Service function chaining for mission critical services

Bruno Sousa[1], Henrique Silva[1], Noé Godinho[1], Marilia Curado[1]

[1] _University of Coimbra, CISUC/LASI, Department of Engineering Informatics, Portugal_

Corresponding author: Bruno Sousa, bmsousa@dei.uc.pt

Services in critical missions, such as firefighting, have demanding service level agreements. The functions of such services must be deployed and managed according to the status and needs of the missions. The proposed Mission Critical Service Function Chaining (MSC-SFC) introduces an architecture relying on Software-Defined Networking (SDN), Service Function Chaining (SFC) and Multiaccess Edge Computing for efficient resource management in dynamic and stringent scenarios like mission critical services (e.g., combatting forest fires). With MSC-SFC, service functions currently assigned to vehicles equipped with computing resources that are leaving the mission can be transferred to nodes based on criteria such as location, latency, or mission duration. In addition to these resource management methods, MSC-SFC also suggests a framework for deploying functions in fire-combat scenarios, ensuring that functions are allocated based on the specific roles of vehicles and first responders in the field. The evaluation results demonstrate SDN and SFC as promising technologies towards the management of service functions in mission critical services, where resources associated with service functions need to be properly managed to reduce impact in performance and security.

Keywords: IoT, mission critical services, multiaccess edge computing, SDN, service function chaining

1. INTRODUCTION

Mission Critical Services (MCS) are relevant for the safety of people, but they have stringent challenges to the underlying communication technologies; for instance, how services can be deployed and how operational/tactical policies can be satisfied, considering the dynamic environments, where nodes can leave the mission due to resource exhaustion like water or fuel in tanks, or where nodes can enter the mission after its start - late entry process that requires configuration of existing communication groups. Mission critical services are organized in a hierarchical fashion, with command control centres managing the overall operations, and commanders controlling the teams of first responders in the field. In addition, such services require stringent Service Level Agreements (SLAs) in terms of availability, reliability, and security.

Public Mobile Radio (PMR) technologies, such as TETRA, are outdated and do not meet the required functionalities, as identified by first responders [1]. Technologies like LTE and 5G have included several features to enhance the support of MCS, such as proximity services within the same security levels as PMR technologies, and for several domains like public safety, health care and critical infrastructures.

To fulfil MCS requirements, Multiaccess Edge Computing (MEC) support in 5G aims to enhance the network, by providing high reliability, low latency, high availability, and higher levels of security. Through MEC it is possible to maintain communications between teams for their safety or for immediate assistance on man-down or vehicle malfunction events. MEC improves reliability, as computation tasks can be distributed and performed close to the user/end devices [2]. MEC allows for dynamic management of resources during missions and according to the human or/and technical resources operating in a specific location. Such management can consider multiple levels, where nodes can be used to support a variety of application types, and can be interconnected at multiple levels [3]. At the *edge level* nodes collect sensor data and control devices. At the *intermediate level*, nodes are responsible for performing data transformation to support real-time processing. Lastly, at the *cloud level*, data aggregation and deep analysis can occur with the support of artificial intelligence techniques.

Service Function Chaining (SFC) outlines the method for decomposing various service functions in a technology-agnostic and network topology-independent manner, which is an important key factor when transitioning to new-generation networks (e.g., from 5G to 6G). SFCs consider the order of flows; that is, functions that must be applied before others, and the nodes on which such functions can be executed. SFC leverages Software-Defined Networks' (SDN) flexibility and automation to decompose services into functions that can be placed at different locations to enhance availability and fault-tolerance. The SFC model with edge and cloud computing enables data-centric approaches, where data consolidation and preprocessing can be performed at the network edge. Processing at the edge saves resources at the core and enables energy savings at the end devices. It also reduces end-to-end latency, since computation-intensive tasks can be performed closer to the user. Data consolidation support at the edge of the network enhances the potential of proximity services in public safety networks, towards the safety of first responders, such as a man-down event or vehicle malfunction [4]. SFC allows to achieve reliability in mission critical services, by decomposing services into multiple functions that can be deployed in distinct and redundant nodes to enhance service availability [5].

The architecture herein proposed for Mission Critical Service Function Chaining (MCS-SFC) relies on the SDN paradigm to enable the management of services in highly dynamic scenarios, with high mobility levels of users/vehicles, with teams being established on the field and on-the-fly, according to the available resources. Some teams may perform a late entry, which means that they enter on the scene in the middle of the mission and need to be on the same communication channels as other teams nearby. However, teams participating in the mission from the start moment can leave the scene due to resource exhaustion (e.g., water, fuel) or to substitute exhausted team members. To the best of our knowledge, no integrated solution for mission critical services has yet been proposed considering SFC, SDN and service migration. We introduce MCS-SFC as an integrated architecture relying on SDN and SFC for resource orchestration in mission critical services, according to the demanding service level agreements and security levels.

Critical services in MCS include voice communications, such as Push-To-Talk (PTT), or in full duplex mode, sensor data acquisition, video streaming services for full situational awareness at Command Control Centres (CCC) [6], according to Portuguese legislation [6]. With respect to the MCS reliability standards, PTT and full duplex communications gain from the decomposition facilitated by SFC, which supports their implementation in vehicles/nodes based on their role in the mission. The goal is to ensure that the communication channels are accessible for mission management in the field. In the same line, other services like sensor data and video streaming become more resilient in the challenging MCS scenarios.

The MCS-SFC architecture is decoupled from specific technologies (e.g., 5G/6G), but benefits from the availability of multiple connections to increase reliability and for enhanced operationality. MCS-SFC is aligned with the hierarchical organization of human and technical resources in mission critical scenarios [6].

The remainder of the article is organized as follows: Section 2 discusses related work and Section 3 presents the MCS-SFC proposal. Section 4 presents the evaluation methodology and the respective results, while Section 5 concludes the paper.

2. RELATED WORK

Bellavista el al. [7] consider reactive and proactive handover approaches to migrate services and data in edge contexts. The proposed scheme mainly targets services implemented as containers, being evaluated in a testbed without considering chaining of functions. The results, in terms of migration time and data loss metrics, demonstrate a gain in performance with a proactive handover approach.

Kiba et al. [8] propose ML-SFCDA that is a graph-based SFC deployment algorithm that considers two types of services: high priority and non-priority. Critical / high priority services can be mapped with the 5G uRLLC, where the survivability of services is ensured through backup resources. The non-priority services can be mapped into the enhanced Mobile Broadband (eMBB) service category of 5G, and are those which are able to tolerate service disruption to a higher level. In this regard, the ML-SFCDA includes a migration-aware algorithm for non-priority services to employ the non-used backup resources of high priority services to increase resource utilization efficiency and at the same time minimize the service interruption. The high priority services of an SFC are deployed and constantly monitored, so that they can be migrated when there are failures. Although the aim is to reduce the migration delay and cost associated with new VNF instantiations, it leads to higher power consumption at the nodes, which is crucial considering that missions can last long periods of time.

