. doi: http://dx.doi.org/10.1002/9781119454205.ch8.

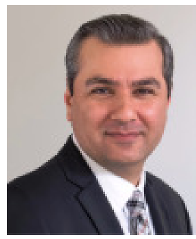
[39] D. Tsolkas and H. Koumaras, "On the Development and Provisioning of Vertical Applications in the Beyond 5G Era," *IEEE Networking Letters*, vol. 4, no. 1, pp. 43–47, 2022. doi:http://dx.doi.org/10.1109/LNET.2022.3142088.

[40] O. D. S. Al-Gburi, "General Overview of 4G and 5G with field measurements and performance comparison," *Faculty of Information Technology and Communication Sciences, Tampere University*, 2021. Online [Available]: https://trepo.tuni.fi/bitstream/handle/10024/125111/AlgburiOmar.pdf?sequence=2&isAllowed=y.

[41] 5G-CARMEN, "Deliverable 5.3 - 5G-CARMEN Final Pilot Report," *H2020 5G-CARMEN Project Consortium*, 2022. Online [Available]: https://5gcarmen.eu/wp-content/uploads/2022/10/5.3-October-2022.pdf.

[42] D. Harutyunyan, R. Behravesh, and N. Slamnik-Kriještorac, "Cost-efficient placement and scaling of 5g core network and mec-enabled application vnfs," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 241–249, 2021. Online [Available]: https://ieeexplore.ieee.org/document/9463941.

Nina Slamnik-Kriještorac is a Postdoctoral Research Fellow at the IMEC research center in Belgium, and at the University of Antwerp. In 2016, she obtained her Master's degree in telecommunications engineering at the Faculty of Electrical Engineering, University of Sarajevo, Bosnia and Herzegovina, and her Ph.D. degree in 2022, at the University of Antwerp-imec, IDLab-Faculty of Applied Engineering. She authored or co-authored more than 30 publications in journals and international conferences. Her current research is mostly based on NFV/SDN-based network architectures with edge computing for vehicular systems, and the management and orchestration of the flexible and programmable next-generation end-to-end network resources and services, with a focus on edge applications.



F. Zarrar Yousaf is a Senior Researcher at NEC Laboratories Europe, Germany. He completed his PhD from TU Dortmund, Germany. His current research interest is on NFV/SDN in the context of 5G/6G networks. He is also actively involved in the ETSI NFV standards organization, where currently he is serving as the Chair of the Solutions WG and is also a Rapporteur of multiple work items. He has extensive experience in the design, development, modeling, simulation and prototyping of communication systems and protocols for optimizing overall network performance. He has a total of 17 granted patents while his research work has been published in several peer reviewed journals, conferences and book chapters.

Girma M. Yilma is a Research Engineer at NEC Laboratories Europe in Heidelberg, Germany. He received his B.Sc. (2010) in Electrical and Computer Engineering from Addis Ababa University, and M.Sc. (2016) in Telecommunications Engineering from University of Trento, Italy.His current research interest is focused around NFV MANO, Orchestration, Cloud Native Networking, NFV/SDN related technologies in the context of 5G networks and beyond.

Rreze Halili obtained her Ph.D. from the Faculty of Applied Engineering at the University of Antwerp, Belgium in 2022. Her doctoral research was on Joint communication and localization for connected vehicles using wireless communications and RTK GNSS. Currently, she holds the position of Principal Research Fellow at the University of Antwerp, IDLab-imec. Her research interests involve localization using wireless communications and satellite technologies, integrated communication and sensing-related topics, smart mobility, radio wave propagation, and electromagnetic radiation exposure.

Marco Liebsch is chief researcher at NEC Laboratories Europe GmbH and is working in the area of 5G mobility management, mobile edge computing and content distribution, mobile cloud networking, and software-defined networking. He received his Ph.D. degree from University of Karlsruhe, Germany, in 2007. He worked in different EU research projects and is contributing to standards in the IETF, ETSI and 3GPP. He has a long record of IETF contributions as well as RFC, journal and conference publications.

Johann M. Marquez-Barja is a Professor at University of Antwerp, as well as a Professor in IMEC, Belgium. He is a member of ACM, and a Senior member of the IEEE Communications Society, IEEE Vehicular Technology Society, and IEEE Education Society where he participates in the board of the Standards Committee. His main research interests are: 5G advanced architectures including edge computing; flexible and programmable future end-to-end networks; IoT communications and applications. He is also interested in vehicular communications, mobility, and smart cities deployments. Prof. Marquez-Barja is co-leading the Citylab Smart City testbed, part of the City of Things programme, and the SmartHighway testbed, both located in Antwerp, Belgium.