Fontoulakis et al. [9] consider the performance of SFCs according to Markov chains and queue theory in multiaccess edge computing environments, but do not consider the requirements of critical services, as opposed to [5], which provides a solution for dynamic chaining of critical services. Despite this, the proposed solution focusses mainly on the reliability of critical telecommunication services.

Mission critical services are associated with real-time services that have demanding QoS requirements, for PTT, video communications (MCVideo), and sensor data (MCData) [10, 11]. In this context, approaches have evaluated the provisioning of Virtual Network Functions (VNFs) [12] at the edge with latency constraints. Other works have mainly focused on the deployment of such services to avoid overloaded networks [10, 11].

Resource provisioning schemes have been proposed for MCS scenarios, considering the delay-sensitive use of fog resources [12], or by power consumption restrictions on virtual machines [13]. Such schemes optimize resource allocation by employing fog colony models, grouping fog resources where a fog control node manages fog cells. Other approaches optimize resource allocation using linear programming to minimize energy consumption considering heterogeneous VMs.

Previous work [6] considered the evaluation of mission critical services through SFC and edge computing; nevertheless, this was with the limitation of not supporting service migration. This manuscript overcomes this issue by presenting an architecture for service function migration, considering different approaches to minimize latency and optimize the lifetime of the mission.

3. MISSION CRITICAL SERVICE FUNCTION CHAINING

This section presents the Mission Critical Service Function Chaining (MCS-SFC) architecture in a compliant fashion with multiaccess edge computing, SDN and SFC technologies.

3.1 Overall architecture

The MCS-SFC architecture for service function chaining of mission critical services employs the multiaccess edge computing model, by considering the hierarchy of teams in the fire combating fields and the respective Command and Control Centre (CCC). IoT can greatly benefit first responders in terms of safety and completing their mission due to increased situational awareness. In fact, safety-related events can be propagated to a CCC in a transparent fashion by resorting to the sensing capabilities of sensors and without the intervention of first responders.

Fig. 1 shows the proposed MCS-SFC architecture considering three layers: edge, intermediate, and cloud layers, according to the hierarchical organization of teams in fire-combat scenarios. In each of these layers, different nodes can be deployed, which have the control and data planes, towards the management of service functions through the SDN paradigm.

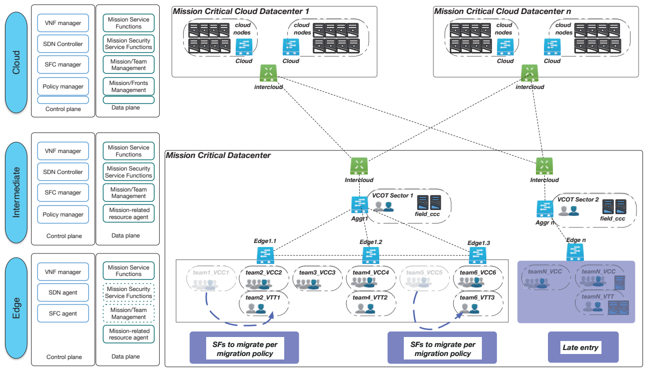


Figure 1 – Overall MCS-SFC architecture

The proposed architecture considers the hierarchical organization established in Portuguese national directive plans to combat fire, and the tactical organization of critical services [6]. In such directives, the human and technical resources (e.g., vehicles) are organized considering the risk phase, being more severe during summer, and according to the event situation and respective evolution. The directives establish the organization of teams and the structure of the command and control centre. Each team is composed of human resources supported by Vehicles for Combating Fire (VCF), Vehicles with Water Tanks (VWT) and depending on the size, teams can also be managed by commanders in the field within the support of Vehicles equipped with devices for Communications (VCC) or supported by Vehicles for Communications and Tactics (VCOT). The vehicles combating fire can be equipped with computing devices, such as Raspberry PI or other types of devices with computing resources to support their teams in terms of mission realization, but also to efficiently support the required communications for mission critical services.

3.2 Mission critical services

The mission critical service functions include voice communications and data. Team leaders, one assigned per vehicle, communicate the current status of their team upon arrival at the scene, when they are without resources, such as fuel or water, when the team is at risk, or when needing additional support. Such communications are carried out through established communication systems, with the support of the respective vehicles. In the unavailability of the communication infrastructure, communications may be limited to teams that are connected to the same multiaccess edge node, or through secondary and more limited communication systems [6] (e.g., SIRESP or in the operational network dedicated to firemen, ROB) [6]. Data communications convey sensor information on the physiological status of first responders. With accelerometers it is possible to detect man-down events, or that the teams' location is risky due to the evolution of the natural conditions like wind direction and speed. Upon availability of such data, specific alarms can be triggered at the CCC. Biosensor data are relevant for assessing the levels of exhaustion of first responders so that they can be replaced by other teams.

Video streaming services (MCVideo) contribute to improving situational awareness. Wearable cameras can be placed in firemen's protection equipment to provide useful information regarding fire and surroundings, to team leaders, operators, and commanders in the CCC. Such cameras must have hands-free interfaces to not interfere with the mission's execution.

Professional radio devices or mobile phones can provide information about the exact location of teams. This information, along with the information provided by other services like video, IoT devices sensing the environment of physiological status of a firemen, enables the identification of risks.

Internet access is helpful for teams involved in combating fire, for accessing risk information from official entities like civil protection, or for mapping services. Navigation is useful to follow the shortest and most secure route or to find the location(s) with logistics related to food or with fuel and water supply.

Mission analysis services have the potential to enhance mission tactics. Access to vehicle sensor meteorological data can establish risk maps in real time. In addition, the information of vehicle sensors (MCData) regarding water usage allows one to predict when a certain vehicle needs to leave the mission to refuel.

The MCS-SFC manages service allocation considering the particularities and roles performed by the vehicles in the mission. In this regard, voice, location, and security-related services should be supported in all vehicles (VCOTs, VCCs, VCFs, VWTs), as they are essential to the mission and are crucial to the safety of first responders. The MC-IoT service should be deployed in vehicles with a high number of first responders, and the video services should be deployed in vehicles dedicated to communications, such as VCOT and VCCs.

3.3 Functions of mission critical services

To support voice and data services, various service functions are considered, according to Table 1, for the services previously identified. The firewall function is considered essential for security services with the optional Intrusion Detection and Prevention Systems (IDPS) function for enhanced security. The expense of service functions is evaluated based on the necessary Million Instructions Per Second (MIPS) operations. Using MIPS is a common procedure for modelling the complexity of services [14].

The service functions of the several services are chained differently to take into account the edge, intermediate, and edge layers for compliance with multiaccess edge computing. Thus, nodes placed at the edge and with low computational resources aim to ensure basic functionalities. In this regard, the firewall function (SF-FW) runs in all nodes, while SF-IDPS runs on the nodes with more resources. This allows service functions to run at the cloud/intermediate levels that require more computational resources.

The mission analysis, due to the computational resources can run on the cloud side, thus the computational cost is not considered in Table 1. The MCS-SFC goal is the functions running in the operational theatre for firefighting or public safety.

The computational cost of each service function models the complexity associated with each function in MIPS and relies on a uniform distribution. The computational cost includes packet processing and memory access costs. The cost values for the various service functions are considered as in [6], where the packet processing cost includes the per packet processing cost and the per byte processing cost in a ratio of the CPU clock frequency, while the memory access cost considers the per packet memory and the per byte memory access in a ratio of time to access memory. This modelling approach was chosen to provide a preliminary evaluation of the MCS-SFC architecture proposal.

Multimedia services are demanding services in terms of resource and latency requirements. Chaining the different functions, such as sf-VidConf, sf-VidAna, sf-VidTx, requires minimization of resource consumption under the constraints of the delay requirements [15].

Table 1 – Mission critical service functions [6]

Service	Function	Description	Computation Cost in MIPS
Voice PTT	SF-VoConf	Configure voice channels	[1, 30]
	SF-VoChan	Voice bridge & channels	[10, 50]
	SF-VoCore	Voice core, recording & logging	[10, 50]
Voice Duplex	SF-VoDown	Voice bridge & channels from CCC to FR	[10, 50]
	SF-VoUp	Voice bridge & channels from FR to CCC	[10, 50]
Location Data	SF-Loc	Retrieve GPS location data	[1, 10]
	SF-LocMap	Map location data	[10, 20]
IoT	SF-IoTacq	Sensors' data acquisition	[1, 10]
	SF-IoTproc	IoT data processing	[10, 20]
	SF-IoTana	IoT analysis for risk maps	[20, 40]
Multimedia	SF-VidConf	Video configuration	[10, 50]
	SF-VidAna	Video real-time analysis and transcoding	[700, 2000]
	SF-VidTx	Video transmission	[50, 200]
Security	SF-FW	Firewall & authentication	[20, 100]
	SF-IDPS	Intrusion Detection Preventation System	[50, 150]
Mission Analysis	SF-MiStore	Storage	N/A
	SF-MiAna	Analysis	N/A

3.4 Service function chaining policies in MCS

SFC policies must be devised to reduce end-to-end latency and maximize operational time, thus avoiding unnecessary processing at the edge layers. In addition, depending on the location of the node, functions such as video analytics, configuration functions, or functions to assess in real-time risk maps can be deployed at the CCC nodes.

Different approaches can be followed to enforce policies through SFCs. In fire-combat s cenarios, t he possibility is to consider the location of a node/resource, the role of a specific node in the mission, and the computational capacities of the node. VCC or VWT vehicles placed on the edge should host data acquisition functions (sf-IoTacq, sf-LocData), mandatory security functions (sf-FW), and support functions for critical communications (sf-VoChan, sf-VoCore, sf-VidTrans). Fig. 2 shows examples of service functions for Voice PTT, Voice Full Duplex, and IoT services.

Voice PTT or Voice Full Duplex have functions deployed at the edge and cloud layers. The functions in the edge layer (e.g., *sf-VoCore*, *sf-VoChain*) are responsible for encoding, while in the cloud layer, functions like *sf-voConf* allow the configuration of groups or other settings.

IoT requires the deployment of service functions in all the layers, where the edge is concerned with the acquisition of data from sensors, while at the intermediate layer, for instance on the multiaccess edge computing node the collected data is processed, aggregated considering the teams at the field or the type of data that is being processed. On the cloud side, the data is used to update risk maps.

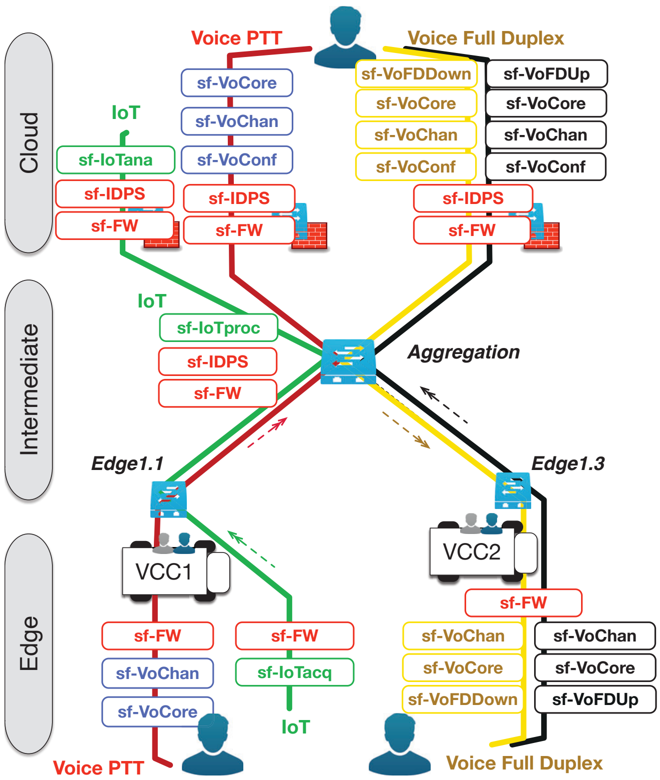


Figure 2 – Example of service function chaining for critical services

Services that have bidirectional flows, like Voice Full Duplex, are a concern for SFCs. In such cases, the policies must consider the direction of flows, as depicted in Fig. 2 in the yellow and black lines. In such cases, some functions can be shared. For instance, voice traffic sent downlink and uplink at the edge can share the firewall function (sf-FW), while at the cloud side, it may include the sf-IDPS. This is relevant for stateful security analysis, in order to track flows of a service sent in both directions.

3.5 Resource management and allocation schemes

MCS-SFC considers different approaches for service function or resource migration, as summarized in Table 2. The first approach is based on distance criteria, where resources are migrated to the closest nodes: *location*. A second approach considers the migration of service functions to nodes providing lower latency levels: *latency*, considering the communications with the CCC located at the cloud layer. Both approaches can lead to the migration of resources between vehicles with heterogeneous characteristics. For instance, resources can migrate from a VCC to a VWT or vice versa. The third approach is mission-orientated: *mission* where migration aims to maximize the mission lifetime, leading to service functions being migrated to nodes that have more resources available.

Table 2 – Resource migration approaches

Algorithm (fAl)	**Description**
disAW - distance/location	Migration of SF to the closest node (e.g., linear distance).
LatAW - latency	Migration of SFs to nodes with lower latency.
misAW - mission	Migration of SFs considering the availability of resources in nodes (e.g., fuel).

MCS-SFC migration approaches must first select the functions to be migrated, as summarized in Table 3. This selection can be based on the priority of services and the availability of computational resources. If resources are not available, some functions can be excluded. For instance, the security functions might be filtered due to resource unavailability and thus not being migrated. The priority of the service functions can be selected based on the criticality of the service in the mission. Voice communications are of the utmost importance and thus have a higher priority in comparison to video streaming or IoT services. The schemes to assign such priority can be static, which establishes a list of services that have priority, and inside such services, the service functions that are mandatory or optional. For instance, the Voice PTT service can have the highest priority in relation to the other services, and in this service, sf-VoCore and sf-VoChan are the functions with the highest priorities.

Table 3 – SF selection policies for migration

Policies	**Description**
noPriority	Migration of SF and VMs in a Best-Effort model.
Priority	Migration of resources prioritising Voice and security SFs.
noSecurity	Migration of resources prioritising Voice and Data SFs without security SFs.

Algorithm 1 specifies the algorithm for the selection of priorities. Another policy that can be employed can ignore the security functions, hence, one can use Algorithm 1, while excluding security functions from the list of service functions to be migrated.

Algorithm 1: Priority selection policy

```
Input: Set of nodes without connection Nc
       Set of available nodes N excluding Nc
       SF migration algorithm MIG
Output: List of SFs Lp
        List of Policies Pp

Lp, Pp = {}
for nc_i ∈ Nc do
    Laux ← SFs in nc_i
    Naux ← Nodes where nc_i was connected
    Paux ← Policies from Naux
    for la_i ∈ Laux do
        switch typeOf(la_i) do
            case voice do priority = 0 ;
            case security do priority = 1;
            case Locdata do priority = 2 ;
            case IoTdata do priority = 3 ;
            case VidStream do priority = 4 ;
            otherwise do priority = Inf ;
        end
        la_i.updateStatus(TO_MIGRATE)
        Lp.insert(priority, la_i)
    end
    Lp = sort(Lp, key = {priority, l_i} for l_i ∈ Lp)
    DoMigrationMCS(MIG, Lp, Pp, nc_i, Naux, Paux)
end
```

Migration approaches are implemented according to Algorithm 1 and Algorithm 2 that details the steps of each approach. For example, the distance / location approach (Method *updDistanceTo()*) uses the Haversine function,which allows one to determine a linear distance, based on geographical coordinates. The mission approach considers as resources the water and fuel in vehicles that provide support to the mission. The probability of outage of such resources is determined and the maximum is considered, since on the failure of a resource, the vehicle needs to leave the mission.

The resource allocation schemes also affect the overall performance of the service. Considering compliance with the MEC model, service functions can be implemented as virtual network functions in different ways:

- A- Each service function is mapped to a virtual machine.
- B- All the service functions of a service are allocated to a virtual machine.

Approach A has the advantage of having lower communication overhead between the service functions, as they are deployed in the same machine. Although approach B provides higher isolation and increased flexibility but with more communication overhead between service functions.

Algorithm 2: Generic migration for *fAl*

```
Function DoMIG(fAl, Lp, Pp, nc_i, Naux, Paux) :
    for l_i ∈ Lp do
        R_i ← GetRequiredResources(l_i)
        Pl_i ← Paux.GetPoliciesOfSF(l_i)
        Nd_i ← Naux.GetLinkedNodesOfSF(l_i)
        for n_j ∈ Nd_i do
            switch fAl do
                case disAw do n_j.updDistanceTo(nc_i);
                case LatAw do n_j.updMinLatency(nc_i);
                case MisAw do n_j.updMissOutage(nc_i);
            end
        end
        Nd_i = sort(Nd_i, key = {fAl_crit, n_i} for n_i ∈ Nd_i)
        for n_j ∈ Nd_i do
            Pn_j ← GetPolicies(n_j)
            if n_j.hasAvailable(R_i) then
                if !Pn_j.contains(Pl_i) then
                    n_j.addPolicies(Pl_i)
                    Pp.insert(Pl_i, n_j)
                end
                MIGRATE(l_i, n_j)
                l_i.updateStatus(MIGRATED)
            else
                l_i.updateStatus(NOT_MIGRATED)
            end
        end
    end
end
Method updDistanceTo(nc_i) : fAl_crit = disTo
    /* Haversine Function                                  */
    this.distTo =
      Haversine(this.getLongitude, this.getLatitude,
      nc_i.getLongitude, nc_i.getLatitude)
end
Method updateMinLatency(nc_i) : fAl_crit = minLat
    minLat = Inf, minLink = NULL
    for l_j ∈ this.getLinks(nc_i) do
        if l_j.isD2D(nc_i) then
            if l_j.latency <= minAux then
                minLat = l_j.latency
                minLink = l_j
            end
        end
    end
    if minLink! = NULL then
        minLink.setMinLat(minLat)
    end
end
Method updMissOutage(timePeriod) : fAl_crit = probOut
    probOut = 0
    for r_j ∈ this.getResources() do
        switch typeOf(r_j) do
            case WATER do
                probWaterOutage =
                  getWaterUsage * timePeriod/CapacityWater
            end
            case FUEL do
                probFuelOutage =
                  getFuelUsage * timePeriod/CapacityFuel
            end
        end
        probOut = max(probWaterOutage, probFuelOutage)
    end
end
```

4. MCS-SFC VALIDATION

This section documents the evaluation and results of MCS-SFC.

4.1 Evaluation methodology and scenario

The evaluation aims to validate algorithms designed in the MFC-SFC architecture, in terms of: 1) resource allocation, 2) resource migration and, 3) policies for service function chaining in a standardized fashion and with SDN support. The allocation of resources mainly affects the computational characteristics required in vehicles, such as CPU processing capabilities and available memory. The resource migration evaluation aims to assess the performance and impact of migration schemes in services, on the late entry event, or when a vehicle needs to refuel. The evaluation scenario is depicted in Fig. 3,

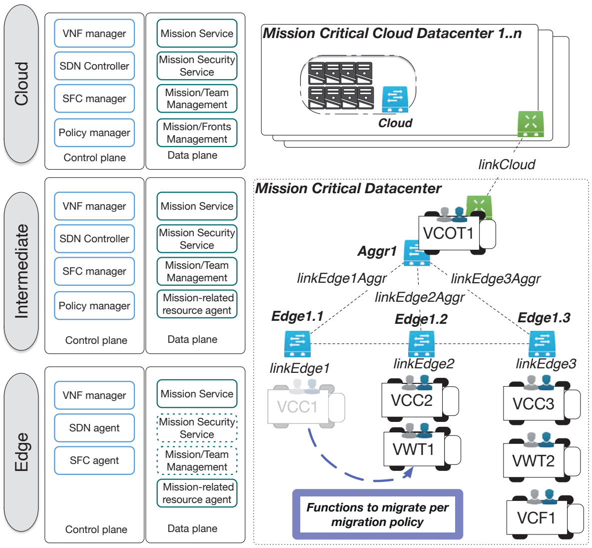


Figure 3 – Evaluation scenario

where the vehicles are placed according to their role in the mission. For example, VCOTs that aggregate communications in the intermediate layer have the role of forwarding traffic to the mission cloud data centre for deep mission analysis, through the *linkCloud* link. Each edge node has a set of four teams, and the vehicles are deployed according to three types of teams:

- minimal with the {VCC} set, 1 per edge (example of Edge 1.1);
- recommended with the {VCC, VWT} set, 1 per edge (example of Edge 1.2);
- complete with the {VCC, VCF, VWT} set, 2 sets per edge (example of Edge 1.3).

MFC-SFC chains the service functions considering the distinct roles and characteristics of the vehicles that support first responders.

LTE settings are considered with a DL/UL bandwidth of 100/50 Mbps and a latency of 30 ms [16], between the intermediate and cloud layers, *linkCloud* link. Each edge has different latency values to assess migration approaches based on latency. Edge 1.1 is configured with 34 ms of latency in *linkEdge1* and *linkEdge1Aggr* links. Edge 1.2 is configured with 36 ms and Edge 1.3 with 37.5 ms, for the respective links.

Table 4 – Capabilities of vehicles (VCOTs do not have water tanks) [17, 18]

Vehicle	**Water** Litres	**Water Debit** (L/min)	**Fuel** Litres	**Fuel Debit** (L/min)
VCOT	–	–	22	77
VCC/VCF	3000	83	900	38
VWT	8000	492	239	38

Vehicles are configured with fixed positions and are associated with capabilities based on their role in the mission and the type of trucks commonly used, such as Volvo trucks [18]. The vehicle configuration sets are summarized in Table 4, considering the technical details available for the various types of vehicle (e.g., the capacity of the water tank).

Table 5 summarizes the migration settings, considering that vehicles migrate at different times and leave the mission simultaneously in a different number (e.g., 1, 5, 10). When vehicles migrate, the service functions that are deployed in each vehicle need to be migrated. The configured SFs are documented in Table 1 for mission critical services.

Table 5 – Migration settings

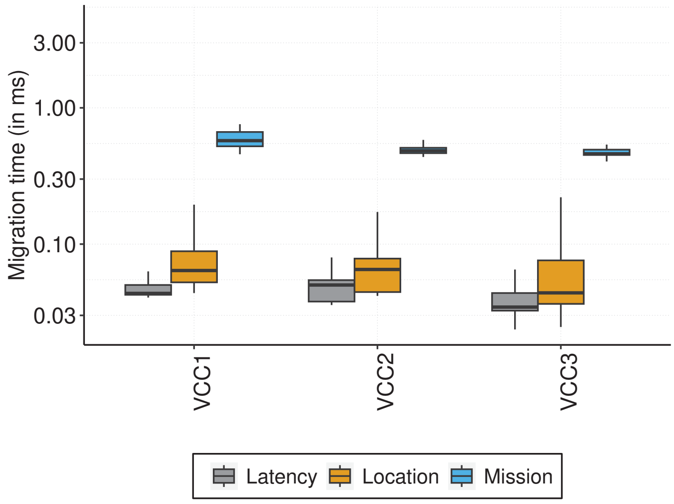
# Migrating	Instants	Vehicles migrating
1	20s	1 VCC
5	{20, 40, 60}s	5 VCCs per instant
10	20s	10 VCCs

Validation is implemented in the CloudSimSDN simulator [19] which supports edge computing and SDN.

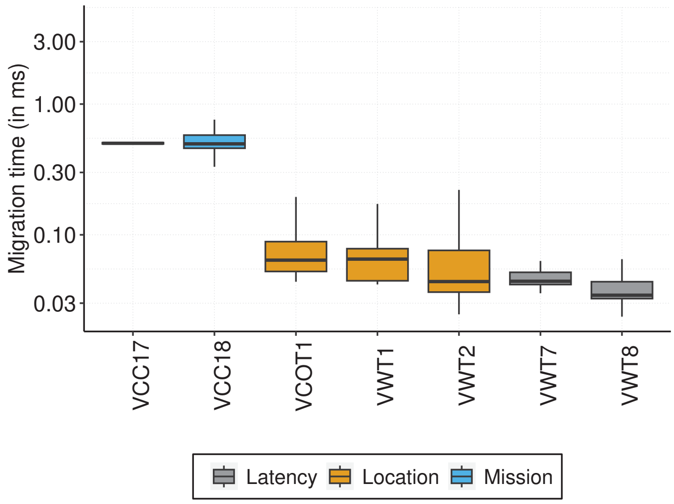
4.2 Migration evaluation results

The results include the migration time of the various service functions according to the migration settings summarized in Table 5. Fig. 4 shows the results for one migration vehicle, Fig. 5 shows the results for five migration vehicles, and Fig. 6 for ten migration vehicles. These figures report the values for the mapping B approach, "A Service function per VM". The results are discussed per migration algorithm to assess the impact of each approach in overall migration.

Fig. 4 shows the results for the different migration approaches when one vehicle leaves the mission. In terms



(a) Migrating vehicles

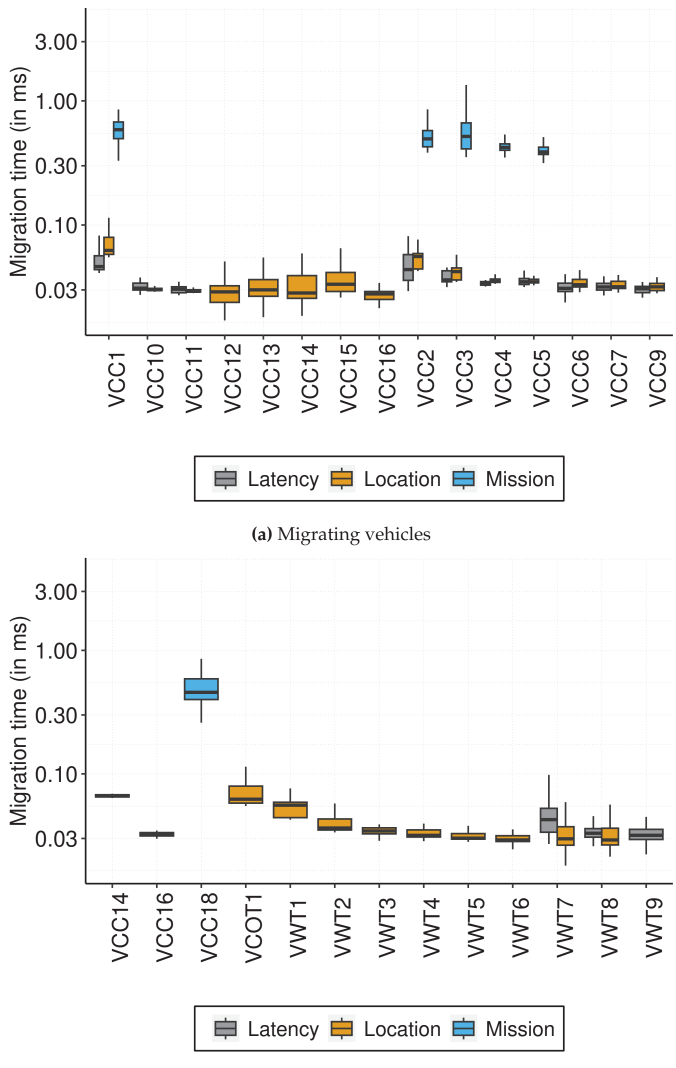


(b) Target vehicles

Figure 4 – One (1) vehicle leaving mission

of performance, the diverse approaches have similar performance in terms of migration time, with the main difference being based on the target hosts that are chosen. The latency chooses VWT7 and VWT8, as they are connected to *linkEdge1* which has latency values of 34 ms. The location approach chooses the VCOT1, VWT1 and VWT2 vehicles, as they are part of the same team (VCC2 → VWT2), thus acting at the same location. The mission chooses vehicles that have more resources in terms of water or fuel, which can accommodate new functions to prolong the mission's life. The mission algorithm tends to have an impact in migration, as more information needs to be collected from all vehicles that are in the mission.

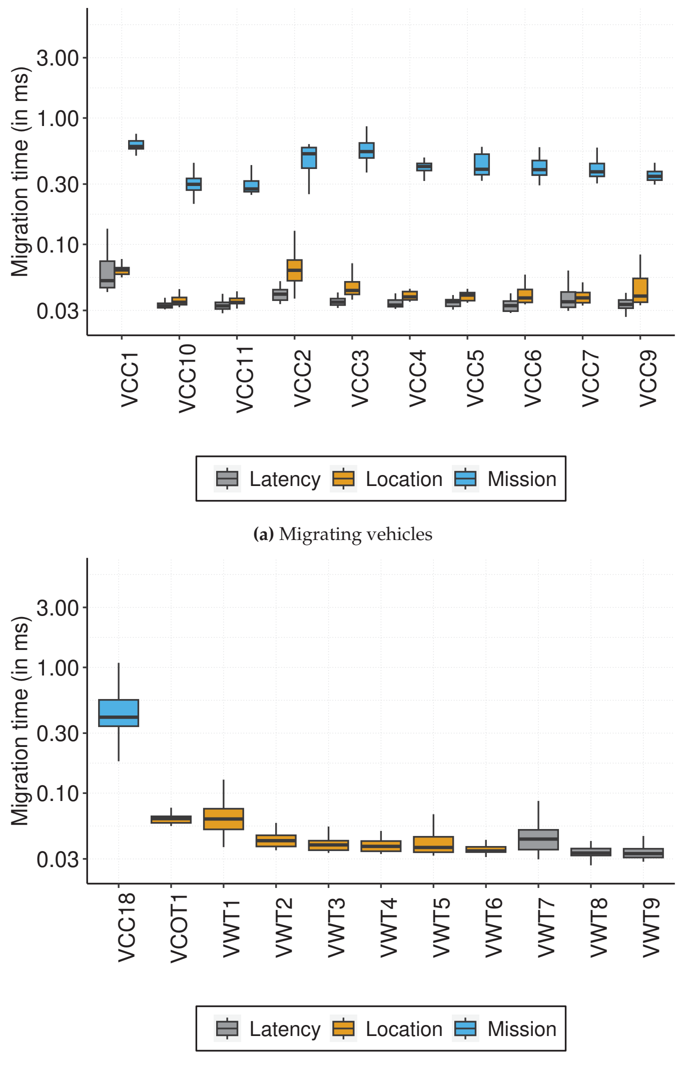
Fig. 5 shows the results for the different migration approaches when five vehicles leave the mission simultaneously. One can observe that there are 15 vehicles that leave the mission and 13 target vehicles. Migration has the same behavior as in the case where one vehicle leaves the mission. However, in the instant 60s (not pictured),



(a) Migrating vehicles

(b) Target vehicles

Figure 5 – Five (5) simultaneous vehicles leaving mission



(a) Migrating vehicles

(b) Target vehicles

Figure 6 – Ten (10) simultaneous vehicles leaving mission

the target hosts cannot accommodate additional service functions, leading to the choice of new target vehicles such as VWT9, VCC14, and VCC16 for the latency approach. With location and mission approaches, it can be seen that some vehicles accommodate a large number of functions, as the migration delay also increases in the VCC18 and VCOT1 target vehicles. This is expected as more service functions need to be assigned in the target vehicles. Also worth mentioning is that vehicles migrating in the first time instance (20s) do not return to the scenario.

Fig. 5 depicts the results for the different migration approaches when ten vehicles leave the mission simultaneously. The performance results are similar with the five (5) simultaneous migration test cases in distinct time instants, the only difference is that the final number of migrating vehicles is ten, as expected. The mission, by concentrating the migrating functions in a specific vehicle, can impact the performance of the distinct services, (video, voice, as documented in Table 1).

The performance of the security, video and voice services is shown in Fig. 7, for the different mapping strategies and also the number of service functions that are migrated, as per Fig. 8. The functions of the security service have higher migration time values, compared to other

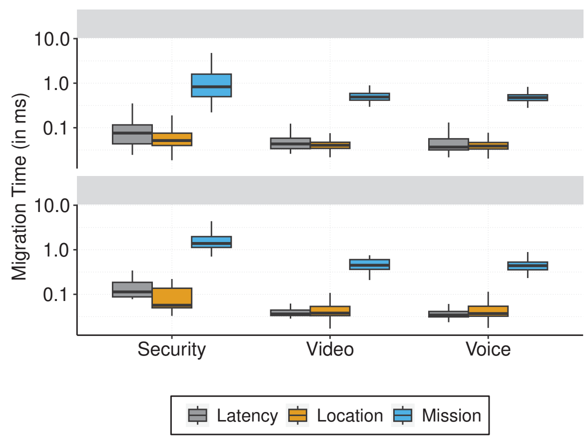


Figure 7 – Migration time of service functions (in milliseconds).

services (e.g., video, voice). This is because each service is configured in the SFC to use security functions such as the firewall and IDPS. The approach to the migration of functions also has an impact on the service migration time. As noted above, the migration algorithm leads to higher values in terms of latency for all the services and mapping approaches. This is due to the fact that all service functions are concentrated in a single vehicle. The benefit of the *Latency* approach is more evident when all the services functions are mapped to a virtual machine, approach A. When migrating a VM in approach A, all the hosted functions are also migrated. Thus, reducing the time required to find a target vehicle to host a service function.

The benefit of the *Latency* approach is when migration is performed according to resource allocation approach B that maps a service function to a virtual machine. Within the resource allocation approach B, the migration time is higher, since all the functions need to migrate towards different virtual machines. The migration time is almost similar between the video and voice services, which demonstrates that MCS-SFC introduces low overhead. The *Location* approach selects the hosts that are nearby, leading to the selection of Vehicles with Water Tanks (VWTs) that are usually in the vicinity of vehicles fighting fires. VWTs are used to refill the water from vehicles in firefighting.

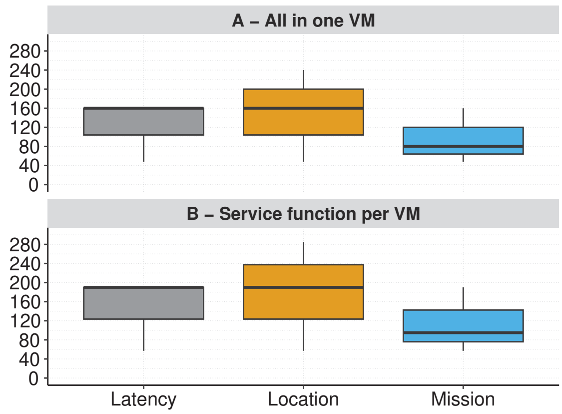


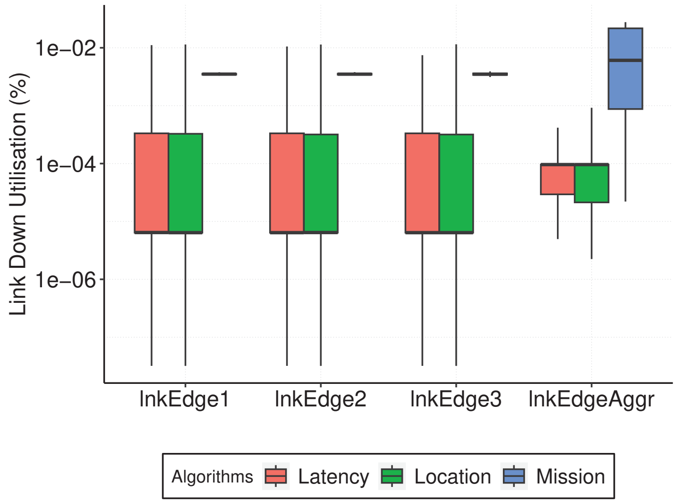
Figure 8 – Number of service functions migrated

Fig. 8 shows the number of service functions that are migrated using resource management and resource allocation approaches. The *Latency* approach selects a smaller number of destinations due to the delay selection criteria. Destination vehicles must support lower or equal latency values compared to the vehicle that is migrating. The *Location* approach is mainly concerned with the location of vehicles, as long as there are available resources.

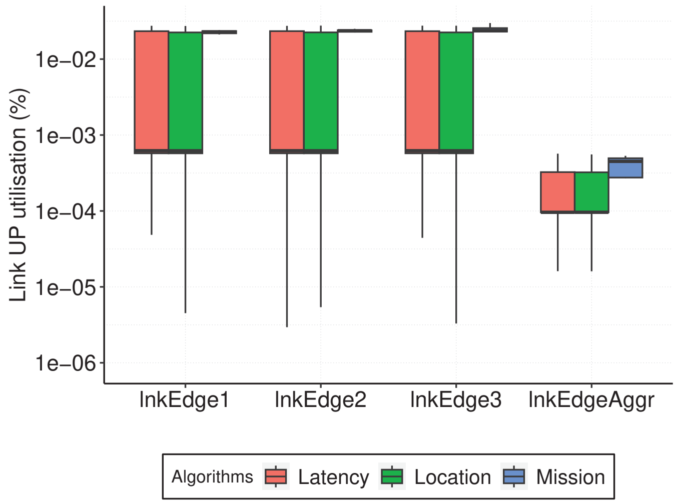
The allocation approach B, with a service function per virtual machine, introduces a higher number of migrations, since the service functions of a service can be distributed in other hosts/vehicles. With approach B, it is easier to find resources for a single service function in comparison to approach A, which concentrates all the functions of a service into a single virtual machine.

4.3 Link utilization evaluation results

Fig. 9a depicts the DL downlink utilization ratios and Fig. 9b shows the UL uplink directions. The link utilization ratio in DL and UL is similar, as depicted in Fig. 9.



(a) Link down



(b) Link up

Figure 9 – Link utilization ratios for the mapping B approach

The mission migration algorithm substantially introduces more overhead in all links providing connection in the edges (i.e., linkEdge1, linkEdge2, linkEdge3), in both directions. As reported in Section 4.2, the mission algorithm tends to migrate all service functions to a single host to maximize the mission's life. This leads to migration of more service functions between virtual machines that can be located in the intermediate layer (i.e., VCOT1), leading to higher usage resource ratios in the *linkEdgeAggr* links.

The reduced usage of edge links occurs because the latency and location algorithms, which tend to use vehicles that are nearby, or in the same edge link. Vehicles can communicate with each other directly without requiring support from edge nodes (e.g., Edge 1.1 in Fig. 3). The mission approach has a higher link usage ratio as some teams do not have water tank vehicles, requiring the migration of VMs between vehicles that are connected to distinct edges. This is more evident in Fig. 9b with the uplink, where in approach B the utilization of the link has values of around 0.02%.

5. CONCLUSIONS

Mission Critical Services (MCS) can benefit with technologies like Service Function Chaining (SFC), Software-Defined Networks (SDN) and Multiaccess Edge Computing (MEC). Such technologies provide the ability to design solutions that can work in current and future wireless technologies (5G and beyond).

Despite the potential of service function chaining and Software-Defined networks, several aspects must be considered for real deployments. Further research on Software-Defined network mechanisms is required to support the dynamic environments of MCS, with nodes constantly changing their connections to the network. Either using centralized or distributed SDN controllers, which allow to configure the data plane through protocols like OpenFlow or P4. The mobility of vehicles leads to frequent changes in the network topologies, which require efficient mechanisms for routing, to determine paths that meet the requirements of services (e.g., lower latency, higher bandwidth). The configuration of routing paths must consider the dynamics in MCS through programmable approaches, like P4. Additionally, the dynamics in MCS requires (re)organization of teams and, consequently, policy adjustment through service function chaining and software-defined network mechanisms. SFC adds additional policy information to each packet to convey the packet flows to a function. This can be performed with programmable approaches such as P4 in cooperation with SDN controllers.

The placement of service functions must be optimized according to service requirements. The MCS-SFC approach demonstrates the feasibility of allocating functions in a distributed fashion according to available resources and per different migration policies. Latency and location tend to introduce better overall performance of services, while the mission approach prolongs the mission life with the cost of impacting the service performance or placing the functions related with a service in a single host.

While such approaches may be applicable to mission-critical scenarios, they require synchronization with SDN mechanisms and with policy manager(s) to aggregate traffic classification rules in the service function chains. Our next steps will be in the direction of implementing such policies using P4 data plane programming integrated with SDN controllers like ONOS for the control plane management and unified SFC policy management.

ACKNOWLEDGEMENT

This work was supported in part by project MH-SDVanet: Multihomed Software-Defined Vehicular Networks (reference PTDC/EEI-COM/5284/2020); in part by the national funds through the FCT - Fundação para a Ciência e a Tecnologia, I.P., in the framework of the Project UIDB/00326/2025 and UIDP/00326/2025.

REFERENCES

[1] International Forum to Advance First Responders Forum. *Common Capability Gaps*. https://www. internationalresponderforum.org/capability_gaps. 2020.

[2] Najib Ahmed Mohammed, Ali Mohammed Mansoor, and Rodina Binti Ahmad. "Mission-Critical Machine-Type Communication: An Overview and Perspectives Towards 5G". In: *IEEE Access* (2019). ISSN: 21693536.

[3] Cheol Ho Hong and Blesson Varghese. "Resource management in Fog/Edge computing: A survey on architectures, infrastructure, and algorithms". In: *ACM Computing Surveys* 52.5 (2019). ISSN: 15577341. DOI: 10.1145/3326066.

[4] Nathan F. Saraiva de Sousa, Danny A. Lachos Perez, Raphael V. Rosa, Mateus A.S. Santos, and Christian Esteve Rothenberg. "Network Service Orchestration: A survey". In: *Computer Communications* 142-143 (2019). ISSN: 1873703X. eprint: 1803.06596.

[5] Shuang Qin, Mengjie Liu, and Gang Feng. "Dynamic Service Chaining for Ultra-Reliable Services in Softwarized Networks". In: *IEEE Transactions on Network and Service Management* 20.3 (2023), pp. 3585–3595. DOI: 10.1109/TNSM.2023.3237587.

[6] Bruno Sousa, Henrique Silva, Noé Godinho, and Marilia Curado. "Service Function Chaining in Wildfire Scenarios". In: *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*. 2021. DOI: 10.1109/NetSoft51509.2021.9492699.

[7] Paolo Bellavista, Antonio Corradi, Luca Foschini, and Domenico Scotece. "Differentiated Service/Data Migration for Edge Services Leveraging Container Characteristics". In: *IEEE Access* 7 (2019), pp. 139746–139758. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2943848.

[8] Godfrey Kibalya, Joan Serrat, Juan-Luis Gorricho, Jonathan Serugunda, and Peiying Zhang. "A multi-stage graph based algorithm for survivable Service Function Chain orchestration with backup resource sharing". In: *Computer Communications* 174.May (June 2021), pp. 42–60. ISSN: 01403664. DOI: 10.1016/j.comcom.2021.04.008.

[9] Emmanouil Fountoulakis, Qi Liao, and Nikolaos Pappas. "An End-to-End Performance Analysis for Service Chaining in a Virtualized Network". In: *IEEE Open Journal of the Communications Society* 1.December 2019 (2020), pp. 148–163. ISSN: 2644-125X. arXiv: 1906.10549.

[10] Sotirios Spantideas, Anastasios Giannopoulos, Marta Amor Cambeiro, Oscar Trullols-Cruces, Eneko Atxutegi, and Panagiotis Trakadas. “Intelligent Mission Critical Services over Beyond 5G Networks: Control Loop and Proactive Overload Detection”. In: *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*. 2023, pp. 1–6. DOI: 10.1109/SmartNets58706.2023.10216134.

[11] Sotirios T. Spantideas, Anastasios E. Giannopoulos, and Panagiotis Trakadas. “Smart Mission Critical Service Management: Architecture, Deployment Options, and Experimental Results”. In: *IEEE Transactions on Network and Service Management* (2024), pp. 1–1. DOI: 10.1109/TNSM.2024.3498348.

[12] Jungmin Son and Rajkumar Buyya. “Latency-aware Virtualized Network Function provisioning for distributed edge clouds”. In: *Journal of Systems and Software* 152 (2019), pp. 24–31. ISSN: 01641212. DOI: 10.1016/j.jss.2019.02.030.

[13] S. Mishra et al. “Sustainable Service Allocation Using a Metaheuristic Technique in a Fog Server for Industrial Applications”. In: *IEEE Transactions on Industrial Informatics* 14.10 (2018), pp. 4497–4506. ISSN: 15513203. DOI: 10.1109/TII.2018.2791619.

[14] B. Booba, X. Joshphin Jasaline Anitha, C. Mohan, and Jeyalaksshmi S. “Hybrid approach for virtual machine allocation in cloud computing”. In: *Sustainable Computing: Informatics and Systems* 41 (2024), p. 100922. ISSN: 2210-5379. DOI: 10.1016/j.suscom.2023.100 922.

[15] Duo Xu, Xin Tian, Khanh Pham, Erik Blasch, and Genshe Chen. “Virtual Network Function Placement for Mapping SFC Requests of UAV-Sourced Video Streaming in Cloud Networks”. In: *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2024, pp. 523–528. DOI: 10.1109/ICCWorkshops59551 .2024.10615913.

[16] Hisham Zorghani and Ivica Kostanic. “Push-to-Talk over LTE Modeling and Performance Evaluation”. In: *Universal Journal of Com. & Network* (2019). DOI: 10.13189/ujcn.2019.070101.

[17] Corpo de Bombeiros Militar do Estado de Goiás. *Manual de Bombeiros Fundamentos de Combate a Incêndio*. https://www.bombeiros.go.gov.br/wp-content/uploads/2015/12/cbmgo-1aedicao-20160921.pdf. 2016.

[18] Volvo. *Redução do consumo de combustível para apenas 24 litros*. https://www.volvotrucks.com.pt/pt-pt/news/magazine-online/2019/sep/Business-Story-Spain.html. 2019.

[19] Jungmin Son, TianZhang Zhang He, and Rajkumar Buyya. “CloudSimSDN-NFV: Modeling and simulation of network function virtualization and service function chaining in edge computing environments”. In: *Software - Practice and Experience* 49 (Dec. 2019). ISSN: 1097024X. DOI: 10.1002/spe.2755.

AUTHORS

Bruno Sousa is an assistant professor in the Department of Informatics Engineering at the University of Coimbra, Portugal, since 2018, from where he got a PhD in informatics engineering in 2014. He is a senior researcher in CISUC and he is an editorial member of Wiley Emerging Transaction of Technology and Frontiers.

Henrique Silva is researcher fellow at the Centre for Informatics and Systems of University of Coimbra (CISUC) since 2018. He concluded his master’s thesis in September 2022 at the Department of Informatics Engineering at the University of Coimbra.

Noé Godinho is a Ph.D. Student at CISUC from the University of Coimbra, Portugal, where he has participated in several projects. He received his bachelor’s and master’s degree in informatics engineering from the same university in 2015 and 2017. His current research interests include resource allocation, routing, multi-objective optimization, quality of service, and combinatorial optimization.

Marilia Curado is a Full Professor at the Department of Informatics Engineering of the University of Coimbra, Portugal, where she obtained the PhD in 2005. She is currently Director of the Laboratory for Informatics and Systems of Pedro Nunes Institute. Her research interests include resilience and quality of service in computer networks, energy efficient communications, internet of things and communications in the Cloud. She is a member of the editorial board in several journals and has been involved in the scientific organization and coordination of several international conferences.